

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**PROPUESTA METODOLÓGICA PARA LA MEJORA DEL
PROCESO DE VERIFICACIÓN Y VALIDACIÓN EN LA
INDUSTRIA DE SERVICIOS TI EN BASE A MODELOS CMMI
CON ITIL**

NATALIA KARIN CARPIO ARNAIZ

TESIS DE GRADO
MAGÍSTER EN INGENIERÍA INFORMÁTICA

Diciembre 2009

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**PROPUESTA METODOLÓGICA PARA LA MEJORA DEL
PROCESO DE VERIFICACIÓN Y VALIDACIÓN EN LA
INDUSTRIA DE SERVICIOS TI EN BASE A MODELOS CMMI
CON ITIL**

NATALIA KARIN CARPIO ARNAIZ

Profesor Guía: Silvana Roncagliolo de la Horra

Programa: Magíster en Ingeniería Informática

Diciembre 2009

Resumen

Cada vez más, a los departamentos de TI se les exigen niveles de calidad y servicio directamente proporcionales a los niveles de calidad y servicio que se exigen a las distintas líneas de negocio de las organizaciones. Esto está provocando un afán por encontrar métodos y técnicas que permitan desarrollar y mantener software de alta calidad, así como implantar y mantener infraestructura con un alto grado de disponibilidad.

Se presenta una propuesta metodológica creada en base a los procesos de Verificación y Validación de CMMI pareados con los Procesos/Funciones de ITIL. En la metodología se exponen las prácticas específicas y se detalla el proceso de cómo implantar la metodología que apoya y mejora los niveles de calidad en el testing funcional de aplicaciones.

A partir de la metodología propuesta se presenta la evaluación de su aplicación a un caso real, definiendo como punto de partida las métricas que se consideran en dicha evaluación, generando mejoras a la propuesta en sí.

Abstract

Everyday, TI Departments are demanded with quality and service levels directly proportional to the quality and service levels demanded to the different business lines of the organizations. This is causing an urge to find methods and techniques that allow to develop and maintain high quality software, such as to implant and keep the infrastructure with a high level of availability.

A methodological proposal is shown, created based on the CMMI verification and validation processes matched against the ITIL processes/functions. In the methodology specific practices are exposed and the quality levels are improved in the functional testing of the applications.

From the proposed methodology, the evaluation of its application to a real case is showed, defining as a start point the metrics considered in that evaluation, generating improvements to the proposal itself.

Índice

1	Introducción.....	9
2	Objetivos	10
3	Estado del Arte	11
3.1	ITIL v2.0.....	11
3.1.1	<i>Descripción (PRamirez-FDonoso, 2006)</i>	<i>11</i>
3.1.2	<i>Objetivos de ITIL (PRamirez-FDonoso, 2006).....</i>	<i>12</i>
3.2	CMMI.....	13
3.2.1	<i>Historia y evolución (JPalacio, 2006)</i>	<i>13</i>
3.2.2	<i>Los Cinco Niveles de Madurez (Axentia, 2006).....</i>	<i>13</i>
3.2.3	<i>Arquitectura del Modelo (Axentia, 2006)</i>	<i>15</i>
3.3	CMMI e ITIL (JRaggio, 2004)	17
3.4	Análisis del Estado del Arte.....	18
4	Solución Propuesta.....	21
4.1	Conocimiento de los modelos.....	21
4.2	Proceso de cruce de los modelos	21
4.3	Definición de métricas para la evaluación final	25
4.3.1	<i>Tiempo</i>	<i>26</i>
4.3.2	<i>Costo.....</i>	<i>26</i>
4.3.3	<i>Alcance</i>	<i>26</i>
5	Implantación de la metodología propuesta y aplicación de plantillas de apoyo	32
5.1	Proceso para aplicar la metodología	32
5.1.1	<i>Proceso “Definir proyecto”</i>	<i>34</i>
5.1.2	<i>Proceso “Administrar externo”.....</i>	<i>37</i>
5.1.3	<i>Proceso “Administrar proceso de pruebas” PARTE 1.....</i>	<i>40</i>
5.1.3.1	<i>Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”.....</i>	<i>45</i>
5.1.3.2	<i>Proceso “Estimar grado automatización y nivel diagnóstico”.....</i>	<i>46</i>
5.1.3.3	<i>Proceso “Definir requerimientos no funcionales”</i>	<i>51</i>
5.1.4	<i>Proceso “Administrar proceso de pruebas” PARTE 2.....</i>	<i>53</i>
5.1.4.1	<i>Proceso “Definir prueba”</i>	<i>57</i>
5.1.4.2	<i>Proceso “Automatizar prueba unitaria”.....</i>	<i>61</i>
5.1.4.3	<i>Proceso “Automatizar prueba integrada”</i>	<i>63</i>
5.1.4.4	<i>Proceso “Generar LB datos”.....</i>	<i>64</i>
5.1.4.5	<i>Proceso “Ejecutar ciclo pruebas”</i>	<i>66</i>
5.1.4.6	<i>Proceso “Verificar ambiente prueba”</i>	<i>68</i>
5.1.4.7	<i>Proceso “Ejecutar conjunto de pruebas”</i>	<i>69</i>
5.1.5	<i>Proceso “Administrar proceso de pruebas” PARTE 3.....</i>	<i>70</i>

5.1.6	Proceso: “Gestionar defectos”.....	71
5.1.7	Proceso: “Realizar auditoría CM/PPQA”	74
5.1.8	Proceso: “Controlar proyecto”.....	79
5.2	Resumen de resultados obtenido (Métricas).....	81
6	Conclusiones	89
7	Bibliografía	90

Tabla de Ilustraciones

Figura 3.01: Modelo de Arquitectura CMMI.....	15
Figura 3.02: Niveles de Madurez CMMI.....	16
Figura 3.03: Representación escalonada. Por Nivele de Madurez CMMI.....	17
Figura 4.01: Tabla de mapeo CMMI e ITIL.....	22
Figura 4.02: Metodología propuesta.....	24
Figura 4.03: Triángulo de restricciones.....	26
Figura 4.04: Métricas definidas para validación de la implantación de metodología propuesta.....	28
Figura 4.05: Métrica “Importancia Metodológica”.....	28
Figura 4.06: Métrica “Propuesta Metodológica”.....	29
Figura 4.07: Tabla “Cálculo de métricas”.....	29
Figura 4.08: Tabla “Apego de la metodología propuesta”.....	30
Figura 5.01: Identificación planillas de apoyo.....	32
Figura 5.02: Proceso “Definir proyecto”.....	34
Figura 5.03: Proceso “Administrar externo”.....	37
Figura 5.04: Proceso “Administrar proceso de pruebas” PARTE 1.....	40
Figura 5.05: Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”.....	45
Figura 5.06: Proceso “Estimar grado automatización y nivel diagnóstico”.....	46
Figura 5.07: Proceso “Habilitar PC para captura”.....	50
Figura 5.08: Proceso “Definir requerimientos no funcionales”.....	51
Figura 5.09: Proceso “Administrar proceso de pruebas” PARTE 2.....	53
Figura 5.10: Proceso “Definir prueba”.....	57
Figura 5.11: Proceso “Automatizar prueba unitaria”.....	61
Figura 5.12: Proceso “Automatizar prueba integrada”.....	63
Figura 5.13: Proceso “Generar LB datos”.....	64
Figura 5.14: Proceso “Ejecutar ciclo pruebas”.....	66
Figura 5.15: Proceso “Verificar ambiente prueba”.....	68
Figura 5.16: Proceso “Ejecutar conjunto de pruebas”.....	69
Figura 5.17: Proceso “Administrar proceso de pruebas” PARTE 3.....	70
Figura 5.18: Proceso “Gestionar defectos”.....	71
Figura 5.19: Proceso “Realizar auditoría CM/PPQA”.....	74
Figura 5.20: “Ciclo de defectos”.....	78
Figura 5.21: Proceso “Controlar proyecto”.....	79
Figura 5.22: Ajuste del modelo en las variables definidas.....	86
Figura 5.23: Costos por ajuste.....	87
Figura 5.24: Tabla comparativa de las métricas con y sin ajustes.....	87
Figura 5.25: Resultados del apego a la metodología.....	88

Glosario

CPU: Caso de Prueba Unitario, corresponde a la descripción de una prueba unitaria que posee un resultado y puede ser constituyente de un Caso de Prueba Integrado (CPI).

CPI: Caso de Prueba Integrado, corresponde a la descripción de aquella prueba cuyo objetivo es validar una funcionalidad o una operación del aplicativo. Está constituida por más de un CPU.

Ciclo: Es una ventana de tiempo definida en la planificación que corresponde a un grupo de iteraciones de pruebas con sus respectivos datos para cada versión liberada.

Iteración: Es una instancia de pruebas para una o un grupo de funcionalidades, interfaces y/o reportes que podría repetirse dentro de un ciclo, esto dependiendo del grado de aprobación, en caso de cumplir con todos los criterios de aceptación para la primera iteración, no es necesario seguir con las siguientes.

Pruebas con Error: Corresponde a aquellas pruebas que luego de ser ejecutadas presentan algún tipo de falla que implica que el estado de ejecución de éstas no sea óptimo.

Pruebas de Aceptación de Usuario: Son aquellas diseñadas y ejecutadas por el usuario para demostrar que la aplicación satisface los requerimientos. En inglés: User Acceptance Testing.

Pruebas de Nuevas Funcionalidades: Corresponde a aquellas pruebas que apuntan a validar aquella operatoria del sistema que no ha sido certificada en ninguna etapa de pruebas anterior.

Pruebas de Regresión: Es un "Conjunto de Pruebas" realizado para identificar "defectos" que causan "Regresión". Si el desarrollador resuelve un defecto, se deben realizar pruebas en los alrededores para verificar que no se haya impactado el resto del código, a esto se le llama regresión. En inglés: Regression Test.

Pruebas Trazadoras: Es un "Conjunto de Pruebas" corridas sobre una cierta "Release" para asegurar que es suficientemente estable para continuar con el "Ciclo" de pruebas activo. Estas pruebas usualmente son un subconjunto del conjunto completo de pruebas, preferiblemente de tipo automatizado, que involucran cada parte del sistema al menos en una forma superficial. Unas buenas pruebas trazadoras también mantienen el sistema corriendo por un lapso de tiempo suficiente como para que se manifiesten problemas "gruesos" de confiabilidad y disponibilidad. En inglés: Smoke Test o Sanity Test.

Requerimientos de Negocio: Corresponde a aquel requerimiento que es generado por el cliente y que representa alguna funcionalidad del sistema a validar. En este caso, el responsable de dicho requerimiento es el equipo de Gestión de Proyecto.

Requerimientos de Prueba Funcional: Corresponde a aquel requerimiento generado a partir de un Requerimiento de Negocio, representa el alcance de las pruebas que se realizarán para abordar la validación del requerimiento de negocio respectivo. Este tipo de requerimiento es de responsabilidad del equipo de Calidad.

1 Introducción

Cada vez más, a los departamentos de TI se les exige niveles de calidad y servicio directamente proporcionales a los niveles de calidad y servicio que se exige a las distintas líneas de negocio de las organizaciones.

Esto está provocando un afán por encontrar métodos y técnicas que permitan desarrollar y mantener software de alta calidad, así como implantar y mantener infraestructuras con un alto grado de disponibilidad.

En los últimos años han aparecido dos líneas metodológicas que están consiguiendo un gran protagonismo por su carácter práctico y por los buenos resultados obtenidos: CMMI e ITIL. La primera se centra en los procesos asociados al desarrollo de software, mientras que la segunda enfatiza en el enfoque a servicios (FSanchez, 2008).

Según revisión en la Web, no existe actualmente una Propuesta tan específica como ésta, sólo se presentan cruces a nivel completo de Metodología CMMI y Mejores Prácticas de ITIL, todas en un marco genérico.

Esta Propuesta apunta específicamente al cruce y presenta una Metodología en la implantación del proceso de Verificación y Validación de CMMI basándose en la Mejores Prácticas de ITIL.

El trabajo presenta una guía clara de cómo enfrentar un proyecto de Testing tanto desde su estimación pasando por el diseño, ejecución, concluyendo por el análisis de los resultados, cierre y revisión del cierre de proyecto en lo que respecta a su Plan de negocio inicial.

Además de presentar la Metodología, se presentan plantillas de apoyo para dicho proceso y se aplica a un proyecto real con el fin de poder refinar la Propuesta y evaluar los resultados en base al apego obtenido al aplicar dicha Metodología, resultando de este trabajo una Propuesta validada y refinada desde su concepción.

2 Objetivos

El objetivo propuesto en este proyecto es:

- Generar una propuesta Metodológica de apoyo para la implantación de Testing en la Industria de TI en base al proceso de CMMI Verificación y Validación con las Mejores Prácticas de ITIL, comprobando su aplicabilidad con un caso real.

Lo anterior se realizará a través de la persecución de los siguientes objetivos específicos:

- Proponer una Metodología de apoyo para la implantación de Testing en la Industria de TI en base al proceso de CMMI Verificación y Validación, cruzándolo con las Mejores Prácticas de ITIL.
- Generar los documentos de apoyo (plantillas) para la aplicación de la Metodología Propuesta, especificando su utilización en cada uno de los procesos que se definen en la Metodología que apoya al proceso de Verificación y Validación de CMMI.
- Generar métricas para evaluar la correcta implementación de la Metodología Propuesta.
- Aplicar la Metodología a un caso real (Proyecto Gestor de Oportunidades), desde su fase de evaluación hasta la ejecución del Testing, para así evaluar la efectividad en la aplicación de la Metodología en base a la definición de métricas para su evaluación.
- Evaluar la aplicación de la Metodología en base a las métricas propuestas.

3 Estado del Arte

Al iniciar el estado del arte es importante presentar los conceptos, así como los orígenes de la metodología y las mejores prácticas, lo que se encuentra en las siguientes secciones.

3.1 ITIL v2.0

3.1.1 Descripción (PRamirez-FDonoso, 2006)

Information Technology Infrastructure Library (ITIL), es una metodología que se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones en sus Sistemas de Información y Tecnologías de Información. Esta metodología fue desarrollada a petición del Gobierno del Reino Unido a finales de los 80 y recoge las mejores prácticas en la gestión de los Sistemas de Información. Desde entonces se ha ido extendiendo su uso en la empresa privada, tanto multinacional como PYME, llegando a ser considerado un estándar de facto para la gestión de esta área de la empresa.

En un entorno donde los periodos de disponibilidad de los servicios son cada vez más amplios, donde las exigencias del cliente son cada vez más elevadas, donde los cambios en los negocios son cada vez más rápidos, es muy importante que los Sistemas de Información estén adecuadamente organizados y alineados con la estrategia del negocio. Igualmente ITIL, ofrece toda una serie de definiciones de conceptos típicos de los Sistemas de Información para garantizar que todos sus conocedores hablen de lo mismo, reduciendo así los tiempos y riesgos por malas interpretaciones.

ITIL, es un set de documentos donde se describen los procesos requeridos para la gestión eficiente y efectiva de los Servicios de Tecnologías de Información dentro de una organización. Son un conjunto de mejores prácticas y estándares en procesos para hacer más eficiente el diseño y administración de las infraestructuras de datos dentro de la organización. Es un “marco de trabajo” (framework) para la Administración de Procesos de TI.

Basándose en el principio de mejora continua, ITIL fue madurado y de los 40 libros originales se redefinieron; se agruparon y eliminaron redundancias y actualmente se presenta en 7 libros:

- 1) Service Support
- 2) Service Delivery
- 3) Security Management
- 4) Application Management
- 5) The Infrastructure Management
- 6) Planning to Implement Service Management
- 7) The Business Perspective

Esta metodología se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones. Garantizando así los niveles de servicio establecidos entre la organización y sus clientes.

El objetivo de ITIL es diseminar las mejores prácticas en la Gestión de Servicios de Tecnologías de Información. Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios IT, al mismo tiempo de garantizar los requerimientos de la información en cuanto a seguridad, mantienen e incrementan sus niveles de fiabilidad, consistencia y calidad.

ITIL brinda una descripción detallada de un número de prácticas importantes en TI, a través de una amplia lista de verificación, tareas, procedimientos y responsabilidades que pueden adaptarse a cualquier organización. ITIL describe una aproximación sistemática y profesional a la Gestión de Servicios TI, haciendo énfasis en la importancia clave de cumplir con los requerimientos del negocio respetando los costos acordados.

3.1.2 Objetivos de ITIL (PRamirez-FDonoso, 2006)

El objetivo que persigue ITIL es diseminar las mejores prácticas en la gestión de servicios de Tecnologías de Información de forma sistemática y coherente. El planteo principal se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos.

Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios de TI, al mismo tiempo que se garantizan los requerimientos de la información en cuanto a seguridad manteniendo e incrementando sus niveles de fiabilidad, consistencia y calidad.

La filosofía ITIL adopta la gestión de procesos y considera que, para lograr los objetivos claves de la Administración de Servicios estos procesos deberían ser usados por las personas y las herramientas efectiva, eficiente y económicamente en el desarrollo de la alta calidad y la innovación de los servicios de TI alineados con los procesos de negocio.

Los estándares ITIL exigen un replanteamiento del área tecnológica y la definición de los elementos y procesos "críticos" dentro de la empresa.

3.2 CMMI

3.2.1 Historia y evolución (JPalacio, 2006)

1984 El Congreso del Gobierno Americano aprobó la creación de un organismo de investigación para el desarrollo de modelos de mejora para los problemas en el desarrollo de los sistemas de software, y evaluar la capacidad de respuesta y fiabilidad de las compañías que suministran software al Departamento de Defensa.

Creación del SEI (Instituto de Ingeniería del Software), fundado por el Departamento de Defensa Americano y la Universidad Carnegie Mellon.

1985 SEI empieza a trabajar en un marco de madurez de procesos que permita evaluar a las empresas productoras de software.

La investigación evoluciona hacia el “Modelo de Madurez de las Capacidades (CMM)”.

1991 En agosto SEI publica la versión 1.0 del Modelo de Madurez de las Capacidades para el Software (SWCMM, Capability Maturity Model for Software).

1993 SEI publica la versión 1.1 de SW-CMM.

1997 Publicación de la versión 1.2.

2000 SW-CMM fue integrado y relevado por el nuevo modelo CMMI.

2006 Se publica CMMI –DEV, V1.2, CMMI for Development, Version 1.2.

2007 Se publica CMMI-ACQ, V1.2, CMMI for Acquisition, Version 1.2, *Improving processes for acquiring better products and services*.

La propuesta metodológica que se expone a continuación, se realiza sobre CMMI-ACQ, ya que las empresas de desarrollo pueden contar con externalización de servicios a la hora de desarrollos de software.

3.2.2 Los Cinco Niveles de Madurez (Axentia, 2006)

En general, los niveles de madurez suelen explicarse en orden creciente; se tomará aquí una dirección distinta y se explicarán exactamente al revés: desde el nivel cinco al nivel uno.

Imagine por un momento que está en una organización de nivel cinco. En este tipo de organizaciones, los procesos son analizados para eliminar las causas comunes de variación, o sea, aquellas que tienen que ver con la misma naturaleza del proceso, no atribuibles a causas externas. Las variaciones en las salidas de los procesos son al azar, pero se encuentran controladas estadísticamente (se puede predecir los resultados de los procesos con cierto nivel de confiabilidad).

Para poder llegar a este nivel, la organización debió primero haber eliminado las causas especiales de variación, aquellas que tienen que ver con causas externas, como por ejemplo falta de entrenamiento del personal, problemas con las herramientas, etc. Este tipo de causas no son aleatorias: Si se examinan los resultados se podrán ver las tendencias que claramente indican que las variaciones tienen un origen concreto. En una organización de nivel cuatro, entonces, las causas especiales de variación son identificadas y eliminadas.

Para poder llegar a identificar causas de variación se necesita tener un proceso estándar: difícil sería poner bajo control estadístico un proceso que no se encuentre mínimamente formalizado.

Así se llega al nivel tres, en el cual los proyectos emplean un proceso productivo adaptado del proceso estándar de la organización. Las actividades técnicas y de gestión son realizadas de acuerdo a políticas, procesos y procedimientos formalizados en algún tipo de estándar organizacional profundamente arraigado en la cultura. La gente está entrenada y dispone de recursos para poder hacer su trabajo. También hay una infraestructura básica (personal, herramientas, etc.) para definir y mejorar el proceso productivo.

Pero para poder llegar a esta situación es necesario pasar por una etapa previa: difícilmente se puede introducir en una organización prácticas estándar relacionadas con la ingeniería del producto (análisis, diseño, etc.) si no se ofrece un contexto en donde ellas puedan ser correctamente ejecutadas. Ese es el foco del nivel dos: poner en orden las prácticas relacionadas con el manejo elemental de los proyectos.

En el nivel dos, los proyectos de la organización siguen algún tipo de proceso para realizar las actividades relacionadas con la gestión del proyecto (planificación, control), para administrar los requerimientos y las configuraciones, y para medir y analizar la calidad de los productos y el desempeño de los procesos. También hay prácticas de aseguramiento de la calidad que permiten garantizar que cada proyecto sigue sus propios estándares.

Y así se llega al nivel uno: La situación aquí es caótica. No existen procesos (no al menos en el sentido del modelo) y la *performance* de los proyectos depende profundamente de la buena voluntad y la capacidad de la gente.

3.2.3 Arquitectura del Modelo (Axentia, 2006)

En la representación por niveles (ver Figura 3.01), cada nivel de madurez contiene varias áreas de proceso, las que a su vez quedan definidas por uno o varios objetivos específicos y un objetivo genérico. Cada uno de ellos tiene vinculado un conjunto de prácticas, llamadas específicas y genéricas respectivamente.

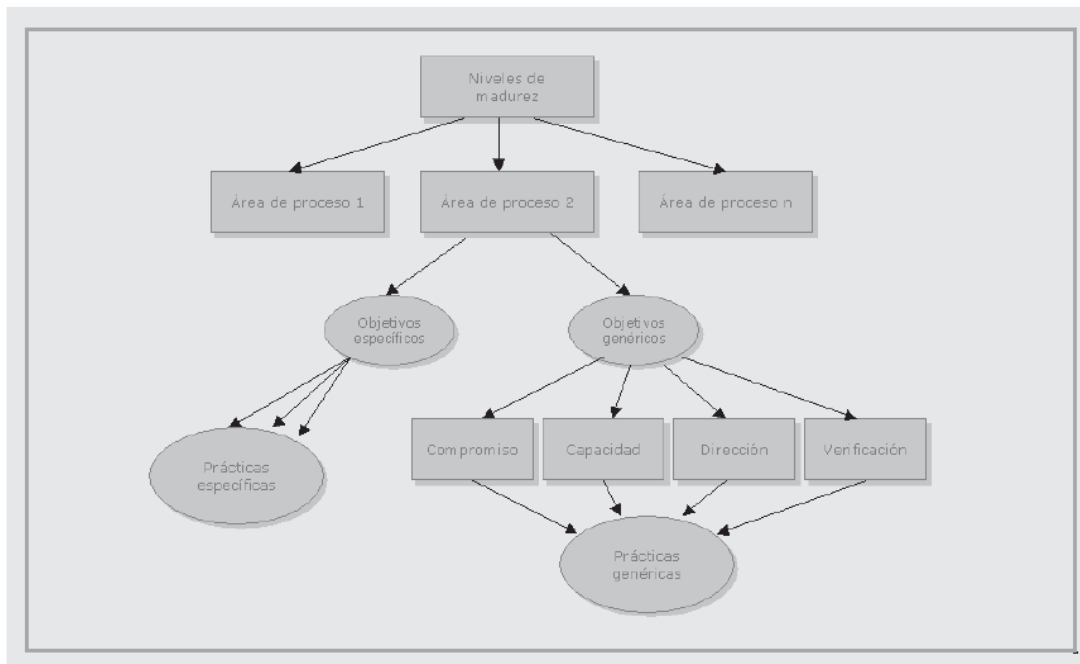


Figura 3.01: Modelo de Arquitectura CMMI

CMMI ofrece una representación sistemática y estructurada para abordar el modelo basado en un proceso de mejora que permite avanzar una etapa a la vez. El logro de cada etapa asegura que el proceso se ha establecido, y cuenta con la infraestructura adecuada que permite avanzar a la siguiente etapa.

Las áreas de proceso del modelo están organizadas en 5 niveles de madurez que representan, cada una, alguna conjetura respecto del proceso de mejora, lo que se muestra en la Figura 3.02.

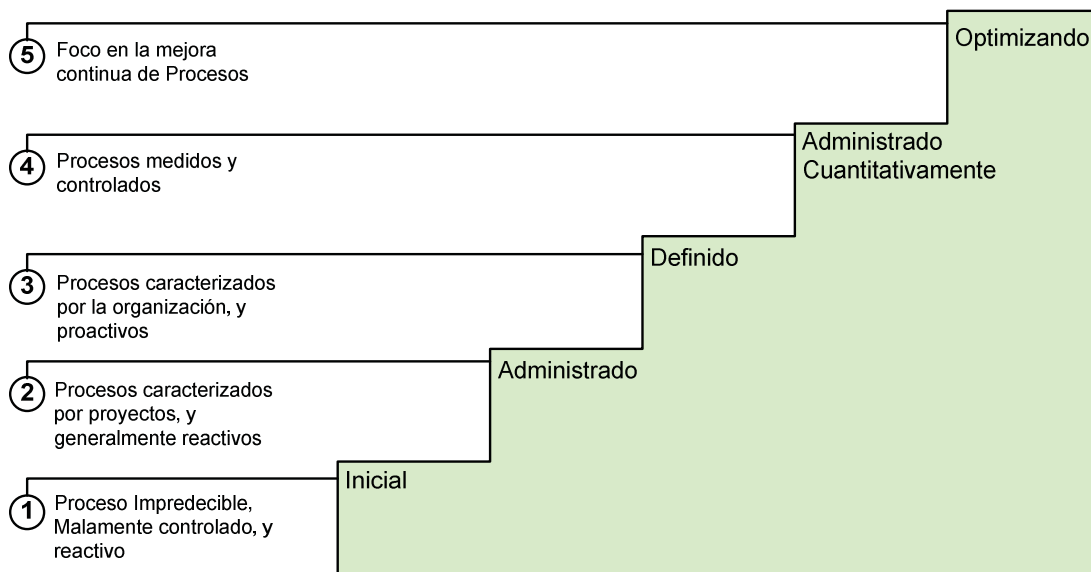


Figura 3.02: Niveles de Madurez CMMI

Esta representación sugiere un camino para la mejora de acuerdo al nivel de madurez de la organización, desde el nivel “Inicial” al nivel “Optimizando”.

El logro de cada nivel de madurez garantiza que un nuevo estado de mejora ha sentado las bases para el siguiente nivel, permitiendo así una mejora incremental.

La representación por niveles de madurez es una buena opción para mejorar un conjunto específico de procesos en cada etapa. Este orden se ha determinado a través de más de una década de investigación y experiencia.

Las áreas de proceso que contiene cada nivel de madurez se muestran en la Figura 3.03.

Nivel	Foco	Áreas de Procesos	Calidad
5 Optimizado	Proceso de Mejoramiento Continuo	Innovación organizacional y Desarrollo Análisis Causal y Resolución	Productividad
4 Administrado Cuantitativamente	Administración Cuantitativa	Ejecución proceso organizacional Administración cuantitativa del proyecto	
3 Definido	Estandarización del Proceso	Desarrollo de Requerimientos Solución Técnica Integración del producto Verificación Validación Foco en proceso organizacional Definición del proceso organizacional + IPPD Entrenamiento organizacional Administración del proyecto integrado + IPPD Administración de Riesgo Análisis de Decisión y Resolución	
2 Administrado	Administración de Proyecto Básico	Administración de Requerimientos Planificación del Proyecto Monitoreo y Control del Proyecto Administración de Acuerdo con Proveedor Medición y Análisis Aseguramiento de la Calidad de Proceso y Producto Administración de la Configuración	Riesgo
1 Inicial			Reelaboración

Figura 3.03: Representación escalonada. Por Nivele de Madurez CMMI

3.3 CMMI e ITIL (JRaggio, 2004)

Como se ha comentado anteriormente CMMI es una colección de buenas prácticas alineadas con el concepto de modelos de madurez. Debido a la herencia de SW/CMM y tal y como se ha visto en la descripción de las áreas de procesos, objetivos y las prácticas, está muy orientado hacia desarrollo y mantenimiento de software.

Si bien CMMI es la evolución e integración de un conjunto de modelos de madurez, el desarrollo y mantenimiento de software, fijan en gran medida sus líneas de actuación. En el otro punto se encuentra ITIL, modelo totalmente orientado a los servicios de mantenimiento y gestión de las operaciones.

ITIL y CMMI son dos modelos de madurez claramente distintos pero no mutuamente excluyentes. Las principales diferencias se deben a que CMMI está focalizado en la madurez de los procesos software a través de una actividad de mejora continua e ITIL se centra en el entendimiento y desarrollo de todas las áreas referentes a la infraestructura de TI, además de incorporar el ciclo de vida del hardware dentro de una organización.

3.4 Análisis del Estado del Arte

El término “nivel de calidad” es aplicable directamente a software o aplicaciones que se desarrollan, pero este término es demasiado subjetivo ya que depende de la definición y/u objetivos de cada empresa que desea implantar este “nivel de calidad”. Actualmente las empresas de TI han involucrado el concepto pero en base a criterios y/o definiciones que nacen por experiencias propias.

Al hacer una revisión en la Web, se pueden encontrar diversos estudios y propuestas de solución en lo que respecta a mejoras al “nivel de calidad” de los productos, pero la mayoría de ellos se enfoca a soluciones generales que condicionan el testing a mejoras en los procesos iniciales de construcción del producto.

La gran mayoría de los análisis y estudios existentes, se basa fundamentalmente en resaltar la sinergia que existe entre CMMI e ITIL, proponiendo distintos puntos de vista a nivel de cómo fusionar o complementar éstos en función de aprovechar ambos en la aplicación de la industria para conseguir mejores “niveles de servicio”.

En general ITIL se puede combinar con CMMI para cubrir todo, pero no aborda el desarrollo de los sistemas de gestión de la calidad. Además, no está orientado a los procesos de desarrollo de software y su uso depende en gran medida de la interpretación. Mientras CMMI es el estándar de calidad de fábrica para los procesos de desarrollo de software, ITIL para muchos es la herramienta de elección para las operaciones y la parte de infraestructura de TI, especialmente para servicios de TI (SThorn 2007).

ITIL y CMMI se aplican de mejor manera a diferentes partes de la organización de TI:

- El uso de CMMI en el desarrollo de aplicaciones
- El uso de CMMI en las TIC. Específicamente en los proyectos de infraestructura.
- Uso de ITIL en las operaciones de TI y servicios

Dentro de las propuestas existentes con un poco más de detalle a nivel de cruces entre la metodología y las mejores prácticas, se encuentra un modelo en el que las mejores prácticas de ITIL ayudan a lograr las metas requeridas para obtener un nivel particular de CMMI, por ejemplo nivel 3 (MJaramillo, 2005).

El modelo apunta a generalizar, a través de los niveles de ITIL como apoyar a CMMI para llegar al nivel 3.

1) Nivel de Servicio de Soporte:

Se lleva a cabo la administración de cada proceso planeado en la primera etapa de CMMI permitiendo así la interacción con las partes involucradas y/o afectadas, alcanzando una eficiente comunicación de usuario y proveedor,

los cuales adquieren un compromiso con el plan de trabajo ya establecido y por último llevan a cabo un mantenimiento continuo del plan.

Este nivel de ITIL también ayuda al Monitoreo y Control del Proyecto, es decir, logra el entendimiento de los procesos para tomar decisiones correctivas adecuadas.

2) Nivel de Entrega de Servicios:

Describe los procesos necesarios de entrega con respecto a la calidad y costo lo cual permite llevar a cabo una eficiente Administración de Proveedores (Etapa de CMMI), la cual analiza todo tipo de adquisición, los contratos, etc.

Por lo que participa en la Administración del Proyecto para su Desarrollo Integral (Etapa de CMMI), donde se considera el grado de involucramiento de quienes administran el proyecto.

3) Nivel de Administración de Seguridad:

Mediante este nivel se logra proteger la tecnología de información de la compañía motivo que contribuye a mejorar la “Administración de Riesgos”, que lleva a cabo CMMI, etapa que persigue evitar los peligros que impidan lograr los objetivos del proyecto.

4) Nivel de Perspectiva del Negocio:

Es indispensable para CMMI ya que mediante este nivel el gerente debe conocer el impacto de la tecnología de información y las medidas que deberá tomar para protegerse de situaciones adversas logrando así la Integración del Equipo de Trabajo, etapa de CMMI que pretende que todas las partes involucradas compartan sus habilidades y experiencias a fin de enriquecer el trabajo de la organización trayendo como resultado la calidad de cada proceso que desarrolla.

5) Nivel de Administración de la Infraestructura de TIC:

Mediante este nivel se logra tener un control de calidad con respecto a todos los servicios de comunicación que proporciona la organización facilitando así la Administración Integral de Proveedores que lleva a cabo CMMI evaluando de esta forma sus procesos y productos y haciendo los ajustes en la relación y en sus contratos evitando así comprar productos o servicios obsoletos para la organización.

6) Nivel de Administración de las Aplicaciones:

Determina las soluciones para cubrir las necesidades de los usuarios de acuerdo a los recursos y presupuesto con los que se cuenta logrando así brindar información importante a la Administración Cuantitativa del Proyecto (Etapa de CMMI) la cual considera el tiempo, mano de obra necesaria, y los costos de los procesos definidos por la organización cumpliendo de esta forma los objetivos establecidos en calidad y eficiencia.

7) Nivel de Planeación para la Administración de Servicios:

Ayuda a las organizaciones a identificar sus fortalezas y debilidades generando así que CMMI logre definir aún mejor sus procesos para un progreso en su madurez.

A partir de los análisis antes expuestos, se desprende que en su mayoría, todo apunta a refundir la metodología y sus mejores prácticas a nivel general, sin embargo es necesario contar con propuestas específicas sobre los procesos de CMMI versus ITIL, con el fin de entregar propuestas metodológicas que apoyen el desarrollo de un proceso específico sin ánimo de estandarizar el desarrollo del proceso, sino más bien orientar en cómo y cuándo aplicar las mejores prácticas en un proceso determinado en base a experiencias vividas.

Por lo anterior se desea proponer una Metodología de apoyo en la implantación del proceso de Testing de software y/o aplicativos en base a los Modelos antes mencionados.

Se analizaron los puntos de semejanza entre dichos Modelos, sin olvidar la diferencia entre sus enfoques, para así identificar los procesos de apoyo específicos al Testing. Se propone una Metodología clara y precisa para el levantamiento y mejora de los procesos existentes, con el fin de solucionar la subjetividad que existe en la implantación del proceso de pruebas, así como también en los procesos de apoyo para entrega de los servicios relacionados al Testing.

La Metodología Propuesta, puede ser aplicada tanto en departamentos de TI que pertenezcan a empresas pequeñas, que no pueden optar a obtener una certificación de los Modelos base, o bien en departamentos de TI que a pesar de contar con presupuesto para optar por estas certificaciones no cuentan con la experiencia en la implantación de servicios de Testing.

El cruce se realizó en base a las prácticas concretas y se utilizó como inicio el análisis de ITIL (PRamirez-FDonoso, 2006), ya que el enfoque a servicios debe ser la base de la Metodología que se desea generar. Sin embargo como el enfoque a nivel de procesos apunta a Verificación y Validación (Testing), se consideró sólo el detalle de los procesos que tienen paridad; es decir, Problem Management (Administración de Problemas) y Application Management (Administración de Aplicaciones).

El proyecto presenta la Metodología creada en base a los procesos de Verificación y Validación de CMMI pareado con los procesos/funciones de ITIL.

Los procesos más importantes que se incluyen en el flujo fueron descritos. El Modelo propuesto tiene inmerso los procesos de apoyo tanto de CMMI como de ITIL (JPalacio, 2006) (Axentia, 2006) (JRaggio, 2004) (OGC, 2007).

4 Solución Propuesta

4.1 Conocimiento de los modelos

Para comenzar se despliega el alcance de la solución propuesta en base a los antecedentes entregados en el Estado del Arte, con el propósito de establecer los puntos en los cuales estos Modelos son semejantes y así poder orientar el análisis y generación de la Metodología de trabajo para el proceso de Testing de Software aplicativos.

Una vez que existe claridad en la definición de los Modelos se comienza con el pareo.

4.2 Proceso de cruce de los modelos

A partir de los 7 libros existentes en ITIL se analizarán:

- Service Support
- Service Delivery
- Application Management

Se realiza un análisis de cada uno de ellos con el fin de encontrar semejanzas a nivel de prácticas concretas con CMMI. La selección de estos libros es en base a conocimiento experto por la certificación obtenida en ITIL. Como la orientación es sobre implantar Testing, se sabe de antemano que estos libros son los que apoyan de manera más directa dicha implantación.

Por su parte el Modelo CMMI cuenta con 22 procesos que incluyen en forma estricta las mejores prácticas a aplicar a nivel de desarrollo de software.

Analizando entonces a nivel de prácticas concretas se pueden mapear algunos procesos de CMMI a ITIL en lo que respecta a soporte y gestión de proyectos.

A continuación se presenta un mapeo de los procesos/funciones de los Modelos CMMI e ITIL. Estos Modelos están enfocados a diferentes áreas, por lo cual guardan escasa relación, pero existen relaciones entre ambos Modelos a nivel de prácticas concretas, aunque pueden generalizarse algunos procesos de soporte y gestión de proyectos.

El Nivel al cual pertenece el Proceso de CMMI en el Modelo está descrito en la tabla como “N+el número del nivel”. La Figura 4.01 muestra los procesos de CMMI pareados a los Procesos/Funciones de ITIL.

Procesos/Funciones ITIL	Áreas de Procesos CMMI
Configuration Management	Configuration Management (N2)
Change Management	Configuration Management (N2)
Release Management	Configuration Management (N2)
Incident Management	
Problem Management	Verification (N3), Causal Analysis & Resolution (N5)
Capacity Management, Availability Management, Continuity Management	Risk Management (N3)
Application Management	Requirements Management (N2), Requirements Development (N3), Technical Solution (N3), Product Integration (N3), Verification (N3), Validation (N3), Integrated Project Management (N3)
Project Management	Project Planning (N2), Project Monitoring and Control (N2), Supplier Agreement Management (N2)

Figura 4.01: Tabla de mapeo CMMI e ITIL

Como se muestra en la Figura 4.01, la base del análisis es ITIL, ya que el enfoque a servicios debe ser la base de la Metodología que se desea generar. Sin embargo como el enfoque a nivel de procesos apunta a Verificación y Validación (Testing), se considerarán sólo el detalle de los procesos que tienen paridad en la tabla anterior; es decir, Problem Management (Administración de Problemas) y Application Management (Administración de Aplicaciones).

No se debe perder de vista que los procesos de Verificación y Validación están soportados en el nivel 3 de CMMI, por lo cual es necesario contar con el apoyo de algunos procesos del nivel anterior (nivel 2) para poder satisfacer en forma correcta las necesidades de la implantación de los procesos de Verificación y Validación del nivel 3.

Los procesos de apoyo para Servicios (ITIL) serán los siguientes:

- Configuration Management (Administración de la Configuración)
- Change Management (Administración del Cambio)
- Release Management (Administración de Release)
- Incident Management (Administración de Incidentes)
- Project Management (Administración de Proyectos)

Los procesos de apoyo para Desarrollo de software (CMMI) serán los siguientes:

- Configuration Management (Administración de la Configuración) → N2
- Project Planning (Planificación de Proyecto) → N2
- Project Monitoring and Control (Monitoreo y Control de Proyecto) → N2
- Supplier Agreement Management (Administración de Proveedores) → N2
- Requirements Management (Administración de Requerimientos) → N2
- Requirement Development (Requerimientos de Desarrollo) → N3
- Technical Solution (Solución Técnica) → N3
- Product Integration (Integración de Producto) → N3
- Integrated Project Management (Administración de Integración de Proyecto) → N3
- Risk Management (Administración de Riesgos) → N3

Es importante mencionar que cuando se habla de procesos de apoyo se entiende que al momento de detallar la Metodología a generar, se nombraran las mejores prácticas y los procesos listados, con el fin de que se tenga claridad de cómo ellos interactúan en el proceso de implantación; sin embargo no se entregará detalle de cómo se deben implantar, a diferencia de los procesos específicos de Verificación y Validación.

La Metodología obtenida es presentada en la Implementación de la misma.

La Figura 4.02 presenta la estructura definida para el Modelo Propuesto y se incluyen los proceso tanto del cruce como los de apoyo de CMMI e ITIL. Cabe destacar que el Modelo Propuesto fue generado en Enterprise Architect - UML CASE Tool - Desktop, Professional and Corporate editions, Version 7.0.

La simbología de los colores utilizados en el Modelo se encuentra en la Figura 4.02 en la parte superior derecha.

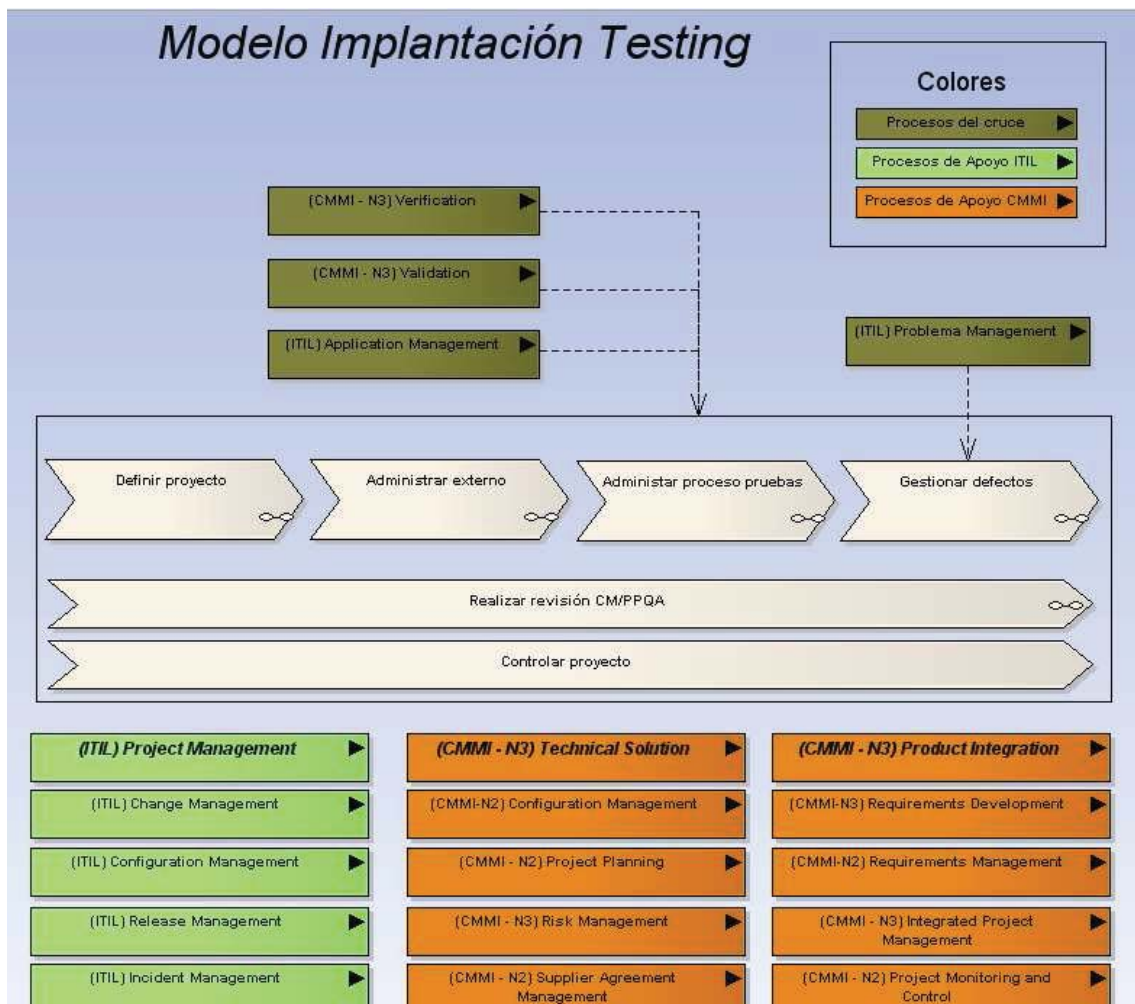


Figura 4.02: Metodología propuesta

En la Metodología se exponen las prácticas específicas y se detalla el proceso de cómo implantar la Metodología. El texto de la Metodología fue apoyado con un Diagrama de Flujo, el cual indica paso a paso como implantar el proceso de Testing en una empresa cualquiera.

A partir de la Metodología Propuesta se presenta la evaluación de su aplicación a un caso real, definiendo como punto de partida las métricas que se considerarán en dicha evaluación.

Al aplicar la Metodología al caso real, se refinaron algunas de las plantillas inicialmente generadas como apoyo.

El proceso de implementación consiguió como objetivo refinar y generalizar aún más los documentos de apoyo, ya que se amplió el espectro inicialmente pensado para tales documentos.

4.3 Definición de métricas para la evaluación final

A continuación se presentan los conceptos utilizados para la definición de métricas:

- Existen cuatro medidas para cuantificar: Caracterizar, Evaluar, Predecir y Mejorar.
 - 1) Medida: Valor asignado a un atributo de una entidad mediante una medición. Por ejemplo: 35.000 líneas de código.
 - 2) Medición: Es el acto de determinar una medida. Por ejemplo: Silvana será la encargada de medir las LDC de cada módulo del sistema.
 - 3) Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Incluye el método de medición. Por ejemplo: La productividad de este proyecto fue de 500 líneas/personas-mes (LDC/pm).
 - 4) Indicador: Es una métrica o combinación de métricas que proporcionan una visión profunda del proceso de software. Por ejemplo: la productividad media de nuestra empresa es de 500 (LDC/pm).

Las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo y el producto para intentar aumentar su calidad (LLeón-GPuglla, 2008).

- Por otro lado es importante saber que muchas veces el gerente de proyecto, se encuentra en la disyuntiva de dar o no el peso correcto a cada una de las variables que afectan el desarrollo de un proyecto en particular.

Existen restricciones tradicionales, como en cualquier empresa, los proyectos necesitan ser ejecutados y entregados bajo ciertas restricciones. Tradicionalmente, estas restricciones han sido alcance, tiempo y costo. Esto también se conoce como el Triángulo de la Gestión de Proyectos, donde cada lado representa una restricción. Un lado del triángulo no puede ser modificado sin impactar a los otros. Un refinamiento posterior de las restricciones separa la calidad del producto del alcance, y hace de la **calidad** una cuarta restricción.

- La restricción de tiempo se refiere a la cantidad de tiempo disponible para completar un proyecto.
- La restricción de costo se refiere a la cantidad presupuestada para el proyecto.
- La restricción de alcance se refiere a lo que se debe hacer para producir el resultado final del proyecto.

Estas tres restricciones son frecuentemente competidoras entre ellas: incrementar el alcance típicamente aumenta el tiempo y el costo, una restricción fuerte de tiempo puede significar un incremento en costos y una reducción en los alcances, así como también un presupuesto limitado puede traducirse en un incremento en tiempo y una reducción de los alcances.

La disciplina de la gestión de proyectos consiste en proporcionar las herramientas y técnicas que permiten al equipo de proyecto (no solamente al gerente del proyecto) organizar su trabajo para cumplir con todas esas restricciones.

4.3.1 Tiempo

El tiempo se descompone para propósitos analíticos en el tiempo requerido para completar los componentes del proyecto que es, a su vez, descompuesto en el tiempo requerido para completar cada tarea que contribuye a la finalización de cada componente. Cuando se realizan tareas utilizando gestión de proyectos, es importante partir el trabajo en pedazos menores para que sean fáciles de seguir.

4.3.2 Costo

El costo de desarrollar un proyecto depende de múltiples variables incluyendo costos de mano de obra, costos de materiales, administración de riesgo, infraestructura (edificios, máquinas, etc.), equipo y utilidades. Cuando se contrata a un consultor independiente para un proyecto, el costo típicamente será determinado por la tarifa de la empresa consultora multiplicada por un estimado del avance del proyecto.

4.3.3 Alcance

Requerimientos especificados para el resultado final. La definición global de lo que se supone que el proyecto debe alcanzar y una descripción específica de lo que el resultado final debe ser o debe realizar. Un componente principal del alcance es la *calidad* del producto final. La cantidad de tiempo dedicado a las tareas individuales determina la *calidad* global del proyecto. Algunas tareas pueden requerir una cantidad dada de tiempo para ser completadas adecuadamente, pero con más tiempo podrían ser completadas excepcionalmente. A lo largo de un proyecto grande, la *calidad* puede tener un impacto muy significativo en el tiempo y en el costo (o viceversa) (PMBOK, 2004).

La Figura 4.03 muestra las restricciones anteriormente descritas. Éstas se pueden representar en un triángulo, el cual aumenta o disminuye sus ángulos dependiendo de la importancia que se le aplique a cada uno de sus vértices (restricciones). Lo anterior refleja que en el caso que se le asigne mayor o menor importancia a uno de sus ángulos, éstos impactarán en la importancia de los demás vértices.

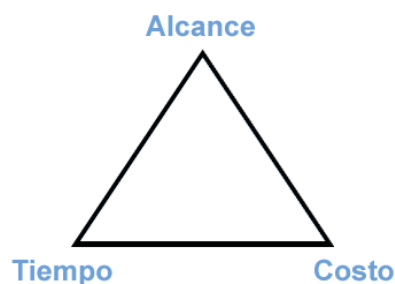


Figura 4.03: Triángulo de restricciones

A partir de los puntos antes descritos; Métricas y Triángulo de restricciones, se definen las siguientes métricas para la medición del éxito en la aplicación de la Metodología Propuesta a nivel de proyecto de Testing:

- El tiempo siempre será la variable menos movable con respecto a las restricciones, ya que todo proyecto tiene como límite una fecha, ya sea por necesidades del cliente o por imposición política (frecuente en empresas públicas).
- Una vez que el tiempo está establecido es importante saber el presupuesto con el que el cliente cuenta, ya que muchas veces es la segunda variable impuesta a nivel de requerimientos. El presupuesto puede estar dado mensual, semanal o incluso diario, dependiendo del tipo de proyecto.
- Finalmente queda el alcance, en el cual el cliente indica su óptimo con respecto al requerimiento en general. A parte de los costos que involucra el desarrollo del proyecto (específicamente lo que se evaluará en la Metodología), se debe considerar la calidad asociada al proceso de desarrollo, el cual también debe ser considerado en esta restricción.

Para el Testing de un aplicativo, teóricamente es prudente considerar a lo menos el 30% del Universo de pruebas posibles; es decir, si se estiman 100 pruebas de negocio para el aplicativo, lo prudente es testear al menos 30 de dichas combinatorias de pruebas de negocio. Más adelante se explicará la forma de estimar el Universo para luego indicar el cálculo del 30% en base a pruebas de negocio base y específicas.

Del 30% que será testeado, se procederá a establecer en el Plan de Pruebas, las políticas para la aceptación del aplicativo; es decir, se indicarán los porcentajes de aceptación para considerar efectivo el esfuerzo aplicado en el Testing.

Por experiencias en proyectos de Testing, lo mínimo recomendable como política de aceptación del 30% testeado será un 80% de aceptación SIN incidentes invalidantes. Muchas veces este porcentaje no es alcanzado, por malas prácticas en el desarrollo del software, así como también por la inexperiencia en el proceso de testeo.

Lo anterior indica que en el ejemplo de las 100 pruebas como Universo estimado, 24 pruebas (de las 30 seleccionadas) deben ser exitosas y no deben existir pruebas con invalidantes al momento de la ejecución. Por lo anteriormente expuesto, entonces se declaran 3 métricas para la evaluación de la implantación de la métrica, las cuales se listan en la Figura 4.04.

N°	Medición	Medida	Métrica
1	Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto.	HH	Esfuerzo Estimado v/s Esfuerzo Real
2	Se medirá el cumplimiento del costo establecido para el Testing del proyecto.	UF	Costo Estimado v/s Costo Real
3	Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.	%	% Establecido v/s % Obtenido

Figura 4.04: Métricas definidas para validación de la implantación de metodología propuesta

Las métricas obtenidas deberán tender en su mayoría a cero una vez calculado su diferencial. Luego de calculado su diferencial, éstos deberán ser analizados por el administrador del proyecto, quien tendrá la noción del proyecto general con respecto a los diferenciales obtenidos. Lo anterior, siempre con la mirada de "un traje de novia con excelencia en calidad al costo estimado, no sirve de nada si es entregado después de la fecha establecida".

Para medir la aplicación de la Metodología Propuesta se sugiere la siguiente métrica:

- 1) Porcentaje de Apego a la Metodología: Esta métrica tiene relación con identificar en que porcentaje se ha aplicado la Metodología Propuesta al concluir el Proyecto de Testing.

Las medidas utilizadas en el cálculo de la métrica se desprenden de la expertise obtenida en el desarrollo de la Propuesta Metodológica, así como también en la aplicación consecutiva de éstas en proyectos diversos para mejorarla.

La Figura 4.05 se refiere a la importancia de la plantilla diseñada para el apoyo al proceso de implantación. Esta importancia está dada por la experiencia en el proceso, pensando siempre en el cumplimiento del objetivo propuesto: apego a la Metodología.

Importancia Metodológica	
Muy Importante	4
Importante	3
Media	2
Baja	1

Figura 4.05: Métrica "Importancia Metodológica"

La Figura 4.06 entrega la relación de la planilla al proceso de implantación bajo la visión de Modelo diseñado del proceso de Testing; es decir, si la planilla apoya al proceso de CMMI, ITIL o es parte del apoyo a un proceso del cruce realizado entre ambos Modelos.

Propuesta Metodológica	
Cruce	5
CMMI	3
ITIL	2

Figura 4.06: Métrica “Propuesta Metodológica”

Una vez definidas las medidas se define la forma en la cual se calculará el apego a la Metodología. A continuación (Figura 4.07) se presenta la tabla con los campos ya asignados para el cálculo de la métrica de apego.

Columna A	Columna B	Columna C		Columna D	Columna E
Estructura	Plantilla	Propuesta Metodológica		Importancia en Metodología	Apego
0. Entradas	Informe de Diseño proveedor(Extracción).doc	CMMI	3	4	12
	PlanificaciónSeguimientoSemanal.mpp	CMMI	3	4	12
	Visión y Alcance.doc	CMMI	3	2	6
1. Análisis	Matriz_de_Evaluacion_de_Propuestas.xls	CMMI	3	2	6
	Estimación Modelo de Servicios.xls	Cruce	5	4	20
	LB_Requerimientos_Calidad.xls	ITIL	2	3	6
2. Diseño	Definición CPL.xls	ITIL	2	4	8
	TF Diseño.xls	ITIL	2	3	6
	Plan_de_CM.doc	CMMI	3	1	3
	Plan_de_Pruebas.doc	CMMI	3	3	9
3. Ejecución	TF Ejecución.xls	Cruce	5	3	15
4. Gestionar Defectos	TF Entidad-Defecto.xls	Cruce	5	3	15
5. Informes	Plan de Negocio.xls	Cruce	5	4	20
	Cierre Proyecto.xlsx	Cruce	5	2	10
	RESULTADO TESTING Ciclo.doc	Cruce	5	2	10

Figura 4.07: Tabla “Cálculo de métricas”

- 1) La Columna A se refiere a la estructura dada para organizar las planilla diseñadas. Concuerdan con el ciclo del proyecto que se testea.
- 2) La Columna B identifica cada una de las plantillas diseñadas para el proceso de implantación del Testing.
- 3) La Columna C clasifica cada planilla diseñada según la medida explicada anteriormente (Propuesta Metodológica).
- 4) La Columna D clasifica cada planilla diseñada según la medida explicada anteriormente (Importancia en Metodología).
- 5) La Columna E es la multiplicación de la Propuesta Metodológica (Columna C) por la Importancia Metodológica (Columna D).
- 6) En la Figura 4.08 se explica la segunda parte de la Métrica en base al apego de la Metodología Propuesta, la cual considera principalmente la aplicación de las métricas propuestas (Figura 4.05 y 4.06).

Columna B	Columna F	Columna G	Columna H	Columna I	Columna J	Columna K	Columna L
Plantilla	Aplicada	Apego por Aplicada	Distribución Importancia	Refinada	Multiplica (Distribución * Refinado)	Apego por Fase	Porcentaje por Estructura
Informe de Diseño proveedor(Extracción).doc							0. Entradas
PlanificacionSeguimientoSemanal.mpp							
Visión y Alcance.doc							
Matriz_de_Evaluacion_de_Propuestas.xls							1. Análisis
Estimación Modelo de Servicios.xls							
LB_Requerimientos_Calidad.xls							
Definición CPI.xls							2. Diseño
TF Diseño.xls							
Plan_de_CM.doc							
Plan_de_Pruebas.doc							3. Ejecución
TF Ejecución.xls							
TF Entidad-Defecto.xls							4. Gestionar Defectos
Plan de Negocio.xls							5. Informes
Cierre Proyecto.xlsx							
RESULTADO TESTING Ciclo.doc							

Figura 4.08: Tabla “Apego de la metodología propuesta”

- 1) La Columna F debe ser llenada con 1 si la planilla fue aplicada en el proceso de Testing o con uno 0 en el caso de que no se haya aplicado.
- 2) La Columna G multiplica la Columna E (Apego) por la Columna F (Aplicada) con el fin de obtener el apego real en la Metodología aplicada por cada una de las plantillas.
- 3) La Columna H es la distribución de la importancia que se calcula considerando Columna G (Apego por Aplicada) dividido por la sumatoria de cada plantilla que compone una estructura (Columna A).
- 4) La Columna I se debe llenar con un 1 para representar que SI se ajustará la planilla propuesta, al momento de su aplicación en lo que respecta a mejoras para su posterior utilización, o bien con un 0,9 en el caso que NO haya sufrido modificaciones la planilla propuesta al momento de su aplicación en lo que respecta a mejoras para su posterior utilización.
- 5) La Columna J es la multiplicación de Columna H (Distribución Importancia) por Columna I (Refinada) para así obtener la medida final en la siguiente columna.
- 6) La Columna K es la sumatoria de toda la columna J de cada plantilla existente por Estructura (Columna A). Esta medición entrega el apego a la Metodología por Estructura definida.
- 7) La Columna L es el recordatorio de la composición por Estructura.

El porcentaje de Apego a la Metodología se calcula sumando todas las medidas obtenidas por estructura (Columna K) dividida por 6 (cantidad de estructuras: Entradas, Análisis, Diseño, Ejecución, Gestionar Defectos e Informes).

Es importante mencionar que se realizó el ejercicio considerando el mínimo recomendable en la utilización de las plantillas en la implementación de la Metodología, llegando a la conclusión de que al aplicar la métrica de Apego a la Metodología se puede observar que el escenario peor es a lo menos un 80% de apego a la Metodología como porcentaje general, ya que sin este porcentaje no se estaría consiguiendo el objetivo de aportar en el proceso de implementación del Testing.

5 Implantación de la metodología propuesta y aplicación de plantillas de apoyo

A partir de la Metodología Propuesta se definen las plantillas de apoyo por proceso en el siguiente orden, considerando siempre el apoyo de los procesos del cruce, procesos de apoyo ITIL y procesos de apoyo CMMI. Es importante mencionar que se creó un nuevo proceso de Control de Proyecto que permitirá ir analizando las métricas propuestas anteriormente.

Estas plantillas apoyan el proceso de Testing (Verificación y Validación).

1. Definir Proyecto
2. Administrar Externo
3. Administrar Proceso Pruebas
4. Gestionar Defectos
5. Realizar Revisión CM/PPQA
6. Controlar Proyecto

Las planillas de apoyo se verán reflejadas como documentos asociados al proceso específico según se muestra en la Figura 5.01.

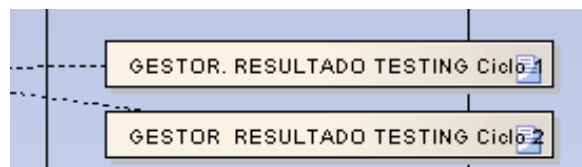


Figura 5.01: Identificación planillas de apoyo

Es importante indicar que el documento contiene el nombre de la planilla de apoyo que se encuentra en los anexos. Además la línea punteada relaciona el documento al proceso que lo utiliza.

5.1 Proceso para aplicar la metodología

La generación de esta Metodología nace de la necesidad y experiencia de establecer un método en el trabajo de la estimación y aplicación del Testing Funcional y No Funcional en un Área de Calidad de una EMPRESA.

Históricamente se sabe que un proyecto nunca es igual a otro, siempre existen diferencias significativas en el desarrollo de éstos, por lo que para la aplicación de la Metodología inicialmente se consideró incorporar como prerrequisitos mínimos los entregables que aplican como entrega por parte del equipo de proyecto. Por lo

anteriormente indicado se hacía cada vez más necesario establecer patrones de semejanza en la aplicación de un método para levantar un “proyecto” de Testing sobre un proyecto de desarrollo de software.

A medida que se analiza la generación de la Metodología se comienzan a exigir como Área, entregables mínimos a considerar en un proyecto de desarrollo; como por ejemplo, documentos de especificación de requerimientos, documentos de diseño, y documentos formales de aceptación de los documentos antes indicados, con el fin de establecer líneas base claras a la hora de estimar y planificar el proyecto de Testing.

Una vez generada la Metodología, se escogió un proyecto real, de una de las áreas con las cuales se llevaban muchos trabajos históricos, lo que ayudó, en gran medida, a contar con los documentos mínimos necesarios.

Al aplicar esta Metodología, se podrá apreciar que muchas veces los documentos mínimos no son considerados formalmente en un proyecto de desarrollo. Esto es una muy mala práctica, ya que no se puede establecer un Método sobre un proyecto que viene mal “engendrado”. A pesar de lo anteriormente expuesto, las plantillas generadas servirán de apoyo para las etapas de Requisitos, Análisis y Diseño de un proyecto de desarrollo de software.

El proceso de aplicación comienza con la exigencia de los documentos mínimos y la definición (estimación y planificación) del proyecto de Testing que se apegará a las condiciones del proyecto de desarrollo.

Se establecieron Métricas que se consideran evaluar al término de la aplicación de la Metodología, con el fin de mostrar el real aporte de su aplicación.

Las Plantillas aplicadas, son presentadas en el ANEXO 1. A medida que se utilizan en la Metodología serán referenciadas. Algunas de ellas se detallarán, con el fin de describir las transformaciones que han sufrido en el proceso de aplicación al proyecto real.

Es importante mencionar que además se incorporaron algunos documentos más que no estaban considerados inicialmente como plantillas, pero que en el proceso se pudieron percibir como gran apoyo.

Se recuerda que los procesos en azul tienen directa relación con la posible utilización de la Metodología para procesos apoyados con herramientas existentes en el mercado; en este caso, específicamente con Mercury de HP.

Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.1 Proceso “Definir proyecto”

En la Figura 5.02 se puede ver que no existen plantillas de apoyo al proceso. En su completitud este proceso apoya en la generación de un nuevo proyecto en el caso de que se utilice una herramienta de Prueba.

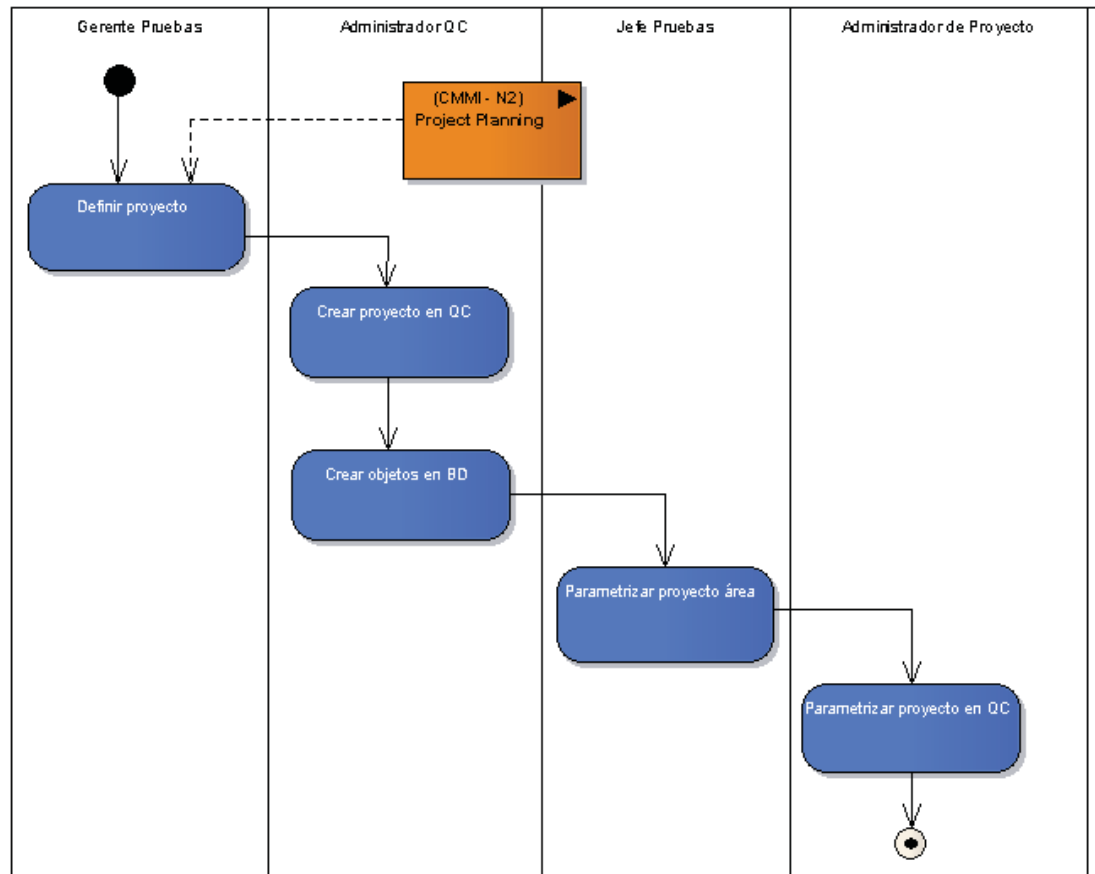


Figura 5.02: Proceso “Definir proyecto”

Proceso	Definir proyecto
Descripción	<p>Una vez que ha sido aprobado por el cliente la propuesta de servicio:</p> <ul style="list-style-type: none"> - Se define el nombre del proyecto y a qué línea de negocios pertenece. - Definición de las áreas de calidad involucradas y los Jefe de Área/Prueba involucrados. - Se debe definir las contrapartes válidas (Jefes de Proyectos) a las cuales se les derivarán los defectos y si accesarán QC. - Se define la política de respaldo y Planes de Pruebas - Checklists serán versionados en Subversion (herramienta de versionamiento) indicando el repositorio. - Se definen los aplicativos y las versiones para cada uno. - Se entrega la información anterior al "Administrador QC" para crear el proyecto en QC.

Proceso	Crear proyecto en QC
Descripción	<p>Se realizan las tareas asociadas a la creación del proyecto en QC, se ajusta una copia de la plantilla Domain: Base y Project: Proyecto basándose en la especificación entregada por el Jefe de Pruebas.</p> <ul style="list-style-type: none"> - Conectarse como Administrador de QC a QC Project Administration - Crear un proyecto dentro del dominio correspondiente. (QC Project Administration- Site Projects). Elegir la opción 'Create a project by copying data from an existing project' - Seleccionar el proyecto plantilla mencionado Domain: Base y Project: Proyecto - Chequear Customization, Requirements, Tests, Test Sets, Public Favorite Views. - Especificar el servidor de BD donde residirá el proyecto. - Asignar el Administrador del Proyecto (Se asume un mismo Administrador para todos los proyectos) - Activar el proyecto (Chequear Activate project) - Dado que los proyectos se guardarán distribuidamente en la BD y en directorios del S.O. no chequear 'Store project's repository in the database' - Dado que no se usará versionamiento, debido a que éste sólo cubre las pruebas y no el resto de las entidades, no chequear 'Create a Version Control database' - Chequear por conectividad a la BD usando el botón Ping. - Crear los usuarios que aún no existen y hayan sido especificados en la Definición del Proyecto .(QC Project Administration - Site Users) - Conectarse a Quality Center como el Administrador de Proyecto al nuevo proyecto. - Crear los aplicativos y releases. Renombrar la carpeta raíz Releases con el nombre del proyecto. - Informar al administrador de la BD, que se generó una nueva BD, con el fin de actualizar el job de respaldo.

Proceso	Crear objetos en BD
Descripción	<p>Crear los objetos cuyos scripts están en el tab Files usando el usuario td, previa especificación de la BD en ellos.</p> <p>Crear sólo las funciones get_domainname y get_projectname pues éstas ya fueron creadas en la BD de Administración de QC.</p>

Proceso	Parametrizar proyecto área
Descripción	<p>En esta tarea se especifica las parametrizaciones propias de cada Área de Calidad:</p> <ul style="list-style-type: none"> - Definir usuarios del área que participarán en el proyecto. - Se identifica si el proyecto necesita customizaciones especiales. Estas pueden implicar: <ol style="list-style-type: none"> (1) Creación de grupos usuarios especiales (2) Creación de nuevos campos para las entidades (3) Creación o mantención de Listas de Lookup.

Proceso	Parametrizar proyecto en QC
Descripción	<p>Una vez que el proyecto ha sido creado se realiza su parametrización:</p> <ul style="list-style-type: none"> - Crear los grupos de usuarios necesarios (QC Project Customization - Groups). - Asignar los usuarios al proyecto (QC Site Administration - Site Users - User Projects) - Si QC se versiona en Subversion y no se está usando sólo la cuenta de Subversion Default, agregar los mappings entre usuarios QC y Subversion. Nótese que si el versionamiento ya estaba activo, debe ser desacoplado, cambiado y vuelto a acoplar. - Realizar las customizaciones (QC Project Customization)

5.1.2 Proceso “Administrar externo”

El proceso “Administrar Externo” se basa fundamentalmente en la búsqueda de externos para el apoyo del Testing. En el caso que sea necesaria la utilización de externos es importante tener realizada la estimación del proyecto es lo que respecta al proceso de Testing.

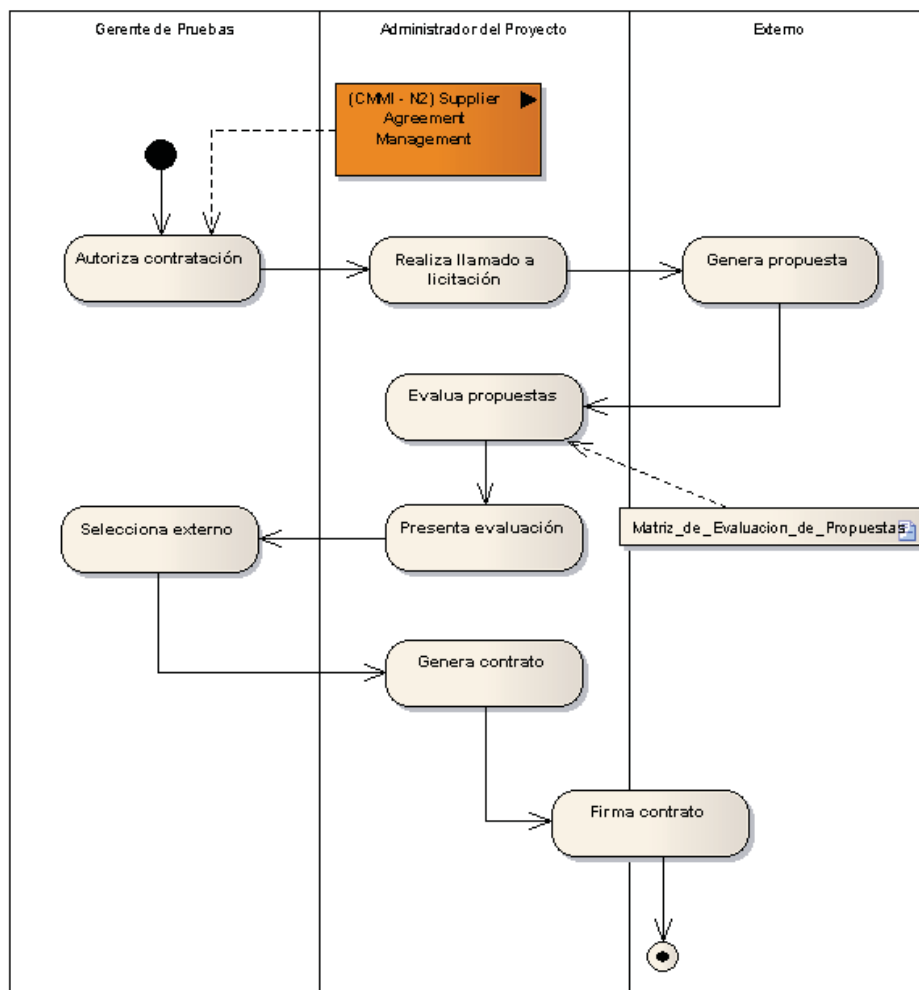


Figura 5.03: Proceso “Administrar externo”

Proceso	Autoriza contratación
Descripción	Una vez que el equipo de Gestión de Proyecto autoriza la utilización de recursos externos. El equipo de proyecto debe tener presente que en estos casos el costo aumenta, ya que contratar una “buena” empresa de Testing involucra experiencia, eficacia y eficiencia en el proceso.

Proceso	Realiza llamado a licitación
Descripción	El equipo de Proyecto entrega la potestad al equipo de Testing en el llamado a licitación, pudiendo ésta ser de carácter público o reservado, dependiendo de la estrategia de empresa por la que se rija el equipo de Testing. Es importante mencionar que un buen llamado a licitación consiste en precisar el servicio que se solicita, tanto en tiempo, costo, calidad y por sobre todo en lo que respecta a los rendimientos tanto del diseño de los casos como en la ejecución de ellos. Otro punto importante a considerar es explicar en detalle la definición de los conceptos base para poder realizar el cálculo del Testing, que en este caso sería la definición de Caso de Prueba Integrado, Casos de Prueba Unitarios o Casos de Negocio. Las definiciones antes mencionadas están especificadas en el Glosario de este documento.

Proceso	Genera propuesta
Descripción	En base al llamado de licitación, el externo genera su propuesta de servicio, en la cual debe especificar claramente el cumplimiento de plazos, restricciones del servicio en caso de necesidades de horas extras y el costo asociado.

Proceso	Evalúa propuestas (*)
Descripción	Una vez que los proveedores de servicio entregan sus propuestas, el equipo de Testing debe comenzar con la evaluación. Es en este punto donde aparece nuestra primera Plantilla de apoyo (ver ANEXO 1 “Matriz_de_Evaluación_Propuestas”).

Proceso	Presentar evaluación
Descripción	Una vez aplicada la matriz de evaluación, es necesario presentar los resultados al equipo de Proyecto y realizar las recomendaciones respecto a dicha evaluación.

Proceso	Selecciona externo
Descripción	El equipo de Proyecto entonces está en condiciones de realizar la selección del externo para que participe en el proceso de Testing, pudiendo ser ésta el diseño y/o ejecución de las Pruebas.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

Proceso	Generar contrato
Descripción	<p>El equipo de Testing está en condiciones de generar el contrato de servicio con el externo.</p> <p>Es importante recomendar que dicho contrato esté orientado a entregables de trabajo relacionados con el proceso del Testing en una primera vista, pero que en segunda línea se especifique claramente la cantidad de HH relacionadas al servicio.</p> <p>Se debe tener especial cuidado en las excepciones que se acuerden y en el aviso y satisfacción del servicio en lo que respecta a horas de trabajo extraordinario.</p>
Proceso	Firma contrato
Descripción	<p>La firma del contrato debe ser entre equipo de Testing y externo, ya que las exigencias con respecto a respuesta en la entrega de servicio son de responsabilidad del equipo de Testing.</p>

5.1.3 Proceso “Administrar proceso de pruebas” PARTE 1

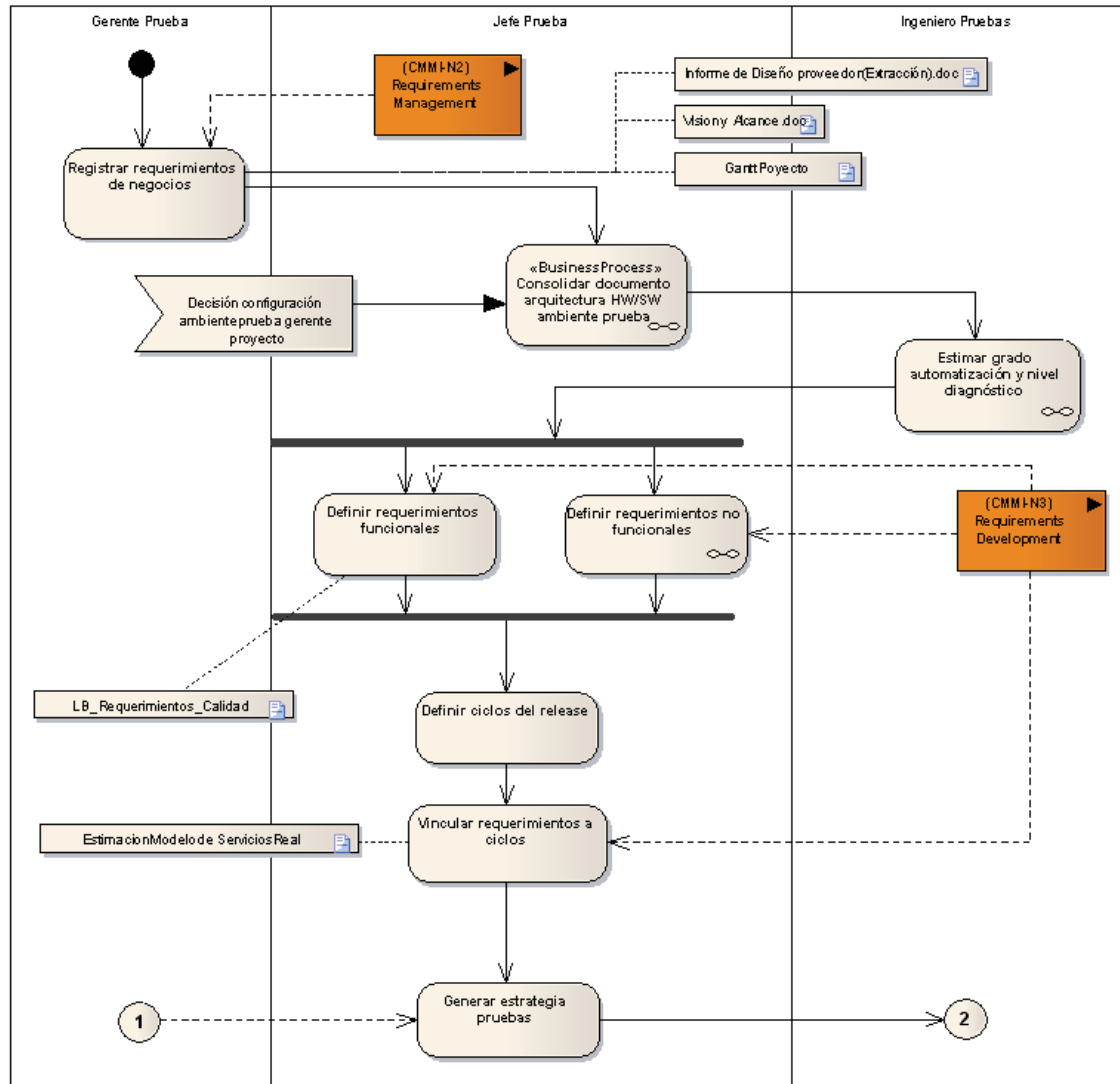


Figura 5.04: Proceso “Administrar proceso de pruebas” PARTE 1

Proceso	Registrar requerimientos de negocios (*)
Descripción	<p>El primer documento que debe aprobar el equipo de proyecto al desarrollador del aplicativo a testear es el de Visión y Alcance, el cual debe especificar claramente lo que se va a desarrollar. Siempre en términos generales en lo que respecta al aplicativo. En la plantilla de apoyo ANEXO 1 “Visión y Alcance” se explica el contenido esperado como mínimo en dicho documento.</p> <p>Luego que el documento de Visión y Alcance está acordado y aprobado por las partes es hora de que el desarrollador comience con el análisis para continuar con el diseño, otro de los input importantes a la hora de analizar qué testear.</p> <p>El proveedor de desarrollo debe entregar al equipo de proyecto, el documento de diseño del aplicativo que se va a testear. A partir de este documento el equipo de Testing debe ser capaz de entender y reflejar en la planilla de requerimientos, todos los requisitos que el aplicativo debe contener.</p> <p>El Informe de diseño debe contener todo lo indicado en la plantilla de apoyo ANEXO 1 “Informe Diseño del Proveedor”.</p> <p>Una vez aprobado el documento de Diseño, el Proveedor ya está en condiciones de declarar tiempos claros y precisos en una Carta Gantt, sobre la cual el equipo de proyecto y el equipo de Testing deben comenzar a analizar los instantes en los cuales el equipo de Testing podrá intervenir, así como también las posibles restricciones de tiempo y costo, con el fin de poder dilucidar cuál será el alcance de la claridad del Testing.</p> <p>La plantilla siguiente muestra una planificación tipo que puede ser utilizada como referencia para analizar las intervenciones del equipo de Testing. Ver ANEXO 1 “Planificación_Seguimiento_Semanal”.</p> <p>A nivel de equipo de Testing, a partir de los 3 documentos antes mencionados, debe ser capaz de definir los requerimientos del aplicativo, por lo anterior se deben realizar los siguientes pasos:</p> <p>Definir el requerimiento como carpeta, especificando:</p> <ol style="list-style-type: none"> 1. Nombre 2. Tipo de Requerimiento como de Negocios. 3. Descripción 4. Autor 5. Prioridad en la implementación:

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<p>1-Baja 2-Media 3-Alta 4-Muy Alta</p> <p>6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)</p> <p>Nota: Es posible agregar Attachments a los Requerimientos también.</p> <p>En el caso de Pruebas No Funcionales es recomendable que los requerimientos de negocios tengan la forma de SLAs estándares.</p> <p>NOTA: LOS REQUERIMIENTOS DE NEGOCIO SON DE ALTO NIVEL, Y SU ANALISIS PUEDE INVOLUCRAR SU DESCOMPOSICION EN REQUERIMIENTOS FUNCIONALES, DESEMPEÑO, SEGURIDAD, ETC.</p>
--	--

Proceso	Consolidar documento arquitectura HW/SW ambiente prueba
Descripción	<p>Si bien el área de calidad no es el encargado de generar este documento en forma íntegra si debiera consolidarlo para sí, previa ejecución de:</p> <ul style="list-style-type: none"> - Recolectar las especificaciones de Arquitectura que generen otras áreas de la Empresa, el Cliente o Terceros. - Generar la especificación del dispositivo GUI, su cantidad, ubicación física y conectividad. <p>El documento debe contener todas las componentes detalladas en el Diagrama "Ambiente_Prueba". Dado que los ambientes para Pruebas Funcionales y No Funcionales son potencialmente distintos se recomienda elaborar un documento para cada uno.</p>

Proceso	Estimar grado automatización y nivel diagnóstico
Descripción	<p>En esta tarea se estima que porcentajes de las Pruebas serán realizadas automática y manualmente. Dicha estimación se realiza basándose en:</p> <ul style="list-style-type: none"> - Diseños Preliminares del Sistema. Por ejemplo si el "Dispositivo GUI" es una PALM y no hay emulador confiable, las Pruebas funcionales deberán ser realizadas manualmente, pero puede que de todos modos sea factible realizar las Pruebas no funcionales. - Pruebas sobre el Aplicativo en el caso que esté disponible usando QTP y LR para las Pruebas funcionales y no funcionales respectivamente. <p>En el caso No Funcional si no es factible realizar las Pruebas automáticamente, probablemente la opción manual sea demasiado cara.</p>

Proceso	Definir requerimientos funcionales (*)
Descripción	<p>La planilla de apoyo "LB_Requerimientos_Calidad" en ANEXO 1 entrega el detalle del llenado para obtener el Universo estimado.</p> <p>Esta plantilla debe estar aprobada por el equipo de proyecto y debe sumar todos los casos de Prueba relacionados a los requerimientos con estado Declarado.</p> <p>Se definen los requerimientos funcionales detallando los atributos mencionados a continuación, si es necesario se crean carpetas que agrupen los requerimientos por temas.</p> <ol style="list-style-type: none"> 1. Nombre 2. Especificar Tipo de Requerimiento (Negocios, Funcional, No Funcional) 3. Descripción 4. Autor 5. Prioridad en la implementación: <ol style="list-style-type: none"> 1-Baja 2-Media 3-Alta 4-Muy Alta 6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)

Proceso	Definir requerimientos No Funcionales
Descripción	<p>Rendimiento, en los Requerimientos de Negocios debería existir la definición de SLA de tiempo de respuesta vinculado a un conjunto de transacciones, sin embargo se dan tres posibles situaciones:</p> <ul style="list-style-type: none"> - El conjunto de transacciones es adecuado para el tiempo asignado a Pruebas de Carga en el proyecto. Usualmente el tiempo alcanza para considerar entre 10 y 30 transacciones. - El conjunto de transacciones es muy amplio y no es factible realizar todas las mediciones en términos económicos. - El conjunto de transacciones ha sido enunciado en forma general y no se han identificado las transacciones.

Proceso	Definir ciclos del release
Descripción	Definir los "Ciclos del Release" indicando los "Conjunto de Pruebas" preliminares que los conforman, por ejemplo el primer "Ciclo de Pruebas" debiera incluir siempre un conjunto de "Pruebas Trazadoras" con el fin de establecer si el ambiente está operativo o no.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

Proceso	Vincular requerimientos a ciclos (*)
Descripción	<p>Esta es la instancia para utilizar la planilla de Apoyo “Estimación Modelo de Servicios” del ANEXO 1. Es importante mencionar que esta plantilla no se podrá utilizar si no está cerrada la planilla de apoyo de LB Requerimientos Calidad, ya que ésta tiene el input más importante para la estimación del servicio.</p> <p>En la plantilla de apoyo además se definen las fases y ciclos de Prueba así como también los recursos relacionados.</p> <p>Se deben vincular los requerimientos a al menos un "Ciclo de Prueba" con el fin de:</p> <ul style="list-style-type: none"> - Determinar en cualquier instante si los requerimientos están siendo cubiertos o no - Generar esbozo de Pruebas en función de los requerimientos asociados a un ciclo. <p>Un requerimiento puede ser vinculado a un Release, en este caso, todos los "Ciclos de Prueba" dentro del Release serán vinculados, y eso implica que debe haber un "Conjunto de Pruebas" al menos dentro de cada Ciclo que valida el requerimiento.</p> <p>La cobertura de Prueba en un ciclo queda definida por la liberación de funcionalidades nuevas y por los defectos ocurridos en el ciclo anterior (en el caso que exista), por lo tanto al principio del proyecto, sólo se vincularán los requerimientos asociados a funcionalidades nuevas al ciclo en que se prueban, y con posterioridad dependiendo de los defectos se vincularán requerimientos a ciclos que contengan Pruebas de Regresión.</p>

Proceso	Generar estrategia pruebas
Descripción	<p>Este documento debe ser un compendio del estado del proyecto hasta este momento, considerando:</p> <ul style="list-style-type: none"> - Roles y personas - Requerimientos de Pruebas - Ciclos de Pruebas - Protocolo de Suspensión de las Pruebas - Conclusiones de Estudio Factibilidad de Automatización. - Referencias a los documentos de Arquitectura de HW/SW. - Formato Plan de Pruebas o Operatoria al respecto - Formato Informe de Resultados de Pruebas. - Manejo de Defectos. - Severidad de Defectos. - Riesgos <p>En el caso de CM-PPQA se incluyen los checklists que se utilizarán en la Revisión. Para extraerlos de QC: Tools->; Excel Report Generator->; Checklist CM-PPQA->; Generate Excel Report->; (Ingresar como parámetro el nombre de la prueba que contiene el Checklist).</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.3.1 Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”

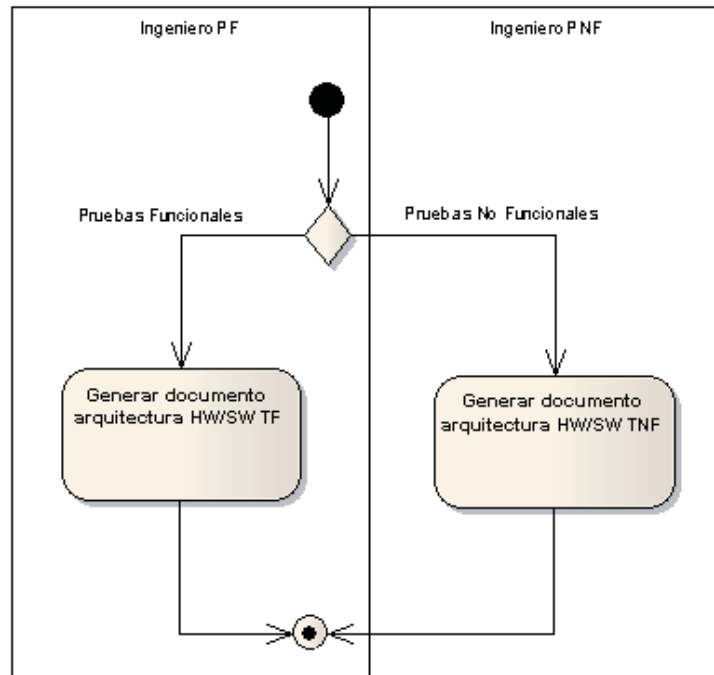


Figura 5.05: Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”

5.1.3.2 Proceso “Estimar grado automatización y nivel diagnóstico”

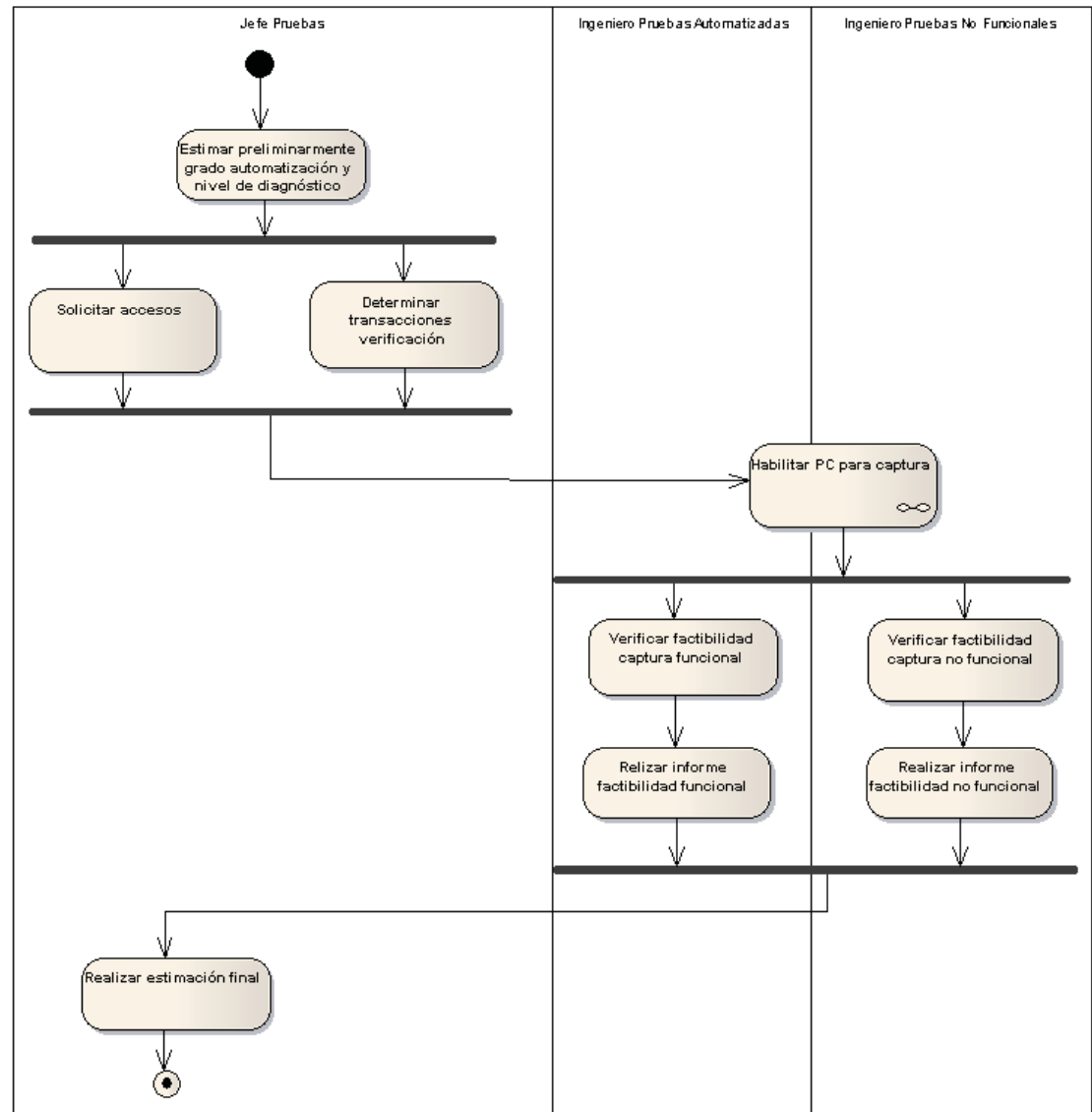


Figura 5.06: Proceso “Estimar grado automatización y nivel diagnóstico”

Proceso	Estimar preliminarmente grado automatización y nivel de diagnóstico
Descripción	<p>El análisis preliminar en esta tarea aborda el Grado de Automatización y el Nivel de Diagnóstico de la siguiente forma:</p> <p>Grado de Automatización</p> <p>En el caso que se disponga del aplicativo, se identifican todas aquellos módulos del aplicativo que no son automatizables con Mercury tanto desde el punto de vista funcional como no funcional, candidatos serían:</p> <ul style="list-style-type: none"> - General: (a) Uso de algoritmos de reconocimiento de imágenes, sonidos u otro distinto a eventos de teclado y mouse. (b) Dispositivos GUI no emulables en PC o notebooks. - Caso Funcional: (a) Reportes (b) Procesos Batch (c) Dispositivos GUI con mala emulación en PC. - Caso No Funcional: (a) Procesos Batch (b) Protocolos a nivel de GUI inexistentes en Mercury o no adquiridos. <p>Nivel de Diagnóstico</p> <p>Tanto en las Pruebas Funcionales como en las No Funcionales existe el mismo concepto, la diferencia es que en el último caso es siempre parte del servicio, en cambio en el caso funcional no.</p> <ul style="list-style-type: none"> - Caso Funcional: Si el servicio es Graybox, se deben verificar los resultados en la Base de Datos. Se debe evaluar si existen impedimentos para acceder la BD como Modelo no documentado. - Caso No Funcional: En este caso, si bien se espera un nivel Whitebox, no es siempre posible, debido a que pueda que no exista el SW para monitorear cada uno de los servidores aplicativos o en el caso que sea Mercury Diagnostics, no se esté dispuesto a hacer la inversión. <p>Si no se disponen del aplicativo se deberá hacer supuestos sobre las transacciones para las cuales no es tan claro si pueden ser automatizadas o no.</p>

Proceso	Solicitar accesos
Descripción	<p>Se debe solicitar:</p> <ul style="list-style-type: none"> - Acceso al aplicativo - Conectividad a Servidor QC en la Organización para QTP en el caso de pruebas funcionales.

Proceso	Determinar transacciones verificación
Descripción	<p>En base a los antecedentes que se disponga, se debe definir que transacciones serán las que se usarán para validar si es posible automatizar ya sea funcional o no funcionalmente. Se debe tener en cuenta al menos:</p> <p>Generales</p> <ul style="list-style-type: none"> - Debería haber al menos 1 transacción por arquitectura de SW distinta a nivel de "Dispositivo GUI" <p>Pruebas No Funcionales</p> <ul style="list-style-type: none"> - Informarse si algún SW encripta o comprime a nivel de "Dispositivo GUI" y si fuera así, solicitar los algoritmos al desarrollador (siempre y cuando sean indispensables para variabilizar los scripts VUGEN posteriormente). - Determinar si la aplicación es mono o multiusuario por dispositivo GUI.

Proceso	Habilitar PC para captura
Descripción	<p>Se contemplan 2 situaciones:</p> <ul style="list-style-type: none"> - Aplicativo se encuentra disponible a través de la red de la organización, en ese caso se utilizarán los computadores destinados para ese fin en dicha organización. - En el caso que no esté disponible en la red organizacional, se usará una máquina virtual que tenga QTP y LR listos para ser instalados.

Proceso	Verificar factibilidad captura funcional
Descripción	<p>La verificación consiste en que para cada transacción de verificación se ejecuta:</p> <ul style="list-style-type: none"> - Identificar los Add-ins que podrían necesitar en base a la documentación o referencias de la GUI. - Generar un script para cada una de las transacciones - Reejecutar el script. - Comprobar que se puedan incorporar puntos de verificación. - Comprobar que se puede variabilizar la entrada. <p>Una vez realizadas las tareas anteriores complete la planilla de Factibilidad QTP.</p>

Proceso	Verificar factibilidad captura No Funcional
Descripción	<p>La verificación consiste en que para cada transacción de verificación se ejecuta:</p> <ul style="list-style-type: none">- Generar un script que implique login, ejecutar transacción y logout.- Detectar correlación y resolverla en el caso que la hubiere.- Determinar si los mensajes son parametrizables.- Reejecutar el script.- Si es posible realizar una prueba con más de un usuario a la vez sobre el mismo "Dispositivo GUI". <p>Nota: Para resolución de problemas ver tab Files.</p>

Proceso	Realizar estimación final
Descripción	<p>Tanto para el caso funcional como no funcional, se consolida la estimación inicial con los resultados obtenidos en la verificación con el aplicativo.</p> <p>El resultado de esta estimación puede producir RFC de un impacto no despreciable en el presupuesto.</p>

5.1.3.2.1 Proceso “Habilitar PC para captura”

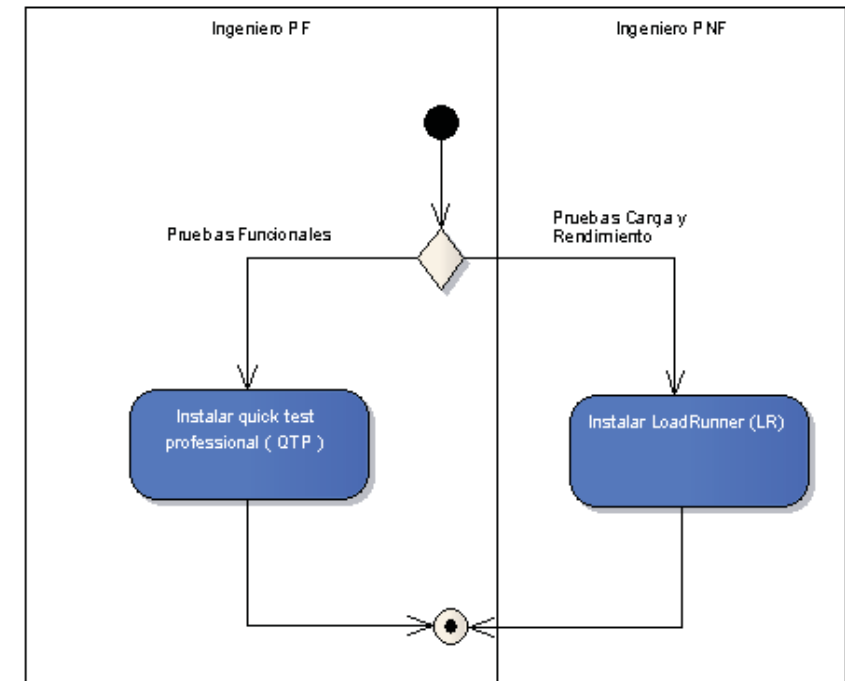


Figura 5.07: Proceso “Habilitar PC para captura”

5.1.3.3 Proceso “Definir requerimientos no funcionales”

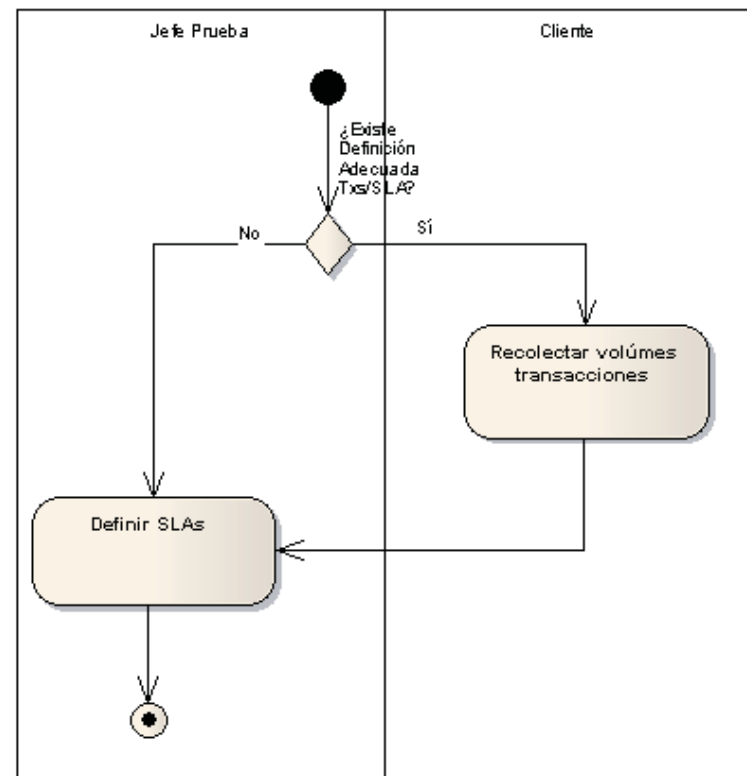


Figura 5.08: Proceso “Definir requerimientos no funcionales”

Proceso	Definir SLAs
Descripción	<ul style="list-style-type: none"> - Definir una carpeta con nombre "Pruebas de Carga y Rendimiento". - En el caso que exista información histórica o estimaciones, agregar attachment tipo URL apuntando a la ubicación de una planilla Excel "Volúmenes de Transacciones" en subversion. - Definir para cada transacción seleccionada un Requerimiento de Testing detallando su SLA, para el cual se debe especificar que acción del usuario gatilla la transacción que va a ser medida (por ejemplo click sobre el botón grabar).

Proceso	Recolectar volúmenes transacciones
Descripción	<p>Recolectar información histórica o estimar en un documento llamado "Volúmenes Transacciones", la siguiente información:</p> <ul style="list-style-type: none"> - Número de usuarios del sistema - % Crecimiento Anual Usuarios del Sistema - % Concurrencia de usuarios máxima - Una hoja de cálculo de transacciones cuyas columnas sean: Id Transacción, Nombre Transacción, Cantidad Transacciones Promedio x Tramo Horario i ($i=1, \dots, n$), % Crecimiento Anual. Ojala se pudieran descomponer la cantidad de transacciones por perfil - Una hoja de cálculo de transacciones cuyas columnas sean: Id Transacción, Nombre Transacción, Cantidad Transacciones Peak x Tramo Horario i ($i=1, \dots, n$), % Crecimiento Anual. Ojala se pudieran descomponer la cantidad de transacciones por perfil.

5.1.4 Proceso “Administrar proceso de pruebas” PARTE 2

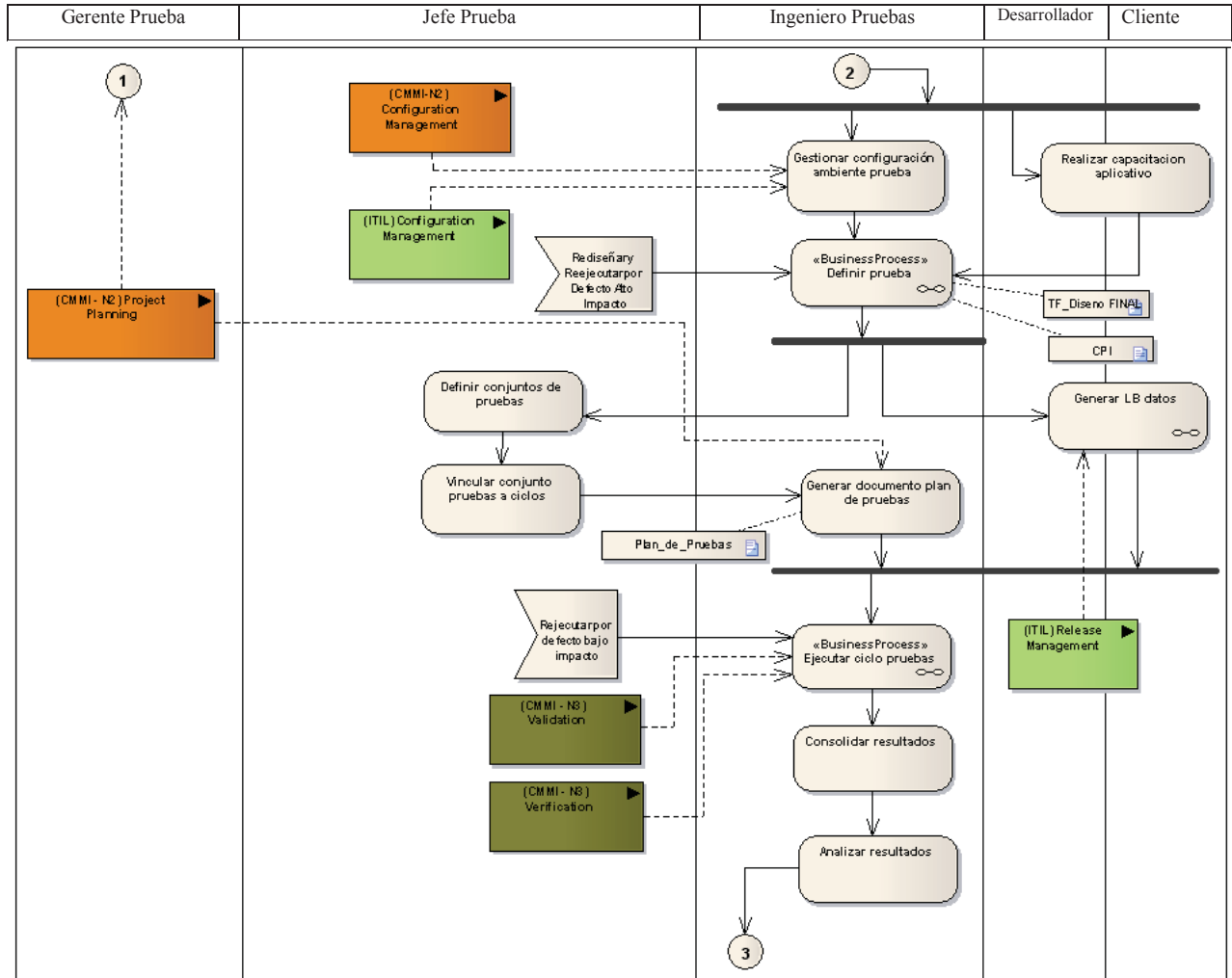


Figura 5.09: Proceso “Administrar proceso de pruebas” PARTE 2

Proceso	Gestionar configuración ambiente prueba
Descripción	<p>Con el objetivo de realizar las Pruebas, además de disponer del "Ambiente de Pruebas" (se debe definir de quien es la responsabilidad de entregar dicho ambiente de Pruebas), Calidad debe adaptar el ambiente para usar Mercury:</p> <ul style="list-style-type: none"> - Verificar que todas las componentes del Ambiente de pruebas que no son responsabilidad de Calidad, están disponibles. - Habilitar Laboratorio(s) Prueba(s): Disponer de "Dispositivos GUI", Instalar SW Calidad, Instalar SW Cliente. (Ver manual de instalación en Tab Files). - Generar los RFC para las conectividades de enlace con los "Servidores de Calidad" y "Dispositivo_GUI". - Generar los RFC para las configuraciones o instalaciones de "SW Calidad" en los "Servidores Aplicativo". - Enviar los RFCs a los responsables. - Verificar si los RFCs fueron correctamente ejecutados. - Verificar Normas de Seguridad impartidas por el Oficial de Seguridad.(Ver archivos adjuntos)

Proceso	Realizar capacitación aplicativo
Descripción	Realizar Capacitación del Aplicativo, idealmente se debiera hacer una Capacitación para Pruebas Funcionales y No Funcionales, pero podría haber casos donde tenga que ser realizada en sesiones separadas.

Proceso	Definir prueba (*)
Descripción	<p>La primera plantilla de apoyo que se utiliza en este proceso es "Definición CPI" ANEXO 1, en la cual se definen todas las combinatorias posibles en base a los criterios especificados en la planilla de apoyo.</p> <p>Una vez seleccionadas las Pruebas a diseñar, se debe preparar la planilla de apoyo "TF Diseño" ANEXO 1, en la cual se debe comenzar a identificar los ID de Prueba y relacionarlos a las Pruebas seleccionadas de la planilla anterior.</p> <p>Al mismo tiempo en este proceso se debe proceder con lo siguiente:</p> <p>Este proceso se basa en:</p> <ul style="list-style-type: none"> - Definición de Requerimientos de Pruebas: De los cuales se derivarán las Pruebas o "carpetas de Pruebas", las cuales serán refinadas con la documentación de Diseño que se tenga del aplicativo. QC tiene la opción de derivar automáticamente de los "Requerimientos de Pruebas" una jerarquía preliminar de Pruebas, la cual debe ser refinada. - La Capacitación Funcional y con otros entregables (sería ideal contar con las grabaciones de

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<p>ejecuciones sobre la GUI del aplicativo) se tiene la información necesaria para detallar los "Pasos" de cada Prueba y generar los "Scripts de Prueba" en el caso de automatización con "Load Runner".</p> <p>y contempla la realización de las siguientes tareas:</p> <ul style="list-style-type: none"> - Definir Estructura Directorios -Definir Pruebas Unitarias - Detallar Pruebas Unitarias - Definir Pruebas Integradas - Generar Scripts - Especificar Necesidad Datos - Detallar Pruebas Integradas <p>Es muy importante tener en mente que en cada ciclo puede ser necesario reejecutar algunas de las anteriores.</p>
--	--

Proceso	Definir conjuntos de pruebas
Descripción	<p>Se definen los conjuntos de prueba, ver ejemplos de estos conjuntos en Diagrama Conjunto_Prueba.</p> <p>En QC, Test Lab:</p> <ul style="list-style-type: none"> - Crear carpeta(s) que contienen "Conjuntos de Pruebas". - Crear cada "Conjunto de Pruebas" en la carpeta que corresponda, e ingresar las "Pruebas" que lo componen usando el tab "Execution Grid". El orden de ejecución se define en el tab "Execution Flow".

Proceso	Vincular Conjunto Pruebas a Ciclos
Descripción	<p>En QC, Test Lab, las carpetas que contienen "Conjuntos de Pruebas" son linkeadas a ciclos. Para esto hacer click derecho sobre la carpeta y "Assign to Cycle".</p>

Proceso	Generar documento plan de pruebas (*)
Descripción	<p>Con todos los procesos anteriores concluidos, y con todas las plantillas de apoyo concretadas, se debe dar pie a la generación del Plan de Pruebas, el cual contiene información de resumen de las decisiones y acuerdos tomados; es decir, es dejar reflejado el acuerdo para dar comienzo formula al proceso de diseño de las Pruebas.</p> <p>Plantilla de apoyo "Plan de Pruebas" ANEXO 1.</p>

Proceso	Generar LB datos
Descripción	<p>Con el fin de realizar las Pruebas es necesario llevar al sistema a un determinado nivel de datos y generar los datos suficientes a utilizar en los escenarios de Prueba, esto implica :</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<ul style="list-style-type: none"> - Generar BDs, las cuales deben ser parte de la LB, la cual debería ser administrada por CM (ver qué pasa en los clientes externos). - Generar Datos en la cantidad necesaria para realizar las Pruebas
--	--

Proceso	Consolidar resultados
Descripción	<p>Esta tarea consiste en determinar si los defectos reportados durante las Pruebas son reducibles a un grupo menor o son preexistentes.</p> <p>En el caso que un grupo de defectos pueda ser reemplazado por un solo defecto:</p> <ul style="list-style-type: none"> - Elegir un defecto del grupo como representativo. - Linkear el resto de los defectos al seleccionado en el paso 1. - Dejar los defectos linkeados en estado "Consolidated".

Proceso	Analizar resultados
Descripción	<p>Esta tarea consiste en determinar si los defectos reportados durante las Pruebas son tales.</p> <ul style="list-style-type: none"> - En el caso que el defecto sea tal, actualizar su estado a "Open". - En el caso que el defecto no aplique, se puede borrar o dejar su estado en "Closed", esto último sólo si es de algún aporte mantener el registro.

5.1.4.1 Proceso “Definir prueba”

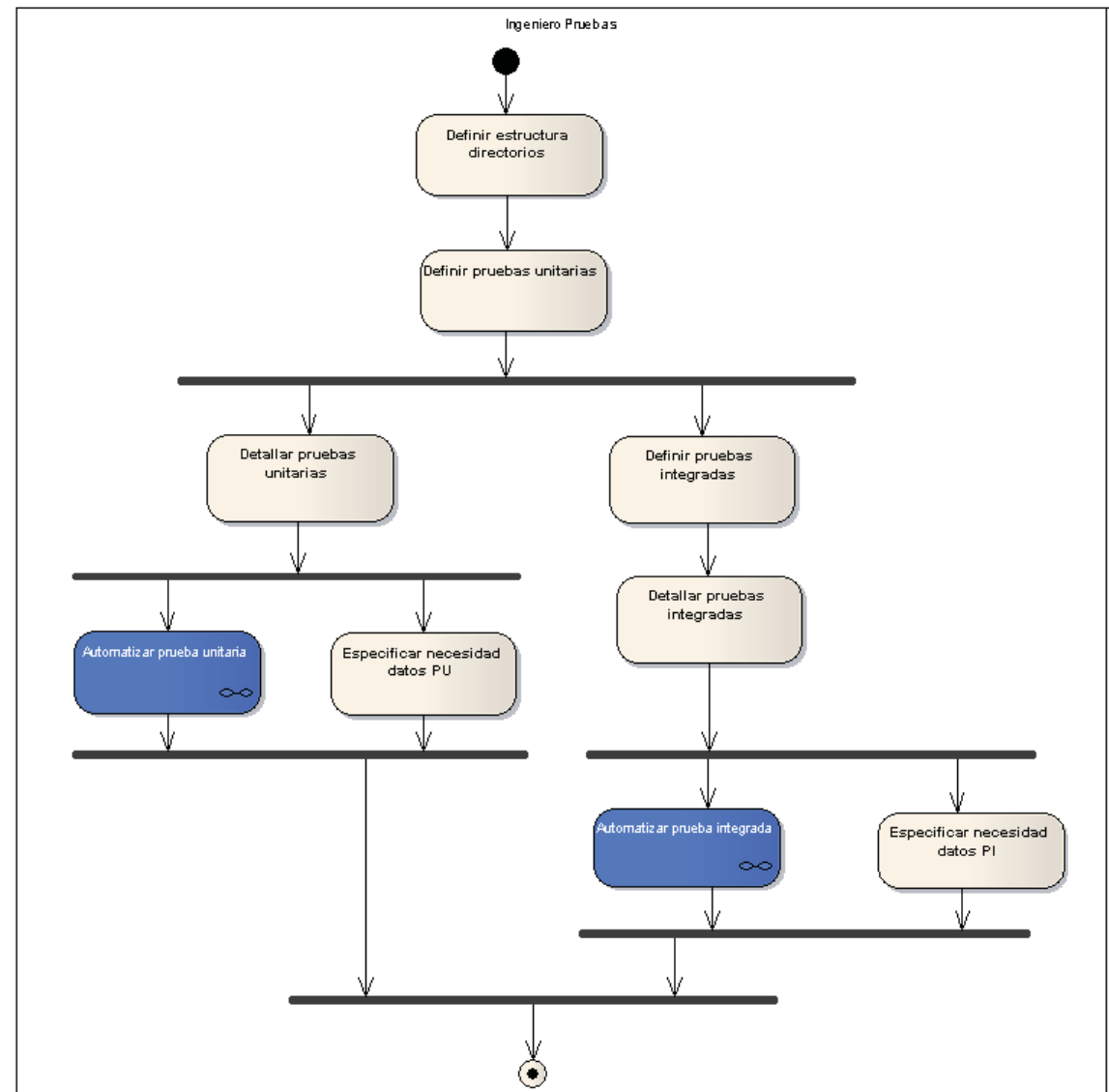


Figura 5.10: Proceso “Definir prueba”

Proceso	Definir estructura directorios
Descripción	<p>Crear los directorios que se necesitan para almacenar los scripts de pruebas, repositorios de objetos y bibliotecas de funciones tanto de QTP como LR. Se recomienda crear en Subversion:</p> <ul style="list-style-type: none"> - Crear directorio de Scripts, se sugiere que el nombre de cada script se un prefijo de a lo más 3 caracteres alfabéticos concatenados con el ID QC de la Pruebas en formato '99999'. - Crear directorio de Repositorios de Objetos (Sólo para uso de QTP, es decir Pruebas Funcionales) - Crear directorio de Bibliotecas de Funciones (Sólo para uso de QTP, es decir Pruebas Funcionales) - Crear directorio de Escenarios de Recuperación <p>Los scripts deben ser actualizados en QC cuando se quiera ejecutarlos y en otro momento no corresponderán necesariamente a la LB.</p> <p>NOTA GENERAL: EL ORDEN EN ESTE DIAGRAMA DE PROCESOS NO IMPLICA QUE UNA TAREA NO SE PUEDA VOLVER A REALIZAR EN UN MOMENTO POSTERIOR CON FINES DE REFINAR UNA DEFINICION YA EXISTENTE, TENIENDO EN MENTE NO OLVIDAR GESTIONAR LA LB, SI CORRESPONDE.</p>

Proceso	Definir pruebas unitarias
Descripción	<p>Una Prueba unitaria en el caso de Pruebas funcionales corresponderá a una unidad funcional indivisible y posiblemente derivable de un requerimiento de Pruebas (si éstos han sido detallados adecuadamente).</p> <p>A este nivel basta con definir la Prueba Unitaria, pero no indicar sus pasos.</p>

Proceso	Detallar pruebas unitarias
Descripción	<p>Esta tarea siempre se realiza para las Pruebas manuales y se evalúa su realización para el caso automático dependiendo del tiempo y del servicio otorgado. Por ejemplo, cuando no se realiza la ejecución por parte de la Organización, si se deberían documentar.</p> <p>Se ingresa la Prueba a QC en Test Plan ->; Design Steps, de acuerdo a la especificación entregada en el Diagrama "Diseño Prueba" para la "Prueba_Unitaria", "Paso_Documentado", "Escenario_Prueba_Unitaria" y "Parametro_Prueba_Unitaria" y "Regla".</p>

Proceso	Especificar necesidad datos PU
Descripción	<p>- Confeccionar documento que indique cual es la necesidad de datos en función de los parámetros que se identifiquen.</p> <p>- Especificar un atributo que permita particionar los escenarios entre los que consideran datos existentes en la Base de Datos y datos generados.</p> <p>- Generar y adjuntar a la Prueba en QC la Planilla Referencial de Datos siguiendo el formato de nombre D999999_01.xls donde los primeros 6 dígitos corresponden al ID generado en QC para la Prueba (Test Plan->; Seleccionar Test->; Tab Attachments). La planilla referencial Excel tiene una hoja con nombre "Global", cuya primera fila contiene los nombres de los parámetros identificados en el punto 1 y tiene tantas filas como escenarios se vayan a probar en la Prueba. Como regla general en el caso manual, debería haber una línea por escenario para hacer más simple la descripción de los pasos y el trabajo del Tester. En el caso automático puede haber más de un escenario manejado por el script.</p> <p>Observación: La planilla referencial Excel es usado como base para las planillas que contendrán los datos reales que se usarán para ejecutar las Pruebas, estas últimas planillas deben ser adjuntadas a los "Test Instance".</p>

Proceso	Definir pruebas integradas
Descripción	<p>Combina 2 o más pruebas unitarias dentro de una sola prueba</p> <p>Esta tarea siempre se realiza para las pruebas manuales y se evalúa su realización para el caso automático dependiendo del tiempo y del servicio otorgado. Por ejemplo, cuando no se realiza la ejecución por parte de organización, si se deberían documentar. (EN EVALUACIÓN DURANTE EL PILOTO)</p>

Proceso	Detallar pruebas integradas
Descripción	Se ingresa la Prueba a QC en Test Plan ->; Design Steps, de acuerdo a la especificación entregada en el Diagrama "Diseño Prueba" para la "Prueba_Integrada".

Proceso	Especificar necesidad datos PI
Descripción	<p>- Se debe especificar un atributo que permita particionar los escenarios entre los que consideran datos existentes en la Base de Datos y datos generados.</p> <p>- Generar y adjuntar a la prueba en QC la Planilla Referencial de Datos siguiendo el formato de nombre D999999_01.xls donde los primeros 6 dígitos corresponden al ID generado en QC para la prueba (Test Plan->; Seleccionar Test->; Tab Attachments). La planilla referencial Excel tiene una hoja de nombre Global para la Prueba unitaria guía y tantas hojas como Pruebas unitarias dependan de la guía en una relación uno es a muchos. Como ejemplo considere el caso de un Maestro-Detalle, el maestro es la Prueba unitaria guía y el detalle es la Prueba unitaria asociada, esta última tendrá una hoja propia.</p> <p>Observación: La planilla referencial Excel es usada como base para las planillas que contendrán los datos reales que se usarán para ejecutar las Pruebas, estas últimas planillas deben ser adjuntadas a los "Test Instance".</p>

5.1.4.2 Proceso “Automatizar prueba unitaria”

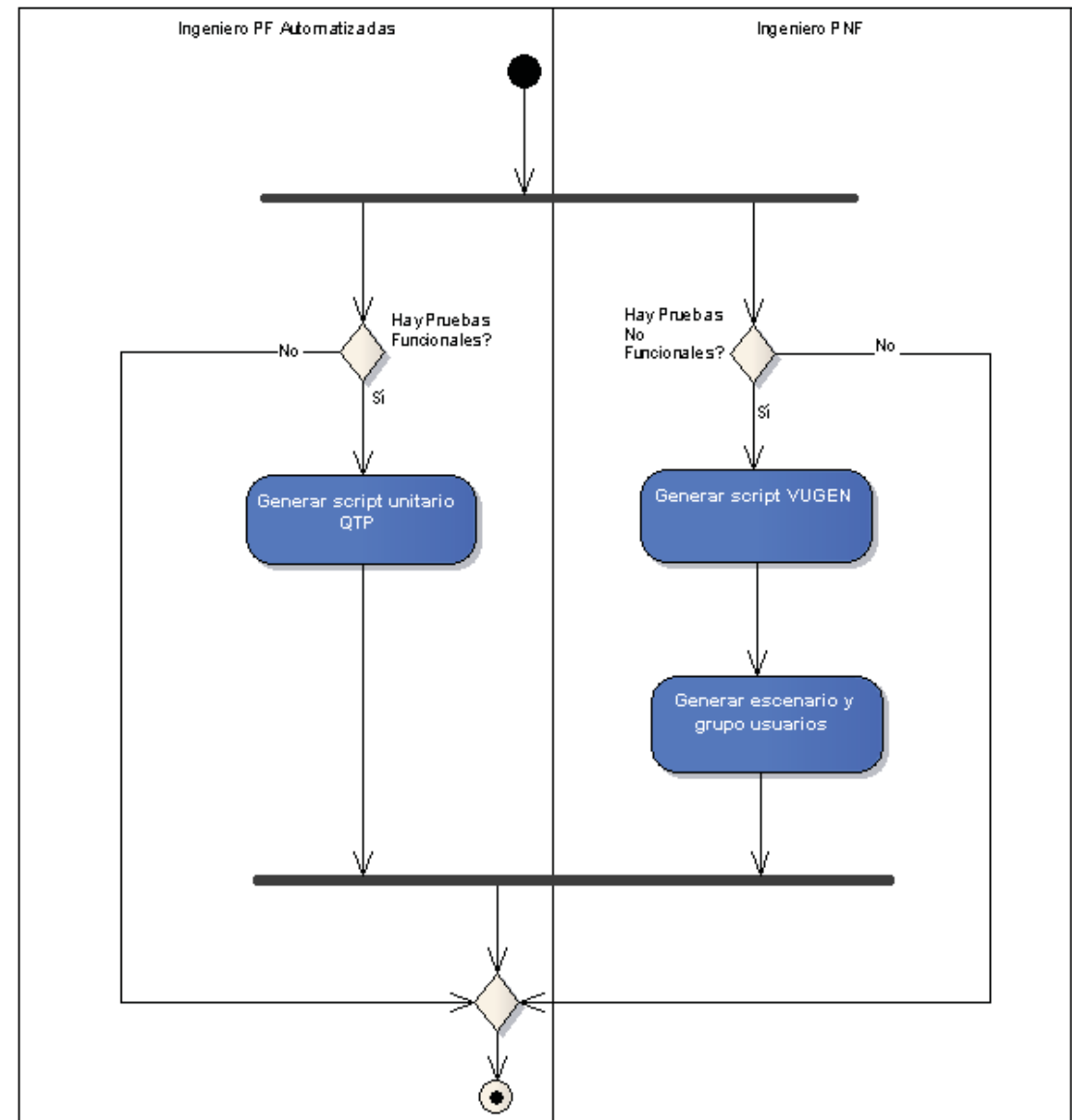


Figura 5.11: Proceso “Automatizar prueba unitaria”

Proceso	Generar script unitario QTP
Descripción	<ul style="list-style-type: none"> - Se graba un "Acción QTP" dentro del script, siguiendo los pasos (recordar que en QTP las validaciones de cualquier tipo son pasos) y parametrización detallada para la prueba en QC. Debido a que puede haber varios flujos de pasos (al menos el caso positivo y el negativo) probablemente se deba intervenir el script en su vista "Keyword-View" o "Script-View". - La "Acción QTP" debe informar siempre a través de Report.ReportEvent si la prueba fue exitosa o falló. - Al final el script se almacena en el directorio que corresponde para que sea versionado. - Se recomienda medir el tiempo de respuesta, pero sólo con el fin de tener una referencia, pues debido a que ni el Ambiente de Pruebas ni las Bases de Datos en él son representativos del Ambiente Productivo en régimen permanente.

Proceso	Generar script VUGEN
Descripción	<ul style="list-style-type: none"> - Se graba el login y el logout dentro de dos acciones. - Para cada transacción acordada con el Cliente como parte de la prueba se graba un script con una acción y dentro de ella se marca el inicio y el término de la "Transaction" la cual mide el tiempo sujeto a SLA. - A continuación se debe depurar el script revisando: Pacing, Log, Think Time, Network - Speed Simulation (útil cuando el ancho de banda disponible es mayor al real) y detección de Correlación y cerciorándose que el script se puede re-ejecutar exitosamente. - Dependiendo del tipo de aplicativo, agregar puntos de Verificación para determinar si la transacción está retornando adecuadamente, pues hay veces que la no disponibilidad no se manifiesta en la caída de la aplicación sino en comportamiento anormal.

Proceso	Generar escenario y grupo usuarios
Descripción	En el caso de Pruebas de Carga y Rendimiento Unitarias, usando LR Controller, se declara un escenario por cada script usando un solo grupo programando tramos de 1, 10, 20, 30 y 50 usuarios respectivamente. La duración de cada tramo debería ser al menos 5 minutos con el fin de ver la estabilidad de la respuesta del sistema, si bien tiempos mayores son recomendables, su uso dependerá del tiempo total asignado a esta prueba.

5.1.4.3 Proceso “Automatizar prueba integrada”

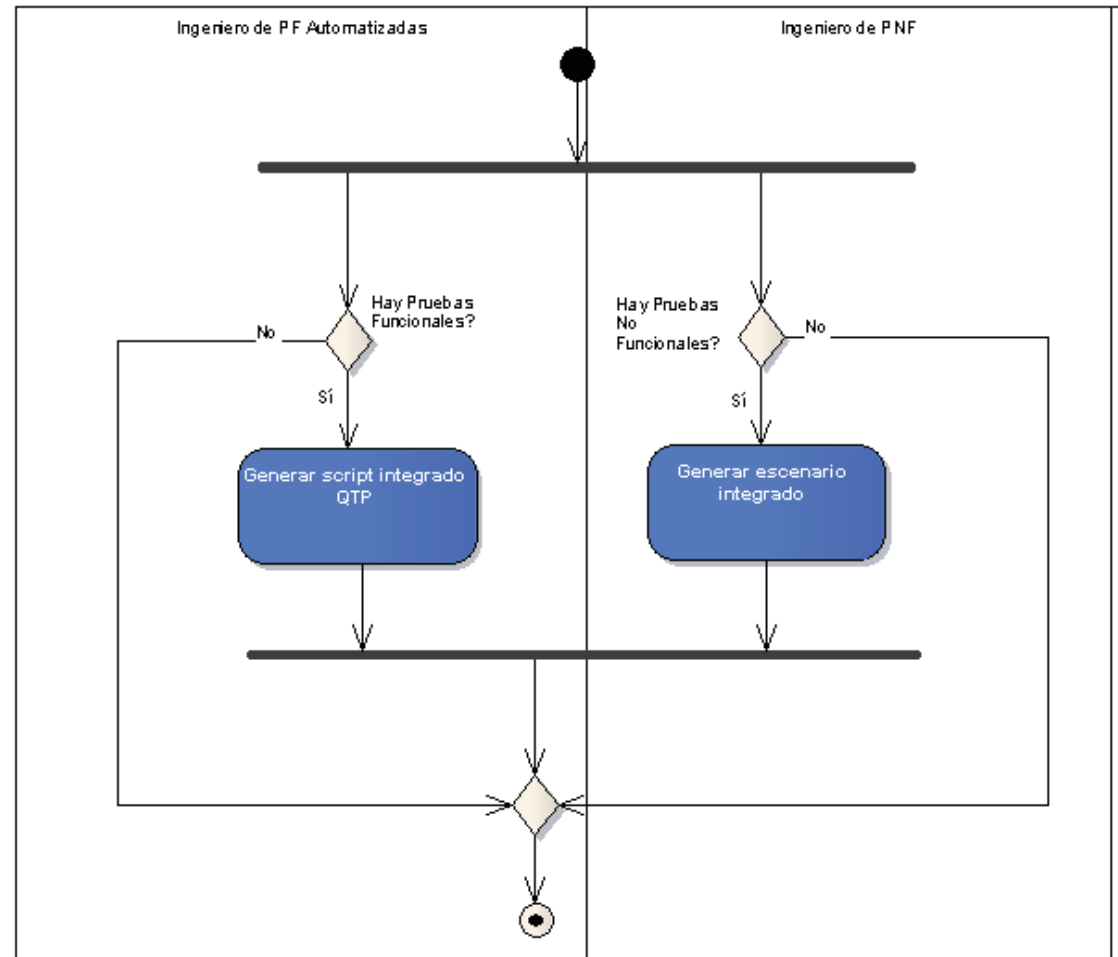


Figura 5.12: Proceso “Automatizar prueba integrada”

Proceso	Generar script integrado QTP
Descripción	En este punto ya se han generado las "Acciones QTP" para las pruebas unitarias, entonces: <ul style="list-style-type: none"> - Se crea un script en el cual se referencia todas aquellas acciones que componen la "Prueba Integrada" siguiendo el flujo definido. - Se declara una planilla de datos por acción, la cual deberá ser llenada.

Proceso	Generar escenario integrado
Descripción	Se genera en Load Runner Controller, el escenario integrado definiendo todos los Grupos de Usuarios que se haya contemplado en la prueba y usando los Script ya generados en VUgen para las Pruebas Unitarias.

5.1.4.4 Proceso "Generar LB datos"

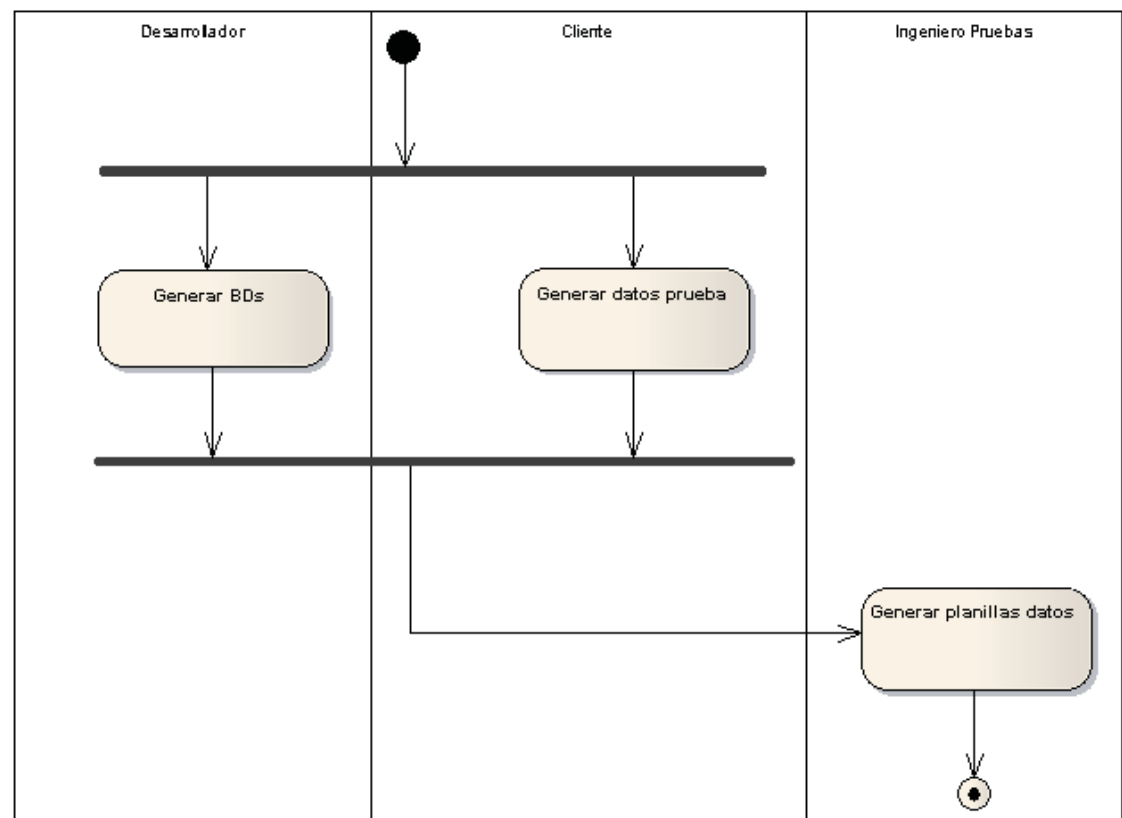


Figura 5.13: Proceso "Generar LB datos"

Proceso	Generar BDs
Descripción	Esta generación depende del tipo de pruebas y se deben considerar los siguientes aspectos: <ul style="list-style-type: none">- Las pruebas no funcionales requerirán un tamaño representativo del aplicativo en régimen permanente.- Se debe velar por la confidencialidad de los datos.- Los procedimientos de Respaldo y Restauración deben existir y estar probados antes del inicio de las pruebas, esto es necesario debido a que muchas veces se necesitan realizar las pruebas más de una vez usando la misma Release del SW. Ejemplo: Se hicieron pruebas no funcionales en las cuales se detectaron problemas en el HW, los cuales fueron resueltos y se requiere probar de nuevo.

Proceso	Generar datos prueba
Descripción	Generación a través de consultas SQL o Recolección de Datos en la cantidad necesaria para realizar las pruebas.

Proceso	Generar planillas datos
Descripción	Usando las Planillas Excel Referenciales adjuntadas a cada prueba durante las actividades "Especificar Necesidad Datos PU" y "Especificar Necesidad Datos PI", generar las Planilla Excel para las pruebas que serán realizadas durante el ciclo en curso.

5.1.4.5 Proceso “Ejecutar ciclo pruebas”

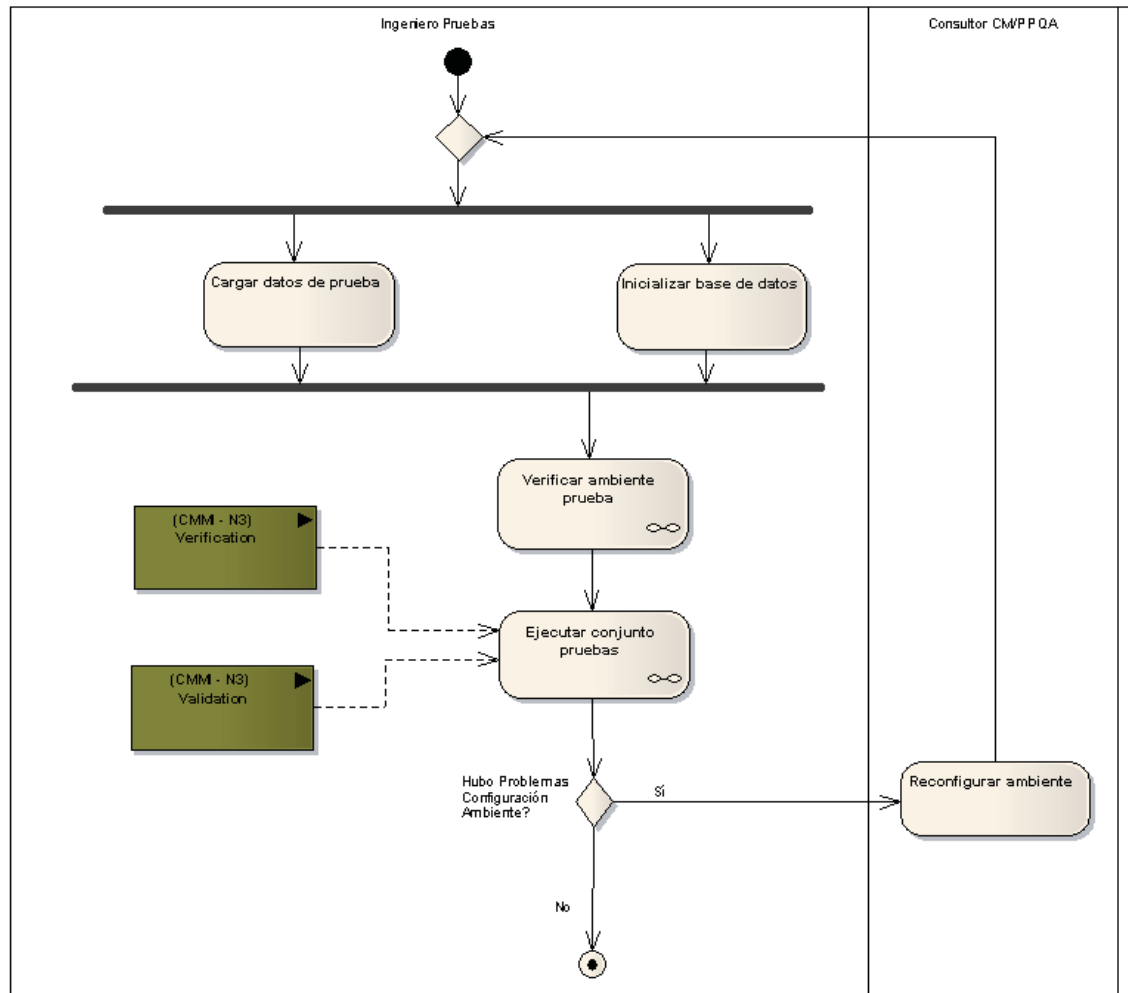


Figura 5.14: Proceso “Ejecutar ciclo pruebas”

Proceso	Cargar datos de prueba
Descripción	Cargar las Planillas de Datos de Prueba generadas en "Generar Planillas Datos" en las "Test Instance" del ciclo.

Proceso	Inicializar base de datos
Descripción	Inicializar la(s) base(s) de datos es setear esta(s) misma(s) en la condición inicial para que comience el Ciclo correspondiente, lo que implica: <ul style="list-style-type: none">- Datos Necesarios y consistentes con los datos que se usarán en las pruebas.- Para el caso no funcional, además se requiere que la(s) Base(s) de Datos tenga(n) un tamaño representativo al que tendría en el estado permanente en producción.

Proceso	Ejecutar conjunto pruebas
Descripción	Se ejecuta cada prueba dentro del Conjunto de Pruebas ya sea manual o automáticamente. El primer Conjunto de Pruebas a ejecutar debería ser algún tipo de "Pruebas Trazadoras". En el caso de Pruebas No Funcionales, es altamente recomendable que las Pruebas de Carga y Rendimiento Unitarias se ejecuten al menos 3 veces, para tener una confiabilidad mínima en la medición de tiempos y comportamiento del sistema.

5.1.4.6 Proceso “Verificar ambiente prueba”

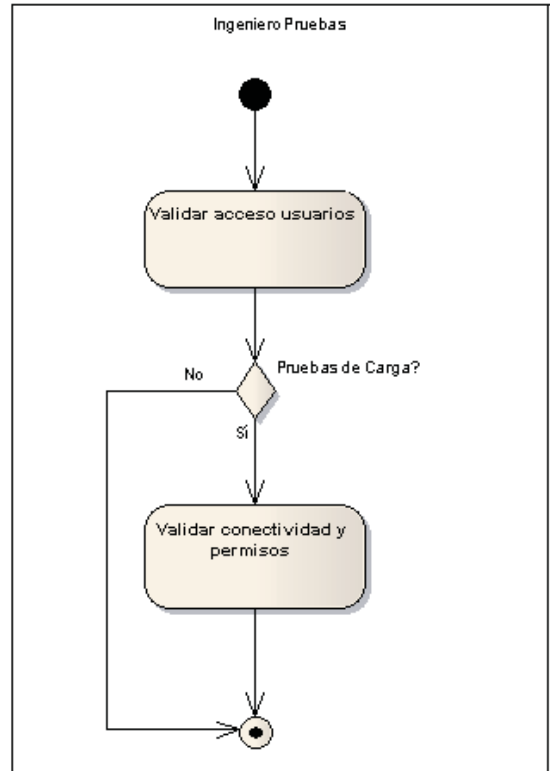


Figura 5.15: Proceso “Verificar ambiente prueba”

Proceso	Validar acceso usuarios
Descripción	Validar que sólo las personas responsables de las pruebas tengan acceso al sistema.

Proceso	Validar conectividad y permisos
Descripción	<p>En el tab Run del LR Controller validar:</p> <ul style="list-style-type: none"> - Conectividad del Controller a Generadores de Carga (Icono Load Generators). - Conectividad y permisos de monitoreo desde el Controller a los servidores del aplicativo y de BDs. (Al agregar las métricas de Windows Resources).

5.1.4.7 Proceso “Ejecutar conjunto de pruebas”

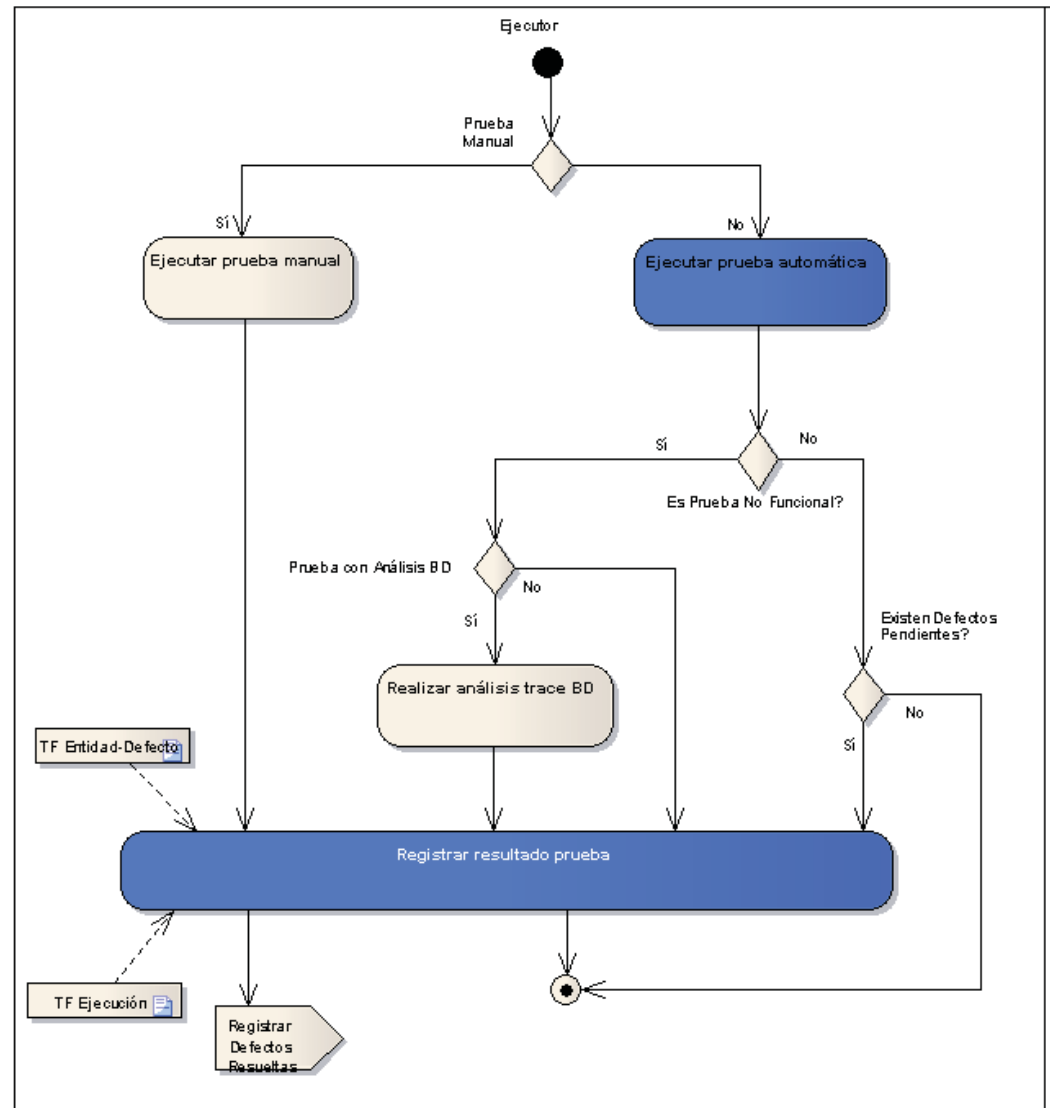


Figura 5.16: Proceso “Ejecutar conjunto de pruebas”

Proceso	Ejecutar prueba automática
Descripción	Para el Caso de Pruebas Funcionales se debe setear en QTP que registre automáticamente cualquier error como defecto.

Proceso	Registrar resultado prueba (*)
Descripción	<p>Una vez que el Analista/Ejecutor ejecuta las pruebas diseñadas, debe ir dejando registro de dichas pruebas en la planilla de “TF Ejecución” (Ver ANEXO1), la cual irá dejando la evidencia de la ejecución.</p> <p>En el caso que la prueba obtenga estado NOK, se debe registrar en la planilla "TF Entidad-Defecto" (Ver ANEXO 1), todos los defectos encontrados en la prueba.</p> <p>Al momento de ejecutar, basta con que un paso no sea el esperado para que el caso de prueba total obtenga el estado NOK, sin embargo, que un paso sea NOK no implica necesariamente que no se puedan seguir testeando los demás pasos.</p>

5.1.5 Proceso “Administrar proceso de pruebas” PARTE 3

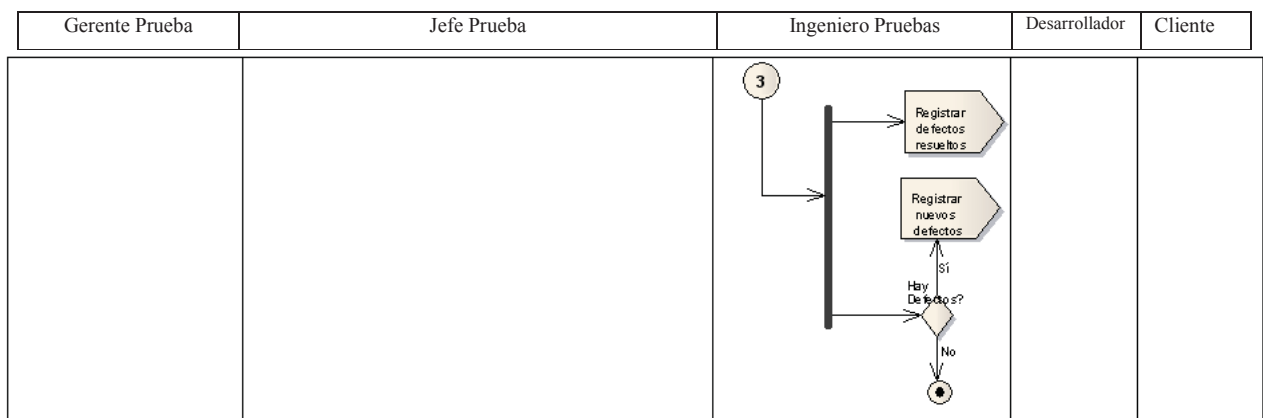


Figura 5.17: Proceso “Administrar proceso de pruebas” PARTE 3

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.6 Proceso: “Gestionar defectos”

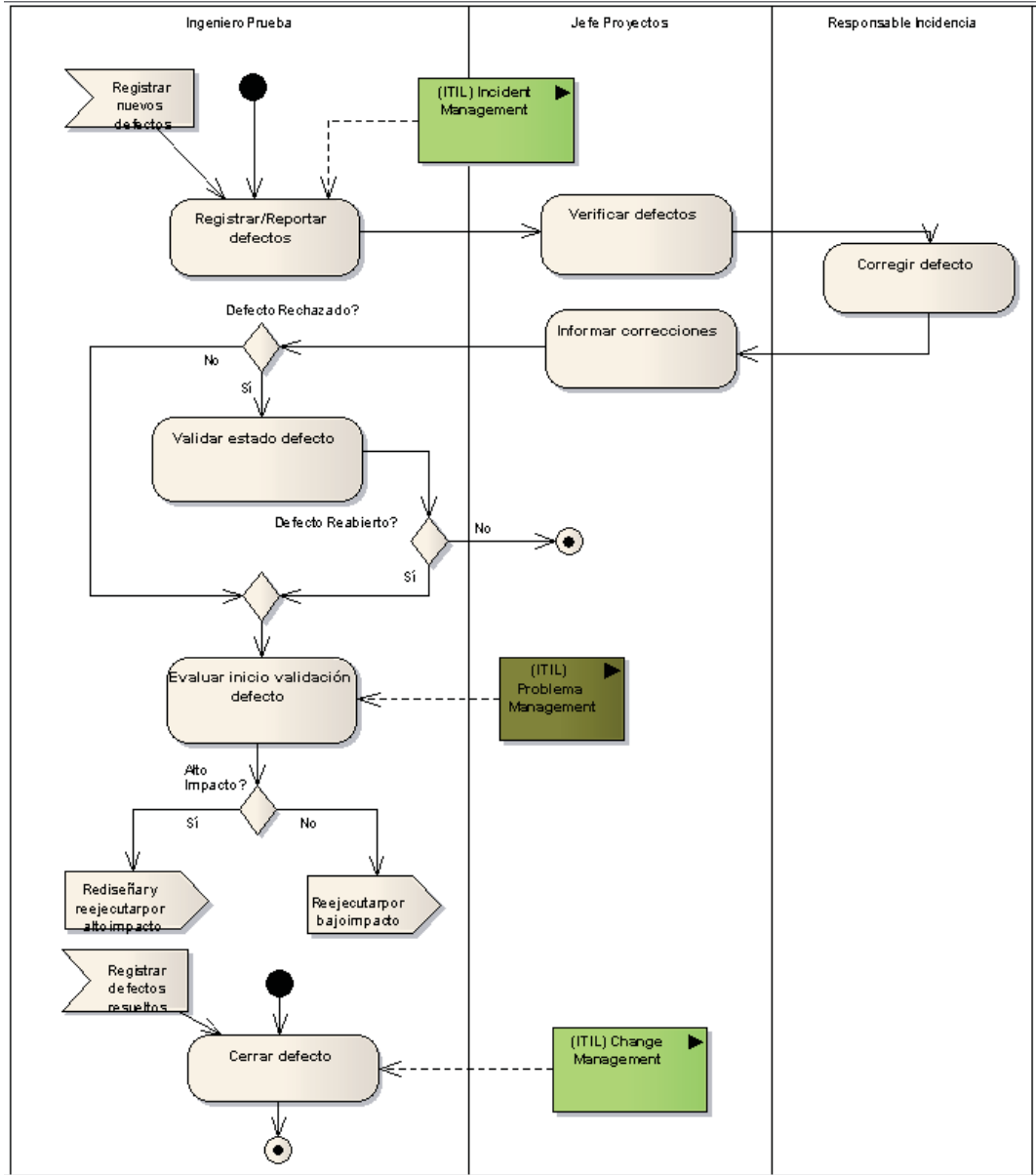


Figura 5.18: Proceso “Gestionar defectos”

Proceso	Registrar/reportar defectos
Descripción	<p>- Ingresar los atributos del defecto, es importante el ingreso del campo Área Calidad (CM, PPQA, TF, TNF) con el fin de cada área pueda posteriormente identificar que defectos debe gestionar. También no se debe identificar al "Responsable de Solución" que puede ser el Desarrollador, Responsable HW, Responsable SW, Responsable Redes.</p> <p>- Vincular las entidades que gatillan el defecto, recuerde que pueden ser las siguientes:</p> <ul style="list-style-type: none"> • Requerimiento • Prueba • Conjunto de Pruebas • Instancia de Pruebas • Ejecución de Pruebas • Paso de Pruebas (En el caso que se esté ejecutando una prueba, lo más recomendable vincular al paso el defecto y adjuntar la evidencia a este último) • A otro defecto (Por deducción) <p>Para que el defecto sea procesado como tal, se lo debe dejar en estado "Open".</p> <p>Reportes de Defectos</p> <p>- CM y PPQA, en Customize->; Excel Report Generator->; CM-PPQA Matriz No Conformidades.</p>

Proceso	Corregir defecto
Descripción	<p>El Jefe de Proyecto o el equipo de desarrollo evalúan si corresponde a un defecto, y de serlo inicia su corrección.</p> <p>Si el defecto no se acepta, se debe setear en estado "Rejected".</p> <p>Si el defecto es aceptado, en el momento de resolverlo se debe setear el estado "Fixed".</p>

Proceso	Validar estado defecto
Descripción	<p>Validar si el estado del defecto rechazado aplica o no. Si aplica, poner el defecto en estado "Closed", en caso contrario en "Reopen" y enviárselo al Desarrollador y/o Jefe de Proyecto.</p>

Proceso	Evaluar inicio validación defecto
Descripción	En función del impacto se decide si modificar el diseño o simplemente reagendar la prueba, el cual es realizado una vez que el Ciclo actual se encuentra terminado: <ul style="list-style-type: none">- Alto impacto =>; Revisar Diseño y reejecutar algunos "Conjuntos de Pruebas"- Bajo Impacto =>; Reejecutar algunos "Conjuntos de Pruebas"

Proceso	Cerrar defecto
Descripción	En el caso que un defecto haya sido resuelto, setear el estado "Closed", esto debería ser hecho en el ciclo que ocurre.

5.1.7 Proceso: “Realizar auditoría CM/PPQA”

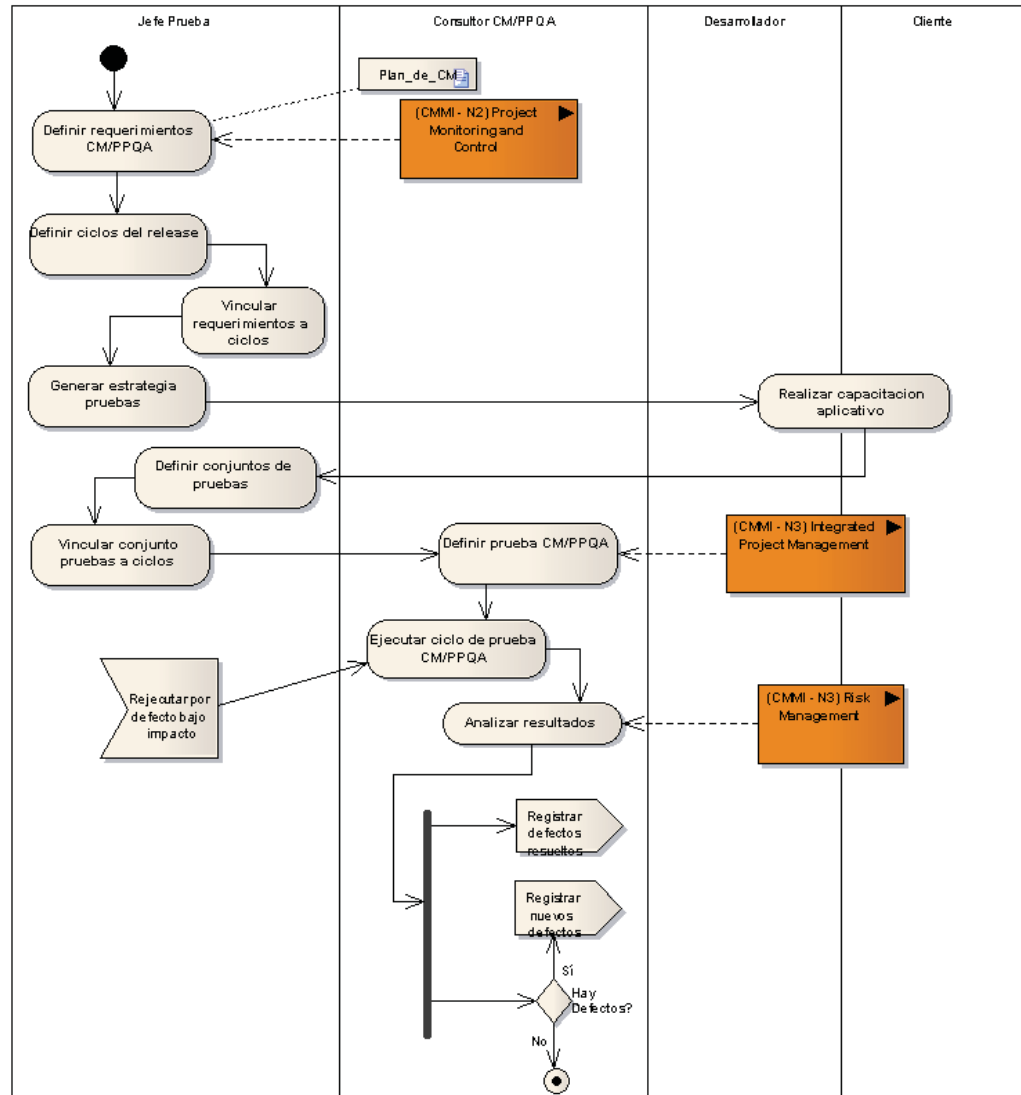


Figura 5.19: Proceso “Realizar auditoría CM/PPQA”

Proceso	Definir requerimientos CM/PPQA (*)
Descripción	<p>La planilla de apoyo "Plan de CM" se encuentra en el ANEXO 1. Dicha planilla apoya a la planificación de perfilamientos y definición de la biblioteca de configuración de los ítems.</p> <p>Propone una estructura base, especificando que debiese tener cada una de ellas.</p> <p>Se definen los requerimientos de CM/PPQA detallando los atributos mencionados a continuación, si es necesario se crean carpetas que agrupen los requerimientos por temas.</p> <ol style="list-style-type: none"> 1. Nombre 2. Especificar Tipo de Requerimiento como de Negocios (CM o PPQA) 3. Descripción 4. Autor 5. Prioridad en la implementación: <ol style="list-style-type: none"> 1-Baja 2-Media 3-Alta 4-Muy Alta 6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)

Proceso	Definir ciclos del release
Descripción	<p>Definir los "Ciclos del Release" indicando los "Conjunto de Pruebas" preliminares que los conforman, por ejemplo el primer "Ciclo de Pruebas" debiera incluir siempre un conjunto de "Pruebas Trazadoras" con el fin de establecer si el ambiente está operativo o no.</p> <p>En QC es posible realizar Attachments de información del Ciclo, como podría ser la planificación del mismo. Pero aparentemente estos elementos no son versionados, por lo que quizá es más recomendable sólo adjuntar Links a Subversion.</p>

Proceso	Vincular requerimientos a ciclos
Descripción	<p>Se debe vincular los requerimientos a al menos un "Ciclo de Prueba" con el fin de:</p> <ul style="list-style-type: none"> - Determinar en cualquier instante si los requerimientos están siendo cubiertos o no - Generar esbozo de pruebas en función de los requerimientos asociados a un ciclo. <p>Un requerimiento puede ser vinculado a un Release, en este caso, todos los "Ciclos de Prueba" dentro del Release serán vinculados, y eso implica que debe haber un "Conjunto de Pruebas" al menos dentro de cada Ciclo que valida el requerimiento.</p> <p>Por el momento, no se realizan Pruebas de Regresión al 100% en todos los ciclos por motivos de</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	recursos. Así la cobertura de prueba en un ciclo queda definida por la liberación de funcionalidades nuevas y por los defectos ocurridos en el ciclo anterior (en el caso que exista), por lo tanto al principio del proyecto, sólo se vincularán los requerimientos asociados a funcionalidades nuevas al ciclo en que se prueban, y con posterioridad dependiendo de los defectos se vincularán requerimientos a ciclos que contengan Pruebas de Regresión.
--	---

Proceso	Generar estrategia pruebas
Descripción	<p>Este documento debe ser un compendio del estado del proyecto hasta este momento, considerando:</p> <ul style="list-style-type: none"> - Roles y personas - Requerimientos de Pruebas - Ciclos de Pruebas - Protocolo de Suspensión de las Pruebas - Conclusiones de Estudio Factibilidad de Automatización. - Referencias a los documentos de Arquitectura de HW/SW. - Formato Plan de Pruebas o Operatoria al respecto - Formato Informe de Resultados de Pruebas. - Manejo de Defectos. - Severidad de Defectos. - Riesgos <p>En el caso de CM-PPQA se incluyen los checklists que se utilizarán en la Revisión.</p>

Proceso	Realizar capacitación aplicativo
Descripción	Realizar Capacitación del Aplicativo, idealmente se debiera hacer una Capacitación para Pruebas Funcionales y No Funcionales, pero podría haber casos donde tenga que ser realizada en sesiones separadas.

Proceso	Definir conjuntos de pruebas
Descripción	<p>Se definen los conjuntos de prueba, ver ejemplos de estos conjuntos en Diagrama Conjunto_Prueba.</p> <p>En QC, Test Lab:</p> <ul style="list-style-type: none"> - Crear carpeta(s) que contienen "Conjuntos de Pruebas". - Crear cada "Conjunto de Pruebas" en la carpeta que corresponda, e ingresar las "Pruebas" que lo componen usando el tab "Execution Grid". El orden de ejecución se define en el tab "Execution Flow".

Proceso	Vincular conjunto pruebas a ciclos
Descripción	En QC, Test Lab, las carpetas que contienen "Conjuntos de Pruebas" son linkeadas a ciclos. Para esto click derecho sobre la carpeta y "Assign to Cycle".

Proceso	Analizar resultados
Descripción	Esta tarea consiste en determinar si los defectos reportados durante las pruebas son tales. - En el caso que el defecto sea tal, actualizar su estado a "Open". - En el caso que el defecto no aplique, se puede borrar o dejar su estado en "Closed", esto último sólo si es de algún aporte mantener el registro.

El ciclo de los defectos o incidentes que se explican en los Modelos anteriores es el siguiente:

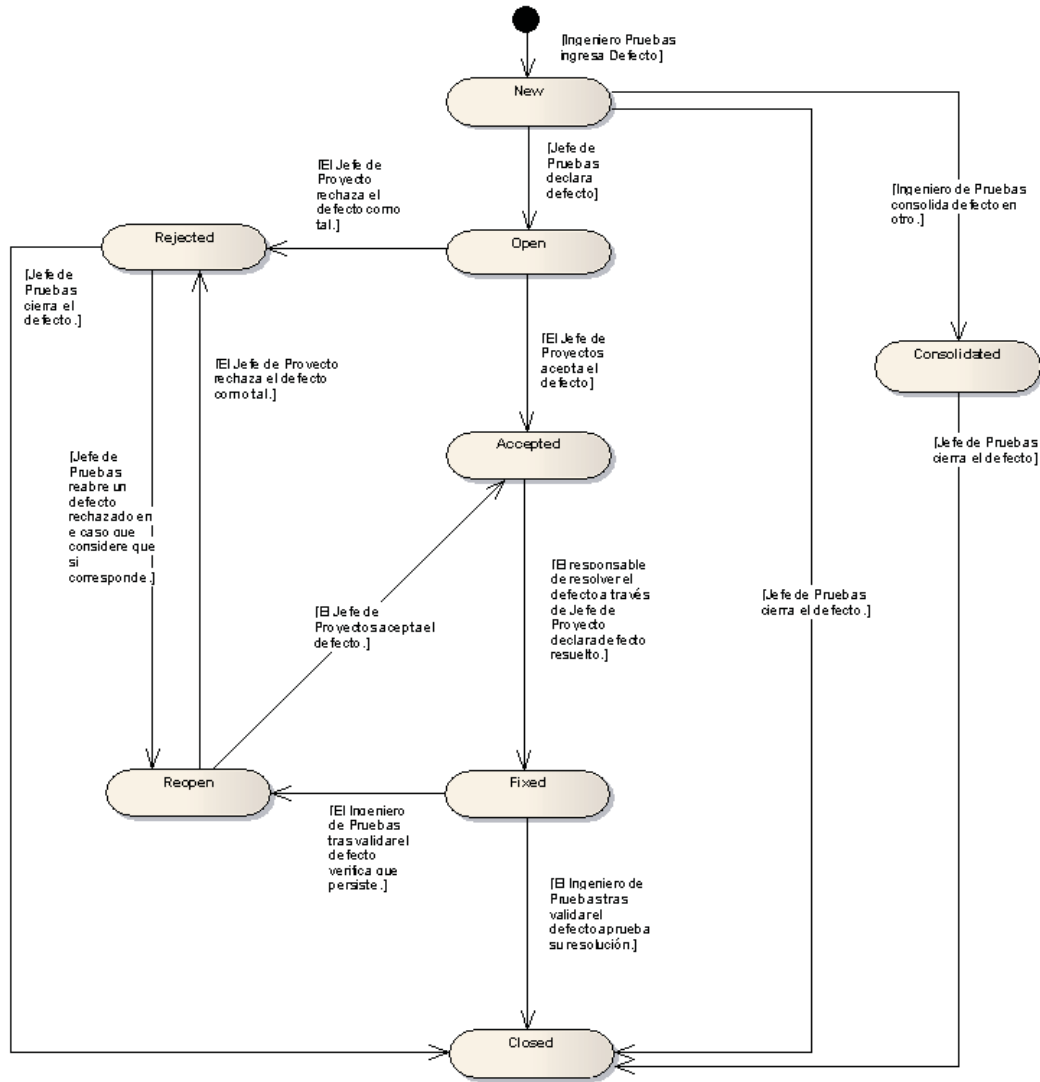


Figura 5.20: “Ciclo de defectos”

5.1.8 Proceso: “Controlar proyecto”

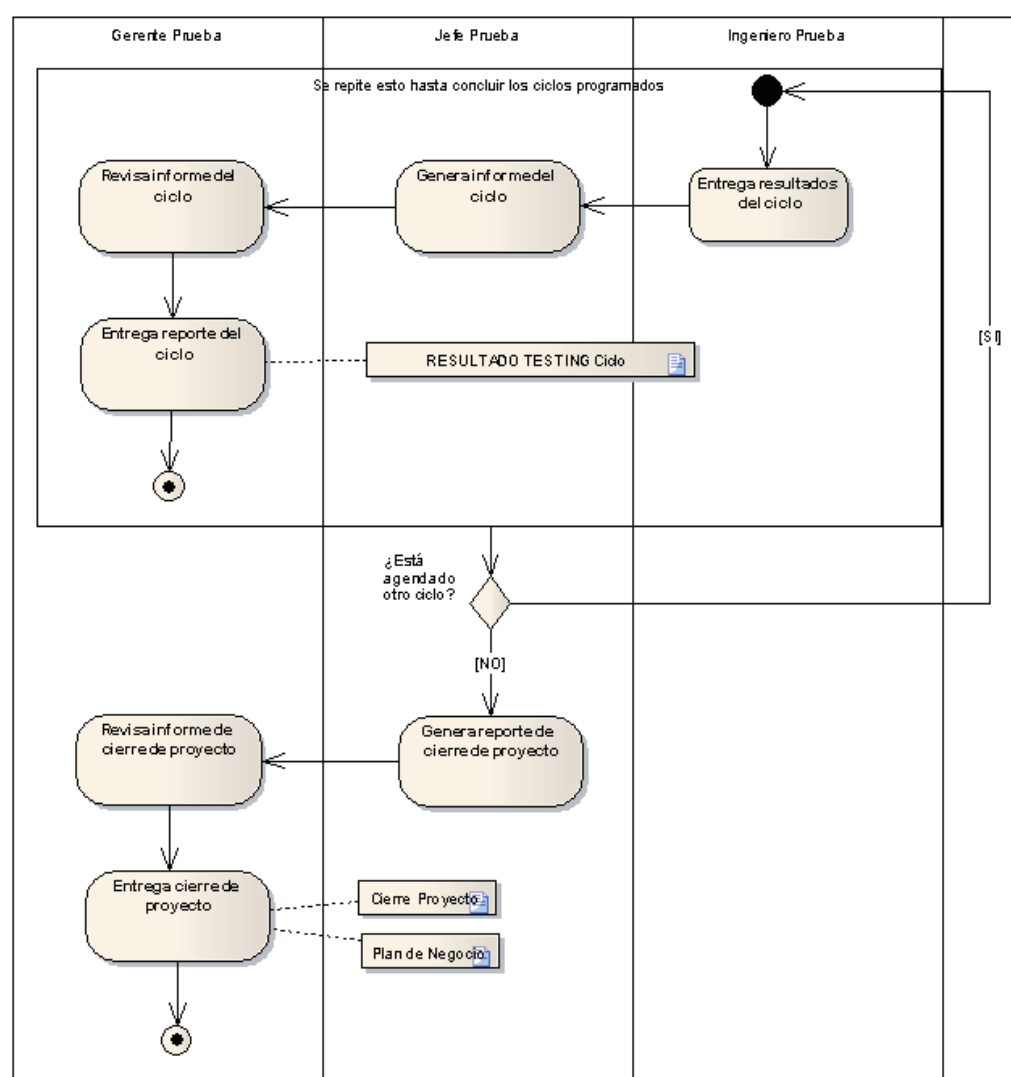


Figura 5.21: Proceso “Controlar proyecto”

Proceso	Entregar resultados del ciclo
Descripción	Una vez ejecutadas las pruebas el Ingeniero de Pruebas entrega los resultados de la ejecución y de los defectos encontrados.
Proceso	Generar informe del ciclo
Descripción	Con la entrega de los resultados el Jefe de Proyecto construye el Informe del Ciclo y lo entrega al Gerente de Pruebas para su revisión
Proceso	Revisar informe del ciclo
Descripción	Cuando el Gerente de proyecto revisa el informe lo libera al equipo de Proyecto para su análisis y revisión en conjunto al equipo de Testing.
Proceso	Entrega reporte del ciclo (*)
Descripción	El Reporte que se genera se basa en la planilla de apoyo “RESULTADO TESTING Ciclo” ANEXO 1. Dicho informe se basa esencialmente en la entrega de los resultados en comparación a las políticas de paso a producción planteadas en la planilla de apoyo “Plan de Pruebas”.
Proceso	Generar reporte de cierre de proyecto (*)
Descripción	En el caso que se terminen de ejecutar todos los ciclos programados o solicitados por emergencia es la hora que el Jefe de Proyecto Genere el reporte de Cierre de Proyecto utilizando la planilla de Apoyo “Cierre Proyecto” y “Plan de Negocio” del ANEXO 1. Luego de generado el reporte, es entregado al Gerente de Proyecto.
Proceso	Revisar informe de cierre de proyecto
Descripción	El Gerente de Proyecto revisa el Informe de Cierre de Proyecto y el Plan de Negocio entregado por el Jefe de Proyecto con el fin de evaluar las diferencias entre lo programado y lo real.
Proceso	Entrega cierre de proyecto
Descripción	Una vez realizada la revisión por el Gerente de Proyecto los entrega y presenta al equipo de proyecto, dejando en conocimiento la desviación u optimización en el caso que exista en el Plan de Negocio junto a los resultados finales de la intervención del Testing.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.2 Resumen de resultados obtenido (Métricas)

A partir de las métricas definidas inicialmente se procede a evaluar cada una de ellas con el fin de poder analizar cómo se comportó la aplicación de la Metodología aplicada al proyecto.

Nº	Medición	Medida	Métrica
1	Se medirá el cumplimiento del esfuerzo establecido para el Testing del producto.	HH	Esfuerzo Estimado v/s Esfuerzo Real
2	Se medirá el cumplimiento del costo establecido para el Testing del proyecto.	UF	Costo Estimado v/s Costo Real
3	Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.	%	% Establecido v/s % Obtenido

- 1) Métrica 1: Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto.

En base al Modelo de Servicio se obtienen los siguientes datos:

Días	HH	
2	13	Jefe de Área
14	112	Jefe de Proyecto
27	218	Externos
43	343	Total Esfuerzos

Según los datos obtenidos del Plan de Negocio se tiene lo siguiente:

Mes 2	Mes 3	Mes 4	
Ago-09	Sep-09	Oct-09	
5.00 hh	11.00 hh	6.50 hh	
90.00 hh	140.00 hh	101.00 hh	
			Total HH planificadas
95.00 hh	151.00 hh	107.50 hh	353.50 hh

Es decir, se planificaron 10 horas más de las inicialmente estimadas.

Lo que realmente se gasta en HH son las siguientes:

Mes 2	Mes 3	Mes 4	
Ago-09	Sep-09	Oct-09	
22.00 hh	24.80 hh	58.00 hh	
72.00 hh	214.70 hh	99.00 hh	
		34.00 hh	
	40.00 hh	9.00 hh	
			Total HH Reales
94.00 hh	279.50 hh	200.00 hh	622.50 hh

Existe una diferencia de 269 HH según la estimación inicial y lo real.

- 2) Métrica 2: Se medirá el cumplimiento del costo establecido para el Testing del proyecto.

En base al Modelo de servicio se obtienen los siguientes datos:

UF	
14	Costo Jefe de Área
107	Costo Jefe de Proyecto
194	Costo Externos
315	Total Costo Proyecto

Según los datos obtenidos del Plan de Negocio, se obtiene lo siguiente:

APROBADO TOTAL COSTO SERVICIO TF	446.33 UF
REAL TOTAL COSTO SERVICIO TF	541.06 UF
DIFERENCIA	-94.73 UF

3) Métrica 3: Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.

Los criterios de aceptación establecidos para este proyecto fueron los siguientes (ver planilla aplicada Plan de Pruebas) son:

- 0% de los Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad INVALIDANTE.
- A lo más, 5% de Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad ALTA.
- A lo más, 10% de Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad Media o Baja.
- A lo menos, 85% de Casos de Prueba Integrados efectivamente probados con estado OK.

Los resultados obtenidos en el ciclo 1 ejecutado son los siguientes:

En base a la necesidad de acortar el ciclo de Testing, finalmente se probó un 62% de las pruebas inicialmente planificadas; es decir, 241 CPI de las 387 CPI planificadas.

El resultado de la ejecución de los 241 CPI arroja los siguientes resultados:

Failed	Passed	Total General	
84	157	241	CPI
35%	65%	100%	Porcentajes

Existen 146 CPI con estado N/A, estos casos son los que NO se ejecutaron en el primer ciclo.

De lo anterior se desprende que el aplicativo tiene un **Nivel de Madurez del 65%** sobre los casos de pruebas **ejecutados**, con los siguientes defectos encontrados:

Cuenta de Criticidad	Criticidad				Total general
	1-Baja	2-Media	3-Alta	4-Invalidante	
Total	14	23	62	7	106
Porcentaje	35%		58%	7%	100%

Se recomienda realizar un CICLO 2 de pruebas, para aclarar si las incidencias encontradas en el ciclo anterior, han sido corregidas o corresponden a errores en la instalación del software, en el ambiente de Testing utilizado. Además no se cumplieron los criterios establecidos en el Plan de Pruebas para el paso a producción.

Los resultados obtenidos en el ciclo 2 ejecutado se describen a continuación:

Se indica la estrategia usada en el ciclo 2 de Certificación del proyecto que considera la redefinición del total de pruebas definidas en el Plan de Pruebas inicialmente aprobado, en base a los cambios solicitados por el proyecto (inicialmente en el ciclo 2 se habían definido 194 CPI a ejecutar).

En base a la nueva estrategia que apunta a minimizar los riesgos de no completar las pruebas del ciclo 1, tal como estaban planificadas, en el ciclo 2 se definió la ejecución de 215 Casos de Prueba que estaban compuestos de la siguiente manera:

Failed	Passed	Total general	
65	150	215	CPI
30%	70%	100%	Porcentajes

De lo anterior se desprende que el aplicativo tiene un **Nivel de Madurez del 70%** sobre el total de casos de pruebas **ejecutados**, con la siguiente distribución de los defectos encontrados:

Cuenta de Criticidad	Criticidad				Total general
	1-Baja	2-Media	3-Alta	4-Invalidante	
Total general	2	27	52	9	90
Porcentajes	32%		58%	10%	100%

Se recomienda realizar un CICLO 3 de pruebas, ya que el Nivel de Madurez obtenido no es suficiente para el Paso a Producción del aplicativo. Además no se cumplieron los criterios establecidos en el Plan de Pruebas para el paso a producción.

Pese a lo anterior, el equipo de proyecto define pasar igualmente el aplicativo a producción con los antecedentes de cuáles son los errores detectados para así declararlos a los clientes finales y realizar seguimiento de su resolución.

Los puntos importantes que se rescatan de la entrevista final con la persona que dirigió la aplicación de la Metodología fueron los siguientes:

- 1) Si en el Modelo se tiene claridad de los costos asociados a cada recurso se puede lograr con más fineza el cálculo de la métrica final.
- 2) Las tasas de rendimiento incorporadas en el Modelo de Servicio distaron mucho de lo real; la tasa de rendimiento de los diseños no fue 5 CPI por hora, lo real se acercó al 2.5 por hora.
- 3) Las tasas de rendimiento incorporadas en el Modelo de Servicio de la ejecución de las pruebas no fue 5 CPI por hora, sino que fue de 3 CPI por hora.
- 4) Además es importante mencionar que los costos utilizados en el Modelo Inicial son distintos a los que finalmente se obtuvieron en el Plan de Negocio. Dichos costos fueron finalmente de 1.01 UF HH del Jefe de Proyecto y de 0.88 UF HH para los Externos, Analistas y Ejecutores.
- 5) A pesar de lo antes expuesto, el proceso fue limpio y claro de aplicar. Se siguiere generar una BD de conocimiento de proyectos con el fin de ir refinando cada vez más la estimación.

En base a las observaciones entregadas anteriormente se realiza el ejercicio de ajustar el Modelo de servicio inicial, obteniendo los resultados que se muestran en la Figura 5.22.

Variables					
Universo de Pruebas	2024	CPI	Cantidad de Gestión LB CM	10	Cantidad
Cobertura	85	Universo Funcionalidades	HH invertidas Gestión LB en TF	1	HH
	58	Funcionalidades a cubrir			
	89	% sobre universo	Tiempo Total de Proyecto	0	Meses
Profundidad	20	% sobre universo		0	Semanas
				0	Días hábiles
Cantidad de Ciclos	405	CPI Ciclo 1 (100%)	Costo total del proyecto	1106	UF
	202	CPI Ciclo 2 (50%)	Costo máximo Propuesta TF	332	UF
Diseños	405	CPI			
Ciclo Extra	202	CPI Ciclo N (50%)	N° de Entregas de SW	1	Entregables de SW
HH Mensuales	160	Horas mes			
HH trabajo Diario Interno	8	HH Dia			
HH trabajo Diario Externo	8	HH Dia			
Jefe de Área			Costos Jefe de Área	161	UF Mes
Cantidad	1			40,3	UF Semana
% Dedicación	5,00%	%		8,05	UF Dia Hábil
Horas mensuales	0,08	HH Mensuales		1,01	UF HH
Costo Mensual					
Costo HH					
Jefe de Proyecto			Costos Jefe de Proyecto	141	UF Mes
Cantidad	1			35,3	UF Semana
% Dedicación	100,00%	Real 50%		7,05	UF Dia Hábil
Horas mensuales	1,6	HH Mensuales		0,88	UF HH
N° Analistas/Ejecutores			Costos HH Externos	141	UF Mes
Cantidad	1			35,2	UF Semana
% Dedicación	100,00%	%		7,04	UF Dia Hábil
Horas mensuales	160	HH Mensuales		0,88	UF HH
Tasa de Rendimiento Diseño	2,5	CPI por HH por Recurso			
	20	CPI por HH trabajo Diario Externo			
Tasa de Rendimiento Ejecución	3	CPI por HH por Recurso			
	24	CPI por HH trabajo Diario Externo			

Figura 5.22: Ajuste del modelo en las variables definidas

En función de los nuevos ajustes se obtienen los costos que se muestran en la Figura 5.23.

Costos Estimación Proyecto			
Cantidad de Pruebas	Días	HH	
	56	452	Total Esfuerzo Proyecto
	2	19	Jefe de Área
	24	189	Jefe de Proyecto
	49	389	Externos
		UF	
		20	Costo Jefe de Área
		167	Costo Jefe de Proyecto
		342	Costo Externos
		528	Total Costo Proyecto

Figura 5.23: Costos por ajuste

De lo anterior se desprende la tabla comparativa (Figura 5.24):

Iteración	Métrica 1: Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto			Métrica 2: Se medirá el cumplimiento del costo establecido para el Testing del proyecto.		
	Planificado	Real	% Desviación	Planificado	Real	% Desviación
Modelo de Servicio Inicial	353.50 HH	622.50 HH	43.2	446.33 UF	541.06 UF	17.5
Modelo de Servicio Ajustado	597 HH	622.50 HH	4.1	528 UF	541.06 UF	2.4

Figura 5.24: Tabla comparativa de las métricas con y sin ajustes

Como se puede ver en la tabla anterior (Figura 5.24), el Modelo se puede ajustar sólo si se precisan los datos de entrada del Modelo de Servicio. Mientras más exactos, más es posible acercarse al real esperado del proyecto.

La métrica de Apego a la Metodología se calcula a continuación:

A	B	C	D	E	F	G	H	I	J	K	L		
Estructura	Plantilla	Propuesta Metodológica	Importancia en Metodología	Apego	Aplicada	Apego por Aplicada	Distribución Importancia	Refinada	Multipl. (Distribución * Refinado)	Apego por Fase	Porcentaje por Estructura		
0. Entradas	Informe de Diseño proveedor(Extracción).doc	CMMI	3	4	12	1	12	0,40	NO	0,9	0,36	0. Entradas	
	PlanificacionSeguimientoSemanal.mpp	CMMI	3	4	12	1	12	0,40	SI	1	0,40		
	Visión y Alcance.doc	CMMI	3	2	6	1	6	0,20	NO	0,9	0,18		0,94
1. Análisis	Matriz_de_Evaluacion_de_Propuestas.xls	CMMI	3	2	6	0	0	0,00	NO	0,9	0,00	1. Análisis	
	Estimación Modelo de Servicios.xls	Cruce	5	4	20	1	20	0,63	SI	1	0,63		
	LB_Requerimientos_Calidad.xls	ITIL	2	3	6	1	6	0,19	NO	0,9	0,17		0,79
2. Diseño	Definición CPI.xls	ITIL	2	4	8	1	8	0,31	NO	0,9	0,28	2. Diseño	
	TF Diseño.xls	ITIL	2	3	6	1	6	0,23	SI	1	0,23		
	Plan_de_CM.doc	CMMI	3	1	3	1	3	0,12	NO	0,9	0,10		
	Plan_de_Pruebas.doc	CMMI	3	3	9	1	9	0,35	NO	0,9	0,31		0,92
3. Ejecución	TF Ejecución.xls	Cruce	5	3	15	1	15	1,00	SI	1	1,00	1,00	3. Ejecución
4. Gestionar Defectos	TF Entidad-Defecto.xls	Cruce	5	3	15	1	15	1,00	SI	1	1,00	1,00	4. Gestionar Defectos
5. Informes	Plan de Negocio.xls	Cruce	5	4	20	1	20	0,50	SI	1	0,50	5. Informes	
	Cierre Proyecto.xlsx	Cruce	5	2	10	1	10	0,25	NO	0,9	0,23		
	RESULTADO TESTING Ciclo.doc	Cruce	5	2	10	1	10	0,25	NO	0,9	0,23		0,95

Figura 5.25: Resultados del apego a la metodología

Métrica de Porcentaje de Apego a la Metodología obtenido = 93%

6 Conclusiones

La Metodología Propuesta se basa fundamentalmente en la necesidad que impone el mercado en entregar servicios de calidad a bajo costo y en los tiempos comprometidos utilizando estándares de mejores prácticas y modelos de servicios que han existido desde hace ya varios años.

Al realizar el cruce de los procesos de CMMI versus las mejores prácticas de ITIL sobre uno de los procesos más importantes a la hora de validar la calidad de un software, se obtiene una Metodología eficiente y consistente de aplicar a cualquier tipo de proyecto de desarrollo. Además la expertise de trabajar en distintos tipos de proyectos en el proceso escogido (Validación y Verificación), apoyó en forma significativa al momento de crear las plantillas de apoyo que ayudan a concretizar la aplicación de la propuesta metodológica a un proyecto real.

El análisis de las métricas en este documento se realiza en función de los dos ciclos ejecutados en el proyecto de Testing, las métricas de Esfuerzo y Costos se realizan en base a los antecedentes entregados por el Jefe de Proyecto de Testing Funcional, en función de que el ciclo debe terminar en la fecha replanificada.

En el cálculo de métricas se puede ver que las variables cuantitativas son posibles de ajustar en función de la precisión y realismo con que se analicen; por ejemplo, las tasas de rendimiento de diseño y ejecución, así como también los costos relacionados al proyecto.

El cálculo de la métrica de Apego a la Metodología ayuda a determinar que en la aplicación de dicha métrica, se establece el grado de utilización efectiva en relación al objetivo de la Metodología. Sin embargo, se pueden recoger mejoras sustanciales en cuanto a la entrevista al jefe de proyecto sobre como fue el proceso de aplicación, ya que él indica que en sí la Metodología le sirvió para mantener el control y orden de los pasos a seguir, además de aumentar la percepción de calidad de servicio por parte del cliente, ya que se había trabajado en otros proyectos con este cliente, sin embargo esta vez las reglas para una mejor entrega del servicio fueron entregadas inicialmente, apoyando el proceso de desarrollo de software en lo que respecta a formalidad en los documentos de diseño y estimación de esfuerzo y planificación más acertada para el desarrollo, ya que consideraron la generación de los documentos necesarios en los momentos reales del ciclo de vida del proyecto y no al final como se estila realizar.

Se propone como trabajo futuro modelar y generar una Base de Conocimiento para registrar los tipos de proyectos y sus mejores ajustes planificados y reales con el fin de mejorar cada vez más el Modelo del Servicio.

También se planteó la iniciativa de llevar a programa la plantilla de Modelado de Servicio, con el fin de que pueda ser accesada desde internet al momento que se desea realizar la estimación de costo o esfuerzo involucrado en un servicio de Testing funcional.

7 Bibliografía

- 1) Axentia (2006), White Paper, Una Introducción a CMMI, Transforming Your IT Organization. (Ver ANEXO 2 entregado en el Informe de Avance), 2006.
- 2) Fernando Sánchez, (2008), CMMI vs. ITIL: dos enfoques complementarios, gerente de Grupo Delaware, <http://www.mkm-pi.com/mkmpi.php?article1817>, Jueves 6 de marzo de 2008.
- 3) Juan Palacio, (2006), Sinopsis de los Modelos SW-CMM y CMMI, <http://www.navegapolis.net/jpalacio@navegapolis.net>, 1.0 Abril – 2006.
- 4) Juan Raggio Pérez, (2004), Universidad Politécnica de Madrid, Facultad de Informática, Estudios de Doctorado, DESARROLLO DE PROCESOS DE GESTIÓN DE SERVICIOS DE EXPLOTACIÓN SIGUIENDO EL MODELO CMMI, 2004.
- 5) Lorena León – Gabriela Puglla, (2008), Métricas de Proceso y proyecto de software, Universidad Técnica particular de Loja. <http://www.slideshare.net/loreknelamorena/mtricas-de-proceso-y-proyecto-de-software>, 2008.
- 6) Mariana Isela Jaramillo González (2005), Utilización de Estándares ITIL para lograr el Nivel 3 de CMMI en una Organización. Universidad Autónoma del Estado de México, <http://docente.uco.mx/juancont/documentos/cap02/6.pdf>, 24 Octubre 2005.
- 7) OGC (2007), ITIL Managing IT Services. Best Practice for Application Management, http://www.ogc.gov.uk/guidance_itol.asp, 2007.
- 8) PMBOK (2004), [PMBOK](#) Tercera Versión en Español [Project Management Institute](#). «Capítulo 1», *Guía de los Fundamentos de la Dirección de Proyectos*, 3ª edición. ISBN 1-930699-73-5. http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos, 2004.
- 9) Raúl Suárez O. y Felipe Donoso Jaurés, (2006), METODOLOGÍA ITIL, Descripción, Funcionamiento y Aplicaciones, PÍA RAMÍREZ BRAVO - FELIPE DONOSO JAURÉS, Santiago, Julio de 2006.
- 10) Serge Thorn Geneva (2007), ITIL and CMMI synergies. <http://sergethorn.blogspot.com/2007/05/itil-and-cmmi-synergies.html>, 23 Mayo 2007.

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**PROPUESTA METODOLÓGICA PARA LA MEJORA DEL
PROCESO DE VERIFICACIÓN Y VALIDACIÓN EN LA
INDUSTRIA DE SERVICIOS TI EN BASE A MODELOS CMMI
CON ITIL**

NATALIA KARIN CARPIO ARNAIZ

TESIS DE GRADO
MAGÍSTER EN INGENIERÍA INFORMÁTICA

Diciembre 2009

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**PROPUESTA METODOLÓGICA PARA LA MEJORA DEL
PROCESO DE VERIFICACIÓN Y VALIDACIÓN EN LA
INDUSTRIA DE SERVICIOS TI EN BASE A MODELOS CMMI
CON ITIL**

NATALIA KARIN CARPIO ARNAIZ

Profesor Guía: Silvana Roncagliolo de la Horra

Programa: Magíster en Ingeniería Informática

Diciembre 2009

Resumen

Cada vez más, a los departamentos de TI se les exigen niveles de calidad y servicio directamente proporcionales a los niveles de calidad y servicio que se exigen a las distintas líneas de negocio de las organizaciones. Esto está provocando un afán por encontrar métodos y técnicas que permitan desarrollar y mantener software de alta calidad, así como implantar y mantener infraestructura con un alto grado de disponibilidad.

Se presenta una propuesta metodológica creada en base a los procesos de Verificación y Validación de CMMI pareados con los Procesos/Funciones de ITIL. En la metodología se exponen las prácticas específicas y se detalla el proceso de cómo implantar la metodología que apoya y mejora los niveles de calidad en el testing funcional de aplicaciones.

A partir de la metodología propuesta se presenta la evaluación de su aplicación a un caso real, definiendo como punto de partida las métricas que se consideran en dicha evaluación, generando mejoras a la propuesta en sí.

Abstract

Everyday, TI Departments are demanded with quality and service levels directly proportional to the quality and service levels demanded to the different business lines of the organizations. This is causing an urge to find methods and techniques that allow to develop and maintain high quality software, such as to implant and keep the infrastructure with a high level of availability.

A methodological proposal is shown, created based on the CMMI verification and validation processes matched against the ITIL processes/functions. In the methodology specific practices are exposed and the quality levels are improved in the functional testing of the applications.

From the proposed methodology, the evaluation of its application to a real case is showed, defining as a start point the metrics considered in that evaluation, generating improvements to the proposal itself.

Índice

1	Introducción.....	9
2	Objetivos	10
3	Estado del Arte	11
3.1	ITIL v2.0.....	11
3.1.1	<i>Descripción (PRamirez-FDonoso, 2006)</i>	<i>11</i>
3.1.2	<i>Objetivos de ITIL (PRamirez-FDonoso, 2006).....</i>	<i>12</i>
3.2	CMMI.....	13
3.2.1	<i>Historia y evolución (JPalacio, 2006)</i>	<i>13</i>
3.2.2	<i>Los Cinco Niveles de Madurez (Axentia, 2006).....</i>	<i>13</i>
3.2.3	<i>Arquitectura del Modelo (Axentia, 2006)</i>	<i>15</i>
3.3	CMMI e ITIL (JRaggio, 2004)	17
3.4	Análisis del Estado del Arte.....	18
4	Solución Propuesta.....	21
4.1	Conocimiento de los modelos.....	21
4.2	Proceso de cruce de los modelos	21
4.3	Definición de métricas para la evaluación final	25
4.3.1	<i>Tiempo</i>	<i>26</i>
4.3.2	<i>Costo.....</i>	<i>26</i>
4.3.3	<i>Alcance</i>	<i>26</i>
5	Implantación de la metodología propuesta y aplicación de plantillas de apoyo	32
5.1	Proceso para aplicar la metodología	32
5.1.1	<i>Proceso “Definir proyecto”</i>	<i>34</i>
5.1.2	<i>Proceso “Administrar externo”.....</i>	<i>37</i>
5.1.3	<i>Proceso “Administrar proceso de pruebas” PARTE 1.....</i>	<i>40</i>
5.1.3.1	<i>Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”.....</i>	<i>45</i>
5.1.3.2	<i>Proceso “Estimar grado automatización y nivel diagnóstico”.....</i>	<i>46</i>
5.1.3.3	<i>Proceso “Definir requerimientos no funcionales”</i>	<i>51</i>
5.1.4	<i>Proceso “Administrar proceso de pruebas” PARTE 2.....</i>	<i>53</i>
5.1.4.1	<i>Proceso “Definir prueba”</i>	<i>57</i>
5.1.4.2	<i>Proceso “Automatizar prueba unitaria”.....</i>	<i>61</i>
5.1.4.3	<i>Proceso “Automatizar prueba integrada”</i>	<i>63</i>
5.1.4.4	<i>Proceso “Generar LB datos”.....</i>	<i>64</i>
5.1.4.5	<i>Proceso “Ejecutar ciclo pruebas”</i>	<i>66</i>
5.1.4.6	<i>Proceso “Verificar ambiente prueba”</i>	<i>68</i>
5.1.4.7	<i>Proceso “Ejecutar conjunto de pruebas”</i>	<i>69</i>
5.1.5	<i>Proceso “Administrar proceso de pruebas” PARTE 3.....</i>	<i>70</i>

5.1.6	Proceso: “Gestionar defectos”.....	71
5.1.7	Proceso: “Realizar auditoría CM/PPQA”	74
5.1.8	Proceso: “Controlar proyecto”.....	79
5.2	Resumen de resultados obtenido (Métricas).....	81
6	Conclusiones	89
7	Bibliografía	90

Tabla de Ilustraciones

Figura 3.01: Modelo de Arquitectura CMMI.....	15
Figura 3.02: Niveles de Madurez CMMI.....	16
Figura 3.03: Representación escalonada. Por Nivele de Madurez CMMI.....	17
Figura 4.01: Tabla de mapeo CMMI e ITIL.....	22
Figura 4.02: Metodología propuesta.....	24
Figura 4.03: Triángulo de restricciones.....	26
Figura 4.04: Métricas definidas para validación de la implantación de metodología propuesta.....	28
Figura 4.05: Métrica “Importancia Metodológica”.....	28
Figura 4.06: Métrica “Propuesta Metodológica”.....	29
Figura 4.07: Tabla “Cálculo de métricas”.....	29
Figura 4.08: Tabla “Apego de la metodología propuesta”.....	30
Figura 5.01: Identificación planillas de apoyo.....	32
Figura 5.02: Proceso “Definir proyecto”.....	34
Figura 5.03: Proceso “Administrar externo”.....	37
Figura 5.04: Proceso “Administrar proceso de pruebas” PARTE 1.....	40
Figura 5.05: Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”.....	45
Figura 5.06: Proceso “Estimar grado automatización y nivel diagnóstico”.....	46
Figura 5.07: Proceso “Habilitar PC para captura”.....	50
Figura 5.08: Proceso “Definir requerimientos no funcionales”.....	51
Figura 5.09: Proceso “Administrar proceso de pruebas” PARTE 2.....	53
Figura 5.10: Proceso “Definir prueba”.....	57
Figura 5.11: Proceso “Automatizar prueba unitaria”.....	61
Figura 5.12: Proceso “Automatizar prueba integrada”.....	63
Figura 5.13: Proceso “Generar LB datos”.....	64
Figura 5.14: Proceso “Ejecutar ciclo pruebas”.....	66
Figura 5.15: Proceso “Verificar ambiente prueba”.....	68
Figura 5.16: Proceso “Ejecutar conjunto de pruebas”.....	69
Figura 5.17: Proceso “Administrar proceso de pruebas” PARTE 3.....	70
Figura 5.18: Proceso “Gestionar defectos”.....	71
Figura 5.19: Proceso “Realizar auditoría CM/PPQA”.....	74
Figura 5.20: “Ciclo de defectos”.....	78
Figura 5.21: Proceso “Controlar proyecto”.....	79
Figura 5.22: Ajuste del modelo en las variables definidas.....	86
Figura 5.23: Costos por ajuste.....	87
Figura 5.24: Tabla comparativa de las métricas con y sin ajustes.....	87
Figura 5.25: Resultados del apego a la metodología.....	88

Glosario

CPU: Caso de Prueba Unitario, corresponde a la descripción de una prueba unitaria que posee un resultado y puede ser constituyente de un Caso de Prueba Integrado (CPI).

CPI: Caso de Prueba Integrado, corresponde a la descripción de aquella prueba cuyo objetivo es validar una funcionalidad o una operación del aplicativo. Está constituida por más de un CPU.

Ciclo: Es una ventana de tiempo definida en la planificación que corresponde a un grupo de iteraciones de pruebas con sus respectivos datos para cada versión liberada.

Iteración: Es una instancia de pruebas para una o un grupo de funcionalidades, interfaces y/o reportes que podría repetirse dentro de un ciclo, esto dependiendo del grado de aprobación, en caso de cumplir con todos los criterios de aceptación para la primera iteración, no es necesario seguir con las siguientes.

Pruebas con Error: Corresponde a aquellas pruebas que luego de ser ejecutadas presentan algún tipo de falla que implica que el estado de ejecución de éstas no sea óptimo.

Pruebas de Aceptación de Usuario: Son aquellas diseñadas y ejecutadas por el usuario para demostrar que la aplicación satisface los requerimientos. En inglés: User Acceptance Testing.

Pruebas de Nuevas Funcionalidades: Corresponde a aquellas pruebas que apuntan a validar aquella operatoria del sistema que no ha sido certificada en ninguna etapa de pruebas anterior.

Pruebas de Regresión: Es un "Conjunto de Pruebas" realizado para identificar "defectos" que causan "Regresión". Si el desarrollador resuelve un defecto, se deben realizar pruebas en los alrededores para verificar que no se haya impactado el resto del código, a esto se le llama regresión. En inglés: Regression Test.

Pruebas Trazadoras: Es un "Conjunto de Pruebas" corridas sobre una cierta "Release" para asegurar que es suficientemente estable para continuar con el "Ciclo" de pruebas activo. Estas pruebas usualmente son un subconjunto del conjunto completo de pruebas, preferiblemente de tipo automatizado, que involucran cada parte del sistema al menos en una forma superficial. Unas buenas pruebas trazadoras también mantienen el sistema corriendo por un lapso de tiempo suficiente como para que se manifiesten problemas "gruesos" de confiabilidad y disponibilidad. En inglés: Smoke Test o Sanity Test.

Requerimientos de Negocio: Corresponde a aquel requerimiento que es generado por el cliente y que representa alguna funcionalidad del sistema a validar. En este caso, el responsable de dicho requerimiento es el equipo de Gestión de Proyecto.

Requerimientos de Prueba Funcional: Corresponde a aquel requerimiento generado a partir de un Requerimiento de Negocio, representa el alcance de las pruebas que se realizarán para abordar la validación del requerimiento de negocio respectivo. Este tipo de requerimiento es de responsabilidad del equipo de Calidad.

1 Introducción

Cada vez más, a los departamentos de TI se les exige niveles de calidad y servicio directamente proporcionales a los niveles de calidad y servicio que se exige a las distintas líneas de negocio de las organizaciones.

Esto está provocando un afán por encontrar métodos y técnicas que permitan desarrollar y mantener software de alta calidad, así como implantar y mantener infraestructuras con un alto grado de disponibilidad.

En los últimos años han aparecido dos líneas metodológicas que están consiguiendo un gran protagonismo por su carácter práctico y por los buenos resultados obtenidos: CMMI e ITIL. La primera se centra en los procesos asociados al desarrollo de software, mientras que la segunda enfatiza en el enfoque a servicios (FSanchez, 2008).

Según revisión en la Web, no existe actualmente una Propuesta tan específica como ésta, sólo se presentan cruces a nivel completo de Metodología CMMI y Mejores Prácticas de ITIL, todas en un marco genérico.

Esta Propuesta apunta específicamente al cruce y presenta una Metodología en la implantación del proceso de Verificación y Validación de CMMI basándose en la Mejores Prácticas de ITIL.

El trabajo presenta una guía clara de cómo enfrentar un proyecto de Testing tanto desde su estimación pasando por el diseño, ejecución, concluyendo por el análisis de los resultados, cierre y revisión del cierre de proyecto en lo que respecta a su Plan de negocio inicial.

Además de presentar la Metodología, se presentan plantillas de apoyo para dicho proceso y se aplica a un proyecto real con el fin de poder refinar la Propuesta y evaluar los resultados en base al apego obtenido al aplicar dicha Metodología, resultando de este trabajo una Propuesta validada y refinada desde su concepción.

2 Objetivos

El objetivo propuesto en este proyecto es:

- Generar una propuesta Metodológica de apoyo para la implantación de Testing en la Industria de TI en base al proceso de CMMI Verificación y Validación con las Mejores Prácticas de ITIL, comprobando su aplicabilidad con un caso real.

Lo anterior se realizará a través de la persecución de los siguientes objetivos específicos:

- Proponer una Metodología de apoyo para la implantación de Testing en la Industria de TI en base al proceso de CMMI Verificación y Validación, cruzándolo con las Mejores Prácticas de ITIL.
- Generar los documentos de apoyo (plantillas) para la aplicación de la Metodología Propuesta, especificando su utilización en cada uno de los procesos que se definen en la Metodología que apoya al proceso de Verificación y Validación de CMMI.
- Generar métricas para evaluar la correcta implementación de la Metodología Propuesta.
- Aplicar la Metodología a un caso real (Proyecto Gestor de Oportunidades), desde su fase de evaluación hasta la ejecución del Testing, para así evaluar la efectividad en la aplicación de la Metodología en base a la definición de métricas para su evaluación.
- Evaluar la aplicación de la Metodología en base a las métricas propuestas.

3 Estado del Arte

Al iniciar el estado del arte es importante presentar los conceptos, así como los orígenes de la metodología y las mejores prácticas, lo que se encuentra en las siguientes secciones.

3.1 ITIL v2.0

3.1.1 Descripción (PRamirez-FDonoso, 2006)

Information Technology Infrastructure Library (ITIL), es una metodología que se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones en sus Sistemas de Información y Tecnologías de Información. Esta metodología fue desarrollada a petición del Gobierno del Reino Unido a finales de los 80 y recoge las mejores prácticas en la gestión de los Sistemas de Información. Desde entonces se ha ido extendiendo su uso en la empresa privada, tanto multinacional como PYME, llegando a ser considerado un estándar de facto para la gestión de esta área de la empresa.

En un entorno donde los periodos de disponibilidad de los servicios son cada vez más amplios, donde las exigencias del cliente son cada vez más elevadas, donde los cambios en los negocios son cada vez más rápidos, es muy importante que los Sistemas de Información estén adecuadamente organizados y alineados con la estrategia del negocio. Igualmente ITIL, ofrece toda una serie de definiciones de conceptos típicos de los Sistemas de Información para garantizar que todos sus conocedores hablen de lo mismo, reduciendo así los tiempos y riesgos por malas interpretaciones.

ITIL, es un set de documentos donde se describen los procesos requeridos para la gestión eficiente y efectiva de los Servicios de Tecnologías de Información dentro de una organización. Son un conjunto de mejores prácticas y estándares en procesos para hacer más eficiente el diseño y administración de las infraestructuras de datos dentro de la organización. Es un “marco de trabajo” (framework) para la Administración de Procesos de TI.

Basándose en el principio de mejora continua, ITIL fue madurado y de los 40 libros originales se redefinieron; se agruparon y eliminaron redundancias y actualmente se presenta en 7 libros:

- 1) Service Support
- 2) Service Delivery
- 3) Security Management
- 4) Application Management
- 5) The Infrastructure Management
- 6) Planning to Implement Service Management
- 7) The Business Perspective

Esta metodología se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones. Garantizando así los niveles de servicio establecidos entre la organización y sus clientes.

El objetivo de ITIL es diseminar las mejores prácticas en la Gestión de Servicios de Tecnologías de Información. Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios IT, al mismo tiempo de garantizar los requerimientos de la información en cuanto a seguridad, mantienen e incrementan sus niveles de fiabilidad, consistencia y calidad.

ITIL brinda una descripción detallada de un número de prácticas importantes en TI, a través de una amplia lista de verificación, tareas, procedimientos y responsabilidades que pueden adaptarse a cualquier organización. ITIL describe una aproximación sistemática y profesional a la Gestión de Servicios TI, haciendo énfasis en la importancia clave de cumplir con los requerimientos del negocio respetando los costos acordados.

3.1.2 Objetivos de ITIL (PRamirez-FDonoso, 2006)

El objetivo que persigue ITIL es diseminar las mejores prácticas en la gestión de servicios de Tecnologías de Información de forma sistemática y coherente. El planteo principal se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos.

Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios de TI, al mismo tiempo que se garantizan los requerimientos de la información en cuanto a seguridad manteniendo e incrementando sus niveles de fiabilidad, consistencia y calidad.

La filosofía ITIL adopta la gestión de procesos y considera que, para lograr los objetivos claves de la Administración de Servicios estos procesos deberían ser usados por las personas y las herramientas efectiva, eficiente y económicamente en el desarrollo de la alta calidad y la innovación de los servicios de TI alineados con los procesos de negocio.

Los estándares ITIL exigen un replanteamiento del área tecnológica y la definición de los elementos y procesos "críticos" dentro de la empresa.

3.2 CMMI

3.2.1 Historia y evolución (JPalacio, 2006)

1984 El Congreso del Gobierno Americano aprobó la creación de un organismo de investigación para el desarrollo de modelos de mejora para los problemas en el desarrollo de los sistemas de software, y evaluar la capacidad de respuesta y fiabilidad de las compañías que suministran software al Departamento de Defensa.

Creación del SEI (Instituto de Ingeniería del Software), fundado por el Departamento de Defensa Americano y la Universidad Carnegie Mellon.

1985 SEI empieza a trabajar en un marco de madurez de procesos que permita evaluar a las empresas productoras de software.

La investigación evoluciona hacia el “Modelo de Madurez de las Capacidades (CMM)”.

1991 En agosto SEI publica la versión 1.0 del Modelo de Madurez de las Capacidades para el Software (SWCMM, Capability Maturity Model for Software).

1993 SEI publica la versión 1.1 de SW-CMM.

1997 Publicación de la versión 1.2.

2000 SW-CMM fue integrado y relevado por el nuevo modelo CMMI.

2006 Se publica CMMI –DEV, V1.2, CMMI for Development, Version 1.2.

2007 Se publica CMMI-ACQ, V1.2, CMMI for Acquisition, Version 1.2, *Improving processes for acquiring better products and services*.

La propuesta metodológica que se expone a continuación, se realiza sobre CMMI-ACQ, ya que las empresas de desarrollo pueden contar con externalización de servicios a la hora de desarrollos de software.

3.2.2 Los Cinco Niveles de Madurez (Axentia, 2006)

En general, los niveles de madurez suelen explicarse en orden creciente; se tomará aquí una dirección distinta y se explicarán exactamente al revés: desde el nivel cinco al nivel uno.

Imagine por un momento que está en una organización de nivel cinco. En este tipo de organizaciones, los procesos son analizados para eliminar las causas comunes de variación, o sea, aquellas que tienen que ver con la misma naturaleza del proceso, no atribuibles a causas externas. Las variaciones en las salidas de los procesos son al azar, pero se encuentran controladas estadísticamente (se puede predecir los resultados de los procesos con cierto nivel de confiabilidad).

Para poder llegar a este nivel, la organización debió primero haber eliminado las causas especiales de variación, aquellas que tienen que ver con causas externas, como por ejemplo falta de entrenamiento del personal, problemas con las herramientas, etc. Este tipo de causas no son aleatorias: Si se examinan los resultados se podrán ver las tendencias que claramente indican que las variaciones tienen un origen concreto. En una organización de nivel cuatro, entonces, las causas especiales de variación son identificadas y eliminadas.

Para poder llegar a identificar causas de variación se necesita tener un proceso estándar: difícil sería poner bajo control estadístico un proceso que no se encuentre mínimamente formalizado.

Así se llega al nivel tres, en el cual los proyectos emplean un proceso productivo adaptado del proceso estándar de la organización. Las actividades técnicas y de gestión son realizadas de acuerdo a políticas, procesos y procedimientos formalizados en algún tipo de estándar organizacional profundamente arraigado en la cultura. La gente está entrenada y dispone de recursos para poder hacer su trabajo. También hay una infraestructura básica (personal, herramientas, etc.) para definir y mejorar el proceso productivo.

Pero para poder llegar a esta situación es necesario pasar por una etapa previa: difícilmente se puede introducir en una organización prácticas estándar relacionadas con la ingeniería del producto (análisis, diseño, etc.) si no se ofrece un contexto en donde ellas puedan ser correctamente ejecutadas. Ese es el foco del nivel dos: poner en orden las prácticas relacionadas con el manejo elemental de los proyectos.

En el nivel dos, los proyectos de la organización siguen algún tipo de proceso para realizar las actividades relacionadas con la gestión del proyecto (planificación, control), para administrar los requerimientos y las configuraciones, y para medir y analizar la calidad de los productos y el desempeño de los procesos. También hay prácticas de aseguramiento de la calidad que permiten garantizar que cada proyecto sigue sus propios estándares.

Y así se llega al nivel uno: La situación aquí es caótica. No existen procesos (no al menos en el sentido del modelo) y la *performance* de los proyectos depende profundamente de la buena voluntad y la capacidad de la gente.

3.2.3 Arquitectura del Modelo (Axentia, 2006)

En la representación por niveles (ver Figura 3.01), cada nivel de madurez contiene varias áreas de proceso, las que a su vez quedan definidas por uno o varios objetivos específicos y un objetivo genérico. Cada uno de ellos tiene vinculado un conjunto de prácticas, llamadas específicas y genéricas respectivamente.

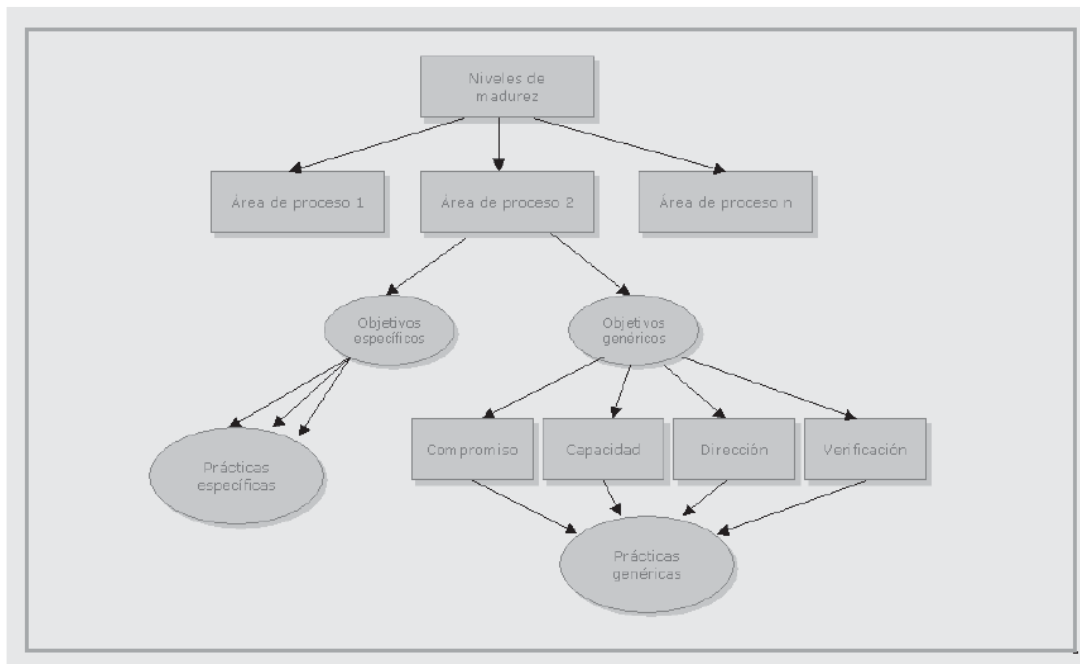


Figura 3.01: Modelo de Arquitectura CMMI

CMMI ofrece una representación sistemática y estructurada para abordar el modelo basado en un proceso de mejora que permite avanzar una etapa a la vez. El logro de cada etapa asegura que el proceso se ha establecido, y cuenta con la infraestructura adecuada que permite avanzar a la siguiente etapa.

Las áreas de proceso del modelo están organizadas en 5 niveles de madurez que representan, cada una, alguna conjetura respecto del proceso de mejora, lo que se muestra en la Figura 3.02.

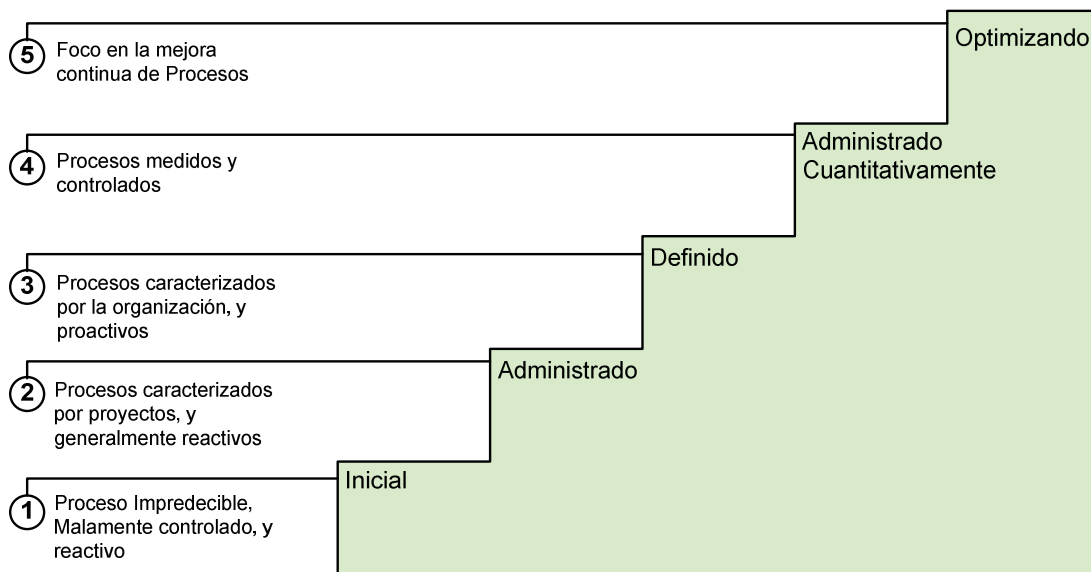


Figura 3.02: Niveles de Madurez CMMI

Esta representación sugiere un camino para la mejora de acuerdo al nivel de madurez de la organización, desde el nivel “Inicial” al nivel “Optimizando”.

El logro de cada nivel de madurez garantiza que un nuevo estado de mejora ha sentado las bases para el siguiente nivel, permitiendo así una mejora incremental.

La representación por niveles de madurez es una buena opción para mejorar un conjunto específico de procesos en cada etapa. Este orden se ha determinado a través de más de una década de investigación y experiencia.

Las áreas de proceso que contiene cada nivel de madurez se muestran en la Figura 3.03.

Nivel	Foco	Áreas de Procesos	Calidad
5 Optimizado	Proceso de Mejoramiento Continuo	Innovación organizacional y Desarrollo Análisis Causal y Resolución	Productividad
4 Administrado Cuantitativamente	Administración Cuantitativa	Ejecución proceso organizacional Administración cuantitativa del proyecto	
3 Definido	Estandarización del Proceso	Desarrollo de Requerimientos Solución Técnica Integración del producto Verificación Validación Foco en proceso organizacional Definición del proceso organizacional + IPPD Entrenamiento organizacional Administración del proyecto integrado + IPPD Administración de Riesgo Análisis de Decisión y Resolución	
2 Administrado	Administración de Proyecto Básico	Administración de Requerimientos Planificación del Proyecto Monitoreo y Control del Proyecto Administración de Acuerdo con Proveedor Medición y Análisis Aseguramiento de la Calidad de Proceso y Producto Administración de la Configuración	
1 Inicial			Riesgo Reelaboración

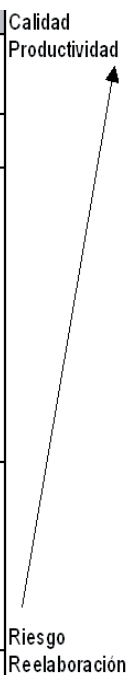


Figura 3.03: Representación escalonada. Por Nivele de Madurez CMMI

3.3 CMMI e ITIL (JRaggio, 2004)

Como se ha comentado anteriormente CMMI es una colección de buenas prácticas alineadas con el concepto de modelos de madurez. Debido a la herencia de SW/CMM y tal y como se ha visto en la descripción de las áreas de procesos, objetivos y las prácticas, está muy orientado hacia desarrollo y mantenimiento de software.

Si bien CMMI es la evolución e integración de un conjunto de modelos de madurez, el desarrollo y mantenimiento de software, fijan en gran medida sus líneas de actuación. En el otro punto se encuentra ITIL, modelo totalmente orientado a los servicios de mantenimiento y gestión de las operaciones.

ITIL y CMMI son dos modelos de madurez claramente distintos pero no mutuamente excluyentes. Las principales diferencias se deben a que CMMI está focalizado en la madurez de los procesos software a través de una actividad de mejora continua e ITIL se centra en el entendimiento y desarrollo de todas las áreas referentes a la infraestructura de TI, además de incorporar el ciclo de vida del hardware dentro de una organización.

3.4 Análisis del Estado del Arte

El término “nivel de calidad” es aplicable directamente a software o aplicaciones que se desarrollan, pero este término es demasiado subjetivo ya que depende de la definición y/u objetivos de cada empresa que desea implantar este “nivel de calidad”. Actualmente las empresas de TI han involucrado el concepto pero en base a criterios y/o definiciones que nacen por experiencias propias.

Al hacer una revisión en la Web, se pueden encontrar diversos estudios y propuestas de solución en lo que respecta a mejoras al “nivel de calidad” de los productos, pero la mayoría de ellos se enfoca a soluciones generales que condicionan el testing a mejoras en los procesos iniciales de construcción del producto.

La gran mayoría de los análisis y estudios existentes, se basa fundamentalmente en resaltar la sinergia que existe entre CMMI e ITIL, proponiendo distintos puntos de vista a nivel de cómo fusionar o complementar éstos en función de aprovechar ambos en la aplicación de la industria para conseguir mejores “niveles de servicio”.

En general ITIL se puede combinar con CMMI para cubrir todo, pero no aborda el desarrollo de los sistemas de gestión de la calidad. Además, no está orientado a los procesos de desarrollo de software y su uso depende en gran medida de la interpretación. Mientras CMMI es el estándar de calidad de fábrica para los procesos de desarrollo de software, ITIL para muchos es la herramienta de elección para las operaciones y la parte de infraestructura de TI, especialmente para servicios de TI (SThorn 2007).

ITIL y CMMI se aplican de mejor manera a diferentes partes de la organización de TI:

- El uso de CMMI en el desarrollo de aplicaciones
- El uso de CMMI en las TIC. Específicamente en los proyectos de infraestructura.
- Uso de ITIL en las operaciones de TI y servicios

Dentro de las propuestas existentes con un poco más de detalle a nivel de cruces entre la metodología y las mejores prácticas, se encuentra un modelo en el que las mejores prácticas de ITIL ayudan a lograr las metas requeridas para obtener un nivel particular de CMMI, por ejemplo nivel 3 (MJaramillo, 2005).

El modelo apunta a generalizar, a través de los niveles de ITIL como apoyar a CMMI para llegar al nivel 3.

1) Nivel de Servicio de Soporte:

Se lleva a cabo la administración de cada proceso planeado en la primera etapa de CMMI permitiendo así la interacción con las partes involucradas y/o afectadas, alcanzando una eficiente comunicación de usuario y proveedor,

los cuales adquieren un compromiso con el plan de trabajo ya establecido y por último llevan a cabo un mantenimiento continuo del plan.

Este nivel de ITIL también ayuda al Monitoreo y Control del Proyecto, es decir, logra el entendimiento de los procesos para tomar decisiones correctivas adecuadas.

2) Nivel de Entrega de Servicios:

Describe los procesos necesarios de entrega con respecto a la calidad y costo lo cual permite llevar a cabo una eficiente Administración de Proveedores (Etapa de CMMI), la cual analiza todo tipo de adquisición, los contratos, etc.

Por lo que participa en la Administración del Proyecto para su Desarrollo Integral (Etapa de CMMI), donde se considera el grado de involucramiento de quienes administran el proyecto.

3) Nivel de Administración de Seguridad:

Mediante este nivel se logra proteger la tecnología de información de la compañía motivo que contribuye a mejorar la “Administración de Riesgos”, que lleva a cabo CMMI, etapa que persigue evitar los peligros que impidan lograr los objetivos del proyecto.

4) Nivel de Perspectiva del Negocio:

Es indispensable para CMMI ya que mediante este nivel el gerente debe conocer el impacto de la tecnología de información y las medidas que deberá tomar para protegerse de situaciones adversas logrando así la Integración del Equipo de Trabajo, etapa de CMMI que pretende que todas las partes involucradas compartan sus habilidades y experiencias a fin de enriquecer el trabajo de la organización trayendo como resultado la calidad de cada proceso que desarrolla.

5) Nivel de Administración de la Infraestructura de TIC:

Mediante este nivel se logra tener un control de calidad con respecto a todos los servicios de comunicación que proporciona la organización facilitando así la Administración Integral de Proveedores que lleva a cabo CMMI evaluando de esta forma sus procesos y productos y haciendo los ajustes en la relación y en sus contratos evitando así comprar productos o servicios obsoletos para la organización.

6) Nivel de Administración de las Aplicaciones:

Determina las soluciones para cubrir las necesidades de los usuarios de acuerdo a los recursos y presupuesto con los que se cuenta logrando así brindar información importante a la Administración Cuantitativa del Proyecto (Etapa de CMMI) la cual considera el tiempo, mano de obra necesaria, y los costos de los procesos definidos por la organización cumpliendo de esta forma los objetivos establecidos en calidad y eficiencia.

7) Nivel de Planeación para la Administración de Servicios:

Ayuda a las organizaciones a identificar sus fortalezas y debilidades generando así que CMMI logre definir aún mejor sus procesos para un progreso en su madurez.

A partir de los análisis antes expuestos, se desprende que en su mayoría, todo apunta a refundir la metodología y sus mejores prácticas a nivel general, sin embargo es necesario contar con propuestas específicas sobre los procesos de CMMI versus ITIL, con el fin de entregar propuestas metodológicas que apoyen el desarrollo de un proceso específico sin ánimo de estandarizar el desarrollo del proceso, sino más bien orientar en cómo y cuándo aplicar las mejores prácticas en un proceso determinado en base a experiencias vividas.

Por lo anterior se desea proponer una Metodología de apoyo en la implantación del proceso de Testing de software y/o aplicativos en base a los Modelos antes mencionados.

Se analizaron los puntos de semejanza entre dichos Modelos, sin olvidar la diferencia entre sus enfoques, para así identificar los procesos de apoyo específicos al Testing. Se propone una Metodología clara y precisa para el levantamiento y mejora de los procesos existentes, con el fin de solucionar la subjetividad que existe en la implantación del proceso de pruebas, así como también en los procesos de apoyo para entrega de los servicios relacionados al Testing.

La Metodología Propuesta, puede ser aplicada tanto en departamentos de TI que pertenezcan a empresas pequeñas, que no pueden optar a obtener una certificación de los Modelos base, o bien en departamentos de TI que a pesar de contar con presupuesto para optar por estas certificaciones no cuentan con la experiencia en la implantación de servicios de Testing.

El cruce se realizó en base a las prácticas concretas y se utilizó como inicio el análisis de ITIL (PRamirez-FDonoso, 2006), ya que el enfoque a servicios debe ser la base de la Metodología que se desea generar. Sin embargo como el enfoque a nivel de procesos apunta a Verificación y Validación (Testing), se consideró sólo el detalle de los procesos que tienen paridad; es decir, Problem Management (Administración de Problemas) y Application Management (Administración de Aplicaciones).

El proyecto presenta la Metodología creada en base a los procesos de Verificación y Validación de CMMI pareado con los procesos/funciones de ITIL.

Los procesos más importantes que se incluyen en el flujo fueron descritos. El Modelo propuesto tiene inmerso los procesos de apoyo tanto de CMMI como de ITIL (JPalacio, 2006) (Axentia, 2006) (JRaggio, 2004) (OGC, 2007).

4 Solución Propuesta

4.1 Conocimiento de los modelos

Para comenzar se despliega el alcance de la solución propuesta en base a los antecedentes entregados en el Estado del Arte, con el propósito de establecer los puntos en los cuales estos Modelos son semejantes y así poder orientar el análisis y generación de la Metodología de trabajo para el proceso de Testing de Software aplicativos.

Una vez que existe claridad en la definición de los Modelos se comienza con el pareo.

4.2 Proceso de cruce de los modelos

A partir de los 7 libros existentes en ITIL se analizarán:

- Service Support
- Service Delivery
- Application Management

Se realiza un análisis de cada uno de ellos con el fin de encontrar semejanzas a nivel de prácticas concretas con CMMI. La selección de estos libros es en base a conocimiento experto por la certificación obtenida en ITIL. Como la orientación es sobre implantar Testing, se sabe de antemano que estos libros son los que apoyan de manera más directa dicha implantación.

Por su parte el Modelo CMMI cuenta con 22 procesos que incluyen en forma estricta las mejores prácticas a aplicar a nivel de desarrollo de software.

Analizando entonces a nivel de prácticas concretas se pueden mapear algunos procesos de CMMI a ITIL en lo que respecta a soporte y gestión de proyectos.

A continuación se presenta un mapeo de los procesos/funciones de los Modelos CMMI e ITIL. Estos Modelos están enfocados a diferentes áreas, por lo cual guardan escasa relación, pero existen relaciones entre ambos Modelos a nivel de prácticas concretas, aunque pueden generalizarse algunos procesos de soporte y gestión de proyectos.

El Nivel al cual pertenece el Proceso de CMMI en el Modelo está descrito en la tabla como “N+el número del nivel”. La Figura 4.01 muestra los procesos de CMMI pareados a los Procesos/Funciones de ITIL.

Procesos/Funciones ITIL	Áreas de Procesos CMMI
Configuration Management	Configuration Management (N2)
Change Management	Configuration Management (N2)
Release Management	Configuration Management (N2)
Incident Management	
Problem Management	Verification (N3), Causal Analysis & Resolution (N5)
Capacity Management, Availability Management, Continuity Management	Risk Management (N3)
Application Management	Requirements Management (N2), Requirements Development (N3), Technical Solution (N3), Product Integration (N3), Verification (N3), Validation (N3), Integrated Project Management (N3)
Project Management	Project Planning (N2), Project Monitoring and Control (N2), Supplier Agreement Management (N2)

Figura 4.01: Tabla de mapeo CMMI e ITIL

Como se muestra en la Figura 4.01, la base del análisis es ITIL, ya que el enfoque a servicios debe ser la base de la Metodología que se desea generar. Sin embargo como el enfoque a nivel de procesos apunta a Verificación y Validación (Testing), se considerarán sólo el detalle de los procesos que tienen paridad en la tabla anterior; es decir, Problem Management (Administración de Problemas) y Application Management (Administración de Aplicaciones).

No se debe perder de vista que los procesos de Verificación y Validación están soportados en el nivel 3 de CMMI, por lo cual es necesario contar con el apoyo de algunos procesos del nivel anterior (nivel 2) para poder satisfacer en forma correcta las necesidades de la implantación de los procesos de Verificación y Validación del nivel 3.

Los procesos de apoyo para Servicios (ITIL) serán los siguientes:

- Configuration Management (Administración de la Configuración)
- Change Management (Administración del Cambio)
- Release Management (Administración de Release)
- Incident Management (Administración de Incidentes)
- Project Management (Administración de Proyectos)

Los procesos de apoyo para Desarrollo de software (CMMI) serán los siguientes:

- Configuration Management (Administración de la Configuración) → N2
- Project Planning (Planificación de Proyecto) → N2
- Project Monitoring and Control (Monitoreo y Control de Proyecto) → N2
- Supplier Agreement Management (Administración de Proveedores) → N2
- Requirements Management (Administración de Requerimientos) → N2
- Requirement Development (Requerimientos de Desarrollo) → N3
- Technical Solution (Solución Técnica) → N3
- Product Integration (Integración de Producto) → N3
- Integrated Project Management (Administración de Integración de Proyecto) → N3
- Risk Management (Administración de Riesgos) → N3

Es importante mencionar que cuando se habla de procesos de apoyo se entiende que al momento de detallar la Metodología a generar, se nombraran las mejores prácticas y los procesos listados, con el fin de que se tenga claridad de cómo ellos interactúan en el proceso de implantación; sin embargo no se entregará detalle de cómo se deben implantar, a diferencia de los procesos específicos de Verificación y Validación.

La Metodología obtenida es presentada en la Implementación de la misma.

La Figura 4.02 presenta la estructura definida para el Modelo Propuesto y se incluyen los proceso tanto del cruce como los de apoyo de CMMI e ITIL. Cabe destacar que el Modelo Propuesto fue generado en Enterprise Architect - UML CASE Tool - Desktop, Professional and Corporate editions, Version 7.0.

La simbología de los colores utilizados en el Modelo se encuentra en la Figura 4.02 en la parte superior derecha.

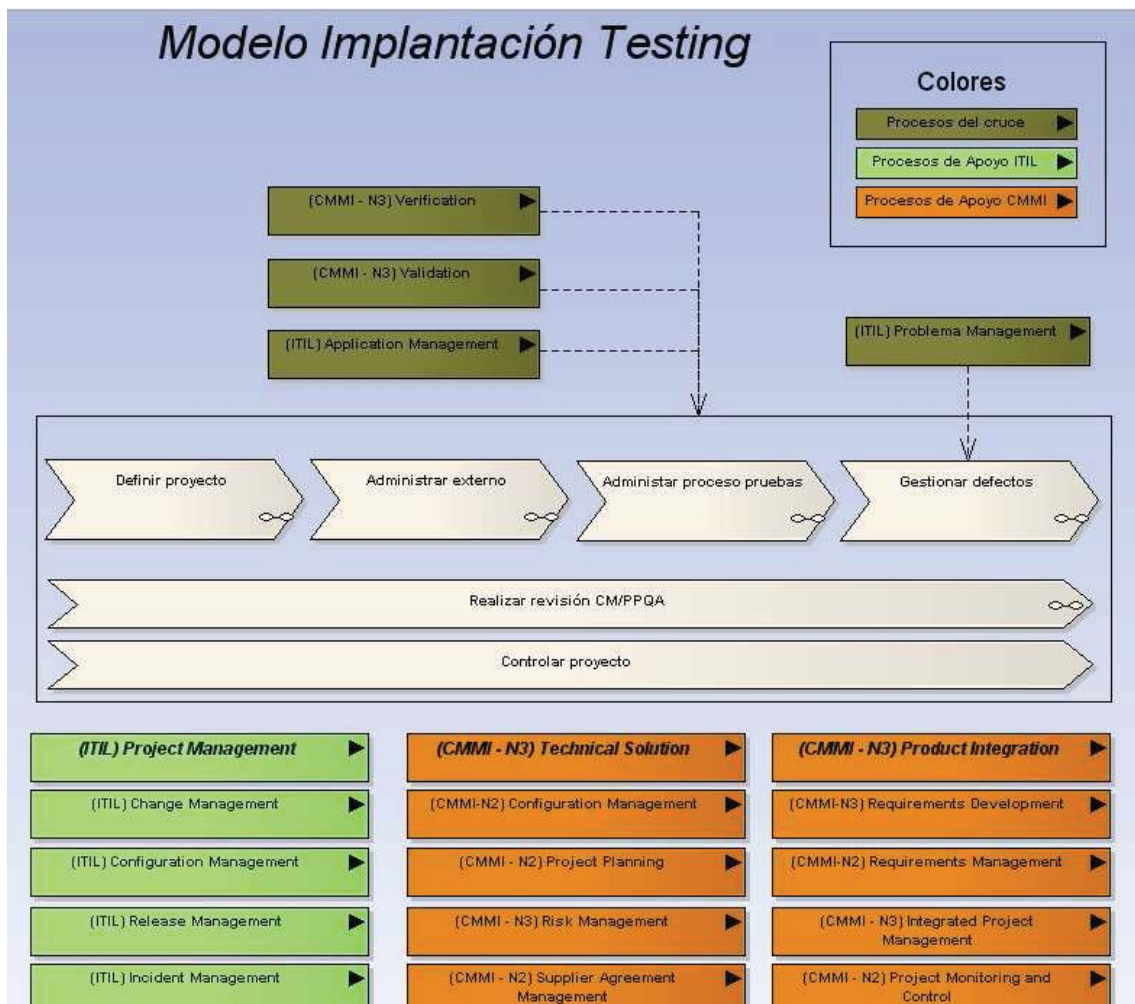


Figura 4.02: Metodología propuesta

En la Metodología se exponen las prácticas específicas y se detalla el proceso de cómo implantar la Metodología. El texto de la Metodología fue apoyado con un Diagrama de Flujo, el cual indica paso a paso como implantar el proceso de Testing en una empresa cualquiera.

A partir de la Metodología Propuesta se presenta la evaluación de su aplicación a un caso real, definiendo como punto de partida las métricas que se considerarán en dicha evaluación.

Al aplicar la Metodología al caso real, se refinaron algunas de las plantillas inicialmente generadas como apoyo.

El proceso de implementación consiguió como objetivo refinar y generalizar aún más los documentos de apoyo, ya que se amplió el espectro inicialmente pensado para tales documentos.

4.3 Definición de métricas para la evaluación final

A continuación se presentan los conceptos utilizados para la definición de métricas:

- Existen cuatro medidas para cuantificar: Caracterizar, Evaluar, Predecir y Mejorar.
 - 1) Medida: Valor asignado a un atributo de una entidad mediante una medición. Por ejemplo: 35.000 líneas de código.
 - 2) Medición: Es el acto de determinar una medida. Por ejemplo: Silvana será la encargada de medir las LDC de cada módulo del sistema.
 - 3) Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Incluye el método de medición. Por ejemplo: La productividad de este proyecto fue de 500 líneas/personas-mes (LDC/pm).
 - 4) Indicador: Es una métrica o combinación de métricas que proporcionan una visión profunda del proceso de software. Por ejemplo: la productividad media de nuestra empresa es de 500 (LDC/pm).

Las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo y el producto para intentar aumentar su calidad (LLeón-GPuglla, 2008).

- Por otro lado es importante saber que muchas veces el gerente de proyecto, se encuentra en la disyuntiva de dar o no el peso correcto a cada una de las variables que afectan el desarrollo de un proyecto en particular.

Existen restricciones tradicionales, como en cualquier empresa, los proyectos necesitan ser ejecutados y entregados bajo ciertas restricciones. Tradicionalmente, estas restricciones han sido alcance, tiempo y costo. Esto también se conoce como el Triángulo de la Gestión de Proyectos, donde cada lado representa una restricción. Un lado del triángulo no puede ser modificado sin impactar a los otros. Un refinamiento posterior de las restricciones separa la calidad del producto del alcance, y hace de la **calidad** una cuarta restricción.

- La restricción de tiempo se refiere a la cantidad de tiempo disponible para completar un proyecto.
- La restricción de costo se refiere a la cantidad presupuestada para el proyecto.
- La restricción de alcance se refiere a lo que se debe hacer para producir el resultado final del proyecto.

Estas tres restricciones son frecuentemente competidoras entre ellas: incrementar el alcance típicamente aumenta el tiempo y el costo, una restricción fuerte de tiempo puede significar un incremento en costos y una reducción en los alcances, así como también un presupuesto limitado puede traducirse en un incremento en tiempo y una reducción de los alcances.

La disciplina de la gestión de proyectos consiste en proporcionar las herramientas y técnicas que permiten al equipo de proyecto (no solamente al gerente del proyecto) organizar su trabajo para cumplir con todas esas restricciones.

4.3.1 Tiempo

El tiempo se descompone para propósitos analíticos en el tiempo requerido para completar los componentes del proyecto que es, a su vez, descompuesto en el tiempo requerido para completar cada tarea que contribuye a la finalización de cada componente. Cuando se realizan tareas utilizando gestión de proyectos, es importante partir el trabajo en pedazos menores para que sean fáciles de seguir.

4.3.2 Costo

El costo de desarrollar un proyecto depende de múltiples variables incluyendo costos de mano de obra, costos de materiales, administración de riesgo, infraestructura (edificios, máquinas, etc.), equipo y utilidades. Cuando se contrata a un consultor independiente para un proyecto, el costo típicamente será determinado por la tarifa de la empresa consultora multiplicada por un estimado del avance del proyecto.

4.3.3 Alcance

Requerimientos especificados para el resultado final. La definición global de lo que se supone que el proyecto debe alcanzar y una descripción específica de lo que el resultado final debe ser o debe realizar. Un componente principal del alcance es la *calidad* del producto final. La cantidad de tiempo dedicado a las tareas individuales determina la *calidad* global del proyecto. Algunas tareas pueden requerir una cantidad dada de tiempo para ser completadas adecuadamente, pero con más tiempo podrían ser completadas excepcionalmente. A lo largo de un proyecto grande, la *calidad* puede tener un impacto muy significativo en el tiempo y en el costo (o viceversa) (PMBOK, 2004).

La Figura 4.03 muestra las restricciones anteriormente descritas. Éstas se pueden representar en un triángulo, el cual aumenta o disminuye sus ángulos dependiendo de la importancia que se le aplique a cada uno de sus vértices (restricciones). Lo anterior refleja que en el caso que se le asigne mayor o menor importancia a uno de sus ángulos, éstos impactarán en la importancia de los demás vértices.

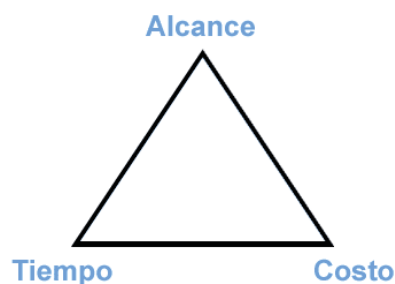


Figura 4.03: Triángulo de restricciones

A partir de los puntos antes descritos; Métricas y Triángulo de restricciones, se definen las siguientes métricas para la medición del éxito en la aplicación de la Metodología Propuesta a nivel de proyecto de Testing:

- El tiempo siempre será la variable menos movable con respecto a las restricciones, ya que todo proyecto tiene como límite una fecha, ya sea por necesidades del cliente o por imposición política (frecuente en empresas públicas).
- Una vez que el tiempo está establecido es importante saber el presupuesto con el que el cliente cuenta, ya que muchas veces es la segunda variable impuesta a nivel de requerimientos. El presupuesto puede estar dado mensual, semanal o incluso diario, dependiendo del tipo de proyecto.
- Finalmente queda el alcance, en el cual el cliente indica su óptimo con respecto al requerimiento en general. A parte de los costos que involucra el desarrollo del proyecto (específicamente lo que se evaluará en la Metodología), se debe considerar la calidad asociada al proceso de desarrollo, el cual también debe ser considerado en esta restricción.

Para el Testing de un aplicativo, teóricamente es prudente considerar a lo menos el 30% del Universo de pruebas posibles; es decir, si se estiman 100 pruebas de negocio para el aplicativo, lo prudente es testear al menos 30 de dichas combinatorias de pruebas de negocio. Más adelante se explicará la forma de estimar el Universo para luego indicar el cálculo del 30% en base a pruebas de negocio base y específicas.

Del 30% que será testeado, se procederá a establecer en el Plan de Pruebas, las políticas para la aceptación del aplicativo; es decir, se indicarán los porcentajes de aceptación para considerar efectivo el esfuerzo aplicado en el Testing.

Por experiencias en proyectos de Testing, lo mínimo recomendable como política de aceptación del 30% testeado será un 80% de aceptación SIN incidentes invalidantes. Muchas veces este porcentaje no es alcanzado, por malas prácticas en el desarrollo del software, así como también por la inexperiencia en el proceso de testeo.

Lo anterior indica que en el ejemplo de las 100 pruebas como Universo estimado, 24 pruebas (de las 30 seleccionadas) deben ser exitosas y no deben existir pruebas con invalidantes al momento de la ejecución. Por lo anteriormente expuesto, entonces se declaran 3 métricas para la evaluación de la implantación de la métrica, las cuales se listan en la Figura 4.04.

N°	Medición	Medida	Métrica
1	Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto.	HH	Esfuerzo Estimado v/s Esfuerzo Real
2	Se medirá el cumplimiento del costo establecido para el Testing del proyecto.	UF	Costo Estimado v/s Costo Real
3	Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.	%	% Establecido v/s % Obtenido

Figura 4.04: Métricas definidas para validación de la implantación de metodología propuesta

Las métricas obtenidas deberán tender en su mayoría a cero una vez calculado su diferencial. Luego de calculado su diferencial, éstos deberán ser analizados por el administrador del proyecto, quien tendrá la noción del proyecto general con respecto a los diferenciales obtenidos. Lo anterior, siempre con la mirada de "un traje de novia con excelencia en calidad al costo estimado, no sirve de nada si es entregado después de la fecha establecida".

Para medir la aplicación de la Metodología Propuesta se sugiere la siguiente métrica:

- 1) Porcentaje de Apego a la Metodología: Esta métrica tiene relación con identificar en que porcentaje se ha aplicado la Metodología Propuesta al concluir el Proyecto de Testing.

Las medidas utilizadas en el cálculo de la métrica se desprenden de la expertise obtenida en el desarrollo de la Propuesta Metodológica, así como también en la aplicación consecutiva de éstas en proyectos diversos para mejorarla.

La Figura 4.05 se refiere a la importancia de la plantilla diseñada para el apoyo al proceso de implantación. Esta importancia está dada por la experiencia en el proceso, pensando siempre en el cumplimiento del objetivo propuesto: apego a la Metodología.

Importancia Metodológica	
Muy Importante	4
Importante	3
Media	2
Baja	1

Figura 4.05: Métrica "Importancia Metodológica"

La Figura 4.06 entrega la relación de la planilla al proceso de implantación bajo la visión de Modelo diseñado del proceso de Testing; es decir, si la planilla apoya al proceso de CMMI, ITIL o es parte del apoyo a un proceso del cruce realizado entre ambos Modelos.

Propuesta Metodológica	
Cruce	5
CMMI	3
ITIL	2

Figura 4.06: Métrica “Propuesta Metodológica”

Una vez definidas las medidas se define la forma en la cual se calculará el apego a la Metodología. A continuación (Figura 4.07) se presenta la tabla con los campos ya asignados para el cálculo de la métrica de apego.

Columna A	Columna B	Columna C		Columna D	Columna E
Estructura	Plantilla	Propuesta Metodológica		Importancia en Metodología	Apego
0. Entradas	Informe de Diseño proveedor(Extracción).doc	CMMI	3	4	12
	PlanificaciónSeguimientoSemanal.mpp	CMMI	3	4	12
	Visión y Alcance.doc	CMMI	3	2	6
1. Análisis	Matriz_de_Evaluacion_de_Propuestas.xls	CMMI	3	2	6
	Estimación Modelo de Servicios.xls	Cruce	5	4	20
	LB_Requerimientos_Calidad.xls	ITIL	2	3	6
2. Diseño	Definición CPL.xls	ITIL	2	4	8
	TF Diseño.xls	ITIL	2	3	6
	Plan_de_CM.doc	CMMI	3	1	3
	Plan_de_Pruebas.doc	CMMI	3	3	9
3. Ejecución	TF Ejecución.xls	Cruce	5	3	15
4. Gestionar Defectos	TF Entidad-Defecto.xls	Cruce	5	3	15
5. Informes	Plan de Negocio.xls	Cruce	5	4	20
	Cierre Proyecto.xlsx	Cruce	5	2	10
	RESULTADO TESTING Ciclo.doc	Cruce	5	2	10

Figura 4.07: Tabla “Cálculo de métricas”

- 1) La Columna A se refiere a la estructura dada para organizar las planilla diseñadas. Concuerdan con el ciclo del proyecto que se testea.
- 2) La Columna B identifica cada una de las plantillas diseñadas para el proceso de implantación del Testing.
- 3) La Columna C clasifica cada planilla diseñada según la medida explicada anteriormente (Propuesta Metodológica).
- 4) La Columna D clasifica cada planilla diseñada según la medida explicada anteriormente (Importancia en Metodología).
- 5) La Columna E es la multiplicación de la Propuesta Metodológica (Columna C) por la Importancia Metodológica (Columna D).
- 6) En la Figura 4.08 se explica la segunda parte de la Métrica en base al apego de la Metodología Propuesta, la cual considera principalmente la aplicación de las métricas propuestas (Figura 4.05 y 4.06).

Columna B	Columna F	Columna G	Columna H	Columna I	Columna J	Columna K	Columna L
Plantilla	Aplicada	Apego por Aplicada	Distribución Importancia	Refinada	Multiplica (Distribución * Refinado)	Apego por Fase	Porcentaje por Estructura
Informe de Diseño proveedor(Extracción).doc							0. Entradas
PlanificacionSeguimientoSemanal.mpp							
Visión y Alcance.doc							
Matriz_de_Evaluacion_de_Propuestas.xls							1. Análisis
Estimación Modelo de Servicios.xls							
LB_Requerimientos_Calidad.xls							
Definición CPI.xls							2. Diseño
TF Diseño.xls							
Plan_de_CM.doc							
Plan_de_Pruebas.doc							3. Ejecución
TF Ejecución.xls							
TF Entidad-Defecto.xls							4. Gestionar Defectos
Plan de Negocio.xls							5. Informes
Cierre Proyecto.xlsx							
RESULTADO TESTING Ciclo.doc							

Figura 4.08: Tabla “Apego de la metodología propuesta”

- 1) La Columna F debe ser llenada con 1 si la planilla fue aplicada en el proceso de Testing o con uno 0 en el caso de que no se haya aplicado.
- 2) La Columna G multiplica la Columna E (Apego) por la Columna F (Aplicada) con el fin de obtener el apego real en la Metodología aplicada por cada una de las plantillas.
- 3) La Columna H es la distribución de la importancia que se calcula considerando Columna G (Apego por Aplicada) dividido por la sumatoria de cada plantilla que compone una estructura (Columna A).
- 4) La Columna I se debe llenar con un 1 para representar que SI se ajustará la planilla propuesta, al momento de su aplicación en lo que respecta a mejoras para su posterior utilización, o bien con un 0,9 en el caso que NO haya sufrido modificaciones la planilla propuesta al momento de su aplicación en lo que respecta a mejoras para su posterior utilización.
- 5) La Columna J es la multiplicación de Columna H (Distribución Importancia) por Columna I (Refinada) para así obtener la medida final en la siguiente columna.
- 6) La Columna K es la sumatoria de toda la columna J de cada plantilla existente por Estructura (Columna A). Esta medición entrega el apego a la Metodología por Estructura definida.
- 7) La Columna L es el recordatorio de la composición por Estructura.

El porcentaje de Apego a la Metodología se calcula sumando todas las medidas obtenidas por estructura (Columna K) dividida por 6 (cantidad de estructuras: Entradas, Análisis, Diseño, Ejecución, Gestionar Defectos e Informes).

Es importante mencionar que se realizó el ejercicio considerando el mínimo recomendable en la utilización de las plantillas en la implementación de la Metodología, llegando a la conclusión de que al aplicar la métrica de Apego a la Metodología se puede observar que el escenario peor es a lo menos un 80% de apego a la Metodología como porcentaje general, ya que sin este porcentaje no se estaría consiguiendo el objetivo de aportar en el proceso de implementación del Testing.

5 Implantación de la metodología propuesta y aplicación de plantillas de apoyo

A partir de la Metodología Propuesta se definen las plantillas de apoyo por proceso en el siguiente orden, considerando siempre el apoyo de los procesos del cruce, procesos de apoyo ITIL y procesos de apoyo CMMI. Es importante mencionar que se creó un nuevo proceso de Control de Proyecto que permitirá ir analizando las métricas propuestas anteriormente.

Estas plantillas apoyan el proceso de Testing (Verificación y Validación).

1. Definir Proyecto
2. Administrar Externo
3. Administrar Proceso Pruebas
4. Gestionar Defectos
5. Realizar Revisión CM/PPQA
6. Controlar Proyecto

Las planillas de apoyo se verán reflejadas como documentos asociados al proceso específico según se muestra en la Figura 5.01.

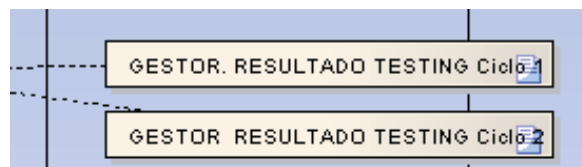


Figura 5.01: Identificación planillas de apoyo

Es importante indicar que el documento contiene el nombre de la planilla de apoyo que se encuentra en los anexos. Además la línea punteada relaciona el documento al proceso que lo utiliza.

5.1 Proceso para aplicar la metodología

La generación de esta Metodología nace de la necesidad y experiencia de establecer un método en el trabajo de la estimación y aplicación del Testing Funcional y No Funcional en un Área de Calidad de una EMPRESA.

Históricamente se sabe que un proyecto nunca es igual a otro, siempre existen diferencias significativas en el desarrollo de éstos, por lo que para la aplicación de la Metodología inicialmente se consideró incorporar como prerrequisitos mínimos los entregables que aplican como entrega por parte del equipo de proyecto. Por lo

anteriormente indicado se hacía cada vez más necesario establecer patrones de semejanza en la aplicación de un método para levantar un “proyecto” de Testing sobre un proyecto de desarrollo de software.

A medida que se analiza la generación de la Metodología se comienzan a exigir como Área, entregables mínimos a considerar en un proyecto de desarrollo; como por ejemplo, documentos de especificación de requerimientos, documentos de diseño, y documentos formales de aceptación de los documentos antes indicados, con el fin de establecer líneas base claras a la hora de estimar y planificar el proyecto de Testing.

Una vez generada la Metodología, se escogió un proyecto real, de una de las áreas con las cuales se llevaban muchos trabajos históricos, lo que ayudó, en gran medida, a contar con los documentos mínimos necesarios.

Al aplicar esta Metodología, se podrá apreciar que muchas veces los documentos mínimos no son considerados formalmente en un proyecto de desarrollo. Esto es una muy mala práctica, ya que no se puede establecer un Método sobre un proyecto que viene mal “engendrado”. A pesar de lo anteriormente expuesto, las plantillas generadas servirán de apoyo para las etapas de Requisitos, Análisis y Diseño de un proyecto de desarrollo de software.

El proceso de aplicación comienza con la exigencia de los documentos mínimos y la definición (estimación y planificación) del proyecto de Testing que se apegará a las condiciones del proyecto de desarrollo.

Se establecieron Métricas que se consideran evaluar al término de la aplicación de la Metodología, con el fin de mostrar el real aporte de su aplicación.

Las Plantillas aplicadas, son presentadas en el ANEXO 1. A medida que se utilizan en la Metodología serán referenciadas. Algunas de ellas se detallarán, con el fin de describir las transformaciones que han sufrido en el proceso de aplicación al proyecto real.

Es importante mencionar que además se incorporaron algunos documentos más que no estaban considerados inicialmente como plantillas, pero que en el proceso se pudieron percibir como gran apoyo.

Se recuerda que los procesos en azul tienen directa relación con la posible utilización de la Metodología para procesos apoyados con herramientas existentes en el mercado; en este caso, específicamente con Mercury de HP.

Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.1 Proceso “Definir proyecto”

En la Figura 5.02 se puede ver que no existen plantillas de apoyo al proceso. En su completitud este proceso apoya en la generación de un nuevo proyecto en el caso de que se utilice una herramienta de Prueba.

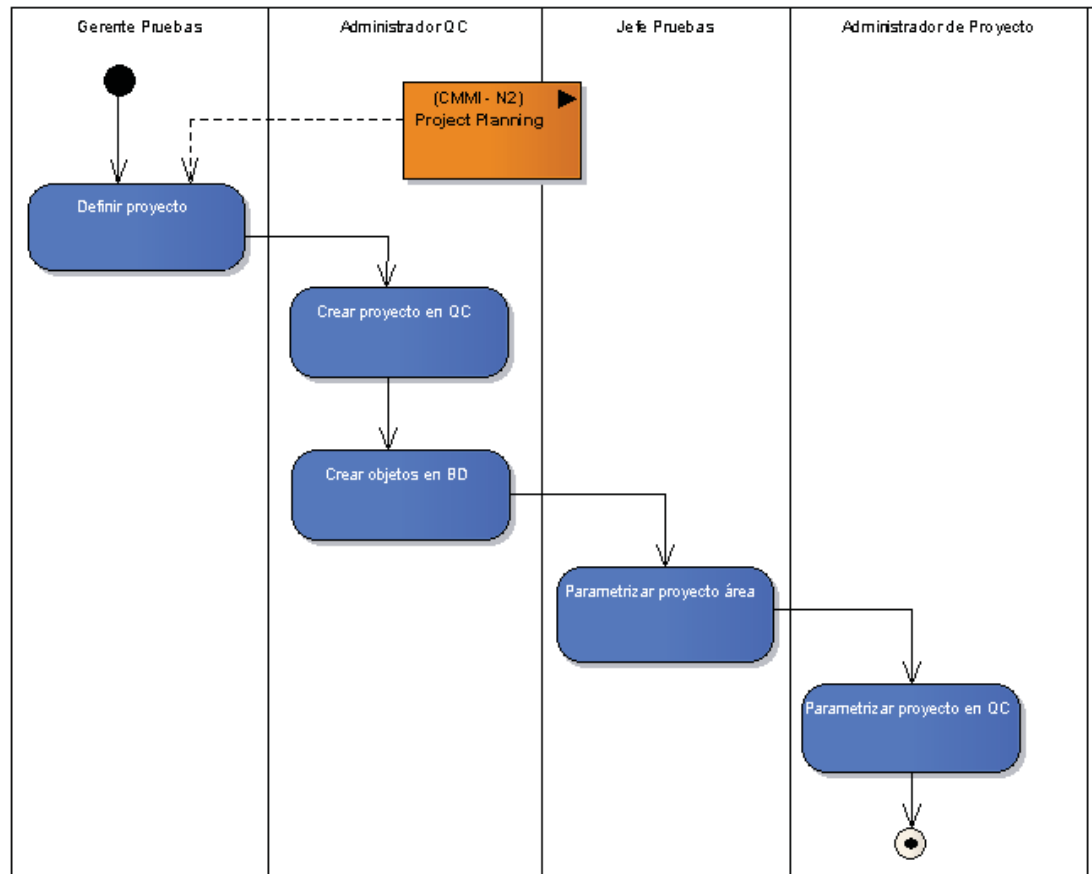


Figura 5.02: Proceso “Definir proyecto”

Proceso	Definir proyecto
Descripción	<p>Una vez que ha sido aprobado por el cliente la propuesta de servicio:</p> <ul style="list-style-type: none"> - Se define el nombre del proyecto y a qué línea de negocios pertenece. - Definición de las áreas de calidad involucradas y los Jefe de Área/Prueba involucrados. - Se debe definir las contrapartes válidas (Jefes de Proyectos) a las cuales se les derivarán los defectos y si accederán QC. - Se define la política de respaldo y Planes de Pruebas - Checklists serán versionados en Subversion (herramienta de versionamiento) indicando el repositorio. - Se definen los aplicativos y las versiones para cada uno. - Se entrega la información anterior al "Administrador QC" para crear el proyecto en QC.

Proceso	Crear proyecto en QC
Descripción	<p>Se realizan las tareas asociadas a la creación del proyecto en QC, se ajusta una copia de la plantilla Domain: Base y Project: Proyecto basándose en la especificación entregada por el Jefe de Pruebas.</p> <ul style="list-style-type: none"> - Conectarse como Administrador de QC a QC Project Administration - Crear un proyecto dentro del dominio correspondiente. (QC Project Administration- Site Projects). Elegir la opción 'Create a project by copying data from an existing project' - Seleccionar el proyecto plantilla mencionado Domain: Base y Project: Proyecto - Chequear Customization, Requirements, Tests, Test Sets, Public Favorite Views. - Especificar el servidor de BD donde residirá el proyecto. - Asignar el Administrador del Proyecto (Se asume un mismo Administrador para todos los proyectos) - Activar el proyecto (Chequear Activate project) - Dado que los proyectos se guardarán distribuidamente en la BD y en directorios del S.O. no chequear 'Store project's repository in the database' - Dado que no se usará versionamiento, debido a que éste sólo cubre las pruebas y no el resto de las entidades, no chequear 'Create a Version Control database' - Chequear por conectividad a la BD usando el botón Ping. - Crear los usuarios que aún no existen y hayan sido especificados en la Definición del Proyecto .(QC Project Administration - Site Users) - Conectarse a Quality Center como el Administrador de Proyecto al nuevo proyecto. - Crear los aplicativos y releases. Renombrar la carpeta raíz Releases con el nombre del proyecto. - Informar al administrador de la BD, que se generó una nueva BD, con el fin de actualizar el job de respaldo.

Proceso	Crear objetos en BD
Descripción	<p>Crear los objetos cuyos scripts están en el tab Files usando el usuario td, previa especificación de la BD en ellos.</p> <p>Crear sólo las funciones get_domainname y get_projectname pues éstas ya fueron creadas en la BD de Administración de QC.</p>

Proceso	Parametrizar proyecto área
Descripción	<p>En esta tarea se especifica las parametrizaciones propias de cada Área de Calidad:</p> <ul style="list-style-type: none"> - Definir usuarios del área que participarán en el proyecto. - Se identifica si el proyecto necesita customizaciones especiales. Estas pueden implicar: <ol style="list-style-type: none"> (1) Creación de grupos usuarios especiales (2) Creación de nuevos campos para las entidades (3) Creación o mantención de Listas de Lookup.

Proceso	Parametrizar proyecto en QC
Descripción	<p>Una vez que el proyecto ha sido creado se realiza su parametrización:</p> <ul style="list-style-type: none"> - Crear los grupos de usuarios necesarios (QC Project Customization - Groups). - Asignar los usuarios al proyecto (QC Site Administration - Site Users - User Projects) - Si QC se versiona en Subversion y no se está usando sólo la cuenta de Subversion Default, agregar los mappings entre usuarios QC y Subversion. Nótese que si el versionamiento ya estaba activo, debe ser desacoplado, cambiado y vuelto a acoplar. - Realizar las customizaciones (QC Project Customization)

5.1.2 Proceso “Administrar externo”

El proceso “Administrar Externo” se basa fundamentalmente en la búsqueda de externos para el apoyo del Testing. En el caso que sea necesaria la utilización de externos es importante tener realizada la estimación del proyecto es lo que respecta al proceso de Testing.

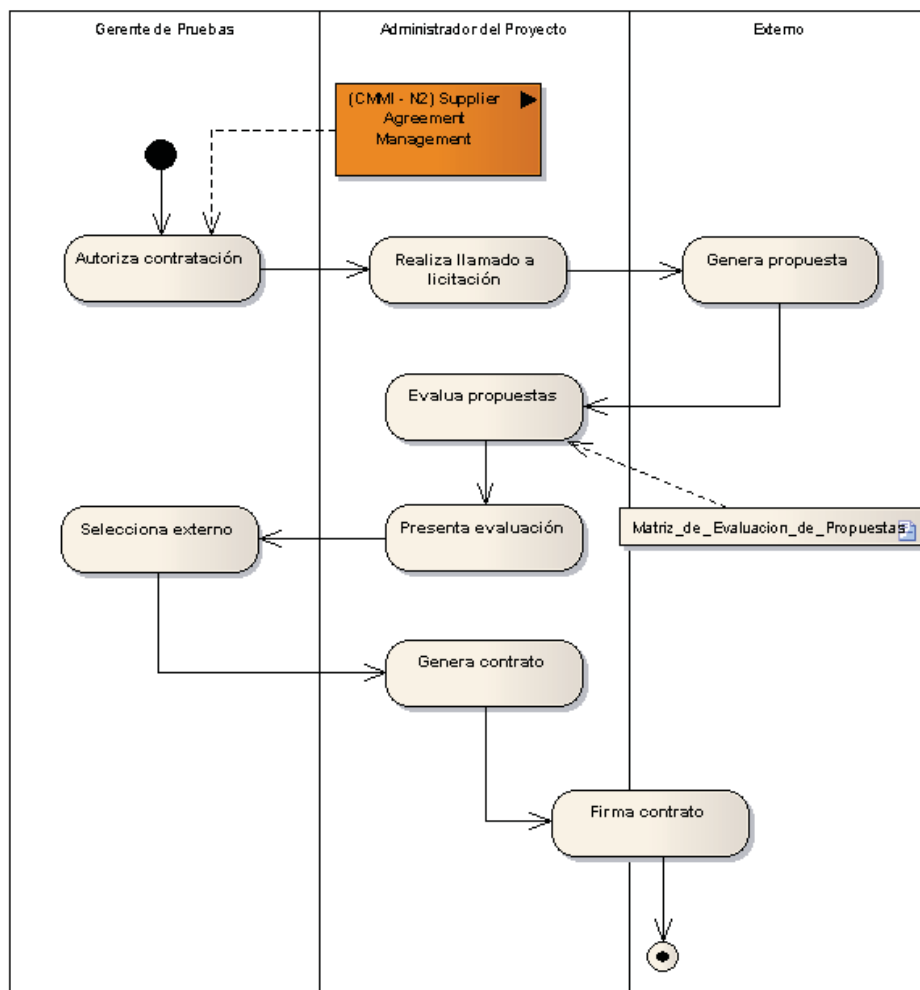


Figura 5.03: Proceso “Administrar externo”

Proceso	Autoriza contratación
Descripción	Una vez que el equipo de Gestión de Proyecto autoriza la utilización de recursos externos. El equipo de proyecto debe tener presente que en estos casos el costo aumenta, ya que contratar una “buena” empresa de Testing involucra experiencia, eficacia y eficiencia en el proceso.

Proceso	Realiza llamado a licitación
Descripción	El equipo de Proyecto entrega la potestad al equipo de Testing en el llamado a licitación, pudiendo ésta ser de carácter público o reservado, dependiendo de la estrategia de empresa por la que se rija el equipo de Testing. Es importante mencionar que un buen llamado a licitación consiste en precisar el servicio que se solicita, tanto en tiempo, costo, calidad y por sobre todo en lo que respecta a los rendimientos tanto del diseño de los casos como en la ejecución de ellos. Otro punto importante a considerar es explicar en detalle la definición de los conceptos base para poder realizar el cálculo del Testing, que en este caso sería la definición de Caso de Prueba Integrado, Casos de Prueba Unitarios o Casos de Negocio. Las definiciones antes mencionadas están especificadas en el Glosario de este documento.

Proceso	Genera propuesta
Descripción	En base al llamado de licitación, el externo genera su propuesta de servicio, en la cual debe especificar claramente el cumplimiento de plazos, restricciones del servicio en caso de necesidades de horas extras y el costo asociado.

Proceso	Evalúa propuestas (*)
Descripción	Una vez que los proveedores de servicio entregan sus propuestas, el equipo de Testing debe comenzar con la evaluación. Es en este punto donde aparece nuestra primera Plantilla de apoyo (ver ANEXO 1 “Matriz_de_Evaluación_Propuestas”).

Proceso	Presentar evaluación
Descripción	Una vez aplicada la matriz de evaluación, es necesario presentar los resultados al equipo de Proyecto y realizar las recomendaciones respecto a dicha evaluación.

Proceso	Selecciona externo
Descripción	El equipo de Proyecto entonces está en condiciones de realizar la selección del externo para que participe en el proceso de Testing, pudiendo ser ésta el diseño y/o ejecución de las Pruebas.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

Proceso	Generar contrato
Descripción	<p>El equipo de Testing está en condiciones de generar el contrato de servicio con el externo.</p> <p>Es importante recomendar que dicho contrato esté orientado a entregables de trabajo relacionados con el proceso del Testing en una primera vista, pero que en segunda línea se especifique claramente la cantidad de HH relacionadas al servicio.</p> <p>Se debe tener especial cuidado en las excepciones que se acuerden y en el aviso y satisfacción del servicio en lo que respecta a horas de trabajo extraordinario.</p>
Proceso	Firma contrato
Descripción	La firma del contrato debe ser entre equipo de Testing y externo, ya que las exigencias con respecto a respuesta en la entrega de servicio son de responsabilidad del equipo de Testing.

5.1.3 Proceso “Administrar proceso de pruebas” PARTE 1

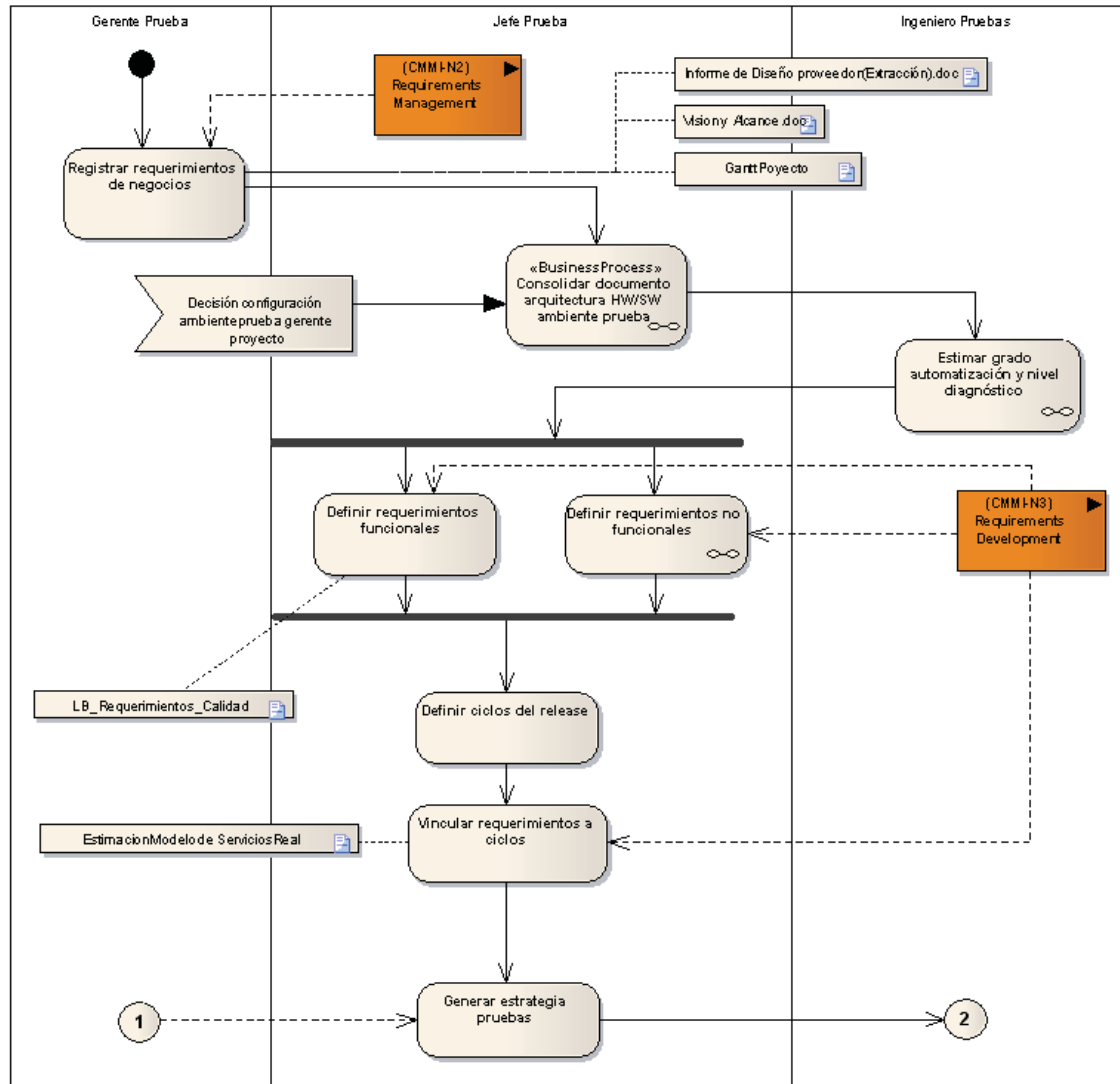


Figura 5.04: Proceso “Administrar proceso de pruebas” PARTE 1

Proceso	Registrar requerimientos de negocios (*)
Descripción	<p>El primer documento que debe aprobar el equipo de proyecto al desarrollador del aplicativo a testear es el de Visión y Alcance, el cual debe especificar claramente lo que se va a desarrollar. Siempre en términos generales en lo que respecta al aplicativo. En la plantilla de apoyo ANEXO 1 “Visión y Alcance” se explica el contenido esperado como mínimo en dicho documento.</p> <p>Luego que el documento de Visión y Alcance está acordado y aprobado por las partes es hora de que el desarrollador comience con el análisis para continuar con el diseño, otro de los input importantes a la hora de analizar qué testear.</p> <p>El proveedor de desarrollo debe entregar al equipo de proyecto, el documento de diseño del aplicativo que se va a testear. A partir de este documento el equipo de Testing debe ser capaz de entender y reflejar en la planilla de requerimientos, todos los requisitos que el aplicativo debe contener.</p> <p>El Informe de diseño debe contener todo lo indicado en la plantilla de apoyo ANEXO 1 “Informe Diseño del Proveedor”.</p> <p>Una vez aprobado el documento de Diseño, el Proveedor ya está en condiciones de declarar tiempos claros y precisos en una Carta Gantt, sobre la cual el equipo de proyecto y el equipo de Testing deben comenzar a analizar los instantes en los cuales el equipo de Testing podrá intervenir, así como también las posibles restricciones de tiempo y costo, con el fin de poder dilucidar cuál será el alcance de la claridad del Testing.</p> <p>La plantilla siguiente muestra una planificación tipo que puede ser utilizada como referencia para analizar las intervenciones del equipo de Testing. Ver ANEXO 1 “Planificación_Seguimiento_Semanal”.</p> <p>A nivel de equipo de Testing, a partir de los 3 documentos antes mencionados, debe ser capaz de definir los requerimientos del aplicativo, por lo anterior se deben realizar los siguientes pasos:</p> <p>Definir el requerimiento como carpeta, especificando:</p> <ol style="list-style-type: none"> 1. Nombre 2. Tipo de Requerimiento como de Negocios. 3. Descripción 4. Autor 5. Prioridad en la implementación:

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<p>1-Baja 2-Media 3-Alta 4-Muy Alta</p> <p>6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)</p> <p>Nota: Es posible agregar Attachments a los Requerimientos también.</p> <p>En el caso de Pruebas No Funcionales es recomendable que los requerimientos de negocios tengan la forma de SLAs estándares.</p> <p>NOTA: LOS REQUERIMIENTOS DE NEGOCIO SON DE ALTO NIVEL, Y SU ANALISIS PUEDE INVOLUCRAR SU DESCOMPOSICION EN REQUERIMIENTOS FUNCIONALES, DESEMPEÑO, SEGURIDAD, ETC.</p>
--	--

Proceso	Consolidar documento arquitectura HW/SW ambiente prueba
Descripción	<p>Si bien el área de calidad no es el encargado de generar este documento en forma íntegra si debiera consolidarlo para sí, previa ejecución de:</p> <ul style="list-style-type: none"> - Recolectar las especificaciones de Arquitectura que generen otras áreas de la Empresa, el Cliente o Terceros. - Generar la especificación del dispositivo GUI, su cantidad, ubicación física y conectividad. <p>El documento debe contener todas las componentes detalladas en el Diagrama "Ambiente_Prueba". Dado que los ambientes para Pruebas Funcionales y No Funcionales son potencialmente distintos se recomienda elaborar un documento para cada uno.</p>

Proceso	Estimar grado automatización y nivel diagnóstico
Descripción	<p>En esta tarea se estima que porcentajes de las Pruebas serán realizadas automática y manualmente. Dicha estimación se realiza basándose en:</p> <ul style="list-style-type: none"> - Diseños Preliminares del Sistema. Por ejemplo si el "Dispositivo GUI" es una PALM y no hay emulador confiable, las Pruebas funcionales deberán ser realizadas manualmente, pero puede que de todos modos sea factible realizar las Pruebas no funcionales. - Pruebas sobre el Aplicativo en el caso que esté disponible usando QTP y LR para las Pruebas funcionales y no funcionales respectivamente. <p>En el caso No Funcional si no es factible realizar las Pruebas automáticamente, probablemente la opción manual sea demasiado cara.</p>

Proceso	Definir requerimientos funcionales (*)
Descripción	<p>La planilla de apoyo "LB_Requerimientos_Calidad" en ANEXO 1 entrega el detalle del llenado para obtener el Universo estimado.</p> <p>Esta plantilla debe estar aprobada por el equipo de proyecto y debe sumar todos los casos de Prueba relacionados a los requerimientos con estado Declarado.</p> <p>Se definen los requerimientos funcionales detallando los atributos mencionados a continuación, si es necesario se crean carpetas que agrupen los requerimientos por temas.</p> <ol style="list-style-type: none"> 1. Nombre 2. Especificar Tipo de Requerimiento (Negocios, Funcional, No Funcional) 3. Descripción 4. Autor 5. Prioridad en la implementación: <ul style="list-style-type: none"> 1-Baja 2-Media 3-Alta 4-Muy Alta 6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)

Proceso	Definir requerimientos No Funcionales
Descripción	<p>Rendimiento, en los Requerimientos de Negocios debería existir la definición de SLA de tiempo de respuesta vinculado a un conjunto de transacciones, sin embargo se dan tres posibles situaciones:</p> <ul style="list-style-type: none"> - El conjunto de transacciones es adecuado para el tiempo asignado a Pruebas de Carga en el proyecto. Usualmente el tiempo alcanza para considerar entre 10 y 30 transacciones. - El conjunto de transacciones es muy amplio y no es factible realizar todas las mediciones en términos económicos. - El conjunto de transacciones ha sido enunciado en forma general y no se han identificado las transacciones.

Proceso	Definir ciclos del release
Descripción	Definir los "Ciclos del Release" indicando los "Conjunto de Pruebas" preliminares que los conforman, por ejemplo el primer "Ciclo de Pruebas" debiera incluir siempre un conjunto de "Pruebas Trazadoras" con el fin de establecer si el ambiente está operativo o no.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

Proceso	Vincular requerimientos a ciclos (*)
Descripción	<p>Esta es la instancia para utilizar la planilla de Apoyo “Estimación Modelo de Servicios” del ANEXO 1. Es importante mencionar que esta plantilla no se podrá utilizar si no está cerrada la planilla de apoyo de LB Requerimientos Calidad, ya que ésta tiene el input más importante para la estimación del servicio.</p> <p>En la plantilla de apoyo además se definen las fases y ciclos de Prueba así como también los recursos relacionados.</p> <p>Se deben vincular los requerimientos a al menos un "Ciclo de Prueba" con el fin de:</p> <ul style="list-style-type: none"> - Determinar en cualquier instante si los requerimientos están siendo cubiertos o no - Generar esbozo de Pruebas en función de los requerimientos asociados a un ciclo. <p>Un requerimiento puede ser vinculado a un Release, en este caso, todos los "Ciclos de Prueba" dentro del Release serán vinculados, y eso implica que debe haber un "Conjunto de Pruebas" al menos dentro de cada Ciclo que valida el requerimiento.</p> <p>La cobertura de Prueba en un ciclo queda definida por la liberación de funcionalidades nuevas y por los defectos ocurridos en el ciclo anterior (en el caso que exista), por lo tanto al principio del proyecto, sólo se vincularán los requerimientos asociados a funcionalidades nuevas al ciclo en que se prueban, y con posterioridad dependiendo de los defectos se vincularán requerimientos a ciclos que contengan Pruebas de Regresión.</p>

Proceso	Generar estrategia pruebas
Descripción	<p>Este documento debe ser un compendio del estado del proyecto hasta este momento, considerando:</p> <ul style="list-style-type: none"> - Roles y personas - Requerimientos de Pruebas - Ciclos de Pruebas - Protocolo de Suspensión de las Pruebas - Conclusiones de Estudio Factibilidad de Automatización. - Referencias a los documentos de Arquitectura de HW/SW. - Formato Plan de Pruebas o Operatoria al respecto - Formato Informe de Resultados de Pruebas. - Manejo de Defectos. - Severidad de Defectos. - Riesgos <p>En el caso de CM-PPQA se incluyen los checklists que se utilizarán en la Revisión. Para extraerlos de QC: Tools->; Excel Report Generator->; Checklist CM-PPQA->; Generate Excel Report->; (Ingresar como parámetro el nombre de la prueba que contiene el Checklist).</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.3.1 Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”

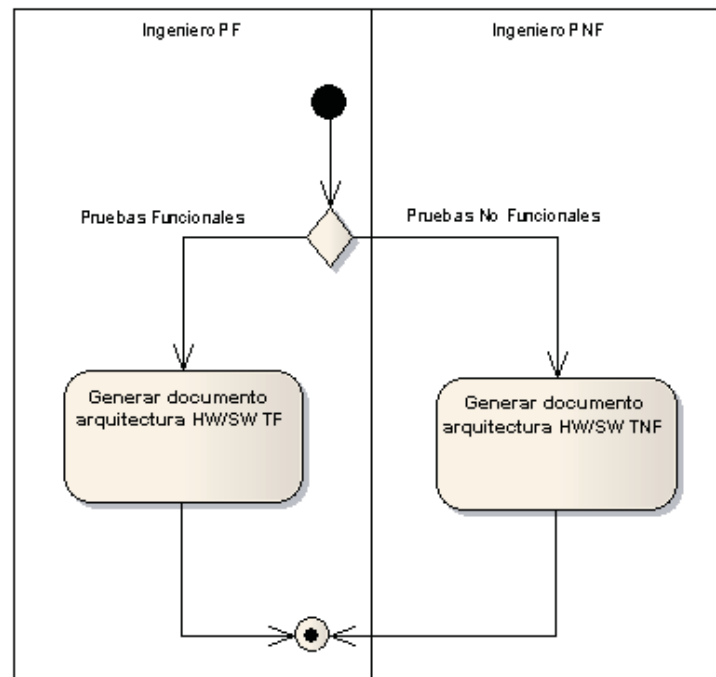


Figura 5.05: Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”

5.1.3.2 Proceso “Estimar grado automatización y nivel diagnóstico”

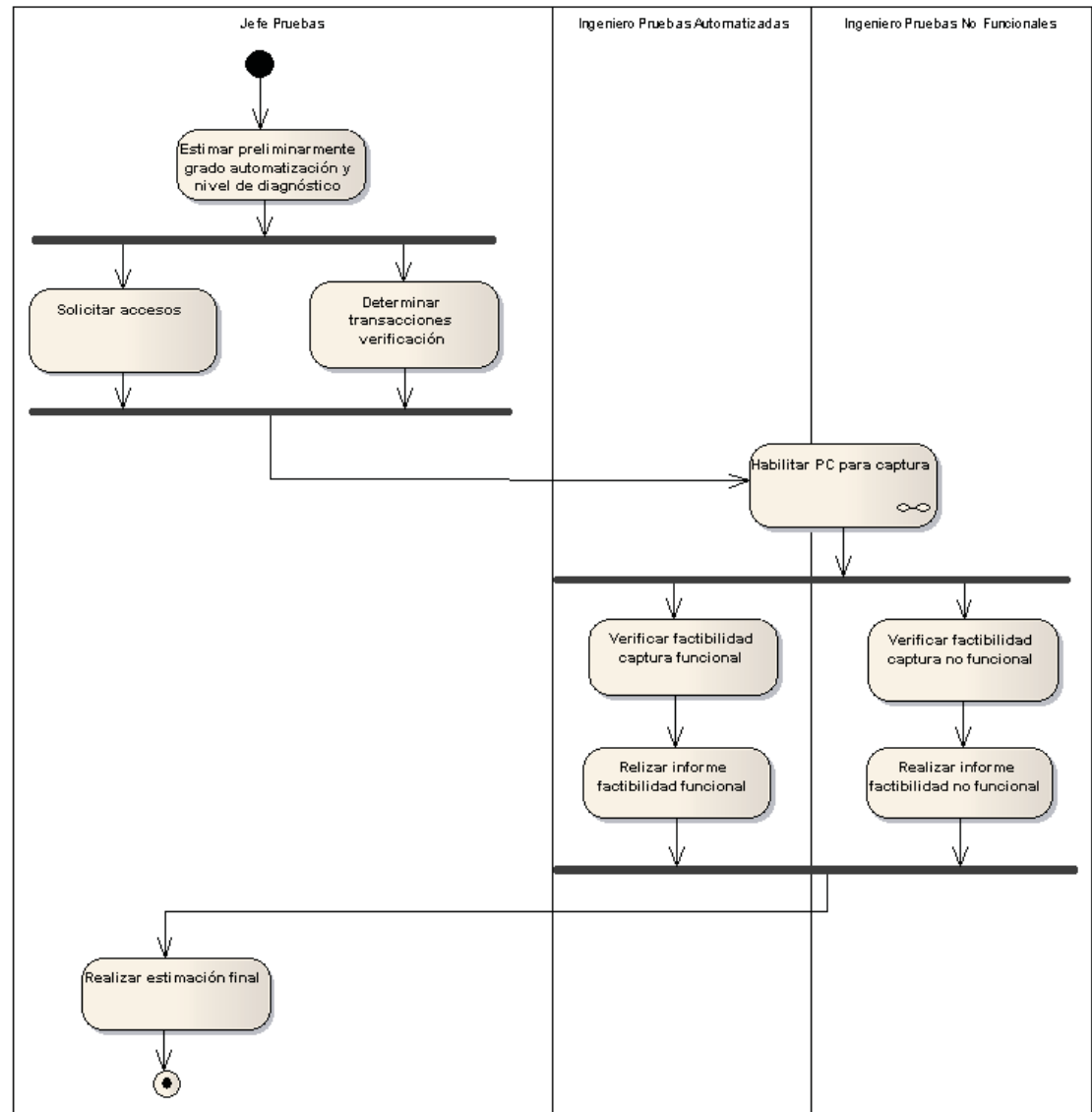


Figura 5.06: Proceso “Estimar grado automatización y nivel diagnóstico”

Proceso	Estimar preliminarmente grado automatización y nivel de diagnóstico
Descripción	<p>El análisis preliminar en esta tarea aborda el Grado de Automatización y el Nivel de Diagnóstico de la siguiente forma:</p> <p>Grado de Automatización</p> <p>En el caso que se disponga del aplicativo, se identifican todas aquellos módulos del aplicativo que no son automatizables con Mercury tanto desde el punto de vista funcional como no funcional, candidatos serían:</p> <ul style="list-style-type: none"> - General: (a) Uso de algoritmos de reconocimiento de imágenes, sonidos u otro distinto a eventos de teclado y mouse. (b) Dispositivos GUI no emulables en PC o notebooks. - Caso Funcional: (a) Reportes (b) Procesos Batch (c) Dispositivos GUI con mala emulación en PC. - Caso No Funcional: (a) Procesos Batch (b) Protocolos a nivel de GUI inexistentes en Mercury o no adquiridos. <p>Nivel de Diagnóstico</p> <p>Tanto en las Pruebas Funcionales como en las No Funcionales existe el mismo concepto, la diferencia es que en el último caso es siempre parte del servicio, en cambio en el caso funcional no.</p> <ul style="list-style-type: none"> - Caso Funcional: Si el servicio es Graybox, se deben verificar los resultados en la Base de Datos. Se debe evaluar si existen impedimentos para acceder la BD como Modelo no documentado. - Caso No Funcional: En este caso, si bien se espera un nivel Whitebox, no es siempre posible, debido a que pueda que no exista el SW para monitorear cada uno de los servidores aplicativos o en el caso que sea Mercury Diagnostics, no se esté dispuesto a hacer la inversión. <p>Si no se disponen del aplicativo se deberá hacer supuestos sobre las transacciones para las cuales no es tan claro si pueden ser automatizadas o no.</p>

Proceso	Solicitar accesos
Descripción	<p>Se debe solicitar:</p> <ul style="list-style-type: none"> - Acceso al aplicativo - Conectividad a Servidor QC en la Organización para QTP en el caso de pruebas funcionales.

Proceso	Determinar transacciones verificación
Descripción	<p>En base a los antecedentes que se disponga, se debe definir que transacciones serán las que se usarán para validar si es posible automatizar ya sea funcional o no funcionalmente. Se debe tener en cuenta al menos:</p> <p>Generales</p> <ul style="list-style-type: none"> - Debería haber al menos 1 transacción por arquitectura de SW distinta a nivel de "Dispositivo GUI" <p>Pruebas No Funcionales</p> <ul style="list-style-type: none"> - Informarse si algún SW encripta o comprime a nivel de "Dispositivo GUI" y si fuera así, solicitar los algoritmos al desarrollador (siempre y cuando sean indispensables para variabilizar los scripts VUGEN posteriormente). - Determinar si la aplicación es mono o multiusuario por dispositivo GUI.

Proceso	Habilitar PC para captura
Descripción	<p>Se contemplan 2 situaciones:</p> <ul style="list-style-type: none"> - Aplicativo se encuentra disponible a través de la red de la organización, en ese caso se utilizarán los computadores destinados para ese fin en dicha organización. - En el caso que no esté disponible en la red organizacional, se usará una máquina virtual que tenga QTP y LR listos para ser instalados.

Proceso	Verificar factibilidad captura funcional
Descripción	<p>La verificación consiste en que para cada transacción de verificación se ejecuta:</p> <ul style="list-style-type: none"> - Identificar los Add-ins que podrían necesitar en base a la documentación o referencias de la GUI. - Generar un script para cada una de las transacciones - Reejecutar el script. - Comprobar que se puedan incorporar puntos de verificación. - Comprobar que se puede variabilizar la entrada. <p>Una vez realizadas las tareas anteriores complete la planilla de Factibilidad QTP.</p>

Proceso	Verificar factibilidad captura No Funcional
Descripción	<p>La verificación consiste en que para cada transacción de verificación se ejecuta:</p> <ul style="list-style-type: none">- Generar un script que implique login, ejecutar transacción y logout.- Detectar correlación y resolverla en el caso que la hubiere.- Determinar si los mensajes son parametrizables.- Reejecutar el script.- Si es posible realizar una prueba con más de un usuario a la vez sobre el mismo "Dispositivo GUI". <p>Nota: Para resolución de problemas ver tab Files.</p>

Proceso	Realizar estimación final
Descripción	<p>Tanto para el caso funcional como no funcional, se consolida la estimación inicial con los resultados obtenidos en la verificación con el aplicativo.</p> <p>El resultado de esta estimación puede producir RFC de un impacto no despreciable en el presupuesto.</p>

5.1.3.2.1 Proceso “Habilitar PC para captura”

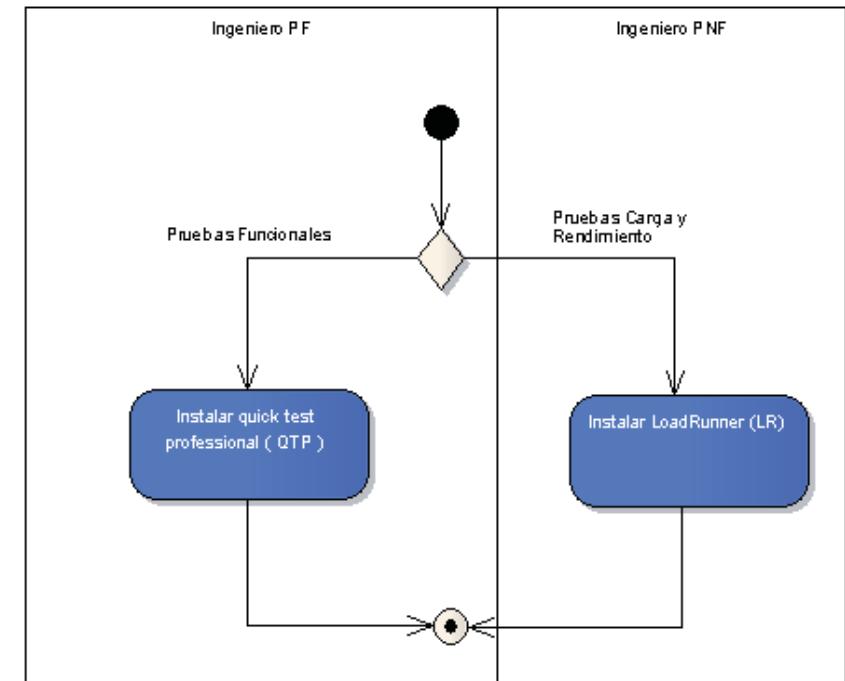


Figura 5.07: Proceso “Habilitar PC para captura”

5.1.3.3 Proceso “Definir requerimientos no funcionales”

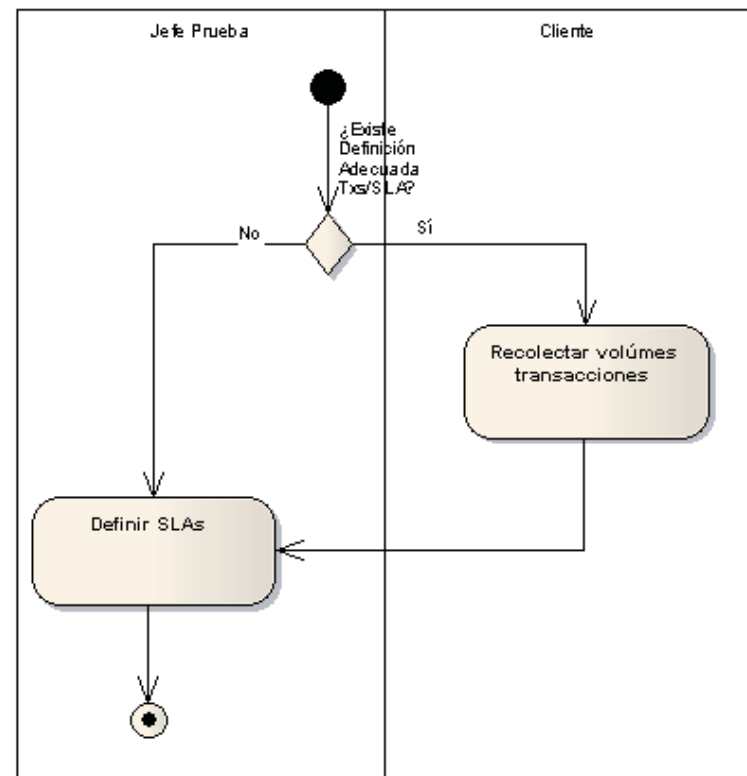


Figura 5.08: Proceso “Definir requerimientos no funcionales”

Proceso	Definir SLAs
Descripción	<ul style="list-style-type: none"> - Definir una carpeta con nombre "Pruebas de Carga y Rendimiento". - En el caso que exista información histórica o estimaciones, agregar attachment tipo URL apuntando a la ubicación de una planilla Excel "Volúmenes de Transacciones" en subversion. - Definir para cada transacción seleccionada un Requerimiento de Testing detallando su SLA, para el cual se debe especificar que acción del usuario gatilla la transacción que va a ser medida (por ejemplo click sobre el botón grabar).

Proceso	Recolectar volúmenes transacciones
Descripción	<p>Recolectar información histórica o estimar en un documento llamado "Volúmenes Transacciones", la siguiente información:</p> <ul style="list-style-type: none"> - Número de usuarios del sistema - % Crecimiento Anual Usuarios del Sistema - % Concurrencia de usuarios máxima - Una hoja de cálculo de transacciones cuyas columnas sean: Id Transacción, Nombre Transacción, Cantidad Transacciones Promedio x Tramo Horario i ($i=1, \dots, n$), % Crecimiento Anual. Ojala se pudieran descomponer la cantidad de transacciones por perfil - Una hoja de cálculo de transacciones cuyas columnas sean: Id Transacción, Nombre Transacción, Cantidad Transacciones Peak x Tramo Horario i ($i=1, \dots, n$), % Crecimiento Anual. Ojala se pudieran descomponer la cantidad de transacciones por perfil.

5.1.4 Proceso “Administrar proceso de pruebas” PARTE 2

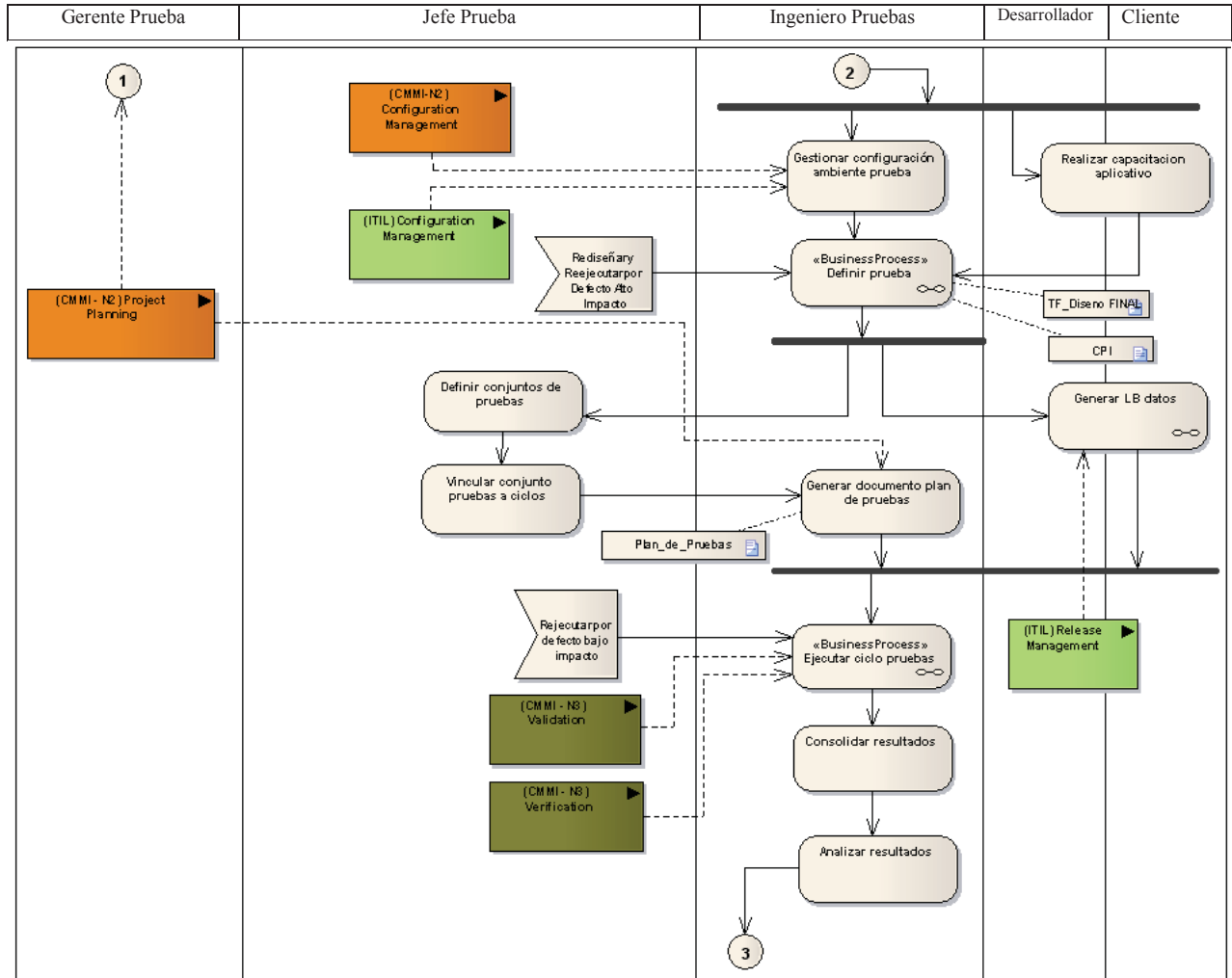


Figura 5.09: Proceso “Administrar proceso de pruebas” PARTE 2

Proceso	Gestionar configuración ambiente prueba
Descripción	<p>Con el objetivo de realizar las Pruebas, además de disponer del "Ambiente de Pruebas" (se debe definir de quien es la responsabilidad de entregar dicho ambiente de Pruebas), Calidad debe adaptar el ambiente para usar Mercury:</p> <ul style="list-style-type: none"> - Verificar que todas las componentes del Ambiente de pruebas que no son responsabilidad de Calidad, están disponibles. - Habilitar Laboratorio(s) Prueba(s): Disponer de "Dispositivos GUI", Instalar SW Calidad, Instalar SW Cliente. (Ver manual de instalación en Tab Files). - Generar los RFC para las conectividades de enlace con los "Servidores de Calidad" y "Dispositivo_GUI". - Generar los RFC para las configuraciones o instalaciones de "SW Calidad" en los "Servidores Aplicativo". - Enviar los RFCs a los responsables. - Verificar si los RFCs fueron correctamente ejecutados. - Verificar Normas de Seguridad impartidas por el Oficial de Seguridad.(Ver archivos adjuntos)

Proceso	Realizar capacitación aplicativo
Descripción	Realizar Capacitación del Aplicativo, idealmente se debiera hacer una Capacitación para Pruebas Funcionales y No Funcionales, pero podría haber casos donde tenga que ser realizada en sesiones separadas.

Proceso	Definir prueba (*)
Descripción	<p>La primera plantilla de apoyo que se utiliza en este proceso es "Definición CPI" ANEXO 1, en la cual se definen todas las combinatorias posibles en base a los criterios especificados en la planilla de apoyo.</p> <p>Una vez seleccionadas las Pruebas a diseñar, se debe preparar la planilla de apoyo "TF Diseño" ANEXO 1, en la cual se debe comenzar a identificar los ID de Prueba y relacionarlos a las Pruebas seleccionadas de la planilla anterior.</p> <p>Al mismo tiempo en este proceso se debe proceder con lo siguiente:</p> <p>Este proceso se basa en:</p> <ul style="list-style-type: none"> - Definición de Requerimientos de Pruebas: De los cuales se derivarán las Pruebas o "carpetas de Pruebas", las cuales serán refinadas con la documentación de Diseño que se tenga del aplicativo. QC tiene la opción de derivar automáticamente de los "Requerimientos de Pruebas" una jerarquía preliminar de Pruebas, la cual debe ser refinada. - La Capacitación Funcional y con otros entregables (sería ideal contar con las grabaciones de

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<p>ejecuciones sobre la GUI del aplicativo) se tiene la información necesaria para detallar los "Pasos" de cada Prueba y generar los "Scripts de Prueba" en el caso de automatización con "Load Runner".</p> <p>y contempla la realización de las siguientes tareas:</p> <ul style="list-style-type: none"> - Definir Estructura Directorios -Definir Pruebas Unitarias - Detallar Pruebas Unitarias - Definir Pruebas Integradas - Generar Scripts - Especificar Necesidad Datos - Detallar Pruebas Integradas <p>Es muy importante tener en mente que en cada ciclo puede ser necesario reejecutar algunas de las anteriores.</p>
--	--

Proceso	Definir conjuntos de pruebas
Descripción	<p>Se definen los conjuntos de prueba, ver ejemplos de estos conjuntos en Diagrama Conjunto_Prueba.</p> <p>En QC, Test Lab:</p> <ul style="list-style-type: none"> - Crear carpeta(s) que contienen "Conjuntos de Pruebas". - Crear cada "Conjunto de Pruebas" en la carpeta que corresponda, e ingresar las "Pruebas" que lo componen usando el tab "Execution Grid". El orden de ejecución se define en el tab "Execution Flow".

Proceso	Vincular Conjunto Pruebas a Ciclos
Descripción	<p>En QC, Test Lab, las carpetas que contienen "Conjuntos de Pruebas" son linkeadas a ciclos. Para esto hacer click derecho sobre la carpeta y "Assign to Cycle".</p>

Proceso	Generar documento plan de pruebas (*)
Descripción	<p>Con todos los procesos anteriores concluidos, y con todas las plantillas de apoyo concretadas, se debe dar pie a la generación del Plan de Pruebas, el cual contiene información de resumen de las decisiones y acuerdos tomados; es decir, es dejar reflejado el acuerdo para dar comienzo formula al proceso de diseño de las Pruebas.</p> <p>Plantilla de apoyo "Plan de Pruebas" ANEXO 1.</p>

Proceso	Generar LB datos
Descripción	<p>Con el fin de realizar las Pruebas es necesario llevar al sistema a un determinado nivel de datos y generar los datos suficientes a utilizar en los escenarios de Prueba, esto implica :</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<ul style="list-style-type: none"> - Generar BDs, las cuales deben ser parte de la LB, la cual debería ser administrada por CM (ver qué pasa en los clientes externos). - Generar Datos en la cantidad necesaria para realizar las Pruebas
--	--

Proceso	Consolidar resultados
Descripción	<p>Esta tarea consiste en determinar si los defectos reportados durante las Pruebas son reducibles a un grupo menor o son preexistentes.</p> <p>En el caso que un grupo de defectos pueda ser reemplazado por un solo defecto:</p> <ul style="list-style-type: none"> - Elegir un defecto del grupo como representativo. - Linkear el resto de los defectos al seleccionado en el paso 1. - Dejar los defectos linkeados en estado "Consolidated".

Proceso	Analizar resultados
Descripción	<p>Esta tarea consiste en determinar si los defectos reportados durante las Pruebas son tales.</p> <ul style="list-style-type: none"> - En el caso que el defecto sea tal, actualizar su estado a "Open". - En el caso que el defecto no aplique, se puede borrar o dejar su estado en "Closed", esto último sólo si es de algún aporte mantener el registro.

5.1.4.1 Proceso “Definir prueba”

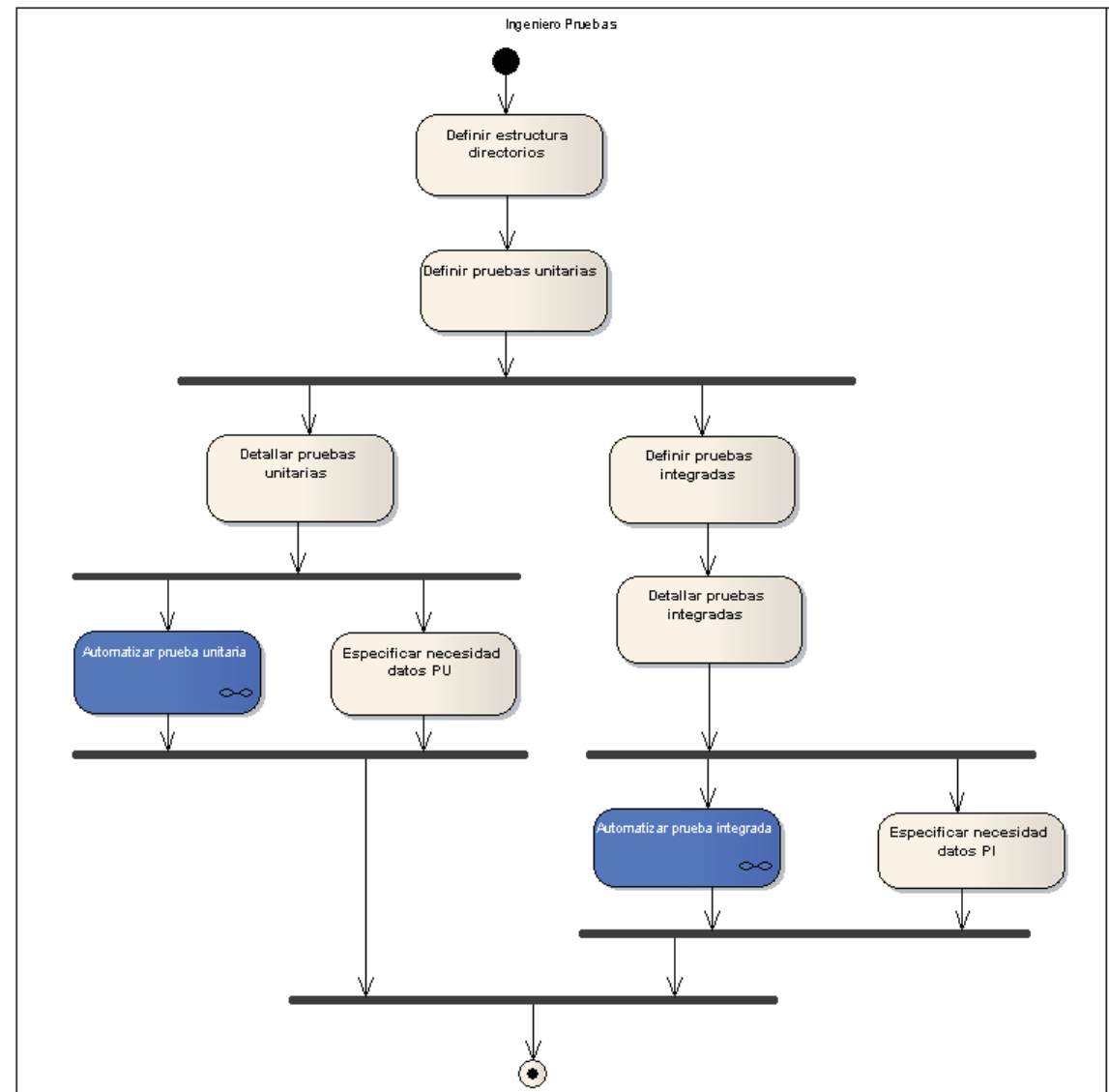


Figura 5.10: Proceso “Definir prueba”

Proceso	Definir estructura directorios
Descripción	<p>Crear los directorios que se necesitan para almacenar los scripts de pruebas, repositorios de objetos y bibliotecas de funciones tanto de QTP como LR. Se recomienda crear en Subversion:</p> <ul style="list-style-type: none"> - Crear directorio de Scripts, se sugiere que el nombre de cada script se un prefijo de a lo más 3 caracteres alfabéticos concatenados con el ID QC de la Pruebas en formato '99999'. - Crear directorio de Repositorios de Objetos (Sólo para uso de QTP, es decir Pruebas Funcionales) - Crear directorio de Bibliotecas de Funciones (Sólo para uso de QTP, es decir Pruebas Funcionales) - Crear directorio de Escenarios de Recuperación <p>Los scripts deben ser actualizados en QC cuando se quiera ejecutarlos y en otro momento no corresponderán necesariamente a la LB.</p> <p>NOTA GENERAL: EL ORDEN EN ESTE DIAGRAMA DE PROCESOS NO IMPLICA QUE UNA TAREA NO SE PUEDA VOLVER A REALIZAR EN UN MOMENTO POSTERIOR CON FINES DE REFINAR UNA DEFINICION YA EXISTENTE, TENIENDO EN MENTE NO OLVIDAR GESTIONAR LA LB, SI CORRESPONDE.</p>

Proceso	Definir pruebas unitarias
Descripción	<p>Una Prueba unitaria en el caso de Pruebas funcionales corresponderá a una unidad funcional indivisible y posiblemente derivable de un requerimiento de Pruebas (si éstos han sido detallados adecuadamente).</p> <p>A este nivel basta con definir la Prueba Unitaria, pero no indicar sus pasos.</p>

Proceso	Detallar pruebas unitarias
Descripción	<p>Esta tarea siempre se realiza para las Pruebas manuales y se evalúa su realización para el caso automático dependiendo del tiempo y del servicio otorgado. Por ejemplo, cuando no se realiza la ejecución por parte de la Organización, si se deberían documentar.</p> <p>Se ingresa la Prueba a QC en Test Plan ->; Design Steps, de acuerdo a la especificación entregada en el Diagrama "Diseño Prueba" para la "Prueba_Unitaria", "Paso_Documentado", "Escenario_Prueba_Unitaria" y "Parametro_Prueba_Unitaria" y "Regla".</p>

Proceso	Especificar necesidad datos PU
Descripción	<p>- Confeccionar documento que indique cual es la necesidad de datos en función de los parámetros que se identifiquen.</p> <p>- Especificar un atributo que permita particionar los escenarios entre los que consideran datos existentes en la Base de Datos y datos generados.</p> <p>- Generar y adjuntar a la Prueba en QC la Planilla Referencial de Datos siguiendo el formato de nombre D999999_01.xls donde los primeros 6 dígitos corresponden al ID generado en QC para la Prueba (Test Plan->; Seleccionar Test->; Tab Attachments). La planilla referencial Excel tiene una hoja con nombre "Global", cuya primera fila contiene los nombres de los parámetros identificados en el punto 1 y tiene tantas filas como escenarios se vayan a probar en la Prueba. Como regla general en el caso manual, debería haber una línea por escenario para hacer más simple la descripción de los pasos y el trabajo del Tester. En el caso automático puede haber más de un escenario manejado por el script.</p> <p>Observación: La planilla referencial Excel es usado como base para las planillas que contendrán los datos reales que se usarán para ejecutar las Pruebas, estas últimas planillas deben ser adjuntadas a los "Test Instance".</p>

Proceso	Definir pruebas integradas
Descripción	<p>Combina 2 o más pruebas unitarias dentro de una sola prueba</p> <p>Esta tarea siempre se realiza para las pruebas manuales y se evalúa su realización para el caso automático dependiendo del tiempo y del servicio otorgado. Por ejemplo, cuando no se realiza la ejecución por parte de organización, si se deberían documentar. (EN EVALUACIÓN DURANTE EL PILOTO)</p>

Proceso	Detallar pruebas integradas
Descripción	Se ingresa la Prueba a QC en Test Plan ->; Design Steps, de acuerdo a la especificación entregada en el Diagrama "Diseño Prueba" para la "Prueba_Integrada".

Proceso	Especificar necesidad datos PI
Descripción	<p>- Se debe especificar un atributo que permita particionar los escenarios entre los que consideran datos existentes en la Base de Datos y datos generados.</p> <p>- Generar y adjuntar a la prueba en QC la Planilla Referencial de Datos siguiendo el formato de nombre D999999_01.xls donde los primeros 6 dígitos corresponden al ID generado en QC para la prueba (Test Plan->; Seleccionar Test->; Tab Attachments). La planilla referencial Excel tiene una hoja de nombre Global para la Prueba unitaria guía y tantas hojas como Pruebas unitarias dependan de la guía en una relación uno es a muchos. Como ejemplo considere el caso de un Maestro-Detalle, el maestro es la Prueba unitaria guía y el detalle es la Prueba unitaria asociada, esta última tendrá una hoja propia.</p> <p>Observación: La planilla referencial Excel es usada como base para las planillas que contendrán los datos reales que se usarán para ejecutar las Pruebas, estas últimas planillas deben ser adjuntadas a los "Test Instance".</p>

5.1.4.2 Proceso “Automatizar prueba unitaria”

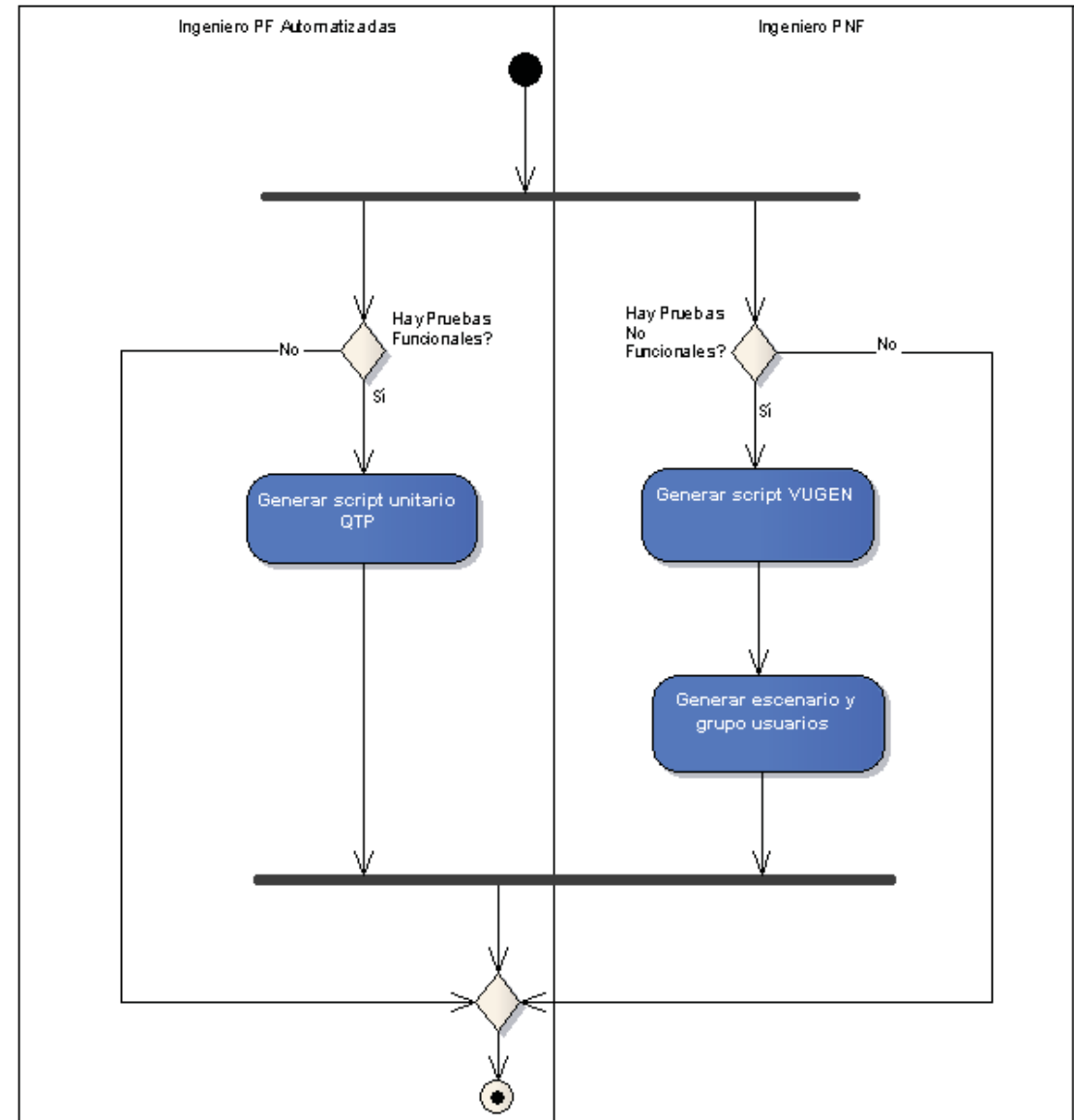


Figura 5.11: Proceso “Automatizar prueba unitaria”

Proceso	Generar script unitario QTP
Descripción	<ul style="list-style-type: none"> - Se graba un "Acción QTP" dentro del script, siguiendo los pasos (recordar que en QTP las validaciones de cualquier tipo son pasos) y parametrización detallada para la prueba en QC. Debido a que puede haber varios flujos de pasos (al menos el caso positivo y el negativo) probablemente se deba intervenir el script en su vista "Keyword-View" o "Script-View". - La "Acción QTP" debe informar siempre a través de Report.ReportEvent si la prueba fue exitosa o falló. - Al final el script se almacena en el directorio que corresponde para que sea versionado. - Se recomienda medir el tiempo de respuesta, pero sólo con el fin de tener una referencia, pues debido a que ni el Ambiente de Pruebas ni las Bases de Datos en él son representativos del Ambiente Productivo en régimen permanente.

Proceso	Generar script VUGEN
Descripción	<ul style="list-style-type: none"> - Se graba el login y el logout dentro de dos acciones. - Para cada transacción acordada con el Cliente como parte de la prueba se graba un script con una acción y dentro de ella se marca el inicio y el término de la "Transaction" la cual mide el tiempo sujeto a SLA. - A continuación se debe depurar el script revisando: Pacing, Log, Think Time, Network - Speed Simulation (útil cuando el ancho de banda disponible es mayor al real) y detección de Correlación y cerciorándose que el script se puede re-ejecutar exitosamente. - Dependiendo del tipo de aplicativo, agregar puntos de Verificación para determinar si la transacción está retornando adecuadamente, pues hay veces que la no disponibilidad no se manifiesta en la caída de la aplicación sino en comportamiento anormal.

Proceso	Generar escenario y grupo usuarios
Descripción	En el caso de Pruebas de Carga y Rendimiento Unitarias, usando LR Controller, se declara un escenario por cada script usando un solo grupo programando tramos de 1, 10, 20, 30 y 50 usuarios respectivamente. La duración de cada tramo debería ser al menos 5 minutos con el fin de ver la estabilidad de la respuesta del sistema, si bien tiempos mayores son recomendables, su uso dependerá del tiempo total asignado a esta prueba.

5.1.4.3 Proceso “Automatizar prueba integrada”

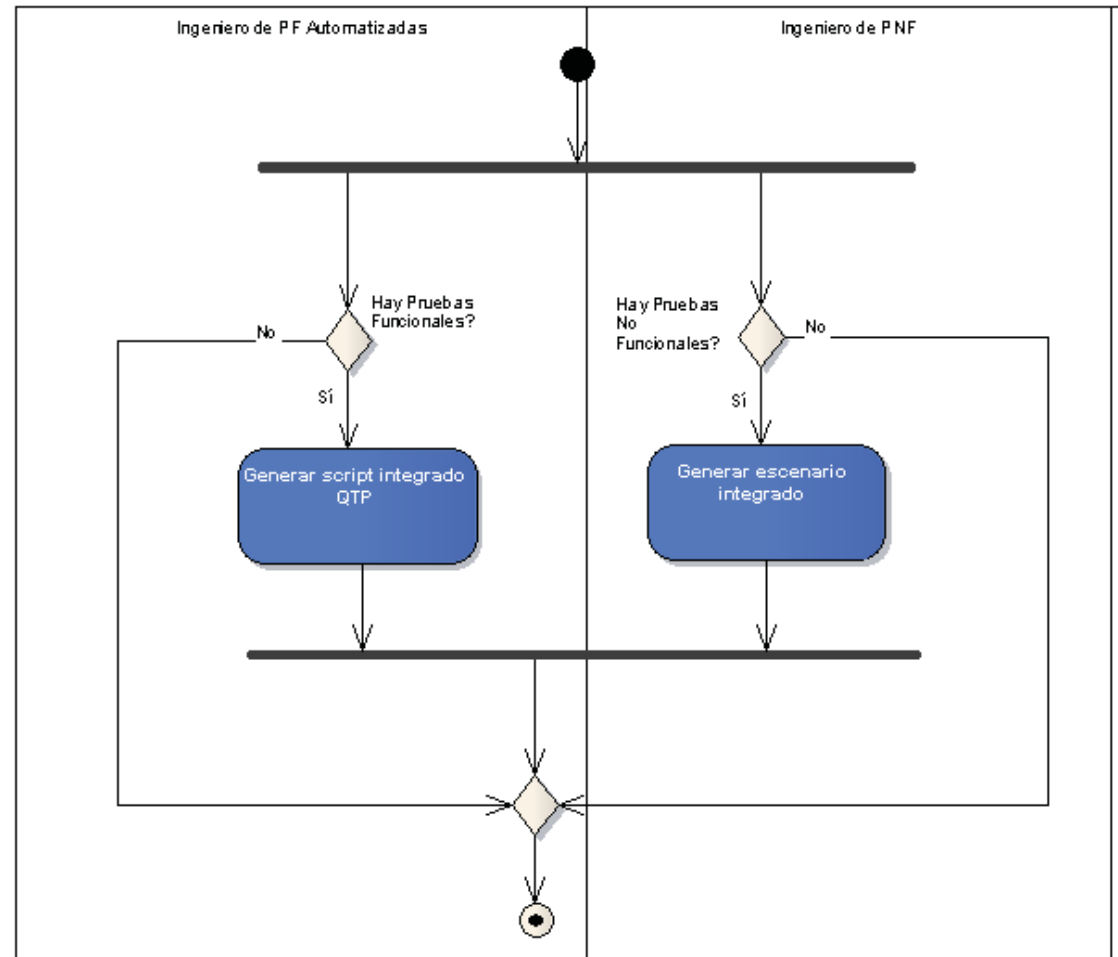


Figura 5.12: Proceso “Automatizar prueba integrada”

Proceso	Generar script integrado QTP
Descripción	En este punto ya se han generado las "Acciones QTP" para las pruebas unitarias, entonces: <ul style="list-style-type: none"> - Se crea un script en el cual se referencia todas aquellas acciones que componen la "Prueba Integrada" siguiendo el flujo definido. - Se declara una planilla de datos por acción, la cual deberá ser llenada.

Proceso	Generar escenario integrado
Descripción	Se genera en Load Runner Controller, el escenario integrado definiendo todos los Grupos de Usuarios que se haya contemplado en la prueba y usando los Script ya generados en VUgen para las Pruebas Unitarias.

5.1.4.4 Proceso "Generar LB datos"

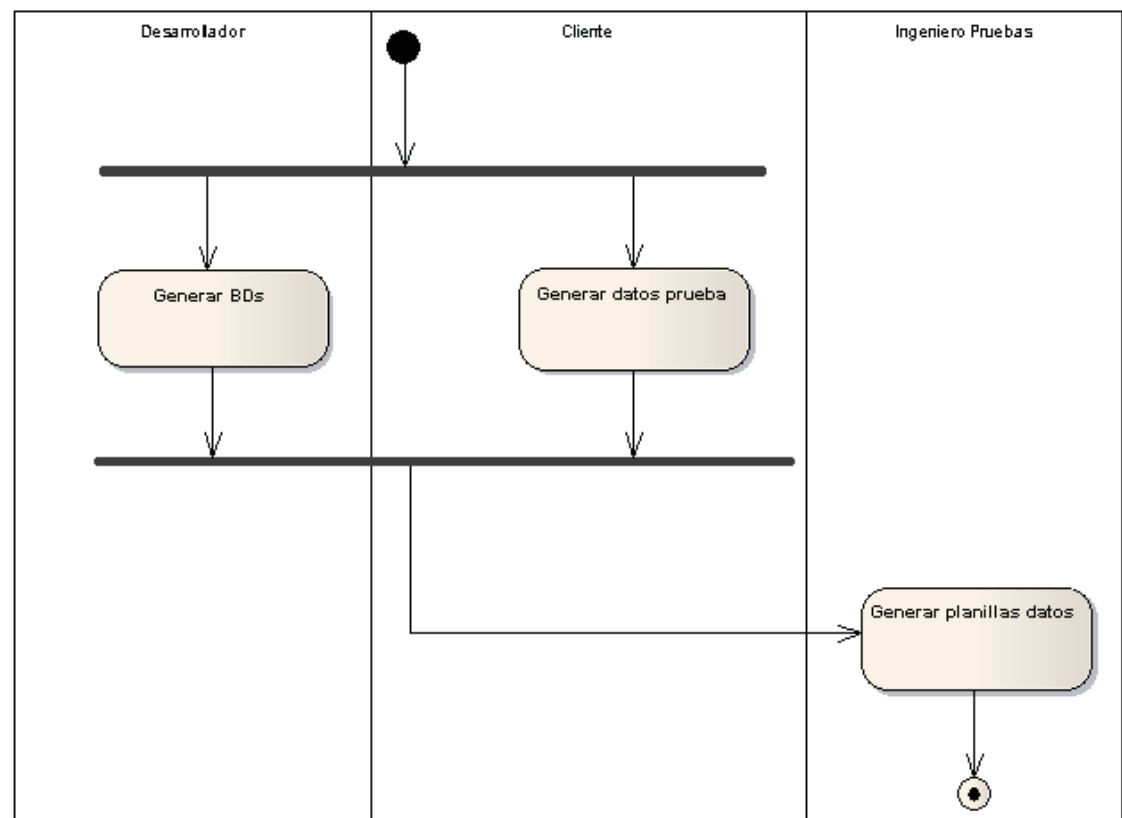


Figura 5.13: Proceso "Generar LB datos"

Proceso	Generar BDs
Descripción	Esta generación depende del tipo de pruebas y se deben considerar los siguientes aspectos: <ul style="list-style-type: none">- Las pruebas no funcionales requerirán un tamaño representativo del aplicativo en régimen permanente.- Se debe velar por la confidencialidad de los datos.- Los procedimientos de Respaldo y Restauración deben existir y estar probados antes del inicio de las pruebas, esto es necesario debido a que muchas veces se necesitan realizar las pruebas más de una vez usando la misma Release del SW. Ejemplo: Se hicieron pruebas no funcionales en las cuales se detectaron problemas en el HW, los cuales fueron resueltos y se requiere probar de nuevo.

Proceso	Generar datos prueba
Descripción	Generación a través de consultas SQL o Recolección de Datos en la cantidad necesaria para realizar las pruebas.

Proceso	Generar planillas datos
Descripción	Usando las Planillas Excel Referenciales adjuntadas a cada prueba durante las actividades "Especificar Necesidad Datos PU" y "Especificar Necesidad Datos PI", generar las Planilla Excel para las pruebas que serán realizadas durante el ciclo en curso.

5.1.4.5 Proceso “Ejecutar ciclo pruebas”

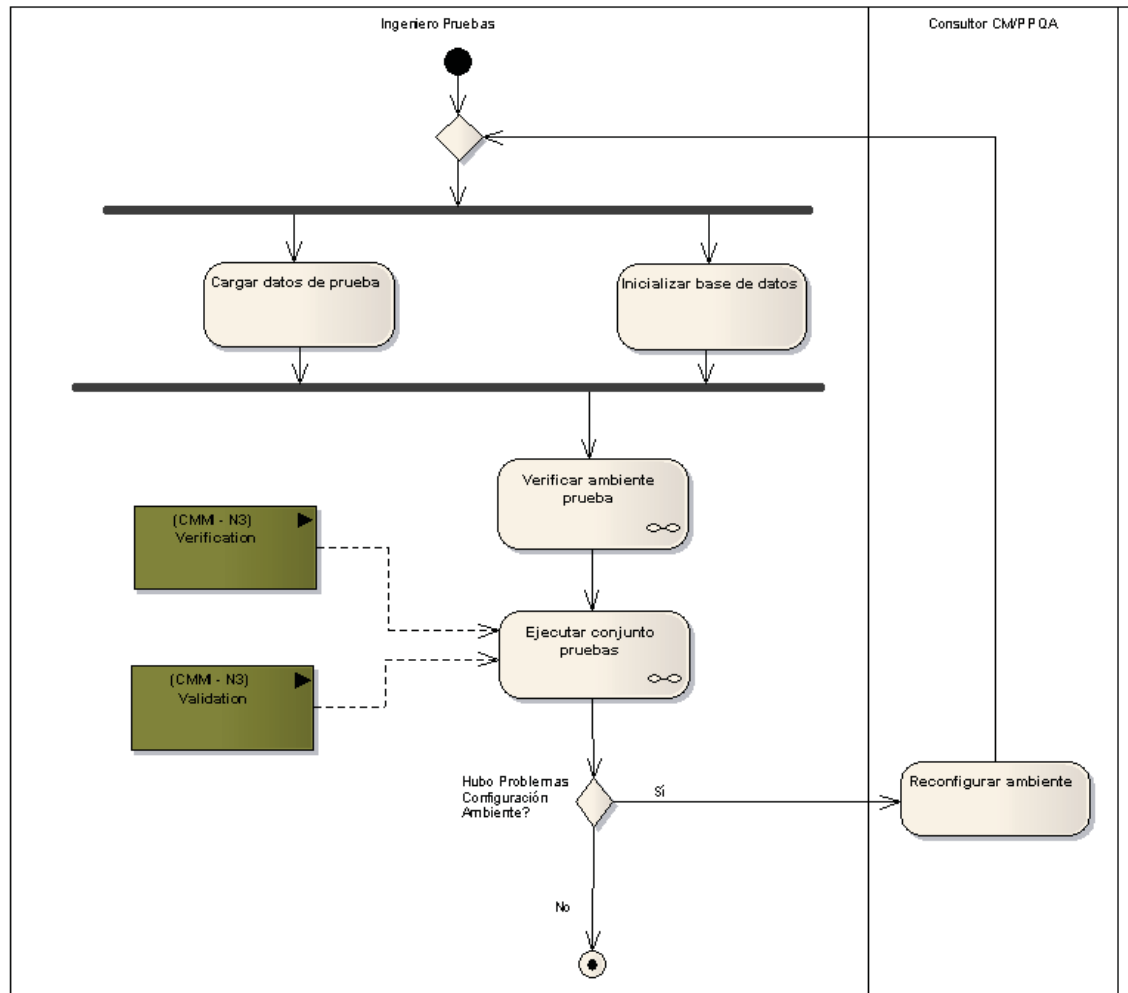


Figura 5.14: Proceso “Ejecutar ciclo pruebas”

Proceso	Cargar datos de prueba
Descripción	Cargar las Planillas de Datos de Prueba generadas en "Generar Planillas Datos" en las "Test Instance" del ciclo.

Proceso	Inicializar base de datos
Descripción	Inicializar la(s) base(s) de datos es setear esta(s) misma(s) en la condición inicial para que comience el Ciclo correspondiente, lo que implica: <ul style="list-style-type: none">- Datos Necesarios y consistentes con los datos que se usarán en las pruebas.- Para el caso no funcional, además se requiere que la(s) Base(s) de Datos tenga(n) un tamaño representativo al que tendría en el estado permanente en producción.

Proceso	Ejecutar conjunto pruebas
Descripción	Se ejecuta cada prueba dentro del Conjunto de Pruebas ya sea manual o automáticamente. El primer Conjunto de Pruebas a ejecutar debería ser algún tipo de "Pruebas Trazadoras". En el caso de Pruebas No Funcionales, es altamente recomendable que las Pruebas de Carga y Rendimiento Unitarias se ejecuten al menos 3 veces, para tener una confiabilidad mínima en la medición de tiempos y comportamiento del sistema.

5.1.4.6 Proceso “Verificar ambiente prueba”

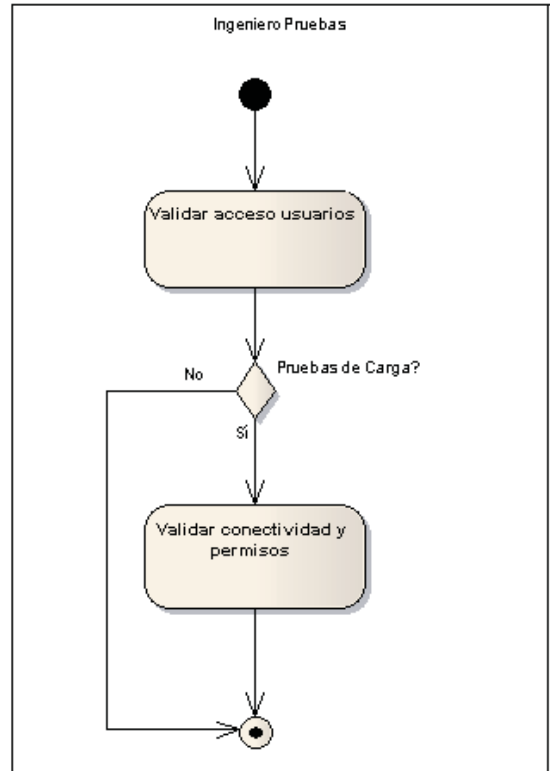


Figura 5.15: Proceso “Verificar ambiente prueba”

Proceso	Validar acceso usuarios
Descripción	Validar que sólo las personas responsables de las pruebas tengan acceso al sistema.

Proceso	Validar conectividad y permisos
Descripción	En el tab Run del LR Controller validar: <ul style="list-style-type: none"> - Conectividad del Controller a Generadores de Carga (Icono Load Generators). - Conectividad y permisos de monitoreo desde el Controller a los servidores del aplicativo y de BDs. (Al agregar las métricas de Windows Resources).

5.1.4.7 Proceso “Ejecutar conjunto de pruebas”

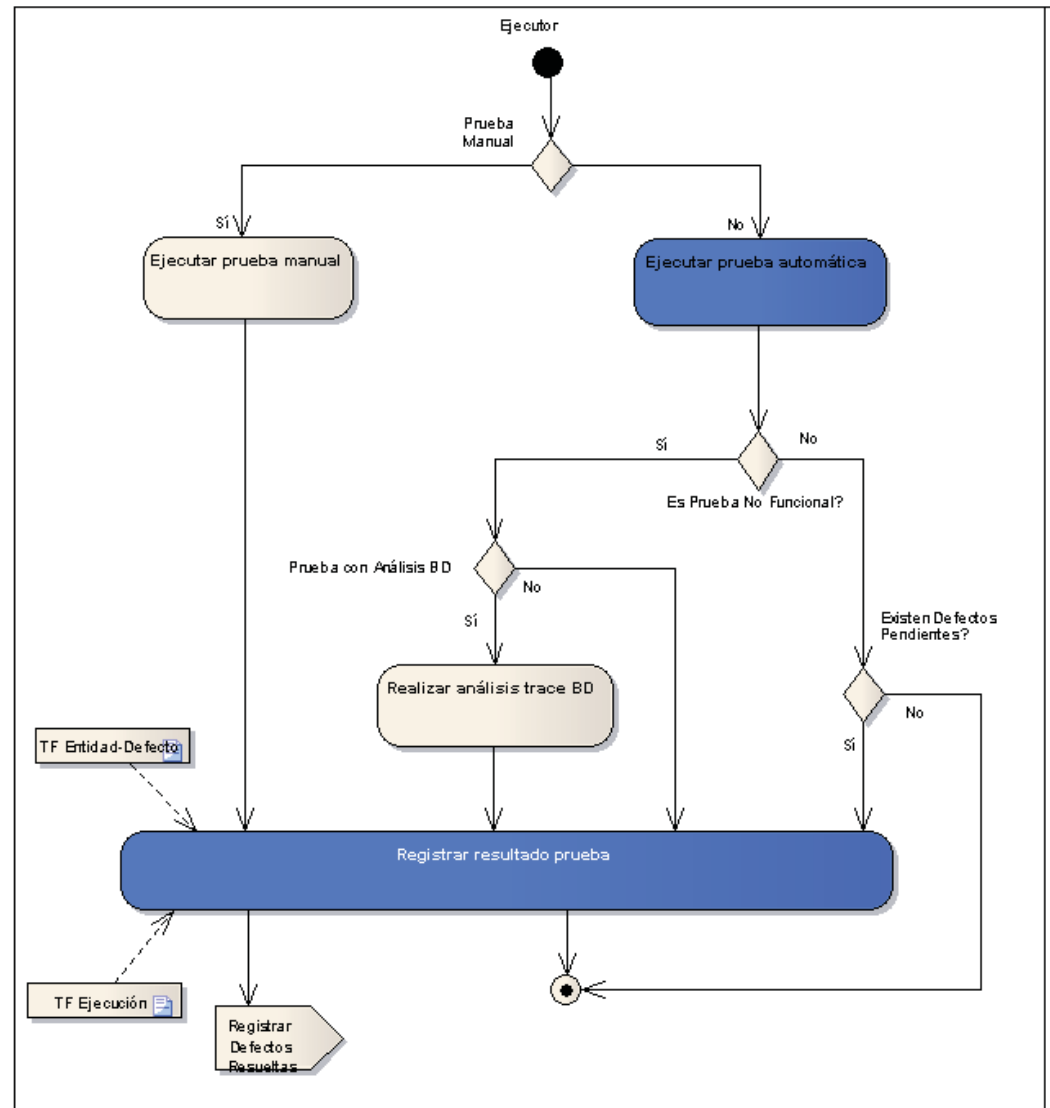


Figura 5.16: Proceso “Ejecutar conjunto de pruebas”

Proceso	Ejecutar prueba automática
Descripción	Para el Caso de Pruebas Funcionales se debe setear en QTP que registre automáticamente cualquier error como defecto.

Proceso	Registrar resultado prueba (*)
Descripción	<p>Una vez que el Analista/Ejecutor ejecuta las pruebas diseñadas, debe ir dejando registro de dichas pruebas en la planilla de “TF Ejecución” (Ver ANEXO1), la cual irá dejando la evidencia de la ejecución.</p> <p>En el caso que la prueba obtenga estado NOK, se debe registrar en la planilla "TF Entidad-Defecto" (Ver ANEXO 1), todos los defectos encontrados en la prueba.</p> <p>Al momento de ejecutar, basta con que un paso no sea el esperado para que el caso de prueba total obtenga el estado NOK, sin embargo, que un paso sea NOK no implica necesariamente que no se puedan seguir testeando los demás pasos.</p>

5.1.5 Proceso “Administrar proceso de pruebas” PARTE 3

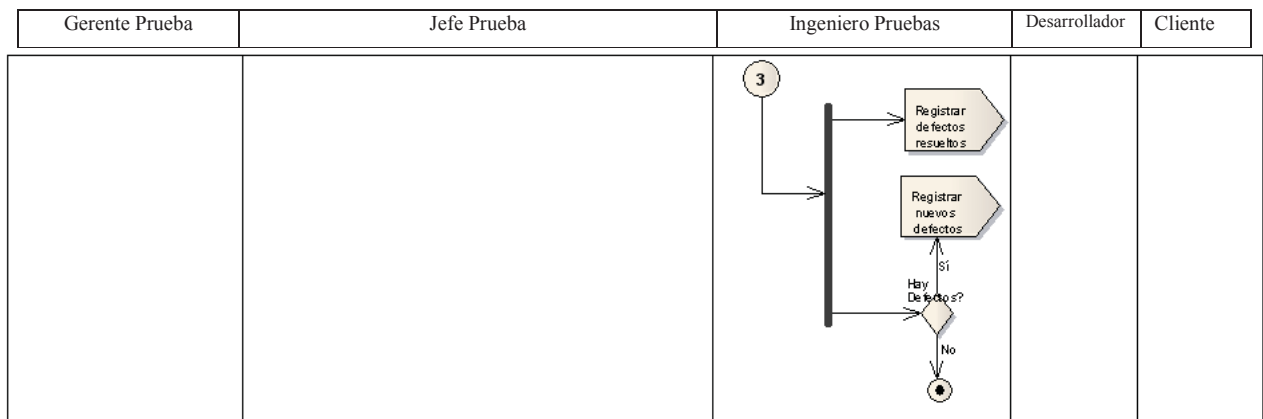


Figura 5.17: Proceso “Administrar proceso de pruebas” PARTE 3

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.6 Proceso: “Gestionar defectos”

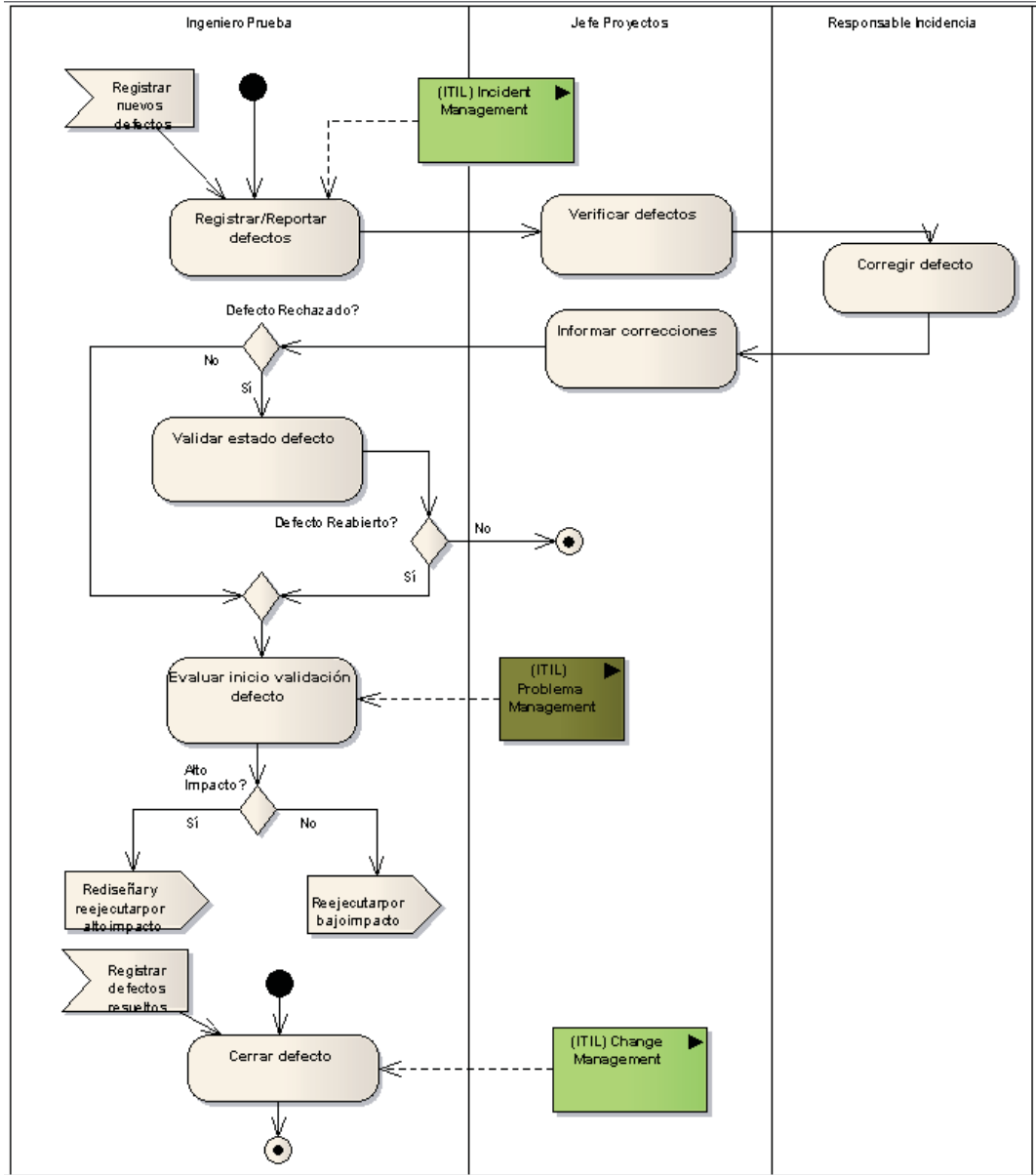


Figura 5.18: Proceso “Gestionar defectos”

Proceso	Registrar/reportar defectos
Descripción	<p>- Ingresar los atributos del defecto, es importante el ingreso del campo Área Calidad (CM, PPQA, TF, TNF) con el fin de cada área pueda posteriormente identificar que defectos debe gestionar. También no se debe identificar al "Responsable de Solución" que puede ser el Desarrollador, Responsable HW, Responsable SW, Responsable Redes.</p> <p>- Vincular las entidades que gatillan el defecto, recuerde que pueden ser las siguientes:</p> <ul style="list-style-type: none"> • Requerimiento • Prueba • Conjunto de Pruebas • Instancia de Pruebas • Ejecución de Pruebas • Paso de Pruebas (En el caso que se esté ejecutando una prueba, lo más recomendable vincular al paso el defecto y adjuntar la evidencia a este último) • A otro defecto (Por deducción) <p>Para que el defecto sea procesado como tal, se lo debe dejar en estado "Open".</p> <p>Reportes de Defectos</p> <p>- CM y PPQA, en Customize->; Excel Report Generator->; CM-PPQA Matriz No Conformidades.</p>

Proceso	Corregir defecto
Descripción	<p>El Jefe de Proyecto o el equipo de desarrollo evalúan si corresponde a un defecto, y de serlo inicia su corrección.</p> <p>Si el defecto no se acepta, se debe setear en estado "Rejected".</p> <p>Si el defecto es aceptado, en el momento de resolverlo se debe setear el estado "Fixed".</p>

Proceso	Validar estado defecto
Descripción	<p>Validar si el estado del defecto rechazado aplica o no. Si aplica, poner el defecto en estado "Closed", en caso contrario en "Reopen" y enviárselo al Desarrollador y/o Jefe de Proyecto.</p>

Proceso	Evaluar inicio validación defecto
Descripción	En función del impacto se decide si modificar el diseño o simplemente reagendar la prueba, el cual es realizado una vez que el Ciclo actual se encuentra terminado: <ul style="list-style-type: none">- Alto impacto =>; Revisar Diseño y reejecutar algunos "Conjuntos de Pruebas"- Bajo Impacto =>; Reejecutar algunos "Conjuntos de Pruebas"

Proceso	Cerrar defecto
Descripción	En el caso que un defecto haya sido resuelto, setear el estado "Closed", esto debería ser hecho en el ciclo que ocurre.

5.1.7 Proceso: “Realizar auditoría CM/PPQA”

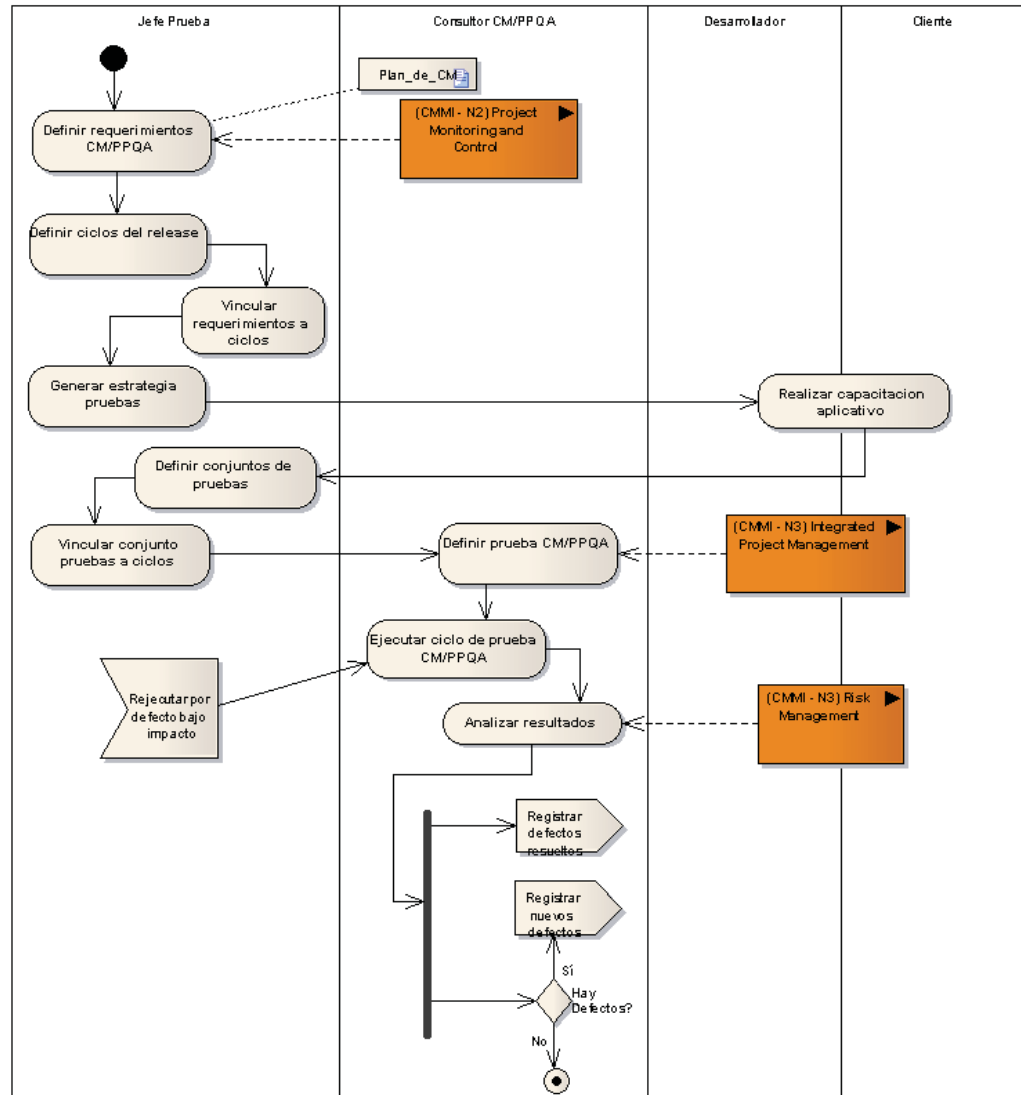


Figura 5.19: Proceso “Realizar auditoría CM/PPQA”

Proceso	Definir requerimientos CM/PPQA (*)
Descripción	<p>La planilla de apoyo "Plan de CM" se encuentra en el ANEXO 1. Dicha planilla apoya a la planificación de perfilamientos y definición de la biblioteca de configuración de los ítems.</p> <p>Propone una estructura base, especificando que debiese tener cada una de ellas.</p> <p>Se definen los requerimientos de CM/PPQA detallando los atributos mencionados a continuación, si es necesario se crean carpetas que agrupen los requerimientos por temas.</p> <ol style="list-style-type: none"> 1. Nombre 2. Especificar Tipo de Requerimiento como de Negocios (CM o PPQA) 3. Descripción 4. Autor 5. Prioridad en la implementación: <ol style="list-style-type: none"> 1-Baja 2-Media 3-Alta 4-Muy Alta 6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)

Proceso	Definir ciclos del release
Descripción	<p>Definir los "Ciclos del Release" indicando los "Conjunto de Pruebas" preliminares que los conforman, por ejemplo el primer "Ciclo de Pruebas" debiera incluir siempre un conjunto de "Pruebas Trazadoras" con el fin de establecer si el ambiente está operativo o no.</p> <p>En QC es posible realizar Attachments de información del Ciclo, como podría ser la planificación del mismo. Pero aparentemente estos elementos no son versionados, por lo que quizá es más recomendable sólo adjuntar Links a Subversion.</p>

Proceso	Vincular requerimientos a ciclos
Descripción	<p>Se debe vincular los requerimientos a al menos un "Ciclo de Prueba" con el fin de:</p> <ul style="list-style-type: none"> - Determinar en cualquier instante si los requerimientos están siendo cubiertos o no - Generar esbozo de pruebas en función de los requerimientos asociados a un ciclo. <p>Un requerimiento puede ser vinculado a un Release, en este caso, todos los "Ciclos de Prueba" dentro del Release serán vinculados, y eso implica que debe haber un "Conjunto de Pruebas" al menos dentro de cada Ciclo que valida el requerimiento.</p> <p>Por el momento, no se realizan Pruebas de Regresión al 100% en todos los ciclos por motivos de</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	recursos. Así la cobertura de prueba en un ciclo queda definida por la liberación de funcionalidades nuevas y por los defectos ocurridos en el ciclo anterior (en el caso que exista), por lo tanto al principio del proyecto, sólo se vincularán los requerimientos asociados a funcionalidades nuevas al ciclo en que se prueban, y con posterioridad dependiendo de los defectos se vincularán requerimientos a ciclos que contengan Pruebas de Regresión.
--	---

Proceso	Generar estrategia pruebas
Descripción	<p>Este documento debe ser un compendio del estado del proyecto hasta este momento, considerando:</p> <ul style="list-style-type: none"> - Roles y personas - Requerimientos de Pruebas - Ciclos de Pruebas - Protocolo de Suspensión de las Pruebas - Conclusiones de Estudio Factibilidad de Automatización. - Referencias a los documentos de Arquitectura de HW/SW. - Formato Plan de Pruebas o Operatoria al respecto - Formato Informe de Resultados de Pruebas. - Manejo de Defectos. - Severidad de Defectos. - Riesgos <p>En el caso de CM-PPQA se incluyen los checklists que se utilizarán en la Revisión.</p>

Proceso	Realizar capacitación aplicativo
Descripción	Realizar Capacitación del Aplicativo, idealmente se debiera hacer una Capacitación para Pruebas Funcionales y No Funcionales, pero podría haber casos donde tenga que ser realizada en sesiones separadas.

Proceso	Definir conjuntos de pruebas
Descripción	<p>Se definen los conjuntos de prueba, ver ejemplos de estos conjuntos en Diagrama Conjunto_Prueba.</p> <p>En QC, Test Lab:</p> <ul style="list-style-type: none"> - Crear carpeta(s) que contienen "Conjuntos de Pruebas". - Crear cada "Conjunto de Pruebas" en la carpeta que corresponda, e ingresar las "Pruebas" que lo componen usando el tab "Execution Grid". El orden de ejecución se define en el tab "Execution Flow".

Proceso	Vincular conjunto pruebas a ciclos
Descripción	En QC, Test Lab, las carpetas que contienen "Conjuntos de Pruebas" son linkeadas a ciclos. Para esto click derecho sobre la carpeta y "Assign to Cycle".

Proceso	Analizar resultados
Descripción	Esta tarea consiste en determinar si los defectos reportados durante las pruebas son tales. - En el caso que el defecto sea tal, actualizar su estado a "Open". - En el caso que el defecto no aplique, se puede borrar o dejar su estado en "Closed", esto último sólo si es de algún aporte mantener el registro.

El ciclo de los defectos o incidentes que se explican en los Modelos anteriores es el siguiente:

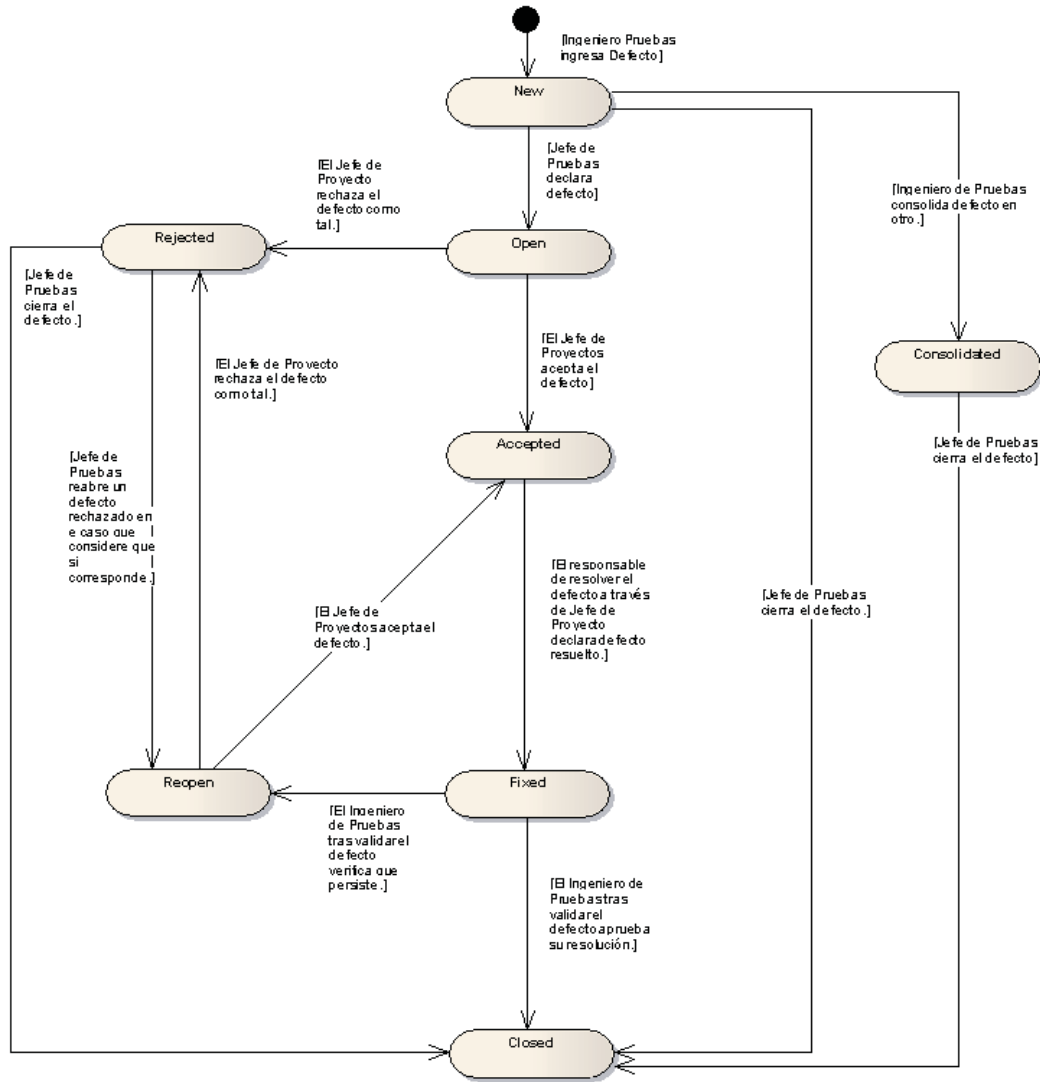


Figura 5.20: “Ciclo de defectos”

5.1.8 Proceso: “Controlar proyecto”

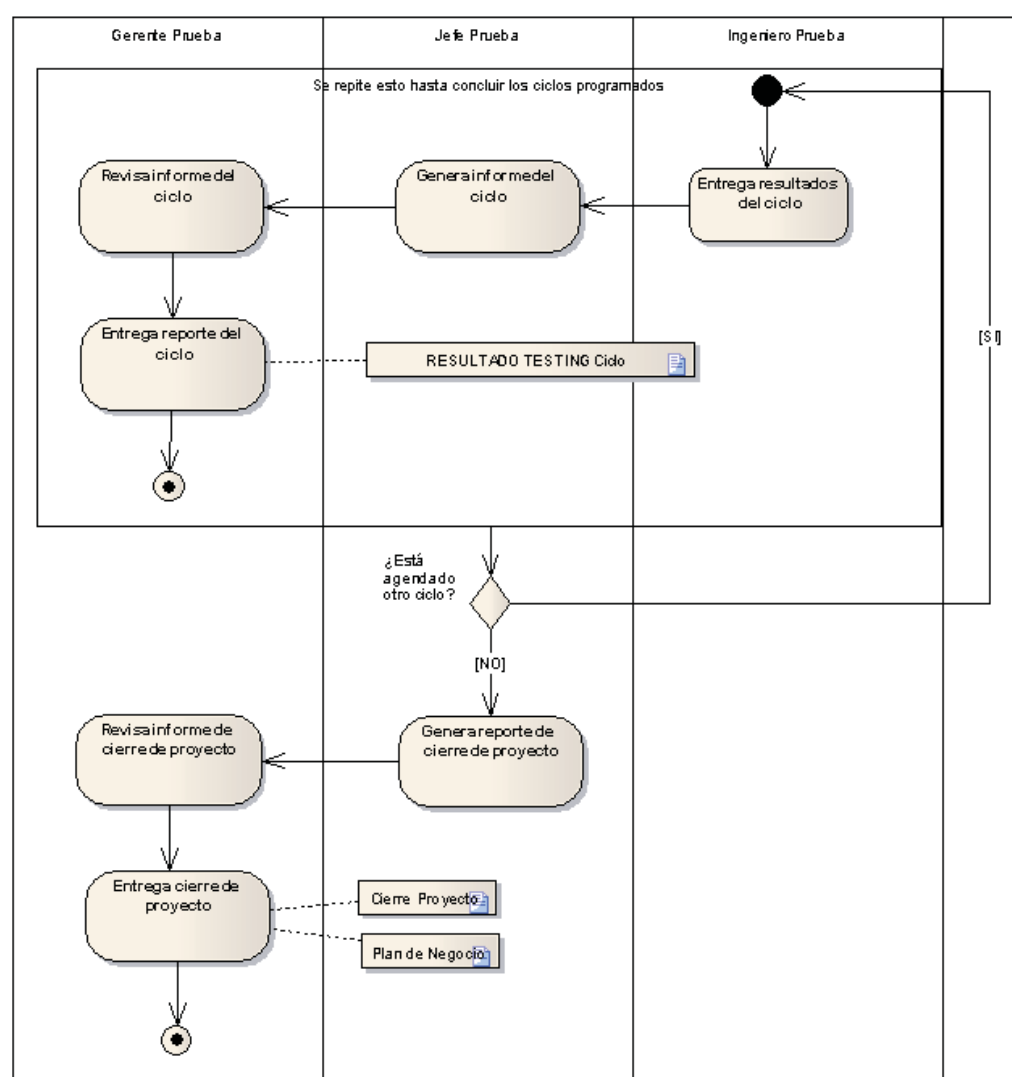


Figura 5.21: Proceso “Controlar proyecto”

Proceso	Entregar resultados del ciclo
Descripción	Una vez ejecutadas las pruebas el Ingeniero de Pruebas entrega los resultados de la ejecución y de los defectos encontrados.
Proceso	Generar informe del ciclo
Descripción	Con la entrega de los resultados el Jefe de Proyecto construye el Informe del Ciclo y lo entrega al Gerente de Pruebas para su revisión
Proceso	Revisar informe del ciclo
Descripción	Cuando el Gerente de proyecto revisa el informe lo libera al equipo de Proyecto para su análisis y revisión en conjunto al equipo de Testing.
Proceso	Entrega reporte del ciclo (*)
Descripción	El Reporte que se genera se basa en la planilla de apoyo “RESULTADO TESTING Ciclo” ANEXO 1. Dicho informe se basa esencialmente en la entrega de los resultados en comparación a las políticas de paso a producción planteadas en la planilla de apoyo “Plan de Pruebas”.
Proceso	Generar reporte de cierre de proyecto (*)
Descripción	En el caso que se terminen de ejecutar todos los ciclos programados o solicitados por emergencia es la hora que el Jefe de Proyecto Genere el reporte de Cierre de Proyecto utilizando la planilla de Apoyo “Cierre Proyecto” y “Plan de Negocio” del ANEXO 1. Luego de generado el reporte, es entregado al Gerente de Proyecto.
Proceso	Revisar informe de cierre de proyecto
Descripción	El Gerente de Proyecto revisa el Informe de Cierre de Proyecto y el Plan de Negocio entregado por el Jefe de Proyecto con el fin de evaluar las diferencias entre lo programado y lo real.
Proceso	Entrega cierre de proyecto
Descripción	Una vez realizada la revisión por el Gerente de Proyecto los entrega y presenta al equipo de proyecto, dejando en conocimiento la desviación u optimización en el caso que exista en el Plan de Negocio junto a los resultados finales de la intervención del Testing.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.2 Resumen de resultados obtenido (Métricas)

A partir de las métricas definidas inicialmente se procede a evaluar cada una de ellas con el fin de poder analizar cómo se comportó la aplicación de la Metodología aplicada al proyecto.

Nº	Medición	Medida	Métrica
1	Se medirá el cumplimiento del esfuerzo establecido para el Testing del producto.	HH	Esfuerzo Estimado v/s Esfuerzo Real
2	Se medirá el cumplimiento del costo establecido para el Testing del proyecto.	UF	Costo Estimado v/s Costo Real
3	Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.	%	% Establecido v/s % Obtenido

- 1) Métrica 1: Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto.

En base al Modelo de Servicio se obtienen los siguientes datos:

Días	HH	
2	13	Jefe de Área
14	112	Jefe de Proyecto
27	218	Externos
43	343	Total Esfuerzos

Según los datos obtenidos del Plan de Negocio se tiene lo siguiente:

Mes 2	Mes 3	Mes 4	
Ago-09	Sep-09	Oct-09	
5.00 hh	11.00 hh	6.50 hh	
90.00 hh	140.00 hh	101.00 hh	
			Total HH planificadas
95.00 hh	151.00 hh	107.50 hh	353.50 hh

Es decir, se planificaron 10 horas más de las inicialmente estimadas.

Lo que realmente se gasta en HH son las siguientes:

Mes 2	Mes 3	Mes 4	
Ago-09	Sep-09	Oct-09	
22.00 hh	24.80 hh	58.00 hh	
72.00 hh	214.70 hh	99.00 hh	
		34.00 hh	
	40.00 hh	9.00 hh	
			Total HH Reales
94.00 hh	279.50 hh	200.00 hh	622.50 hh

Existe una diferencia de 269 HH según la estimación inicial y lo real.

- 2) Métrica 2: Se medirá el cumplimiento del costo establecido para el Testing del proyecto.

En base al Modelo de servicio se obtienen los siguientes datos:

UF	
14	Costo Jefe de Área
107	Costo Jefe de Proyecto
194	Costo Externos
315	Total Costo Proyecto

Según los datos obtenidos del Plan de Negocio, se obtiene lo siguiente:

APROBADO TOTAL COSTO SERVICIO TF	446.33 UF
REAL TOTAL COSTO SERVICIO TF	541.06 UF
DIFERENCIA	-94.73 UF

3) Métrica 3: Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.

Los criterios de aceptación establecidos para este proyecto fueron los siguientes (ver planilla aplicada Plan de Pruebas) son:

- 0% de los Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad INVALIDANTE.
- A lo más, 5% de Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad ALTA.
- A lo más, 10% de Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad Media o Baja.
- A lo menos, 85% de Casos de Prueba Integrados efectivamente probados con estado OK.

Los resultados obtenidos en el ciclo 1 ejecutado son los siguientes:

En base a la necesidad de acortar el ciclo de Testing, finalmente se probó un 62% de las pruebas inicialmente planificadas; es decir, 241 CPI de las 387 CPI planificadas.

El resultado de la ejecución de los 241 CPI arroja los siguientes resultados:

Failed	Passed	Total General	
84	157	241	CPI
35%	65%	100%	Porcentajes

Existen 146 CPI con estado N/A, estos casos son los que NO se ejecutaron en el primer ciclo.

De lo anterior se desprende que el aplicativo tiene un **Nivel de Madurez del 65%** sobre los casos de pruebas **ejecutados**, con los siguientes defectos encontrados:

Cuenta de Criticidad	Criticidad				Total general
	1-Baja	2-Media	3-Alta	4-Invalidante	
Total	14	23	62	7	106
Porcentaje	35%		58%	7%	100%

Se recomienda realizar un CICLO 2 de pruebas, para aclarar si las incidencias encontradas en el ciclo anterior, han sido corregidas o corresponden a errores en la instalación del software, en el ambiente de Testing utilizado. Además no se cumplieron los criterios establecidos en el Plan de Pruebas para el paso a producción.

Los resultados obtenidos en el ciclo 2 ejecutado se describen a continuación:

Se indica la estrategia usada en el ciclo 2 de Certificación del proyecto que considera la redefinición del total de pruebas definidas en el Plan de Pruebas inicialmente aprobado, en base a los cambios solicitados por el proyecto (inicialmente en el ciclo 2 se habían definido 194 CPI a ejecutar).

En base a la nueva estrategia que apunta a minimizar los riesgos de no completar las pruebas del ciclo 1, tal como estaban planificadas, en el ciclo 2 se definió la ejecución de 215 Casos de Prueba que estaban compuestos de la siguiente manera:

Failed	Passed	Total general	
65	150	215	CPI
30%	70%	100%	Porcentajes

De lo anterior se desprende que el aplicativo tiene un **Nivel de Madurez del 70%** sobre el total de casos de pruebas **ejecutados**, con la siguiente distribución de los defectos encontrados:

Cuenta de Criticidad	Criticidad				Total general
	1-Baja	2-Media	3-Alta	4-Invalidante	
Total general	2	27	52	9	90
Porcentajes	32%		58%	10%	100%

Se recomienda realizar un CICLO 3 de pruebas, ya que el Nivel de Madurez obtenido no es suficiente para el Paso a Producción del aplicativo. Además no se cumplieron los criterios establecidos en el Plan de Pruebas para el paso a producción.

Pese a lo anterior, el equipo de proyecto define pasar igualmente el aplicativo a producción con los antecedentes de cuáles son los errores detectados para así declararlos a los clientes finales y realizar seguimiento de su resolución.

Los puntos importantes que se rescatan de la entrevista final con la persona que dirigió la aplicación de la Metodología fueron los siguientes:

- 1) Si en el Modelo se tiene claridad de los costos asociados a cada recurso se puede lograr con más fineza el cálculo de la métrica final.
- 2) Las tasas de rendimiento incorporadas en el Modelo de Servicio distaron mucho de lo real; la tasa de rendimiento de los diseños no fue 5 CPI por hora, lo real se acercó al 2.5 por hora.
- 3) Las tasas de rendimiento incorporadas en el Modelo de Servicio de la ejecución de las pruebas no fue 5 CPI por hora, sino que fue de 3 CPI por hora.
- 4) Además es importante mencionar que los costos utilizados en el Modelo Inicial son distintos a los que finalmente se obtuvieron en el Plan de Negocio. Dichos costos fueron finalmente de 1.01 UF HH del Jefe de Proyecto y de 0.88 UF HH para los Externos, Analistas y Ejecutores.
- 5) A pesar de lo antes expuesto, el proceso fue limpio y claro de aplicar. Se siguiere generar una BD de conocimiento de proyectos con el fin de ir refinando cada vez más la estimación.

En base a las observaciones entregadas anteriormente se realiza el ejercicio de ajustar el Modelo de servicio inicial, obteniendo los resultados que se muestran en la Figura 5.22.

Variables					
Universo de Pruebas	2024	CPI	Cantidad de Gestión LB CM	10	Cantidad
Cobertura	85	Universo Funcionalidades	HH invertidas Gestión LB en TF	1	HH
	58	Funcionalidades a cubrir			
	89	% sobre universo	Tiempo Total de Proyecto	0	Meses
Profundidad	20	% sobre universo		0	Semanas
				0	Días hábiles
Cantidad de Ciclos	405	CPI Ciclo 1 (100%)	Costo total del proyecto	1106	UF
	202	CPI Ciclo 2 (50%)	Costo máximo Propuesta TF	332	UF
Diseños	405	CPI			
Ciclo Extra	202	CPI Ciclo N (50%)	Nº de Entregas de SW	1	Entregables de SW
HH Mensuales	160	Horas mes			
HH trabajo Diario Interno	8	HH Dia			
HH trabajo Diario Externo	8	HH Dia			
Jefe de Área			Costos Jefe de Área	161	UF Mes
Cantidad	1			40,3	UF Semana
% Dedicación	5,00%	%		8,05	UF Dia Hábil
Horas mensuales	0,08	HH Mensuales		1,01	UF HH
Costo Mensual					
Costo HH					
Jefe de Proyecto			Costos Jefe de Proyecto	141	UF Mes
Cantidad	1			35,3	UF Semana
% Dedicación	100,00%	Real 50%		7,05	UF Dia Hábil
Horas mensuales	1,6	HH Mensuales		0,88	UF HH
Nº Analistas/Ejecutores			Costos HH Externos	141	UF Mes
Cantidad	1			35,2	UF Semana
% Dedicación	100,00%	%		7,04	UF Dia Hábil
Horas mensuales	160	HH Mensuales		0,88	UF HH
Tasa de Rendimiento Diseño	2,5	CPI por HH por Recurso			
	20	CPI por HH trabajo Diario Externo			
Tasa de Rendimiento Ejecución	3	CPI por HH por Recurso			
	24	CPI por HH trabajo Diario Externo			

Figura 5.22: Ajuste del modelo en las variables definidas

La métrica de Apego a la Metodología se calcula a continuación:

A	B	C	D	E	F	G	H	I	J	K	L		
Estructura	Plantilla	Propuesta Metodológica	Importancia en Metodología	Apego	Aplicada	Apego por Aplicada	Distribución Importancia	Refinada	Multipl. (Distribución * Refinado)	Apego por Fase	Porcentaje por Estructura		
0. Entradas	Informe de Diseño proveedor(Extracción).doc	CMMI	3	4	12	1	12	0,40	NO	0,9	0,36	0. Entradas	
	PlanificacionSeguimientoSemanal.mpp	CMMI	3	4	12	1	12	0,40	SI	1	0,40		
	Visión y Alcance.doc	CMMI	3	2	6	1	6	0,20	NO	0,9	0,18		0,94
1. Análisis	Matriz_de_Evaluacion_de_Propuestas.xls	CMMI	3	2	6	0	0	0,00	NO	0,9	0,00	1. Análisis	
	Estimación Modelo de Servicios.xls	Cruce	5	4	20	1	20	0,63	SI	1	0,63		
	LB_Requerimientos_Calidad.xls	ITIL	2	3	6	1	6	0,19	NO	0,9	0,17		0,79
2. Diseño	Definición CPI.xls	ITIL	2	4	8	1	8	0,31	NO	0,9	0,28	2. Diseño	
	TF Diseño.xls	ITIL	2	3	6	1	6	0,23	SI	1	0,23		
	Plan_de_CM.doc	CMMI	3	1	3	1	3	0,12	NO	0,9	0,10		
	Plan_de_Pruebas.doc	CMMI	3	3	9	1	9	0,35	NO	0,9	0,31		0,92
3. Ejecución	TF Ejecución.xls	Cruce	5	3	15	1	15	1,00	SI	1	1,00	1,00	3. Ejecución
4. Gestionar Defectos	TF Entidad-Defecto.xls	Cruce	5	3	15	1	15	1,00	SI	1	1,00	1,00	4. Gestionar Defectos
5. Informes	Plan de Negocio.xls	Cruce	5	4	20	1	20	0,50	SI	1	0,50	5. Informes	
	Cierre Proyecto.xlsx	Cruce	5	2	10	1	10	0,25	NO	0,9	0,23		
	RESULTADO TESTING Ciclo.doc	Cruce	5	2	10	1	10	0,25	NO	0,9	0,23		0,95

Figura 5.25: Resultados del apego a la metodología

Métrica de Porcentaje de Apego a la Metodología obtenido = 93%

6 Conclusiones

La Metodología Propuesta se basa fundamentalmente en la necesidad que impone el mercado en entregar servicios de calidad a bajo costo y en los tiempos comprometidos utilizando estándares de mejores prácticas y modelos de servicios que han existido desde hace ya varios años.

Al realizar el cruce de los procesos de CMMI versus las mejores prácticas de ITIL sobre uno de los procesos más importantes a la hora de validar la calidad de un software, se obtiene una Metodología eficiente y consistente de aplicar a cualquier tipo de proyecto de desarrollo. Además la expertise de trabajar en distintos tipos de proyectos en el proceso escogido (Validación y Verificación), apoyó en forma significativa al momento de crear las plantillas de apoyo que ayudan a concretizar la aplicación de la propuesta metodológica a un proyecto real.

El análisis de las métricas en este documento se realiza en función de los dos ciclos ejecutados en el proyecto de Testing, las métricas de Esfuerzo y Costos se realizan en base a los antecedentes entregados por el Jefe de Proyecto de Testing Funcional, en función de que el ciclo debe terminar en la fecha replanificada.

En el cálculo de métricas se puede ver que las variables cuantitativas son posibles de ajustar en función de la precisión y realismo con que se analicen; por ejemplo, las tasas de rendimiento de diseño y ejecución, así como también los costos relacionados al proyecto.

El cálculo de la métrica de Apego a la Metodología ayuda a determinar que en la aplicación de dicha métrica, se establece el grado de utilización efectiva en relación al objetivo de la Metodología. Sin embargo, se pueden recoger mejoras sustanciales en cuanto a la entrevista al jefe de proyecto sobre como fue el proceso de aplicación, ya que él indica que en sí la Metodología le sirvió para mantener el control y orden de los pasos a seguir, además de aumentar la percepción de calidad de servicio por parte del cliente, ya que se había trabajado en otros proyectos con este cliente, sin embargo esta vez las reglas para una mejor entrega del servicio fueron entregadas inicialmente, apoyando el proceso de desarrollo de software en lo que respecta a formalidad en los documentos de diseño y estimación de esfuerzo y planificación más acertada para el desarrollo, ya que consideraron la generación de los documentos necesarios en los momentos reales del ciclo de vida del proyecto y no al final como se estila realizar.

Se propone como trabajo futuro modelar y generar una Base de Conocimiento para registrar los tipos de proyectos y sus mejores ajustes planificados y reales con el fin de mejorar cada vez más el Modelo del Servicio.

También se planteó la iniciativa de llevar a programa la plantilla de Modelado de Servicio, con el fin de que pueda ser accesada desde internet al momento que se desea realizar la estimación de costo o esfuerzo involucrado en un servicio de Testing funcional.

7 Bibliografía

- 1) Axentia (2006), White Paper, Una Introducción a CMMI, Transforming Your IT Organization. (Ver ANEXO 2 entregado en el Informe de Avance), 2006.
- 2) Fernando Sánchez, (2008), CMMI vs. ITIL: dos enfoques complementarios, gerente de Grupo Delaware, <http://www.mkm-pi.com/mkmpi.php?article1817>, Jueves 6 de marzo de 2008.
- 3) Juan Palacio, (2006), Sinopsis de los Modelos SW-CMM y CMMI, <http://www.navegapolis.net/jpalacio@navegapolis.net>, 1.0 Abril – 2006.
- 4) Juan Raggio Pérez, (2004), Universidad Politécnica de Madrid, Facultad de Informática, Estudios de Doctorado, DESARROLLO DE PROCESOS DE GESTIÓN DE SERVICIOS DE EXPLOTACIÓN SIGUIENDO EL MODELO CMMI, 2004.
- 5) Lorena León – Gabriela Puglla, (2008), Métricas de Proceso y proyecto de software, Universidad Técnica particular de Loja. <http://www.slideshare.net/loreknelamorena/mtricas-de-proceso-y-proyecto-de-software>, 2008.
- 6) Mariana Isela Jaramillo González (2005), Utilización de Estándares ITIL para lograr el Nivel 3 de CMMI en una Organización. Universidad Autónoma del Estado de México, <http://docente.uco.mx/juancont/documentos/cap02/6.pdf>, 24 Octubre 2005.
- 7) OGC (2007), ITIL Managing IT Services. Best Practice for Application Management, http://www.ogc.gov.uk/guidance_itol.asp, 2007.
- 8) PMBOK (2004), [PMBOK](#) Tercera Versión en Español [Project Management Institute](#). «Capítulo 1», *Guía de los Fundamentos de la Dirección de Proyectos*, 3ª edición. ISBN 1-930699-73-5. http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos, 2004.
- 9) Raúl Suárez O. y Felipe Donoso Jaurés, (2006), METODOLOGÍA ITIL, Descripción, Funcionamiento y Aplicaciones, PÍA RAMÍREZ BRAVO - FELIPE DONOSO JAURÉS, Santiago, Julio de 2006.
- 10) Serge Thorn Geneva (2007), ITIL and CMMI synergies. <http://sergethorn.blogspot.com/2007/05/itil-and-cmmi-synergies.html>, 23 Mayo 2007.

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**PROPUESTA METODOLÓGICA PARA LA MEJORA DEL
PROCESO DE VERIFICACIÓN Y VALIDACIÓN EN LA
INDUSTRIA DE SERVICIOS TI EN BASE A MODELOS CMMI
CON ITIL**

NATALIA KARIN CARPIO ARNAIZ

TESIS DE GRADO
MAGÍSTER EN INGENIERÍA INFORMÁTICA

Diciembre 2009

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**PROPUESTA METODOLÓGICA PARA LA MEJORA DEL
PROCESO DE VERIFICACIÓN Y VALIDACIÓN EN LA
INDUSTRIA DE SERVICIOS TI EN BASE A MODELOS CMMI
CON ITIL**

NATALIA KARIN CARPIO ARNAIZ

Profesor Guía: Silvana Roncagliolo de la Horra

Programa: Magíster en Ingeniería Informática

Diciembre 2009

Resumen

Cada vez más, a los departamentos de TI se les exigen niveles de calidad y servicio directamente proporcionales a los niveles de calidad y servicio que se exigen a las distintas líneas de negocio de las organizaciones. Esto está provocando un afán por encontrar métodos y técnicas que permitan desarrollar y mantener software de alta calidad, así como implantar y mantener infraestructura con un alto grado de disponibilidad.

Se presenta una propuesta metodológica creada en base a los procesos de Verificación y Validación de CMMI pareados con los Procesos/Funciones de ITIL. En la metodología se exponen las prácticas específicas y se detalla el proceso de cómo implantar la metodología que apoya y mejora los niveles de calidad en el testing funcional de aplicaciones.

A partir de la metodología propuesta se presenta la evaluación de su aplicación a un caso real, definiendo como punto de partida las métricas que se consideran en dicha evaluación, generando mejoras a la propuesta en sí.

Abstract

Everyday, TI Departments are demanded with quality and service levels directly proportional to the quality and service levels demanded to the different business lines of the organizations. This is causing an urge to find methods and techniques that allow to develop and maintain high quality software, such as to implant and keep the infrastructure with a high level of availability.

A methodological proposal is shown, created based on the CMMI verification and validation processes matched against the ITIL processes/functions. In the methodology specific practices are exposed and the quality levels are improved in the functional testing of the applications.

From the proposed methodology, the evaluation of its application to a real case is showed, defining as a start point the metrics considered in that evaluation, generating improvements to the proposal itself.

Índice

1	Introducción.....	9
2	Objetivos	10
3	Estado del Arte	11
3.1	ITIL v2.0.....	11
3.1.1	<i>Descripción (PRamirez-FDonoso, 2006)</i>	<i>11</i>
3.1.2	<i>Objetivos de ITIL (PRamirez-FDonoso, 2006).....</i>	<i>12</i>
3.2	CMMI.....	13
3.2.1	<i>Historia y evolución (JPalacio, 2006)</i>	<i>13</i>
3.2.2	<i>Los Cinco Niveles de Madurez (Axentia, 2006).....</i>	<i>13</i>
3.2.3	<i>Arquitectura del Modelo (Axentia, 2006)</i>	<i>15</i>
3.3	CMMI e ITIL (JRaggio, 2004)	17
3.4	Análisis del Estado del Arte.....	18
4	Solución Propuesta.....	21
4.1	Conocimiento de los modelos.....	21
4.2	Proceso de cruce de los modelos	21
4.3	Definición de métricas para la evaluación final	25
4.3.1	<i>Tiempo</i>	<i>26</i>
4.3.2	<i>Costo.....</i>	<i>26</i>
4.3.3	<i>Alcance</i>	<i>26</i>
5	Implantación de la metodología propuesta y aplicación de plantillas de apoyo	32
5.1	Proceso para aplicar la metodología	32
5.1.1	<i>Proceso “Definir proyecto”</i>	<i>34</i>
5.1.2	<i>Proceso “Administrar externo”.....</i>	<i>37</i>
5.1.3	<i>Proceso “Administrar proceso de pruebas” PARTE 1.....</i>	<i>40</i>
5.1.3.1	<i>Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”.....</i>	<i>45</i>
5.1.3.2	<i>Proceso “Estimar grado automatización y nivel diagnóstico”.....</i>	<i>46</i>
5.1.3.3	<i>Proceso “Definir requerimientos no funcionales”</i>	<i>51</i>
5.1.4	<i>Proceso “Administrar proceso de pruebas” PARTE 2.....</i>	<i>53</i>
5.1.4.1	<i>Proceso “Definir prueba”</i>	<i>57</i>
5.1.4.2	<i>Proceso “Automatizar prueba unitaria”.....</i>	<i>61</i>
5.1.4.3	<i>Proceso “Automatizar prueba integrada”</i>	<i>63</i>
5.1.4.4	<i>Proceso “Generar LB datos”.....</i>	<i>64</i>
5.1.4.5	<i>Proceso “Ejecutar ciclo pruebas”</i>	<i>66</i>
5.1.4.6	<i>Proceso “Verificar ambiente prueba”</i>	<i>68</i>
5.1.4.7	<i>Proceso “Ejecutar conjunto de pruebas”</i>	<i>69</i>
5.1.5	<i>Proceso “Administrar proceso de pruebas” PARTE 3.....</i>	<i>70</i>

5.1.6	Proceso: “Gestionar defectos”.....	71
5.1.7	Proceso: “Realizar auditoría CM/PPQA”	74
5.1.8	Proceso: “Controlar proyecto”.....	79
5.2	Resumen de resultados obtenido (Métricas).....	81
6	Conclusiones	89
7	Bibliografía	90

Tabla de Ilustraciones

Figura 3.01: Modelo de Arquitectura CMMI.....	15
Figura 3.02: Niveles de Madurez CMMI.....	16
Figura 3.03: Representación escalonada. Por Nivele de Madurez CMMI.....	17
Figura 4.01: Tabla de mapeo CMMI e ITIL.....	22
Figura 4.02: Metodología propuesta.....	24
Figura 4.03: Triángulo de restricciones.....	26
Figura 4.04: Métricas definidas para validación de la implantación de metodología propuesta.....	28
Figura 4.05: Métrica “Importancia Metodológica”.....	28
Figura 4.06: Métrica “Propuesta Metodológica”.....	29
Figura 4.07: Tabla “Cálculo de métricas”.....	29
Figura 4.08: Tabla “Apego de la metodología propuesta”.....	30
Figura 5.01: Identificación planillas de apoyo.....	32
Figura 5.02: Proceso “Definir proyecto”.....	34
Figura 5.03: Proceso “Administrar externo”.....	37
Figura 5.04: Proceso “Administrar proceso de pruebas” PARTE 1.....	40
Figura 5.05: Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”.....	45
Figura 5.06: Proceso “Estimar grado automatización y nivel diagnóstico”.....	46
Figura 5.07: Proceso “Habilitar PC para captura”.....	50
Figura 5.08: Proceso “Definir requerimientos no funcionales”.....	51
Figura 5.09: Proceso “Administrar proceso de pruebas” PARTE 2.....	53
Figura 5.10: Proceso “Definir prueba”.....	57
Figura 5.11: Proceso “Automatizar prueba unitaria”.....	61
Figura 5.12: Proceso “Automatizar prueba integrada”.....	63
Figura 5.13: Proceso “Generar LB datos”.....	64
Figura 5.14: Proceso “Ejecutar ciclo pruebas”.....	66
Figura 5.15: Proceso “Verificar ambiente prueba”.....	68
Figura 5.16: Proceso “Ejecutar conjunto de pruebas”.....	69
Figura 5.17: Proceso “Administrar proceso de pruebas” PARTE 3.....	70
Figura 5.18: Proceso “Gestionar defectos”.....	71
Figura 5.19: Proceso “Realizar auditoría CM/PPQA”.....	74
Figura 5.20: “Ciclo de defectos”.....	78
Figura 5.21: Proceso “Controlar proyecto”.....	79
Figura 5.22: Ajuste del modelo en las variables definidas.....	86
Figura 5.23: Costos por ajuste.....	87
Figura 5.24: Tabla comparativa de las métricas con y sin ajustes.....	87
Figura 5.25: Resultados del apego a la metodología.....	88

Glosario

CPU: Caso de Prueba Unitario, corresponde a la descripción de una prueba unitaria que posee un resultado y puede ser constituyente de un Caso de Prueba Integrado (CPI).

CPI: Caso de Prueba Integrado, corresponde a la descripción de aquella prueba cuyo objetivo es validar una funcionalidad o una operación del aplicativo. Está constituida por más de un CPU.

Ciclo: Es una ventana de tiempo definida en la planificación que corresponde a un grupo de iteraciones de pruebas con sus respectivos datos para cada versión liberada.

Iteración: Es una instancia de pruebas para una o un grupo de funcionalidades, interfaces y/o reportes que podría repetirse dentro de un ciclo, esto dependiendo del grado de aprobación, en caso de cumplir con todos los criterios de aceptación para la primera iteración, no es necesario seguir con las siguientes.

Pruebas con Error: Corresponde a aquellas pruebas que luego de ser ejecutadas presentan algún tipo de falla que implica que el estado de ejecución de éstas no sea óptimo.

Pruebas de Aceptación de Usuario: Son aquellas diseñadas y ejecutadas por el usuario para demostrar que la aplicación satisface los requerimientos. En inglés: User Acceptance Testing.

Pruebas de Nuevas Funcionalidades: Corresponde a aquellas pruebas que apuntan a validar aquella operatoria del sistema que no ha sido certificada en ninguna etapa de pruebas anterior.

Pruebas de Regresión: Es un "Conjunto de Pruebas" realizado para identificar "defectos" que causan "Regresión". Si el desarrollador resuelve un defecto, se deben realizar pruebas en los alrededores para verificar que no se haya impactado el resto del código, a esto se le llama regresión. En inglés: Regression Test.

Pruebas Trazadoras: Es un "Conjunto de Pruebas" corridas sobre una cierta "Release" para asegurar que es suficientemente estable para continuar con el "Ciclo" de pruebas activo. Estas pruebas usualmente son un subconjunto del conjunto completo de pruebas, preferiblemente de tipo automatizado, que involucran cada parte del sistema al menos en una forma superficial. Unas buenas pruebas trazadoras también mantienen el sistema corriendo por un lapso de tiempo suficiente como para que se manifiesten problemas "gruesos" de confiabilidad y disponibilidad. En inglés: Smoke Test o Sanity Test.

Requerimientos de Negocio: Corresponde a aquel requerimiento que es generado por el cliente y que representa alguna funcionalidad del sistema a validar. En este caso, el responsable de dicho requerimiento es el equipo de Gestión de Proyecto.

Requerimientos de Prueba Funcional: Corresponde a aquel requerimiento generado a partir de un Requerimiento de Negocio, representa el alcance de las pruebas que se realizarán para abordar la validación del requerimiento de negocio respectivo. Este tipo de requerimiento es de responsabilidad del equipo de Calidad.

1 Introducción

Cada vez más, a los departamentos de TI se les exige niveles de calidad y servicio directamente proporcionales a los niveles de calidad y servicio que se exige a las distintas líneas de negocio de las organizaciones.

Esto está provocando un afán por encontrar métodos y técnicas que permitan desarrollar y mantener software de alta calidad, así como implantar y mantener infraestructuras con un alto grado de disponibilidad.

En los últimos años han aparecido dos líneas metodológicas que están consiguiendo un gran protagonismo por su carácter práctico y por los buenos resultados obtenidos: CMMI e ITIL. La primera se centra en los procesos asociados al desarrollo de software, mientras que la segunda enfatiza en el enfoque a servicios (FSanchez, 2008).

Según revisión en la Web, no existe actualmente una Propuesta tan específica como ésta, sólo se presentan cruces a nivel completo de Metodología CMMI y Mejores Prácticas de ITIL, todas en un marco genérico.

Esta Propuesta apunta específicamente al cruce y presenta una Metodología en la implantación del proceso de Verificación y Validación de CMMI basándose en la Mejores Prácticas de ITIL.

El trabajo presenta una guía clara de cómo enfrentar un proyecto de Testing tanto desde su estimación pasando por el diseño, ejecución, concluyendo por el análisis de los resultados, cierre y revisión del cierre de proyecto en lo que respecta a su Plan de negocio inicial.

Además de presentar la Metodología, se presentan plantillas de apoyo para dicho proceso y se aplica a un proyecto real con el fin de poder refinar la Propuesta y evaluar los resultados en base al apego obtenido al aplicar dicha Metodología, resultando de este trabajo una Propuesta validada y refinada desde su concepción.

2 Objetivos

El objetivo propuesto en este proyecto es:

- Generar una propuesta Metodológica de apoyo para la implantación de Testing en la Industria de TI en base al proceso de CMMI Verificación y Validación con las Mejores Prácticas de ITIL, comprobando su aplicabilidad con un caso real.

Lo anterior se realizará a través de la persecución de los siguientes objetivos específicos:

- Proponer una Metodología de apoyo para la implantación de Testing en la Industria de TI en base al proceso de CMMI Verificación y Validación, cruzándolo con las Mejores Prácticas de ITIL.
- Generar los documentos de apoyo (plantillas) para la aplicación de la Metodología Propuesta, especificando su utilización en cada uno de los procesos que se definen en la Metodología que apoya al proceso de Verificación y Validación de CMMI.
- Generar métricas para evaluar la correcta implementación de la Metodología Propuesta.
- Aplicar la Metodología a un caso real (Proyecto Gestor de Oportunidades), desde su fase de evaluación hasta la ejecución del Testing, para así evaluar la efectividad en la aplicación de la Metodología en base a la definición de métricas para su evaluación.
- Evaluar la aplicación de la Metodología en base a las métricas propuestas.

3 Estado del Arte

Al iniciar el estado del arte es importante presentar los conceptos, así como los orígenes de la metodología y las mejores prácticas, lo que se encuentra en las siguientes secciones.

3.1 ITIL v2.0

3.1.1 Descripción (PRamirez-FDonoso, 2006)

Information Technology Infrastructure Library (ITIL), es una metodología que se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones en sus Sistemas de Información y Tecnologías de Información. Esta metodología fue desarrollada a petición del Gobierno del Reino Unido a finales de los 80 y recoge las mejores prácticas en la gestión de los Sistemas de Información. Desde entonces se ha ido extendiendo su uso en la empresa privada, tanto multinacional como PYME, llegando a ser considerado un estándar de facto para la gestión de esta área de la empresa.

En un entorno donde los periodos de disponibilidad de los servicios son cada vez más amplios, donde las exigencias del cliente son cada vez más elevadas, donde los cambios en los negocios son cada vez más rápidos, es muy importante que los Sistemas de Información estén adecuadamente organizados y alineados con la estrategia del negocio. Igualmente ITIL, ofrece toda una serie de definiciones de conceptos típicos de los Sistemas de Información para garantizar que todos sus conocedores hablen de lo mismo, reduciendo así los tiempos y riesgos por malas interpretaciones.

ITIL, es un set de documentos donde se describen los procesos requeridos para la gestión eficiente y efectiva de los Servicios de Tecnologías de Información dentro de una organización. Son un conjunto de mejores prácticas y estándares en procesos para hacer más eficiente el diseño y administración de las infraestructuras de datos dentro de la organización. Es un “marco de trabajo” (framework) para la Administración de Procesos de TI.

Basándose en el principio de mejora continua, ITIL fue madurado y de los 40 libros originales se redefinieron; se agruparon y eliminaron redundancias y actualmente se presenta en 7 libros:

- 1) Service Support
- 2) Service Delivery
- 3) Security Management
- 4) Application Management
- 5) The Infrastructure Management
- 6) Planning to Implement Service Management
- 7) The Business Perspective

Esta metodología se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones. Garantizando así los niveles de servicio establecidos entre la organización y sus clientes.

El objetivo de ITIL es diseminar las mejores prácticas en la Gestión de Servicios de Tecnologías de Información. Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios IT, al mismo tiempo de garantizar los requerimientos de la información en cuanto a seguridad, mantienen e incrementan sus niveles de fiabilidad, consistencia y calidad.

ITIL brinda una descripción detallada de un número de prácticas importantes en TI, a través de una amplia lista de verificación, tareas, procedimientos y responsabilidades que pueden adaptarse a cualquier organización. ITIL describe una aproximación sistemática y profesional a la Gestión de Servicios TI, haciendo énfasis en la importancia clave de cumplir con los requerimientos del negocio respetando los costos acordados.

3.1.2 Objetivos de ITIL (PRamirez-FDonoso, 2006)

El objetivo que persigue ITIL es diseminar las mejores prácticas en la gestión de servicios de Tecnologías de Información de forma sistemática y coherente. El planteo principal se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos.

Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios de TI, al mismo tiempo que se garantizan los requerimientos de la información en cuanto a seguridad manteniendo e incrementando sus niveles de fiabilidad, consistencia y calidad.

La filosofía ITIL adopta la gestión de procesos y considera que, para lograr los objetivos claves de la Administración de Servicios estos procesos deberían ser usados por las personas y las herramientas efectiva, eficiente y económicamente en el desarrollo de la alta calidad y la innovación de los servicios de TI alineados con los procesos de negocio.

Los estándares ITIL exigen un replanteamiento del área tecnológica y la definición de los elementos y procesos "críticos" dentro de la empresa.

3.2 CMMI

3.2.1 Historia y evolución (JPalacio, 2006)

1984 El Congreso del Gobierno Americano aprobó la creación de un organismo de investigación para el desarrollo de modelos de mejora para los problemas en el desarrollo de los sistemas de software, y evaluar la capacidad de respuesta y fiabilidad de las compañías que suministran software al Departamento de Defensa.

Creación del SEI (Instituto de Ingeniería del Software), fundado por el Departamento de Defensa Americano y la Universidad Carnegie Mellon.

1985 SEI empieza a trabajar en un marco de madurez de procesos que permita evaluar a las empresas productoras de software.

La investigación evoluciona hacia el “Modelo de Madurez de las Capacidades (CMM)”.

1991 En agosto SEI publica la versión 1.0 del Modelo de Madurez de las Capacidades para el Software (SWCMM, Capability Maturity Model for Software).

1993 SEI publica la versión 1.1 de SW-CMM.

1997 Publicación de la versión 1.2.

2000 SW-CMM fue integrado y relevado por el nuevo modelo CMMI.

2006 Se publica CMMI –DEV, V1.2, CMMI for Development, Version 1.2.

2007 Se publica CMMI-ACQ, V1.2, CMMI for Acquisition, Version 1.2, *Improving processes for acquiring better products and services*.

La propuesta metodológica que se expone a continuación, se realiza sobre CMMI-ACQ, ya que las empresas de desarrollo pueden contar con externalización de servicios a la hora de desarrollos de software.

3.2.2 Los Cinco Niveles de Madurez (Axentia, 2006)

En general, los niveles de madurez suelen explicarse en orden creciente; se tomará aquí una dirección distinta y se explicarán exactamente al revés: desde el nivel cinco al nivel uno.

Imagine por un momento que está en una organización de nivel cinco. En este tipo de organizaciones, los procesos son analizados para eliminar las causas comunes de variación, o sea, aquellas que tienen que ver con la misma naturaleza del proceso, no atribuibles a causas externas. Las variaciones en las salidas de los procesos son al azar, pero se encuentran controladas estadísticamente (se puede predecir los resultados de los procesos con cierto nivel de confiabilidad).

Para poder llegar a este nivel, la organización debió primero haber eliminado las causas especiales de variación, aquellas que tienen que ver con causas externas, como por ejemplo falta de entrenamiento del personal, problemas con las herramientas, etc. Este tipo de causas no son aleatorias: Si se examinan los resultados se podrán ver las tendencias que claramente indican que las variaciones tienen un origen concreto. En una organización de nivel cuatro, entonces, las causas especiales de variación son identificadas y eliminadas.

Para poder llegar a identificar causas de variación se necesita tener un proceso estándar: difícil sería poner bajo control estadístico un proceso que no se encuentre mínimamente formalizado.

Así se llega al nivel tres, en el cual los proyectos emplean un proceso productivo adaptado del proceso estándar de la organización. Las actividades técnicas y de gestión son realizadas de acuerdo a políticas, procesos y procedimientos formalizados en algún tipo de estándar organizacional profundamente arraigado en la cultura. La gente está entrenada y dispone de recursos para poder hacer su trabajo. También hay una infraestructura básica (personal, herramientas, etc.) para definir y mejorar el proceso productivo.

Pero para poder llegar a esta situación es necesario pasar por una etapa previa: difícilmente se puede introducir en una organización prácticas estándar relacionadas con la ingeniería del producto (análisis, diseño, etc.) si no se ofrece un contexto en donde ellas puedan ser correctamente ejecutadas. Ese es el foco del nivel dos: poner en orden las prácticas relacionadas con el manejo elemental de los proyectos.

En el nivel dos, los proyectos de la organización siguen algún tipo de proceso para realizar las actividades relacionadas con la gestión del proyecto (planificación, control), para administrar los requerimientos y las configuraciones, y para medir y analizar la calidad de los productos y el desempeño de los procesos. También hay prácticas de aseguramiento de la calidad que permiten garantizar que cada proyecto sigue sus propios estándares.

Y así se llega al nivel uno: La situación aquí es caótica. No existen procesos (no al menos en el sentido del modelo) y la *performance* de los proyectos depende profundamente de la buena voluntad y la capacidad de la gente.

3.2.3 Arquitectura del Modelo (Axentia, 2006)

En la representación por niveles (ver Figura 3.01), cada nivel de madurez contiene varias áreas de proceso, las que a su vez quedan definidas por uno o varios objetivos específicos y un objetivo genérico. Cada uno de ellos tiene vinculado un conjunto de prácticas, llamadas específicas y genéricas respectivamente.

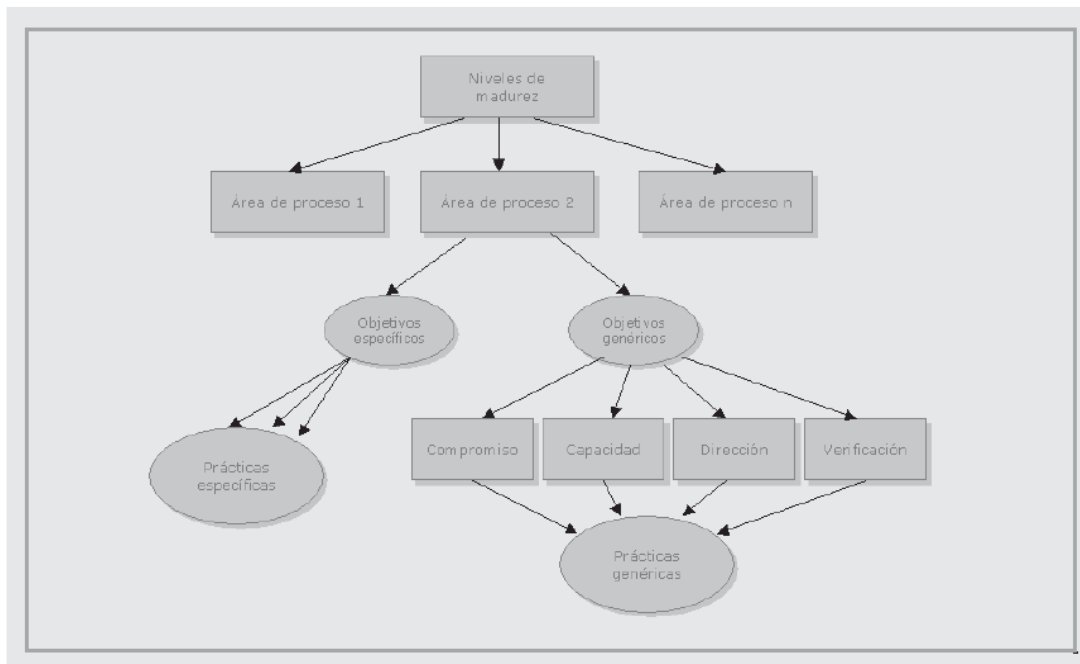


Figura 3.01: Modelo de Arquitectura CMMI

CMMI ofrece una representación sistemática y estructurada para abordar el modelo basado en un proceso de mejora que permite avanzar una etapa a la vez. El logro de cada etapa asegura que el proceso se ha establecido, y cuenta con la infraestructura adecuada que permite avanzar a la siguiente etapa.

Las áreas de proceso del modelo están organizadas en 5 niveles de madurez que representan, cada una, alguna conjetura respecto del proceso de mejora, lo que se muestra en la Figura 3.02.

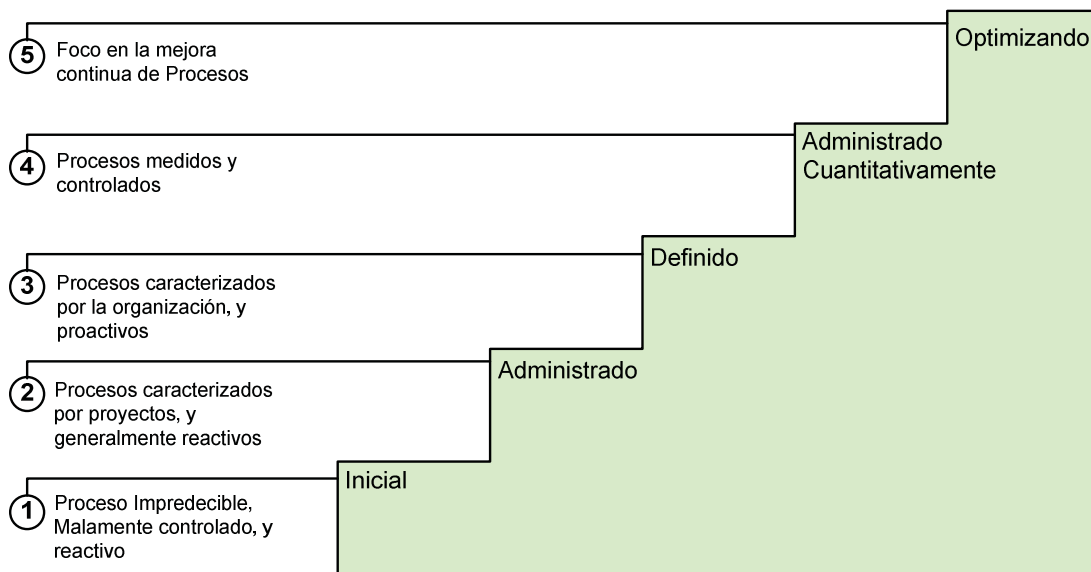


Figura 3.02: Niveles de Madurez CMMI

Esta representación sugiere un camino para la mejora de acuerdo al nivel de madurez de la organización, desde el nivel “Inicial” al nivel “Optimizando”.

El logro de cada nivel de madurez garantiza que un nuevo estado de mejora ha sentado las bases para el siguiente nivel, permitiendo así una mejora incremental.

La representación por niveles de madurez es una buena opción para mejorar un conjunto específico de procesos en cada etapa. Este orden se ha determinado a través de más de una década de investigación y experiencia.

Las áreas de proceso que contiene cada nivel de madurez se muestran en la Figura 3.03.

Nivel	Foco	Áreas de Procesos	Calidad
5 Optimizado	Proceso de Mejoramiento Continuo	Innovación organizacional y Desarrollo Análisis Causal y Resolución	Productividad
4 Administrado Cuantitativamente	Administración Cuantitativa	Ejecución proceso organizacional Administración cuantitativa del proyecto	
3 Definido	Estandarización del Proceso	Desarrollo de Requerimientos Solución Técnica Integración del producto Verificación Validación Foco en proceso organizacional Definición del proceso organizacional + IPPD Entrenamiento organizacional Administración del proyecto integrado + IPPD Administración de Riesgo Análisis de Decisión y Resolución	
2 Administrado	Administración de Proyecto Básico	Administración de Requerimientos Planificación del Proyecto Monitoreo y Control del Proyecto Administración de Acuerdo con Proveedor Medición y Análisis Aseguramiento de la Calidad de Proceso y Producto Administración de la Configuración	Riesgo
1 Inicial			Reelaboración

Figura 3.03: Representación escalonada. Por Nivele de Madurez CMMI

3.3 CMMI e ITIL (JRaggio, 2004)

Como se ha comentado anteriormente CMMI es una colección de buenas prácticas alineadas con el concepto de modelos de madurez. Debido a la herencia de SW/CMM y tal y como se ha visto en la descripción de las áreas de procesos, objetivos y las prácticas, está muy orientado hacia desarrollo y mantenimiento de software.

Si bien CMMI es la evolución e integración de un conjunto de modelos de madurez, el desarrollo y mantenimiento de software, fijan en gran medida sus líneas de actuación. En el otro punto se encuentra ITIL, modelo totalmente orientado a los servicios de mantenimiento y gestión de las operaciones.

ITIL y CMMI son dos modelos de madurez claramente distintos pero no mutuamente excluyentes. Las principales diferencias se deben a que CMMI está focalizado en la madurez de los procesos software a través de una actividad de mejora continua e ITIL se centra en el entendimiento y desarrollo de todas las áreas referentes a la infraestructura de TI, además de incorporar el ciclo de vida del hardware dentro de una organización.

3.4 Análisis del Estado del Arte

El término “nivel de calidad” es aplicable directamente a software o aplicaciones que se desarrollan, pero este término es demasiado subjetivo ya que depende de la definición y/u objetivos de cada empresa que desea implantar este “nivel de calidad”. Actualmente las empresas de TI han involucrado el concepto pero en base a criterios y/o definiciones que nacen por experiencias propias.

Al hacer una revisión en la Web, se pueden encontrar diversos estudios y propuestas de solución en lo que respecta a mejoras al “nivel de calidad” de los productos, pero la mayoría de ellos se enfoca a soluciones generales que condicionan el testing a mejoras en los procesos iniciales de construcción del producto.

La gran mayoría de los análisis y estudios existentes, se basa fundamentalmente en resaltar la sinergia que existe entre CMMI e ITIL, proponiendo distintos puntos de vista a nivel de cómo fusionar o complementar éstos en función de aprovechar ambos en la aplicación de la industria para conseguir mejores “niveles de servicio”.

En general ITIL se puede combinar con CMMI para cubrir todo, pero no aborda el desarrollo de los sistemas de gestión de la calidad. Además, no está orientado a los procesos de desarrollo de software y su uso depende en gran medida de la interpretación. Mientras CMMI es el estándar de calidad de fábrica para los procesos de desarrollo de software, ITIL para muchos es la herramienta de elección para las operaciones y la parte de infraestructura de TI, especialmente para servicios de TI (SThorn 2007).

ITIL y CMMI se aplican de mejor manera a diferentes partes de la organización de TI:

- El uso de CMMI en el desarrollo de aplicaciones
- El uso de CMMI en las TIC. Específicamente en los proyectos de infraestructura.
- Uso de ITIL en las operaciones de TI y servicios

Dentro de las propuestas existentes con un poco más de detalle a nivel de cruces entre la metodología y las mejores prácticas, se encuentra un modelo en el que las mejores prácticas de ITIL ayudan a lograr las metas requeridas para obtener un nivel particular de CMMI, por ejemplo nivel 3 (MJaramillo, 2005).

El modelo apunta a generalizar, a través de los niveles de ITIL como apoyar a CMMI para llegar al nivel 3.

1) Nivel de Servicio de Soporte:

Se lleva a cabo la administración de cada proceso planeado en la primera etapa de CMMI permitiendo así la interacción con las partes involucradas y/o afectadas, alcanzando una eficiente comunicación de usuario y proveedor,

los cuales adquieren un compromiso con el plan de trabajo ya establecido y por último llevan a cabo un mantenimiento continuo del plan.

Este nivel de ITIL también ayuda al Monitoreo y Control del Proyecto, es decir, logra el entendimiento de los procesos para tomar decisiones correctivas adecuadas.

2) Nivel de Entrega de Servicios:

Describe los procesos necesarios de entrega con respecto a la calidad y costo lo cual permite llevar a cabo una eficiente Administración de Proveedores (Etapa de CMMI), la cual analiza todo tipo de adquisición, los contratos, etc.

Por lo que participa en la Administración del Proyecto para su Desarrollo Integral (Etapa de CMMI), donde se considera el grado de involucramiento de quienes administran el proyecto.

3) Nivel de Administración de Seguridad:

Mediante este nivel se logra proteger la tecnología de información de la compañía motivo que contribuye a mejorar la “Administración de Riesgos”, que lleva a cabo CMMI, etapa que persigue evitar los peligros que impidan lograr los objetivos del proyecto.

4) Nivel de Perspectiva del Negocio:

Es indispensable para CMMI ya que mediante este nivel el gerente debe conocer el impacto de la tecnología de información y las medidas que deberá tomar para protegerse de situaciones adversas logrando así la Integración del Equipo de Trabajo, etapa de CMMI que pretende que todas las partes involucradas compartan sus habilidades y experiencias a fin de enriquecer el trabajo de la organización trayendo como resultado la calidad de cada proceso que desarrolla.

5) Nivel de Administración de la Infraestructura de TIC:

Mediante este nivel se logra tener un control de calidad con respecto a todos los servicios de comunicación que proporciona la organización facilitando así la Administración Integral de Proveedores que lleva a cabo CMMI evaluando de esta forma sus procesos y productos y haciendo los ajustes en la relación y en sus contratos evitando así comprar productos o servicios obsoletos para la organización.

6) Nivel de Administración de las Aplicaciones:

Determina las soluciones para cubrir las necesidades de los usuarios de acuerdo a los recursos y presupuesto con los que se cuenta logrando así brindar información importante a la Administración Cuantitativa del Proyecto (Etapa de CMMI) la cual considera el tiempo, mano de obra necesaria, y los costos de los procesos definidos por la organización cumpliendo de esta forma los objetivos establecidos en calidad y eficiencia.

7) Nivel de Planeación para la Administración de Servicios:

Ayuda a las organizaciones a identificar sus fortalezas y debilidades generando así que CMMI logre definir aún mejor sus procesos para un progreso en su madurez.

A partir de los análisis antes expuestos, se desprende que en su mayoría, todo apunta a refundir la metodología y sus mejores prácticas a nivel general, sin embargo es necesario contar con propuestas específicas sobre los procesos de CMMI versus ITIL, con el fin de entregar propuestas metodológicas que apoyen el desarrollo de un proceso específico sin ánimo de estandarizar el desarrollo del proceso, sino más bien orientar en cómo y cuándo aplicar las mejores prácticas en un proceso determinado en base a experiencias vividas.

Por lo anterior se desea proponer una Metodología de apoyo en la implantación del proceso de Testing de software y/o aplicativos en base a los Modelos antes mencionados.

Se analizaron los puntos de semejanza entre dichos Modelos, sin olvidar la diferencia entre sus enfoques, para así identificar los procesos de apoyo específicos al Testing. Se propone una Metodología clara y precisa para el levantamiento y mejora de los procesos existentes, con el fin de solucionar la subjetividad que existe en la implantación del proceso de pruebas, así como también en los procesos de apoyo para entrega de los servicios relacionados al Testing.

La Metodología Propuesta, puede ser aplicada tanto en departamentos de TI que pertenezcan a empresas pequeñas, que no pueden optar a obtener una certificación de los Modelos base, o bien en departamentos de TI que a pesar de contar con presupuesto para optar por estas certificaciones no cuentan con la experiencia en la implantación de servicios de Testing.

El cruce se realizó en base a las prácticas concretas y se utilizó como inicio el análisis de ITIL (PRamirez-FDonoso, 2006), ya que el enfoque a servicios debe ser la base de la Metodología que se desea generar. Sin embargo como el enfoque a nivel de procesos apunta a Verificación y Validación (Testing), se consideró sólo el detalle de los procesos que tienen paridad; es decir, Problem Management (Administración de Problemas) y Application Management (Administración de Aplicaciones).

El proyecto presenta la Metodología creada en base a los procesos de Verificación y Validación de CMMI pareado con los procesos/funciones de ITIL.

Los procesos más importantes que se incluyen en el flujo fueron descritos. El Modelo propuesto tiene inmerso los procesos de apoyo tanto de CMMI como de ITIL (JPalacio, 2006) (Axentia, 2006) (JRaggio, 2004) (OGC, 2007).

4 Solución Propuesta

4.1 Conocimiento de los modelos

Para comenzar se despliega el alcance de la solución propuesta en base a los antecedentes entregados en el Estado del Arte, con el propósito de establecer los puntos en los cuales estos Modelos son semejantes y así poder orientar el análisis y generación de la Metodología de trabajo para el proceso de Testing de Software aplicativos.

Una vez que existe claridad en la definición de los Modelos se comienza con el pareo.

4.2 Proceso de cruce de los modelos

A partir de los 7 libros existentes en ITIL se analizarán:

- Service Support
- Service Delivery
- Application Management

Se realiza un análisis de cada uno de ellos con el fin de encontrar semejanzas a nivel de prácticas concretas con CMMI. La selección de estos libros es en base a conocimiento experto por la certificación obtenida en ITIL. Como la orientación es sobre implantar Testing, se sabe de antemano que estos libros son los que apoyan de manera más directa dicha implantación.

Por su parte el Modelo CMMI cuenta con 22 procesos que incluyen en forma estricta las mejores prácticas a aplicar a nivel de desarrollo de software.

Analizando entonces a nivel de prácticas concretas se pueden mapear algunos procesos de CMMI a ITIL en lo que respecta a soporte y gestión de proyectos.

A continuación se presenta un mapeo de los procesos/funciones de los Modelos CMMI e ITIL. Estos Modelos están enfocados a diferentes áreas, por lo cual guardan escasa relación, pero existen relaciones entre ambos Modelos a nivel de prácticas concretas, aunque pueden generalizarse algunos procesos de soporte y gestión de proyectos.

El Nivel al cual pertenece el Proceso de CMMI en el Modelo está descrito en la tabla como “N+el número del nivel”. La Figura 4.01 muestra los procesos de CMMI pareados a los Procesos/Funciones de ITIL.

Procesos/Funciones ITIL	Áreas de Procesos CMMI
Configuration Management	Configuration Management (N2)
Change Management	Configuration Management (N2)
Release Management	Configuration Management (N2)
Incident Management	
Problem Management	Verification (N3), Causal Analysis & Resolution (N5)
Capacity Management, Availability Management, Continuity Management	Risk Management (N3)
Application Management	Requirements Management (N2), Requirements Development (N3), Technical Solution (N3), Product Integration (N3), Verification (N3), Validation (N3), Integrated Project Management (N3)
Project Management	Project Planning (N2), Project Monitoring and Control (N2), Supplier Agreement Management (N2)

Figura 4.01: Tabla de mapeo CMMI e ITIL

Como se muestra en la Figura 4.01, la base del análisis es ITIL, ya que el enfoque a servicios debe ser la base de la Metodología que se desea generar. Sin embargo como el enfoque a nivel de procesos apunta a Verificación y Validación (Testing), se considerarán sólo el detalle de los procesos que tienen paridad en la tabla anterior; es decir, Problem Management (Administración de Problemas) y Application Management (Administración de Aplicaciones).

No se debe perder de vista que los procesos de Verificación y Validación están soportados en el nivel 3 de CMMI, por lo cual es necesario contar con el apoyo de algunos procesos del nivel anterior (nivel 2) para poder satisfacer en forma correcta las necesidades de la implantación de los procesos de Verificación y Validación del nivel 3.

Los procesos de apoyo para Servicios (ITIL) serán los siguientes:

- Configuration Management (Administración de la Configuración)
- Change Management (Administración del Cambio)
- Release Management (Administración de Release)
- Incident Management (Administración de Incidentes)
- Project Management (Administración de Proyectos)

Los procesos de apoyo para Desarrollo de software (CMMI) serán los siguientes:

- Configuration Management (Administración de la Configuración) → N2
- Project Planning (Planificación de Proyecto) → N2
- Project Monitoring and Control (Monitoreo y Control de Proyecto) → N2
- Supplier Agreement Management (Administración de Proveedores) → N2
- Requirements Management (Administración de Requerimientos) → N2
- Requirement Development (Requerimientos de Desarrollo) → N3
- Technical Solution (Solución Técnica) → N3
- Product Integration (Integración de Producto) → N3
- Integrated Project Management (Administración de Integración de Proyecto) → N3
- Risk Management (Administración de Riesgos) → N3

Es importante mencionar que cuando se habla de procesos de apoyo se entiende que al momento de detallar la Metodología a generar, se nombraran las mejores prácticas y los procesos listados, con el fin de que se tenga claridad de cómo ellos interactúan en el proceso de implantación; sin embargo no se entregará detalle de cómo se deben implantar, a diferencia de los procesos específicos de Verificación y Validación.

La Metodología obtenida es presentada en la Implementación de la misma.

La Figura 4.02 presenta la estructura definida para el Modelo Propuesto y se incluyen los proceso tanto del cruce como los de apoyo de CMMI e ITIL. Cabe destacar que el Modelo Propuesto fue generado en Enterprise Architect - UML CASE Tool - Desktop, Professional and Corporate editions, Version 7.0.

La simbología de los colores utilizados en el Modelo se encuentra en la Figura 4.02 en la parte superior derecha.

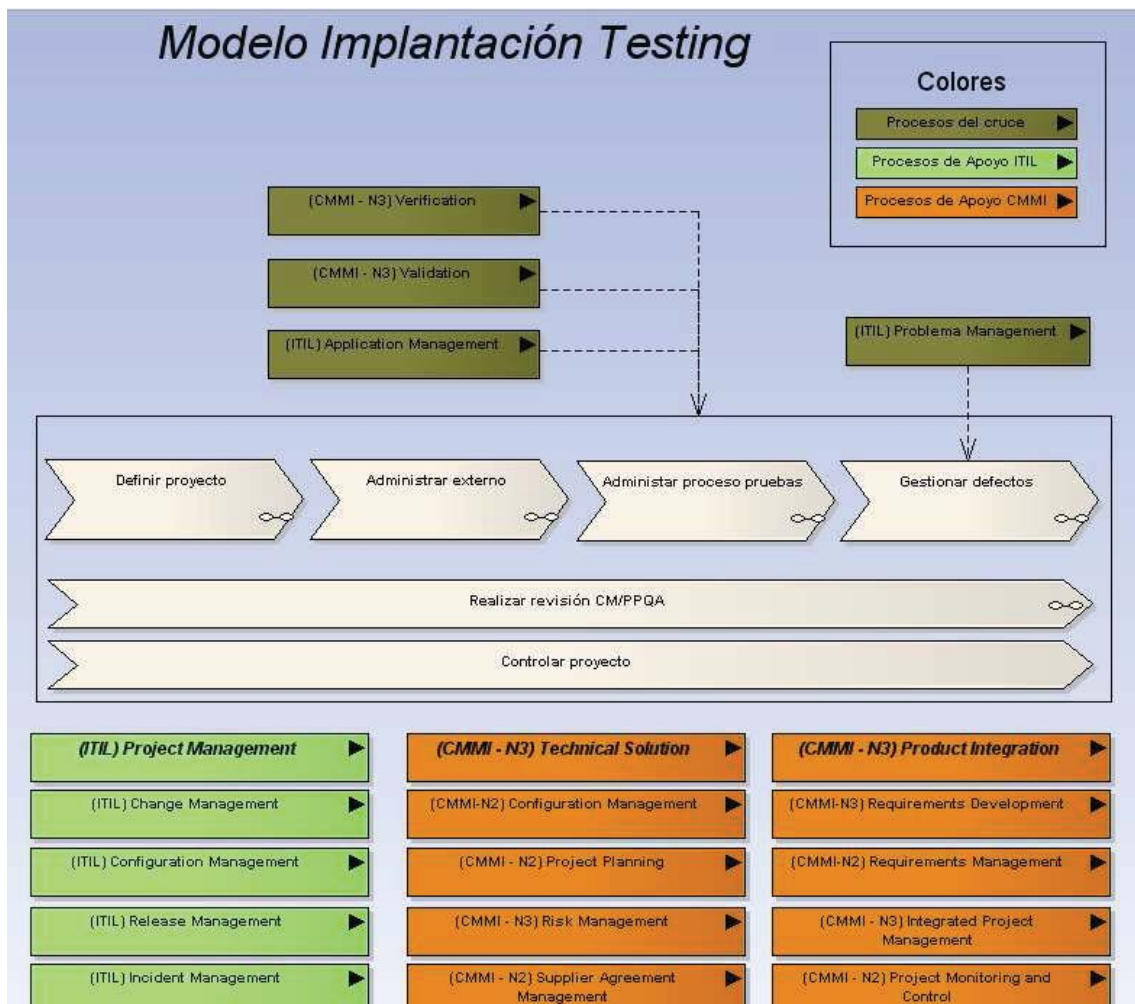


Figura 4.02: Metodología propuesta

En la Metodología se exponen las prácticas específicas y se detalla el proceso de cómo implantar la Metodología. El texto de la Metodología fue apoyado con un Diagrama de Flujo, el cual indica paso a paso como implantar el proceso de Testing en una empresa cualquiera.

A partir de la Metodología Propuesta se presenta la evaluación de su aplicación a un caso real, definiendo como punto de partida las métricas que se considerarán en dicha evaluación.

Al aplicar la Metodología al caso real, se refinaron algunas de las plantillas inicialmente generadas como apoyo.

El proceso de implementación consiguió como objetivo refinar y generalizar aún más los documentos de apoyo, ya que se amplió el espectro inicialmente pensado para tales documentos.

4.3 Definición de métricas para la evaluación final

A continuación se presentan los conceptos utilizados para la definición de métricas:

- Existen cuatro medidas para cuantificar: Caracterizar, Evaluar, Predecir y Mejorar.
 - 1) Medida: Valor asignado a un atributo de una entidad mediante una medición. Por ejemplo: 35.000 líneas de código.
 - 2) Medición: Es el acto de determinar una medida. Por ejemplo: Silvana será la encargada de medir las LDC de cada módulo del sistema.
 - 3) Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Incluye el método de medición. Por ejemplo: La productividad de este proyecto fue de 500 líneas/personas-mes (LDC/pm).
 - 4) Indicador: Es una métrica o combinación de métricas que proporcionan una visión profunda del proceso de software. Por ejemplo: la productividad media de nuestra empresa es de 500 (LDC/pm).

Las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo y el producto para intentar aumentar su calidad (LLeón-GPuglla, 2008).

- Por otro lado es importante saber que muchas veces el gerente de proyecto, se encuentra en la disyuntiva de dar o no el peso correcto a cada una de las variables que afectan el desarrollo de un proyecto en particular.

Existen restricciones tradicionales, como en cualquier empresa, los proyectos necesitan ser ejecutados y entregados bajo ciertas restricciones. Tradicionalmente, estas restricciones han sido alcance, tiempo y costo. Esto también se conoce como el Triángulo de la Gestión de Proyectos, donde cada lado representa una restricción. Un lado del triángulo no puede ser modificado sin impactar a los otros. Un refinamiento posterior de las restricciones separa la calidad del producto del alcance, y hace de la **calidad** una cuarta restricción.

- La restricción de tiempo se refiere a la cantidad de tiempo disponible para completar un proyecto.
- La restricción de costo se refiere a la cantidad presupuestada para el proyecto.
- La restricción de alcance se refiere a lo que se debe hacer para producir el resultado final del proyecto.

Estas tres restricciones son frecuentemente competidoras entre ellas: incrementar el alcance típicamente aumenta el tiempo y el costo, una restricción fuerte de tiempo puede significar un incremento en costos y una reducción en los alcances, así como también un presupuesto limitado puede traducirse en un incremento en tiempo y una reducción de los alcances.

La disciplina de la gestión de proyectos consiste en proporcionar las herramientas y técnicas que permiten al equipo de proyecto (no solamente al gerente del proyecto) organizar su trabajo para cumplir con todas esas restricciones.

4.3.1 Tiempo

El tiempo se descompone para propósitos analíticos en el tiempo requerido para completar los componentes del proyecto que es, a su vez, descompuesto en el tiempo requerido para completar cada tarea que contribuye a la finalización de cada componente. Cuando se realizan tareas utilizando gestión de proyectos, es importante partir el trabajo en pedazos menores para que sean fáciles de seguir.

4.3.2 Costo

El costo de desarrollar un proyecto depende de múltiples variables incluyendo costos de mano de obra, costos de materiales, administración de riesgo, infraestructura (edificios, máquinas, etc.), equipo y utilidades. Cuando se contrata a un consultor independiente para un proyecto, el costo típicamente será determinado por la tarifa de la empresa consultora multiplicada por un estimado del avance del proyecto.

4.3.3 Alcance

Requerimientos especificados para el resultado final. La definición global de lo que se supone que el proyecto debe alcanzar y una descripción específica de lo que el resultado final debe ser o debe realizar. Un componente principal del alcance es la *calidad* del producto final. La cantidad de tiempo dedicado a las tareas individuales determina la *calidad* global del proyecto. Algunas tareas pueden requerir una cantidad dada de tiempo para ser completadas adecuadamente, pero con más tiempo podrían ser completadas excepcionalmente. A lo largo de un proyecto grande, la *calidad* puede tener un impacto muy significativo en el tiempo y en el costo (o viceversa) (PMBOK, 2004).

La Figura 4.03 muestra las restricciones anteriormente descritas. Éstas se pueden representar en un triángulo, el cual aumenta o disminuye sus ángulos dependiendo de la importancia que se le aplique a cada uno de sus vértices (restricciones). Lo anterior refleja que en el caso que se le asigne mayor o menor importancia a uno de sus ángulos, éstos impactarán en la importancia de los demás vértices.

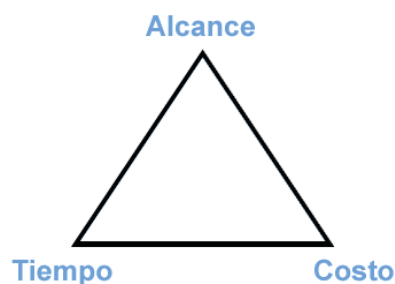


Figura 4.03: Triángulo de restricciones

A partir de los puntos antes descritos; Métricas y Triángulo de restricciones, se definen las siguientes métricas para la medición del éxito en la aplicación de la Metodología Propuesta a nivel de proyecto de Testing:

- El tiempo siempre será la variable menos movable con respecto a las restricciones, ya que todo proyecto tiene como límite una fecha, ya sea por necesidades del cliente o por imposición política (frecuente en empresas públicas).
- Una vez que el tiempo está establecido es importante saber el presupuesto con el que el cliente cuenta, ya que muchas veces es la segunda variable impuesta a nivel de requerimientos. El presupuesto puede estar dado mensual, semanal o incluso diario, dependiendo del tipo de proyecto.
- Finalmente queda el alcance, en el cual el cliente indica su óptimo con respecto al requerimiento en general. A parte de los costos que involucra el desarrollo del proyecto (específicamente lo que se evaluará en la Metodología), se debe considerar la calidad asociada al proceso de desarrollo, el cual también debe ser considerado en esta restricción.

Para el Testing de un aplicativo, teóricamente es prudente considerar a lo menos el 30% del Universo de pruebas posibles; es decir, si se estiman 100 pruebas de negocio para el aplicativo, lo prudente es testear al menos 30 de dichas combinatorias de pruebas de negocio. Más adelante se explicará la forma de estimar el Universo para luego indicar el cálculo del 30% en base a pruebas de negocio base y específicas.

Del 30% que será testeado, se procederá a establecer en el Plan de Pruebas, las políticas para la aceptación del aplicativo; es decir, se indicarán los porcentajes de aceptación para considerar efectivo el esfuerzo aplicado en el Testing.

Por experiencias en proyectos de Testing, lo mínimo recomendable como política de aceptación del 30% testeado será un 80% de aceptación SIN incidentes invalidantes. Muchas veces este porcentaje no es alcanzado, por malas prácticas en el desarrollo del software, así como también por la inexperiencia en el proceso de testeo.

Lo anterior indica que en el ejemplo de las 100 pruebas como Universo estimado, 24 pruebas (de las 30 seleccionadas) deben ser exitosas y no deben existir pruebas con invalidantes al momento de la ejecución. Por lo anteriormente expuesto, entonces se declaran 3 métricas para la evaluación de la implantación de la métrica, las cuales se listan en la Figura 4.04.

N°	Medición	Medida	Métrica
1	Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto.	HH	Esfuerzo Estimado v/s Esfuerzo Real
2	Se medirá el cumplimiento del costo establecido para el Testing del proyecto.	UF	Costo Estimado v/s Costo Real
3	Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.	%	% Establecido v/s % Obtenido

Figura 4.04: Métricas definidas para validación de la implantación de metodología propuesta

Las métricas obtenidas deberán tender en su mayoría a cero una vez calculado su diferencial. Luego de calculado su diferencial, éstos deberán ser analizados por el administrador del proyecto, quien tendrá la noción del proyecto general con respecto a los diferenciales obtenidos. Lo anterior, siempre con la mirada de "un traje de novia con excelencia en calidad al costo estimado, no sirve de nada si es entregado después de la fecha establecida".

Para medir la aplicación de la Metodología Propuesta se sugiere la siguiente métrica:

- 1) Porcentaje de Apego a la Metodología: Esta métrica tiene relación con identificar en que porcentaje se ha aplicado la Metodología Propuesta al concluir el Proyecto de Testing.

Las medidas utilizadas en el cálculo de la métrica se desprenden de la expertise obtenida en el desarrollo de la Propuesta Metodológica, así como también en la aplicación consecutiva de éstas en proyectos diversos para mejorarla.

La Figura 4.05 se refiere a la importancia de la plantilla diseñada para el apoyo al proceso de implantación. Esta importancia está dada por la experiencia en el proceso, pensando siempre en el cumplimiento del objetivo propuesto: apego a la Metodología.

Importancia Metodológica	
Muy Importante	4
Importante	3
Media	2
Baja	1

Figura 4.05: Métrica "Importancia Metodológica"

La Figura 4.06 entrega la relación de la planilla al proceso de implantación bajo la visión de Modelo diseñado del proceso de Testing; es decir, si la planilla apoya al proceso de CMMI, ITIL o es parte del apoyo a un proceso del cruce realizado entre ambos Modelos.

Propuesta Metodológica	
Cruce	5
CMMI	3
ITIL	2

Figura 4.06: Métrica “Propuesta Metodológica”

Una vez definidas las medidas se define la forma en la cual se calculará el apego a la Metodología. A continuación (Figura 4.07) se presenta la tabla con los campos ya asignados para el cálculo de la métrica de apego.

Columna A	Columna B	Columna C		Columna D	Columna E
Estructura	Plantilla	Propuesta Metodológica		Importancia en Metodología	Apego
0. Entradas	Informe de Diseño proveedor(Extracción).doc	CMMI	3	4	12
	PlanificaciónSeguimientoSemanal.mpp	CMMI	3	4	12
	Visión y Alcance.doc	CMMI	3	2	6
1. Análisis	Matriz_de_Evaluacion_de_Propuestas.xls	CMMI	3	2	6
	Estimación Modelo de Servicios.xls	Cruce	5	4	20
	LB_Requerimientos_Calidad.xls	ITIL	2	3	6
2. Diseño	Definición CPL.xls	ITIL	2	4	8
	TF Diseño.xls	ITIL	2	3	6
	Plan_de_CM.doc	CMMI	3	1	3
	Plan_de_Pruebas.doc	CMMI	3	3	9
3. Ejecución	TF Ejecución.xls	Cruce	5	3	15
4. Gestionar Defectos	TF Entidad-Defecto.xls	Cruce	5	3	15
5. Informes	Plan de Negocio.xls	Cruce	5	4	20
	Cierre Proyecto.xlsx	Cruce	5	2	10
	RESULTADO TESTING Ciclo.doc	Cruce	5	2	10

Figura 4.07: Tabla “Cálculo de métricas”

- 1) La Columna A se refiere a la estructura dada para organizar las planilla diseñadas. Concuerdan con el ciclo del proyecto que se testea.
- 2) La Columna B identifica cada una de las plantillas diseñadas para el proceso de implantación del Testing.
- 3) La Columna C clasifica cada planilla diseñada según la medida explicada anteriormente (Propuesta Metodológica).
- 4) La Columna D clasifica cada planilla diseñada según la medida explicada anteriormente (Importancia en Metodología).
- 5) La Columna E es la multiplicación de la Propuesta Metodológica (Columna C) por la Importancia Metodológica (Columna D).
- 6) En la Figura 4.08 se explica la segunda parte de la Métrica en base al apego de la Metodología Propuesta, la cual considera principalmente la aplicación de las métricas propuestas (Figura 4.05 y 4.06).

Columna B	Columna F	Columna G	Columna H	Columna I	Columna J	Columna K	Columna L
Plantilla	Aplicada	Apego por Aplicada	Distribución Importancia	Refinada	Multiplica (Distribución * Refinado)	Apego por Fase	Porcentaje por Estructura
Informe de Diseño proveedor(Extracción).doc							0. Entradas
PlanificacionSeguimientoSemanal.mpp							
Visión y Alcance.doc							
Matriz_de_Evaluacion_de_Propuestas.xls							1. Análisis
Estimación Modelo de Servicios.xls							
LB_Requerimientos_Calidad.xls							
Definición CPI.xls							2. Diseño
TF Diseño.xls							
Plan_de_CM.doc							
Plan_de_Pruebas.doc							3. Ejecución
TF Ejecución.xls							
TF Entidad-Defecto.xls							4. Gestionar Defectos
Plan de Negocio.xls							5. Informes
Cierre Proyecto.xlsx							
RESULTADO TESTING Ciclo.doc							

Figura 4.08: Tabla “Apego de la metodología propuesta”

- 1) La Columna F debe ser llenada con 1 si la planilla fue aplicada en el proceso de Testing o con uno 0 en el caso de que no se haya aplicado.
- 2) La Columna G multiplica la Columna E (Apego) por la Columna F (Aplicada) con el fin de obtener el apego real en la Metodología aplicada por cada una de las plantillas.
- 3) La Columna H es la distribución de la importancia que se calcula considerando Columna G (Apego por Aplicada) dividido por la sumatoria de cada plantilla que compone una estructura (Columna A).
- 4) La Columna I se debe llenar con un 1 para representar que SI se ajustará la planilla propuesta, al momento de su aplicación en lo que respecta a mejoras para su posterior utilización, o bien con un 0,9 en el caso que NO haya sufrido modificaciones la planilla propuesta al momento de su aplicación en lo que respecta a mejoras para su posterior utilización.
- 5) La Columna J es la multiplicación de Columna H (Distribución Importancia) por Columna I (Refinada) para así obtener la medida final en la siguiente columna.
- 6) La Columna K es la sumatoria de toda la columna J de cada plantilla existente por Estructura (Columna A). Esta medición entrega el apego a la Metodología por Estructura definida.
- 7) La Columna L es el recordatorio de la composición por Estructura.

El porcentaje de Apego a la Metodología se calcula sumando todas las medidas obtenidas por estructura (Columna K) dividida por 6 (cantidad de estructuras: Entradas, Análisis, Diseño, Ejecución, Gestionar Defectos e Informes).

Es importante mencionar que se realizó el ejercicio considerando el mínimo recomendable en la utilización de las plantillas en la implementación de la Metodología, llegando a la conclusión de que al aplicar la métrica de Apego a la Metodología se puede observar que el escenario peor es a lo menos un 80% de apego a la Metodología como porcentaje general, ya que sin este porcentaje no se estaría consiguiendo el objetivo de aportar en el proceso de implementación del Testing.

5 Implantación de la metodología propuesta y aplicación de plantillas de apoyo

A partir de la Metodología Propuesta se definen las plantillas de apoyo por proceso en el siguiente orden, considerando siempre el apoyo de los procesos del cruce, procesos de apoyo ITIL y procesos de apoyo CMMI. Es importante mencionar que se creó un nuevo proceso de Control de Proyecto que permitirá ir analizando las métricas propuestas anteriormente.

Estas plantillas apoyan el proceso de Testing (Verificación y Validación).

1. Definir Proyecto
2. Administrar Externo
3. Administrar Proceso Pruebas
4. Gestionar Defectos
5. Realizar Revisión CM/PPQA
6. Controlar Proyecto

Las planillas de apoyo se verán reflejadas como documentos asociados al proceso específico según se muestra en la Figura 5.01.

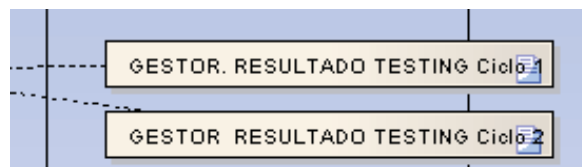


Figura 5.01: Identificación planillas de apoyo

Es importante indicar que el documento contiene el nombre de la planilla de apoyo que se encuentra en los anexos. Además la línea punteada relaciona el documento al proceso que lo utiliza.

5.1 Proceso para aplicar la metodología

La generación de esta Metodología nace de la necesidad y experiencia de establecer un método en el trabajo de la estimación y aplicación del Testing Funcional y No Funcional en un Área de Calidad de una EMPRESA.

Históricamente se sabe que un proyecto nunca es igual a otro, siempre existen diferencias significativas en el desarrollo de éstos, por lo que para la aplicación de la Metodología inicialmente se consideró incorporar como prerrequisitos mínimos los entregables que aplican como entrega por parte del equipo de proyecto. Por lo

anteriormente indicado se hacía cada vez más necesario establecer patrones de semejanza en la aplicación de un método para levantar un “proyecto” de Testing sobre un proyecto de desarrollo de software.

A medida que se analiza la generación de la Metodología se comienzan a exigir como Área, entregables mínimos a considerar en un proyecto de desarrollo; como por ejemplo, documentos de especificación de requerimientos, documentos de diseño, y documentos formales de aceptación de los documentos antes indicados, con el fin de establecer líneas base claras a la hora de estimar y planificar el proyecto de Testing.

Una vez generada la Metodología, se escogió un proyecto real, de una de las áreas con las cuales se llevaban muchos trabajos históricos, lo que ayudó, en gran medida, a contar con los documentos mínimos necesarios.

Al aplicar esta Metodología, se podrá apreciar que muchas veces los documentos mínimos no son considerados formalmente en un proyecto de desarrollo. Esto es una muy mala práctica, ya que no se puede establecer un Método sobre un proyecto que viene mal “engendrado”. A pesar de lo anteriormente expuesto, las plantillas generadas servirán de apoyo para las etapas de Requisitos, Análisis y Diseño de un proyecto de desarrollo de software.

El proceso de aplicación comienza con la exigencia de los documentos mínimos y la definición (estimación y planificación) del proyecto de Testing que se apegará a las condiciones del proyecto de desarrollo.

Se establecieron Métricas que se consideran evaluar al término de la aplicación de la Metodología, con el fin de mostrar el real aporte de su aplicación.

Las Plantillas aplicadas, son presentadas en el ANEXO 1. A medida que se utilizan en la Metodología serán referenciadas. Algunas de ellas se detallarán, con el fin de describir las transformaciones que han sufrido en el proceso de aplicación al proyecto real.

Es importante mencionar que además se incorporaron algunos documentos más que no estaban considerados inicialmente como plantillas, pero que en el proceso se pudieron percibir como gran apoyo.

Se recuerda que los procesos en azul tienen directa relación con la posible utilización de la Metodología para procesos apoyados con herramientas existentes en el mercado; en este caso, específicamente con Mercury de HP.

Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.1 Proceso “Definir proyecto”

En la Figura 5.02 se puede ver que no existen plantillas de apoyo al proceso. En su completitud este proceso apoya en la generación de un nuevo proyecto en el caso de que se utilice una herramienta de Prueba.

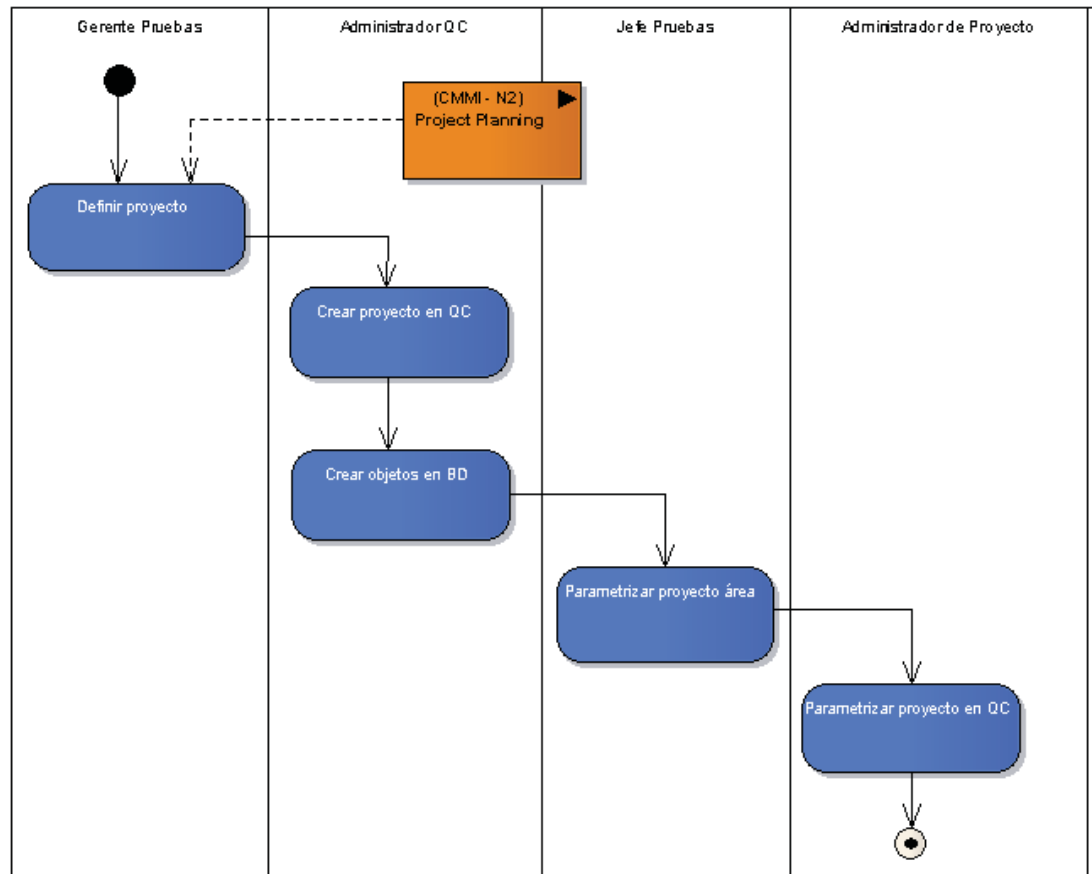


Figura 5.02: Proceso “Definir proyecto”

Proceso	Definir proyecto
Descripción	<p>Una vez que ha sido aprobado por el cliente la propuesta de servicio:</p> <ul style="list-style-type: none"> - Se define el nombre del proyecto y a qué línea de negocios pertenece. - Definición de las áreas de calidad involucradas y los Jefe de Área/Prueba involucrados. - Se debe definir las contrapartes válidas (Jefes de Proyectos) a las cuales se les derivarán los defectos y si accesarán QC. - Se define la política de respaldo y Planes de Pruebas - Checklists serán versionados en Subversion (herramienta de versionamiento) indicando el repositorio. - Se definen los aplicativos y las versiones para cada uno. - Se entrega la información anterior al "Administrador QC" para crear el proyecto en QC.

Proceso	Crear proyecto en QC
Descripción	<p>Se realizan las tareas asociadas a la creación del proyecto en QC, se ajusta una copia de la plantilla Domain: Base y Project: Proyecto basándose en la especificación entregada por el Jefe de Pruebas.</p> <ul style="list-style-type: none"> - Conectarse como Administrador de QC a QC Project Administration - Crear un proyecto dentro del dominio correspondiente. (QC Project Administration- Site Projects). Elegir la opción 'Create a project by copying data from an existing project' - Seleccionar el proyecto plantilla mencionado Domain: Base y Project: Proyecto - Chequear Customization, Requirements, Tests, Test Sets, Public Favorite Views. - Especificar el servidor de BD donde residirá el proyecto. - Asignar el Administrador del Proyecto (Se asume un mismo Administrador para todos los proyectos) - Activar el proyecto (Chequear Activate project) - Dado que los proyectos se guardarán distribuidamente en la BD y en directorios del S.O. no chequear 'Store project's repository in the database' - Dado que no se usará versionamiento, debido a que éste sólo cubre las pruebas y no el resto de las entidades, no chequear 'Create a Version Control database' - Chequear por conectividad a la BD usando el botón Ping. - Crear los usuarios que aún no existen y hayan sido especificados en la Definición del Proyecto .(QC Project Administration - Site Users) - Conectarse a Quality Center como el Administrador de Proyecto al nuevo proyecto. - Crear los aplicativos y releases. Renombrar la carpeta raíz Releases con el nombre del proyecto. - Informar al administrador de la BD, que se generó una nueva BD, con el fin de actualizar el job de respaldo.

Proceso	Crear objetos en BD
Descripción	<p>Crear los objetos cuyos scripts están en el tab Files usando el usuario td, previa especificación de la BD en ellos.</p> <p>Crear sólo las funciones get_domainname y get_projectname pues éstas ya fueron creadas en la BD de Administración de QC.</p>

Proceso	Parametrizar proyecto área
Descripción	<p>En esta tarea se especifica las parametrizaciones propias de cada Área de Calidad:</p> <ul style="list-style-type: none"> - Definir usuarios del área que participarán en el proyecto. - Se identifica si el proyecto necesita customizaciones especiales. Estas pueden implicar: <ol style="list-style-type: none"> (1) Creación de grupos usuarios especiales (2) Creación de nuevos campos para las entidades (3) Creación o mantención de Listas de Lookup.

Proceso	Parametrizar proyecto en QC
Descripción	<p>Una vez que el proyecto ha sido creado se realiza su parametrización:</p> <ul style="list-style-type: none"> - Crear los grupos de usuarios necesarios (QC Project Customization - Groups). - Asignar los usuarios al proyecto (QC Site Administration - Site Users - User Projects) - Si QC se versiona en Subversion y no se está usando sólo la cuenta de Subversion Default, agregar los mappings entre usuarios QC y Subversion. Nótese que si el versionamiento ya estaba activo, debe ser desacoplado, cambiado y vuelto a acoplar. - Realizar las customizaciones (QC Project Customization)

5.1.2 Proceso “Administrar externo”

El proceso “Administrar Externo” se basa fundamentalmente en la búsqueda de externos para el apoyo del Testing. En el caso que sea necesaria la utilización de externos es importante tener realizada la estimación del proyecto es lo que respecta al proceso de Testing.

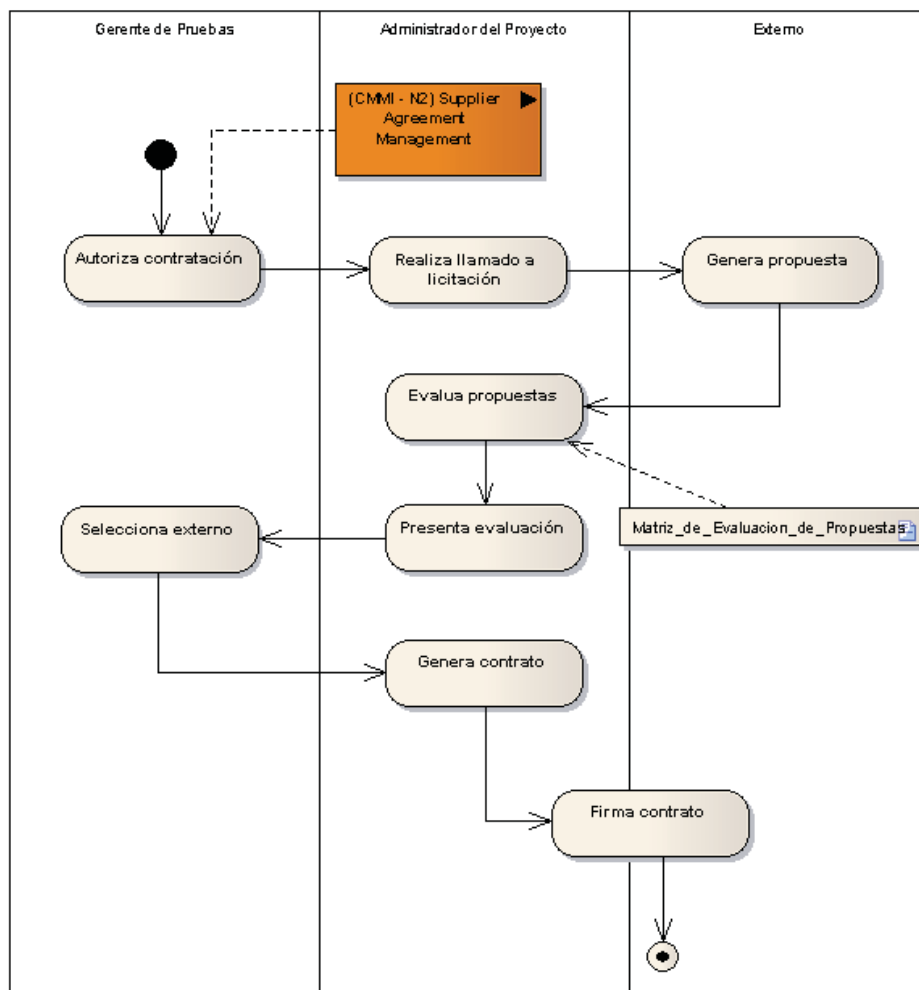


Figura 5.03: Proceso “Administrar externo”

Proceso	Autoriza contratación
Descripción	Una vez que el equipo de Gestión de Proyecto autoriza la utilización de recursos externos. El equipo de proyecto debe tener presente que en estos casos el costo aumenta, ya que contratar una “buena” empresa de Testing involucra experiencia, eficacia y eficiencia en el proceso.

Proceso	Realiza llamado a licitación
Descripción	El equipo de Proyecto entrega la potestad al equipo de Testing en el llamado a licitación, pudiendo ésta ser de carácter público o reservado, dependiendo de la estrategia de empresa por la que se rija el equipo de Testing. Es importante mencionar que un buen llamado a licitación consiste en precisar el servicio que se solicita, tanto en tiempo, costo, calidad y por sobre todo en lo que respecta a los rendimientos tanto del diseño de los casos como en la ejecución de ellos. Otro punto importante a considerar es explicar en detalle la definición de los conceptos base para poder realizar el cálculo del Testing, que en este caso sería la definición de Caso de Prueba Integrado, Casos de Prueba Unitarios o Casos de Negocio. Las definiciones antes mencionadas están especificadas en el Glosario de este documento.

Proceso	Genera propuesta
Descripción	En base al llamado de licitación, el externo genera su propuesta de servicio, en la cual debe especificar claramente el cumplimiento de plazos, restricciones del servicio en caso de necesidades de horas extras y el costo asociado.

Proceso	Evalúa propuestas (*)
Descripción	Una vez que los proveedores de servicio entregan sus propuestas, el equipo de Testing debe comenzar con la evaluación. Es en este punto donde aparece nuestra primera Plantilla de apoyo (ver ANEXO 1 “Matriz_de_Evaluación_Propuestas”).

Proceso	Presentar evaluación
Descripción	Una vez aplicada la matriz de evaluación, es necesario presentar los resultados al equipo de Proyecto y realizar las recomendaciones respecto a dicha evaluación.

Proceso	Selecciona externo
Descripción	El equipo de Proyecto entonces está en condiciones de realizar la selección del externo para que participe en el proceso de Testing, pudiendo ser ésta el diseño y/o ejecución de las Pruebas.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

Proceso	Generar contrato
Descripción	<p>El equipo de Testing está en condiciones de generar el contrato de servicio con el externo.</p> <p>Es importante recomendar que dicho contrato esté orientado a entregables de trabajo relacionados con el proceso del Testing en una primera vista, pero que en segunda línea se especifique claramente la cantidad de HH relacionadas al servicio.</p> <p>Se debe tener especial cuidado en las excepciones que se acuerden y en el aviso y satisfacción del servicio en lo que respecta a horas de trabajo extraordinario.</p>
Proceso	Firma contrato
Descripción	La firma del contrato debe ser entre equipo de Testing y externo, ya que las exigencias con respecto a respuesta en la entrega de servicio son de responsabilidad del equipo de Testing.

5.1.3 Proceso “Administrar proceso de pruebas” PARTE 1

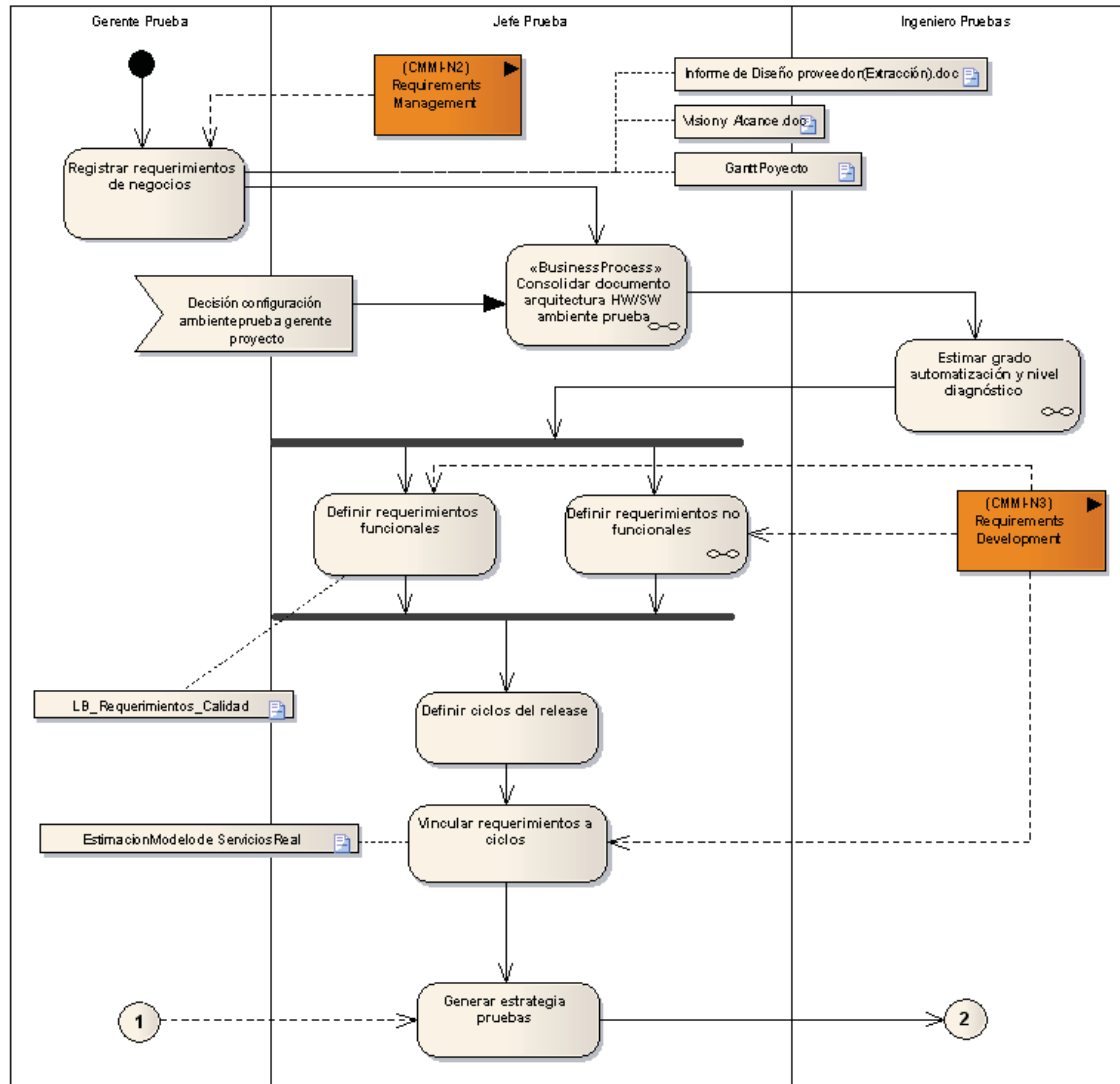


Figura 5.04: Proceso “Administrar proceso de pruebas” PARTE 1

Proceso	Registrar requerimientos de negocios (*)
Descripción	<p>El primer documento que debe aprobar el equipo de proyecto al desarrollador del aplicativo a testear es el de Visión y Alcance, el cual debe especificar claramente lo que se va a desarrollar. Siempre en términos generales en lo que respecta al aplicativo. En la plantilla de apoyo ANEXO 1 “Visión y Alcance” se explica el contenido esperado como mínimo en dicho documento.</p> <p>Luego que el documento de Visión y Alcance está acordado y aprobado por las partes es hora de que el desarrollador comience con el análisis para continuar con el diseño, otro de los input importantes a la hora de analizar qué testear.</p> <p>El proveedor de desarrollo debe entregar al equipo de proyecto, el documento de diseño del aplicativo que se va a testear. A partir de este documento el equipo de Testing debe ser capaz de entender y reflejar en la planilla de requerimientos, todos los requisitos que el aplicativo debe contener.</p> <p>El Informe de diseño debe contener todo lo indicado en la plantilla de apoyo ANEXO 1 “Informe Diseño del Proveedor”.</p> <p>Una vez aprobado el documento de Diseño, el Proveedor ya está en condiciones de declarar tiempos claros y precisos en una Carta Gantt, sobre la cual el equipo de proyecto y el equipo de Testing deben comenzar a analizar los instantes en los cuales el equipo de Testing podrá intervenir, así como también las posibles restricciones de tiempo y costo, con el fin de poder dilucidar cuál será el alcance de la claridad del Testing.</p> <p>La plantilla siguiente muestra una planificación tipo que puede ser utilizada como referencia para analizar las intervenciones del equipo de Testing. Ver ANEXO 1 “Planificación_Seguimiento_Semanal”.</p> <p>A nivel de equipo de Testing, a partir de los 3 documentos antes mencionados, debe ser capaz de definir los requerimientos del aplicativo, por lo anterior se deben realizar los siguientes pasos:</p> <p>Definir el requerimiento como carpeta, especificando:</p> <ol style="list-style-type: none"> 1. Nombre 2. Tipo de Requerimiento como de Negocios. 3. Descripción 4. Autor 5. Prioridad en la implementación:

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<p>1-Baja 2-Media 3-Alta 4-Muy Alta</p> <p>6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)</p> <p>Nota: Es posible agregar Attachments a los Requerimientos también.</p> <p>En el caso de Pruebas No Funcionales es recomendable que los requerimientos de negocios tengan la forma de SLAs estándares.</p> <p>NOTA: LOS REQUERIMIENTOS DE NEGOCIO SON DE ALTO NIVEL, Y SU ANALISIS PUEDE INVOLUCRAR SU DESCOMPOSICION EN REQUERIMIENTOS FUNCIONALES, DESEMPEÑO, SEGURIDAD, ETC.</p>
--	--

Proceso	Consolidar documento arquitectura HW/SW ambiente prueba
Descripción	<p>Si bien el área de calidad no es el encargado de generar este documento en forma íntegra si debiera consolidarlo para sí, previa ejecución de:</p> <ul style="list-style-type: none"> - Recolectar las especificaciones de Arquitectura que generen otras áreas de la Empresa, el Cliente o Terceros. - Generar la especificación del dispositivo GUI, su cantidad, ubicación física y conectividad. <p>El documento debe contener todas las componentes detalladas en el Diagrama "Ambiente_Prueba". Dado que los ambientes para Pruebas Funcionales y No Funcionales son potencialmente distintos se recomienda elaborar un documento para cada uno.</p>

Proceso	Estimar grado automatización y nivel diagnóstico
Descripción	<p>En esta tarea se estima que porcentajes de las Pruebas serán realizadas automática y manualmente. Dicha estimación se realiza basándose en:</p> <ul style="list-style-type: none"> - Diseños Preliminares del Sistema. Por ejemplo si el "Dispositivo GUI" es una PALM y no hay emulador confiable, las Pruebas funcionales deberán ser realizadas manualmente, pero puede que de todos modos sea factible realizar las Pruebas no funcionales. - Pruebas sobre el Aplicativo en el caso que esté disponible usando QTP y LR para las Pruebas funcionales y no funcionales respectivamente. <p>En el caso No Funcional si no es factible realizar las Pruebas automáticamente, probablemente la opción manual sea demasiado cara.</p>

Proceso	Definir requerimientos funcionales (*)
Descripción	<p>La planilla de apoyo "LB_Requerimientos_Calidad" en ANEXO 1 entrega el detalle del llenado para obtener el Universo estimado.</p> <p>Esta plantilla debe estar aprobada por el equipo de proyecto y debe sumar todos los casos de Prueba relacionados a los requerimientos con estado Declarado.</p> <p>Se definen los requerimientos funcionales detallando los atributos mencionados a continuación, si es necesario se crean carpetas que agrupen los requerimientos por temas.</p> <ol style="list-style-type: none"> 1. Nombre 2. Especificar Tipo de Requerimiento (Negocios, Funcional, No Funcional) 3. Descripción 4. Autor 5. Prioridad en la implementación: <ol style="list-style-type: none"> 1-Baja 2-Media 3-Alta 4-Muy Alta 6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)

Proceso	Definir requerimientos No Funcionales
Descripción	<p>Rendimiento, en los Requerimientos de Negocios debería existir la definición de SLA de tiempo de respuesta vinculado a un conjunto de transacciones, sin embargo se dan tres posibles situaciones:</p> <ul style="list-style-type: none"> - El conjunto de transacciones es adecuado para el tiempo asignado a Pruebas de Carga en el proyecto. Usualmente el tiempo alcanza para considerar entre 10 y 30 transacciones. - El conjunto de transacciones es muy amplio y no es factible realizar todas las mediciones en términos económicos. - El conjunto de transacciones ha sido enunciado en forma general y no se han identificado las transacciones.

Proceso	Definir ciclos del release
Descripción	Definir los "Ciclos del Release" indicando los "Conjunto de Pruebas" preliminares que los conforman, por ejemplo el primer "Ciclo de Pruebas" debiera incluir siempre un conjunto de "Pruebas Trazadoras" con el fin de establecer si el ambiente está operativo o no.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

Proceso	Vincular requerimientos a ciclos (*)
Descripción	<p>Esta es la instancia para utilizar la planilla de Apoyo “Estimación Modelo de Servicios” del ANEXO 1. Es importante mencionar que esta plantilla no se podrá utilizar si no está cerrada la planilla de apoyo de LB Requerimientos Calidad, ya que ésta tiene el input más importante para la estimación del servicio.</p> <p>En la plantilla de apoyo además se definen las fases y ciclos de Prueba así como también los recursos relacionados.</p> <p>Se deben vincular los requerimientos a al menos un "Ciclo de Prueba" con el fin de:</p> <ul style="list-style-type: none"> - Determinar en cualquier instante si los requerimientos están siendo cubiertos o no - Generar esbozo de Pruebas en función de los requerimientos asociados a un ciclo. <p>Un requerimiento puede ser vinculado a un Release, en este caso, todos los "Ciclos de Prueba" dentro del Release serán vinculados, y eso implica que debe haber un "Conjunto de Pruebas" al menos dentro de cada Ciclo que valida el requerimiento.</p> <p>La cobertura de Prueba en un ciclo queda definida por la liberación de funcionalidades nuevas y por los defectos ocurridos en el ciclo anterior (en el caso que exista), por lo tanto al principio del proyecto, sólo se vincularán los requerimientos asociados a funcionalidades nuevas al ciclo en que se prueban, y con posterioridad dependiendo de los defectos se vincularán requerimientos a ciclos que contengan Pruebas de Regresión.</p>

Proceso	Generar estrategia pruebas
Descripción	<p>Este documento debe ser un compendio del estado del proyecto hasta este momento, considerando:</p> <ul style="list-style-type: none"> - Roles y personas - Requerimientos de Pruebas - Ciclos de Pruebas - Protocolo de Suspensión de las Pruebas - Conclusiones de Estudio Factibilidad de Automatización. - Referencias a los documentos de Arquitectura de HW/SW. - Formato Plan de Pruebas o Operatoria al respecto - Formato Informe de Resultados de Pruebas. - Manejo de Defectos. - Severidad de Defectos. - Riesgos <p>En el caso de CM-PPQA se incluyen los checklists que se utilizarán en la Revisión. Para extraerlos de QC: Tools->; Excel Report Generator->; Checklist CM-PPQA->; Generate Excel Report->; (Ingresar como parámetro el nombre de la prueba que contiene el Checklist).</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.3.1 Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”

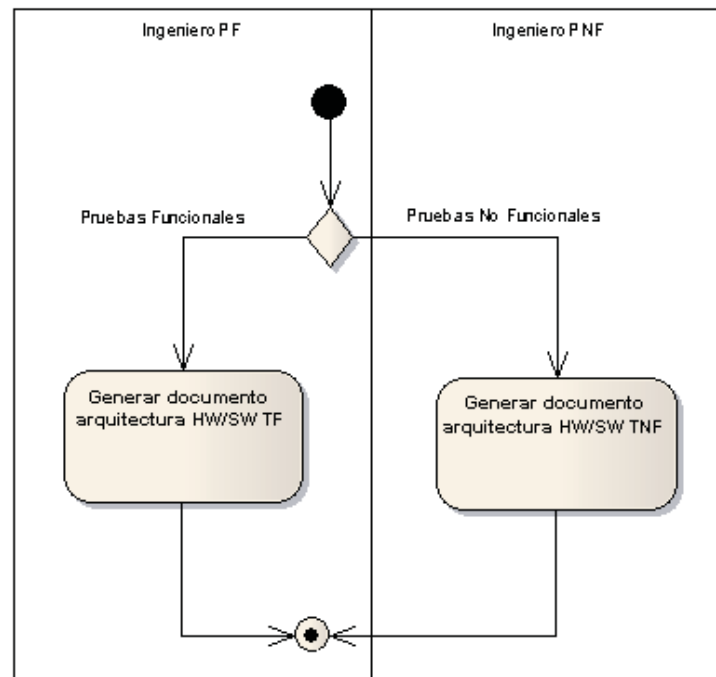


Figura 5.05: Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”

5.1.3.2 Proceso “Estimar grado automatización y nivel diagnóstico”

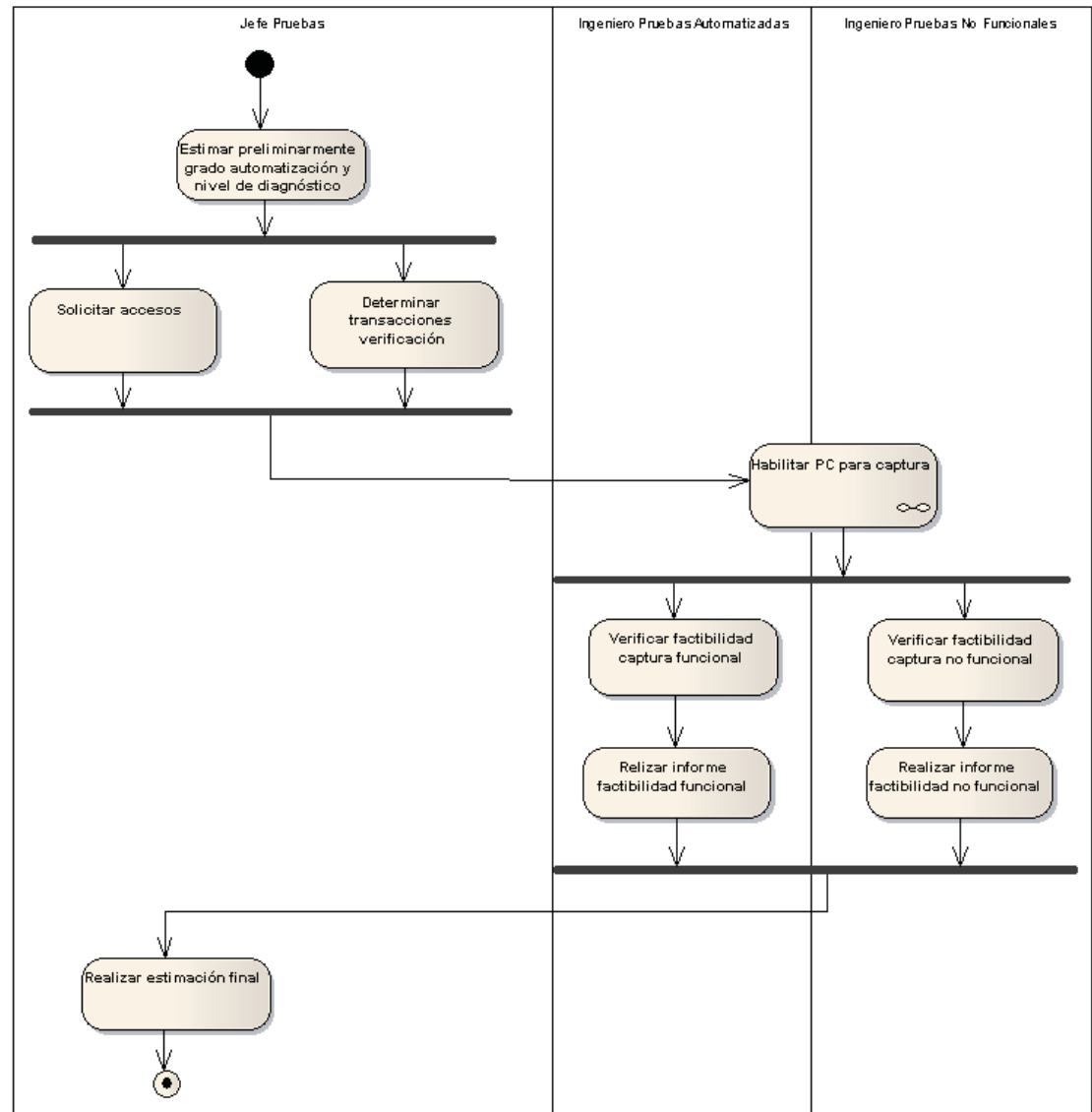


Figura 5.06: Proceso “Estimar grado automatización y nivel diagnóstico”

Proceso	Estimar preliminarmente grado automatización y nivel de diagnóstico
Descripción	<p>El análisis preliminar en esta tarea aborda el Grado de Automatización y el Nivel de Diagnóstico de la siguiente forma:</p> <p>Grado de Automatización</p> <p>En el caso que se disponga del aplicativo, se identifican todas aquellos módulos del aplicativo que no son automatizables con Mercury tanto desde el punto de vista funcional como no funcional, candidatos serían:</p> <ul style="list-style-type: none"> - General: (a) Uso de algoritmos de reconocimiento de imágenes, sonidos u otro distinto a eventos de teclado y mouse. (b) Dispositivos GUI no emulables en PC o notebooks. - Caso Funcional: (a) Reportes (b) Procesos Batch (c) Dispositivos GUI con mala emulación en PC. - Caso No Funcional: (a) Procesos Batch (b) Protocolos a nivel de GUI inexistentes en Mercury o no adquiridos. <p>Nivel de Diagnóstico</p> <p>Tanto en las Pruebas Funcionales como en las No Funcionales existe el mismo concepto, la diferencia es que en el último caso es siempre parte del servicio, en cambio en el caso funcional no.</p> <ul style="list-style-type: none"> - Caso Funcional: Si el servicio es Graybox, se deben verificar los resultados en la Base de Datos. Se debe evaluar si existen impedimentos para acceder la BD como Modelo no documentado. - Caso No Funcional: En este caso, si bien se espera un nivel Whitebox, no es siempre posible, debido a que pueda que no exista el SW para monitorear cada uno de los servidores aplicativos o en el caso que sea Mercury Diagnostics, no se esté dispuesto a hacer la inversión. <p>Si no se disponen del aplicativo se deberá hacer supuestos sobre las transacciones para las cuales no es tan claro si pueden ser automatizadas o no.</p>

Proceso	Solicitar accesos
Descripción	<p>Se debe solicitar:</p> <ul style="list-style-type: none"> - Acceso al aplicativo - Conectividad a Servidor QC en la Organización para QTP en el caso de pruebas funcionales.

Proceso	Determinar transacciones verificación
Descripción	<p>En base a los antecedentes que se disponga, se debe definir que transacciones serán las que se usarán para validar si es posible automatizar ya sea funcional o no funcionalmente. Se debe tener en cuenta al menos:</p> <p>Generales</p> <ul style="list-style-type: none"> - Debería haber al menos 1 transacción por arquitectura de SW distinta a nivel de "Dispositivo GUI" <p>Pruebas No Funcionales</p> <ul style="list-style-type: none"> - Informarse si algún SW encripta o comprime a nivel de "Dispositivo GUI" y si fuera así, solicitar los algoritmos al desarrollador (siempre y cuando sean indispensables para variabilizar los scripts VUGEN posteriormente). - Determinar si la aplicación es mono o multiusuario por dispositivo GUI.

Proceso	Habilitar PC para captura
Descripción	<p>Se contemplan 2 situaciones:</p> <ul style="list-style-type: none"> - Aplicativo se encuentra disponible a través de la red de la organización, en ese caso se utilizarán los computadores destinados para ese fin en dicha organización. - En el caso que no esté disponible en la red organizacional, se usará una máquina virtual que tenga QTP y LR listos para ser instalados.

Proceso	Verificar factibilidad captura funcional
Descripción	<p>La verificación consiste en que para cada transacción de verificación se ejecuta:</p> <ul style="list-style-type: none"> - Identificar los Add-ins que podrían necesitar en base a la documentación o referencias de la GUI. - Generar un script para cada una de las transacciones - Reejecutar el script. - Comprobar que se puedan incorporar puntos de verificación. - Comprobar que se puede variabilizar la entrada. <p>Una vez realizadas las tareas anteriores complete la planilla de Factibilidad QTP.</p>

Proceso	Verificar factibilidad captura No Funcional
Descripción	<p>La verificación consiste en que para cada transacción de verificación se ejecuta:</p> <ul style="list-style-type: none">- Generar un script que implique login, ejecutar transacción y logout.- Detectar correlación y resolverla en el caso que la hubiere.- Determinar si los mensajes son parametrizables.- Reejecutar el script.- Si es posible realizar una prueba con más de un usuario a la vez sobre el mismo "Dispositivo GUI". <p>Nota: Para resolución de problemas ver tab Files.</p>

Proceso	Realizar estimación final
Descripción	<p>Tanto para el caso funcional como no funcional, se consolida la estimación inicial con los resultados obtenidos en la verificación con el aplicativo.</p> <p>El resultado de esta estimación puede producir RFC de un impacto no despreciable en el presupuesto.</p>

5.1.3.2.1 Proceso “Habilitar PC para captura”

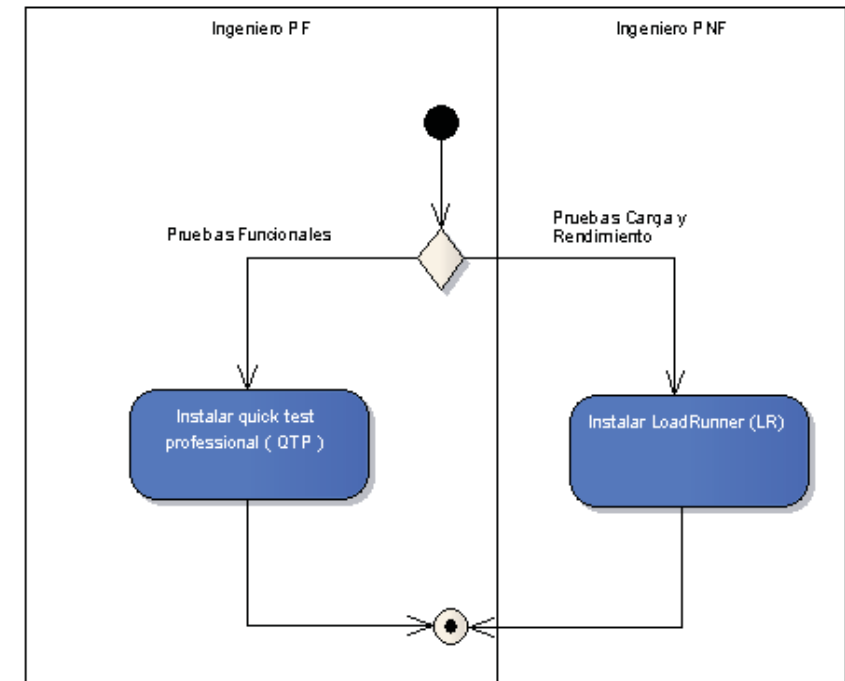


Figura 5.07: Proceso “Habilitar PC para captura”

5.1.3.3 Proceso “Definir requerimientos no funcionales”

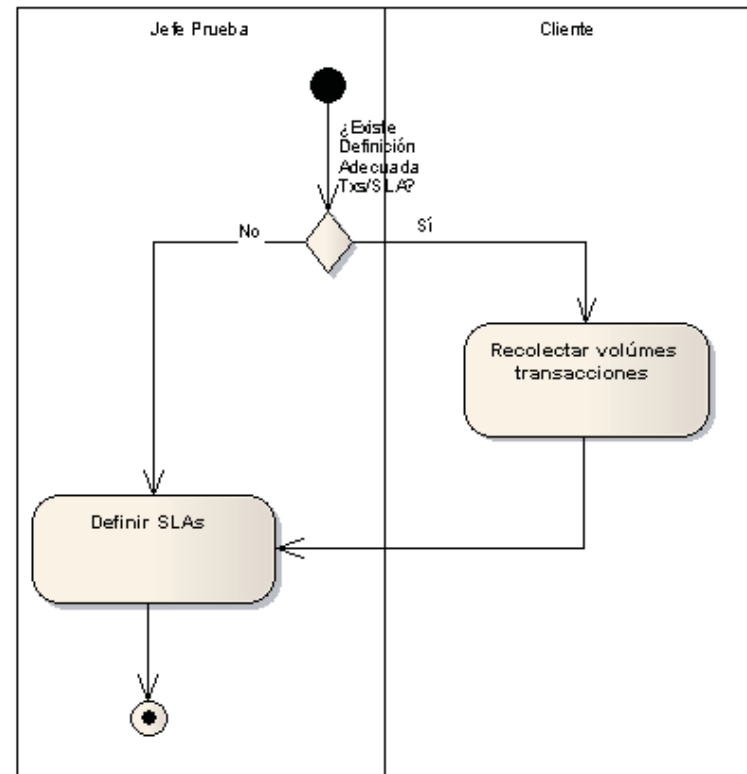


Figura 5.08: Proceso “Definir requerimientos no funcionales”

Proceso	Definir SLAs
Descripción	<ul style="list-style-type: none"> - Definir una carpeta con nombre "Pruebas de Carga y Rendimiento". - En el caso que exista información histórica o estimaciones, agregar attachment tipo URL apuntando a la ubicación de una planilla Excel "Volúmenes de Transacciones" en subversion. - Definir para cada transacción seleccionada un Requerimiento de Testing detallando su SLA, para el cual se debe especificar que acción del usuario gatilla la transacción que va a ser medida (por ejemplo click sobre el botón grabar).

Proceso	Recolectar volúmenes transacciones
Descripción	<p>Recolectar información histórica o estimar en un documento llamado "Volúmenes Transacciones", la siguiente información:</p> <ul style="list-style-type: none"> - Número de usuarios del sistema - % Crecimiento Anual Usuarios del Sistema - % Concurrencia de usuarios máxima - Una hoja de cálculo de transacciones cuyas columnas sean: Id Transacción, Nombre Transacción, Cantidad Transacciones Promedio x Tramo Horario i ($i=1, \dots, n$), % Crecimiento Anual. Ojala se pudieran descomponer la cantidad de transacciones por perfil - Una hoja de cálculo de transacciones cuyas columnas sean: Id Transacción, Nombre Transacción, Cantidad Transacciones Peak x Tramo Horario i ($i=1, \dots, n$), % Crecimiento Anual. Ojala se pudieran descomponer la cantidad de transacciones por perfil.

5.1.4 Proceso “Administrar proceso de pruebas” PARTE 2

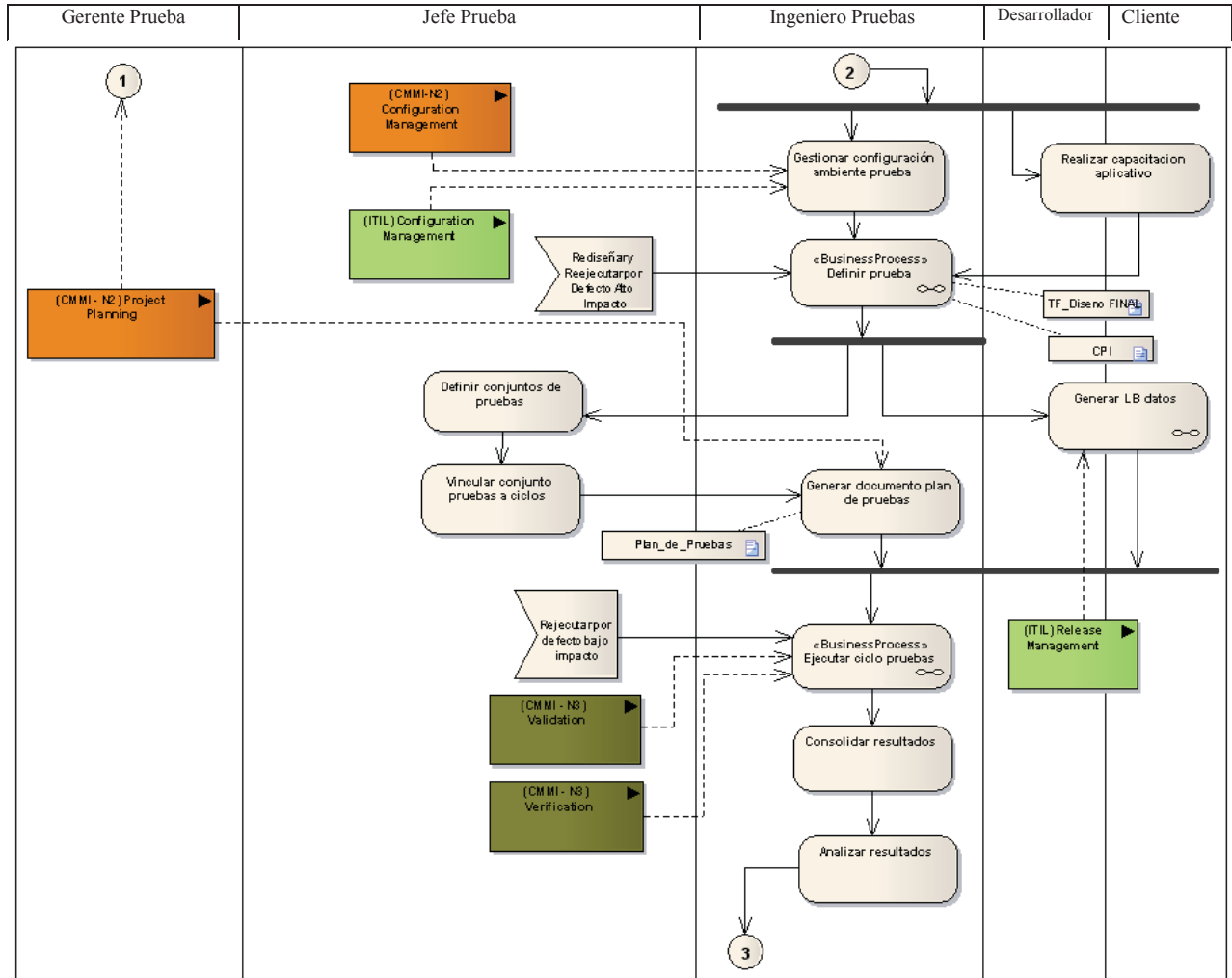


Figura 5.09: Proceso “Administrar proceso de pruebas” PARTE 2

Proceso	Gestionar configuración ambiente prueba
Descripción	<p>Con el objetivo de realizar las Pruebas, además de disponer del "Ambiente de Pruebas" (se debe definir de quien es la responsabilidad de entregar dicho ambiente de Pruebas), Calidad debe adaptar el ambiente para usar Mercury:</p> <ul style="list-style-type: none"> - Verificar que todas las componentes del Ambiente de pruebas que no son responsabilidad de Calidad, están disponibles. - Habilitar Laboratorio(s) Prueba(s): Disponer de "Dispositivos GUI", Instalar SW Calidad, Instalar SW Cliente. (Ver manual de instalación en Tab Files). - Generar los RFC para las conectividades de enlace con los "Servidores de Calidad" y "Dispositivo_GUI". - Generar los RFC para las configuraciones o instalaciones de "SW Calidad" en los "Servidores Aplicativo". - Enviar los RFCs a los responsables. - Verificar si los RFCs fueron correctamente ejecutados. - Verificar Normas de Seguridad impartidas por el Oficial de Seguridad.(Ver archivos adjuntos)

Proceso	Realizar capacitación aplicativo
Descripción	Realizar Capacitación del Aplicativo, idealmente se debiera hacer una Capacitación para Pruebas Funcionales y No Funcionales, pero podría haber casos donde tenga que ser realizada en sesiones separadas.

Proceso	Definir prueba (*)
Descripción	<p>La primera plantilla de apoyo que se utiliza en este proceso es "Definición CPI" ANEXO 1, en la cual se definen todas las combinatorias posibles en base a los criterios especificados en la planilla de apoyo.</p> <p>Una vez seleccionadas las Pruebas a diseñar, se debe preparar la planilla de apoyo "TF Diseño" ANEXO 1, en la cual se debe comenzar a identificar los ID de Prueba y relacionarlos a las Pruebas seleccionadas de la planilla anterior.</p> <p>Al mismo tiempo en este proceso se debe proceder con lo siguiente:</p> <p>Este proceso se basa en:</p> <ul style="list-style-type: none"> - Definición de Requerimientos de Pruebas: De los cuales se derivarán las Pruebas o "carpetas de Pruebas", las cuales serán refinadas con la documentación de Diseño que se tenga del aplicativo. QC tiene la opción de derivar automáticamente de los "Requerimientos de Pruebas" una jerarquía preliminar de Pruebas, la cual debe ser refinada. - La Capacitación Funcional y con otros entregables (sería ideal contar con las grabaciones de

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<p>ejecuciones sobre la GUI del aplicativo) se tiene la información necesaria para detallar los "Pasos" de cada Prueba y generar los "Scripts de Prueba" en el caso de automatización con "Load Runner".</p> <p>y contempla la realización de las siguientes tareas:</p> <ul style="list-style-type: none"> - Definir Estructura Directorios -Definir Pruebas Unitarias - Detallar Pruebas Unitarias - Definir Pruebas Integradas - Generar Scripts - Especificar Necesidad Datos - Detallar Pruebas Integradas <p>Es muy importante tener en mente que en cada ciclo puede ser necesario reejecutar algunas de las anteriores.</p>
--	--

Proceso	Definir conjuntos de pruebas
Descripción	<p>Se definen los conjuntos de prueba, ver ejemplos de estos conjuntos en Diagrama Conjunto_Prueba.</p> <p>En QC, Test Lab:</p> <ul style="list-style-type: none"> - Crear carpeta(s) que contienen "Conjuntos de Pruebas". - Crear cada "Conjunto de Pruebas" en la carpeta que corresponda, e ingresar las "Pruebas" que lo componen usando el tab "Execution Grid". El orden de ejecución se define en el tab "Execution Flow".

Proceso	Vincular Conjunto Pruebas a Ciclos
Descripción	En QC, Test Lab, las carpetas que contienen "Conjuntos de Pruebas" son linkeadas a ciclos. Para esto hacer click derecho sobre la carpeta y "Assign to Cycle".

Proceso	Generar documento plan de pruebas (*)
Descripción	<p>Con todos los procesos anteriores concluidos, y con todas las plantillas de apoyo concretadas, se debe dar pie a la generación del Plan de Pruebas, el cual contiene información de resumen de las decisiones y acuerdos tomados; es decir, es dejar reflejado el acuerdo para dar comienzo formula al proceso de diseño de las Pruebas.</p> <p>Plantilla de apoyo "Plan de Pruebas" ANEXO 1.</p>

Proceso	Generar LB datos
Descripción	Con el fin de realizar las Pruebas es necesario llevar al sistema a un determinado nivel de datos y generar los datos suficientes a utilizar en los escenarios de Prueba, esto implica :

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<ul style="list-style-type: none"> - Generar BDs, las cuales deben ser parte de la LB, la cual debería ser administrada por CM (ver qué pasa en los clientes externos). - Generar Datos en la cantidad necesaria para realizar las Pruebas
--	--

Proceso	Consolidar resultados
Descripción	<p>Esta tarea consiste en determinar si los defectos reportados durante las Pruebas son reducibles a un grupo menor o son preexistentes.</p> <p>En el caso que un grupo de defectos pueda ser reemplazado por un solo defecto:</p> <ul style="list-style-type: none"> - Elegir un defecto del grupo como representativo. - Linkear el resto de los defectos al seleccionado en el paso 1. - Dejar los defectos linkeados en estado "Consolidated".

Proceso	Analizar resultados
Descripción	<p>Esta tarea consiste en determinar si los defectos reportados durante las Pruebas son tales.</p> <ul style="list-style-type: none"> - En el caso que el defecto sea tal, actualizar su estado a "Open". - En el caso que el defecto no aplique, se puede borrar o dejar su estado en "Closed", esto último sólo si es de algún aporte mantener el registro.

5.1.4.1 Proceso “Definir prueba”

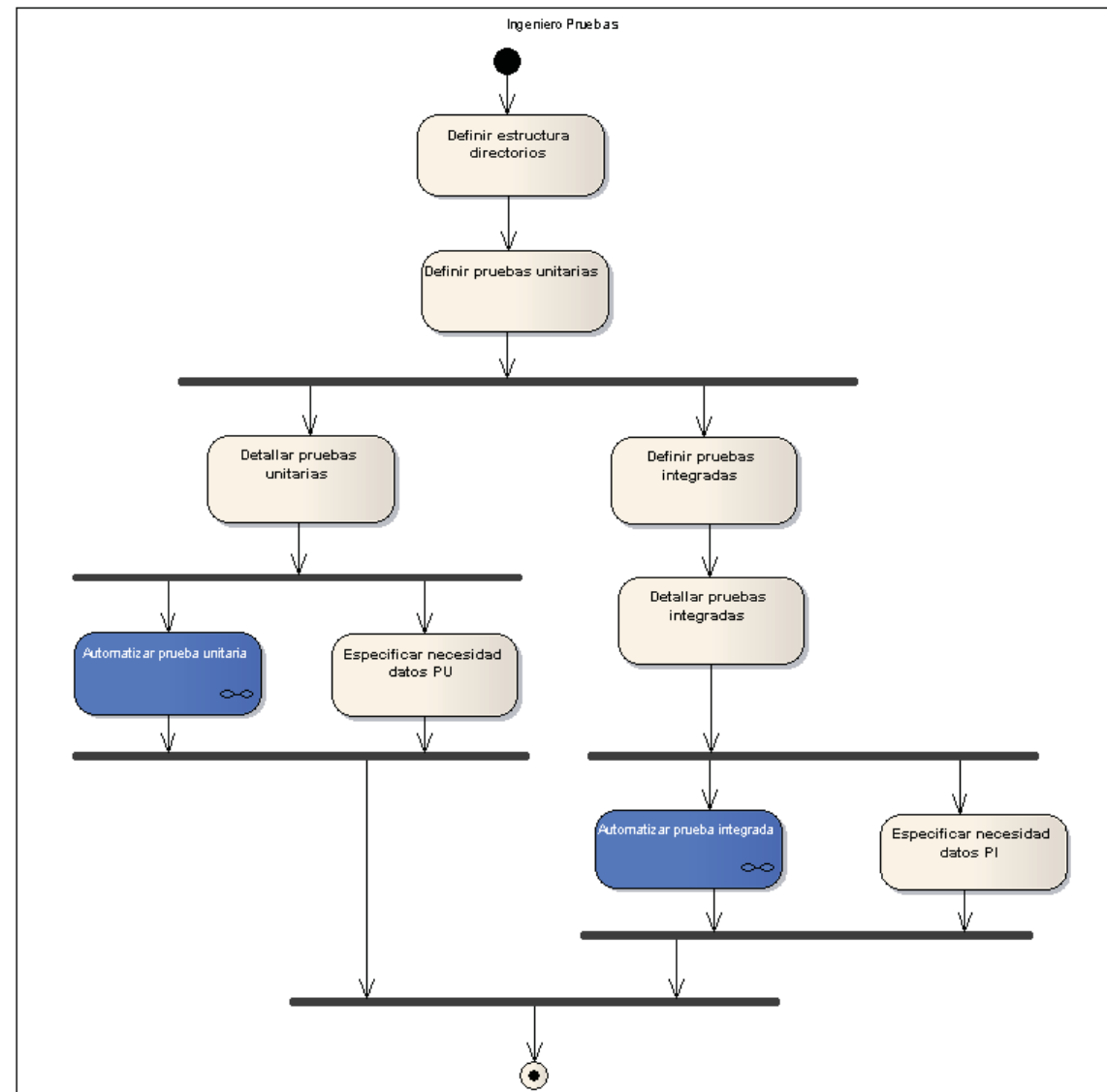


Figura 5.10: Proceso “Definir prueba”

Proceso	Definir estructura directorios
Descripción	<p>Crear los directorios que se necesitan para almacenar los scripts de pruebas, repositorios de objetos y bibliotecas de funciones tanto de QTP como LR. Se recomienda crear en Subversion:</p> <ul style="list-style-type: none"> - Crear directorio de Scripts, se sugiere que el nombre de cada script se un prefijo de a lo más 3 caracteres alfabéticos concatenados con el ID QC de la Pruebas en formato '99999'. - Crear directorio de Repositorios de Objetos (Sólo para uso de QTP, es decir Pruebas Funcionales) - Crear directorio de Bibliotecas de Funciones (Sólo para uso de QTP, es decir Pruebas Funcionales) - Crear directorio de Escenarios de Recuperación <p>Los scripts deben ser actualizados en QC cuando se quiera ejecutarlos y en otro momento no corresponderán necesariamente a la LB.</p> <p>NOTA GENERAL: EL ORDEN EN ESTE DIAGRAMA DE PROCESOS NO IMPLICA QUE UNA TAREA NO SE PUEDA VOLVER A REALIZAR EN UN MOMENTO POSTERIOR CON FINES DE REFINAR UNA DEFINICION YA EXISTENTE, TENIENDO EN MENTE NO OLVIDAR GESTIONAR LA LB, SI CORRESPONDE.</p>

Proceso	Definir pruebas unitarias
Descripción	<p>Una Prueba unitaria en el caso de Pruebas funcionales corresponderá a una unidad funcional indivisible y posiblemente derivable de un requerimiento de Pruebas (si éstos han sido detallados adecuadamente).</p> <p>A este nivel basta con definir la Prueba Unitaria, pero no indicar sus pasos.</p>

Proceso	Detallar pruebas unitarias
Descripción	<p>Esta tarea siempre se realiza para las Pruebas manuales y se evalúa su realización para el caso automático dependiendo del tiempo y del servicio otorgado. Por ejemplo, cuando no se realiza la ejecución por parte de la Organización, si se deberían documentar.</p> <p>Se ingresa la Prueba a QC en Test Plan ->; Design Steps, de acuerdo a la especificación entregada en el Diagrama "Diseño Prueba" para la "Prueba_Unitaria", "Paso_Documentado", "Escenario_Prueba_Unitaria" y "Parametro_Prueba_Unitaria" y "Regla".</p>

Proceso	Especificar necesidad datos PU
Descripción	<p>- Confeccionar documento que indique cual es la necesidad de datos en función de los parámetros que se identifiquen.</p> <p>- Especificar un atributo que permita particionar los escenarios entre los que consideran datos existentes en la Base de Datos y datos generados.</p> <p>- Generar y adjuntar a la Prueba en QC la Planilla Referencial de Datos siguiendo el formato de nombre D999999_01.xls donde los primeros 6 dígitos corresponden al ID generado en QC para la Prueba (Test Plan->; Seleccionar Test->; Tab Attachments). La planilla referencial Excel tiene una hoja con nombre "Global", cuya primera fila contiene los nombres de los parámetros identificados en el punto 1 y tiene tantas filas como escenarios se vayan a probar en la Prueba. Como regla general en el caso manual, debería haber una línea por escenario para hacer más simple la descripción de los pasos y el trabajo del Tester. En el caso automático puede haber más de un escenario manejado por el script.</p> <p>Observación: La planilla referencial Excel es usado como base para las planillas que contendrán los datos reales que se usarán para ejecutar las Pruebas, estas últimas planillas deben ser adjuntadas a los "Test Instance".</p>

Proceso	Definir pruebas integradas
Descripción	<p>Combina 2 o más pruebas unitarias dentro de una sola prueba</p> <p>Esta tarea siempre se realiza para las pruebas manuales y se evalúa su realización para el caso automático dependiendo del tiempo y del servicio otorgado. Por ejemplo, cuando no se realiza la ejecución por parte de organización, si se deberían documentar. (EN EVALUACIÓN DURANTE EL PILOTO)</p>

Proceso	Detallar pruebas integradas
Descripción	Se ingresa la Prueba a QC en Test Plan ->; Design Steps, de acuerdo a la especificación entregada en el Diagrama "Diseño Prueba" para la "Prueba_Integrada".

Proceso	Especificar necesidad datos PI
Descripción	<p>- Se debe especificar un atributo que permita particionar los escenarios entre los que consideran datos existentes en la Base de Datos y datos generados.</p> <p>- Generar y adjuntar a la prueba en QC la Planilla Referencial de Datos siguiendo el formato de nombre D999999_01.xls donde los primeros 6 dígitos corresponden al ID generado en QC para la prueba (Test Plan->; Seleccionar Test->; Tab Attachments). La planilla referencial Excel tiene una hoja de nombre Global para la Prueba unitaria guía y tantas hojas como Pruebas unitarias dependan de la guía en una relación uno es a muchos. Como ejemplo considere el caso de un Maestro-Detalle, el maestro es la Prueba unitaria guía y el detalle es la Prueba unitaria asociada, esta última tendrá una hoja propia.</p> <p>Observación: La planilla referencial Excel es usada como base para las planillas que contendrán los datos reales que se usarán para ejecutar las Pruebas, estas últimas planillas deben ser adjuntadas a los "Test Instance".</p>

5.1.4.2 Proceso “Automatizar prueba unitaria”

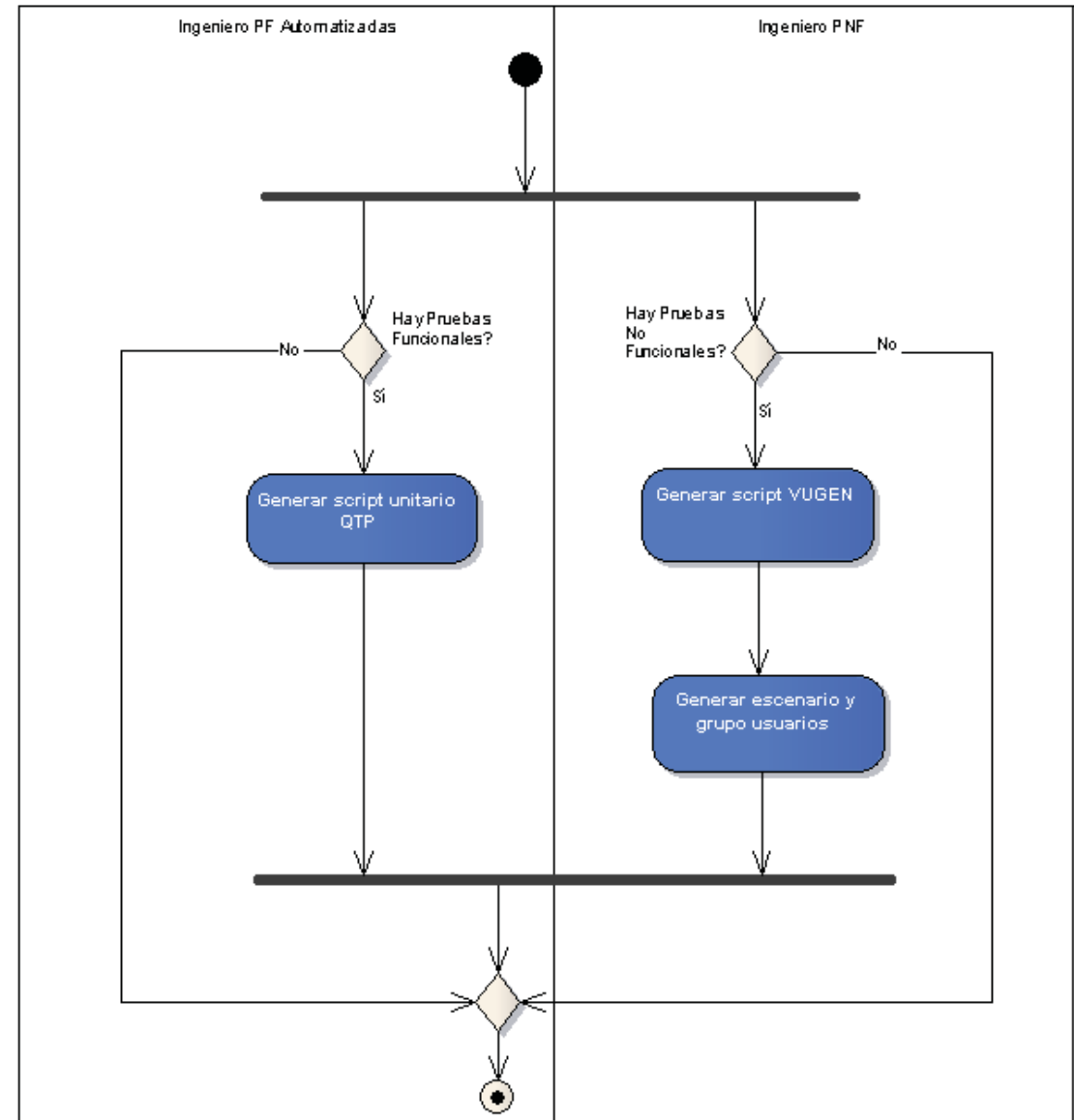


Figura 5.11: Proceso “Automatizar prueba unitaria”

Proceso	Generar script unitario QTP
Descripción	<ul style="list-style-type: none"> - Se graba un "Acción QTP" dentro del script, siguiendo los pasos (recordar que en QTP las validaciones de cualquier tipo son pasos) y parametrización detallada para la prueba en QC. Debido a que puede haber varios flujos de pasos (al menos el caso positivo y el negativo) probablemente se deba intervenir el script en su vista "Keyword-View" o "Script-View". - La "Acción QTP" debe informar siempre a través de Report.ReportEvent si la prueba fue exitosa o falló. - Al final el script se almacena en el directorio que corresponde para que sea versionado. - Se recomienda medir el tiempo de respuesta, pero sólo con el fin de tener una referencia, pues debido a que ni el Ambiente de Pruebas ni las Bases de Datos en él son representativos del Ambiente Productivo en régimen permanente.

Proceso	Generar script VUGEN
Descripción	<ul style="list-style-type: none"> - Se graba el login y el logout dentro de dos acciones. - Para cada transacción acordada con el Cliente como parte de la prueba se graba un script con una acción y dentro de ella se marca el inicio y el término de la "Transaction" la cual mide el tiempo sujeto a SLA. - A continuación se debe depurar el script revisando: Pacing, Log, Think Time, Network - Speed Simulation (útil cuando el ancho de banda disponible es mayor al real) y detección de Correlación y cerciorándose que el script se puede re-ejecutar exitosamente. - Dependiendo del tipo de aplicativo, agregar puntos de Verificación para determinar si la transacción está retornando adecuadamente, pues hay veces que la no disponibilidad no se manifiesta en la caída de la aplicación sino en comportamiento anormal.

Proceso	Generar escenario y grupo usuarios
Descripción	En el caso de Pruebas de Carga y Rendimiento Unitarias, usando LR Controller, se declara un escenario por cada script usando un solo grupo programando tramos de 1, 10, 20, 30 y 50 usuarios respectivamente. La duración de cada tramo debería ser al menos 5 minutos con el fin de ver la estabilidad de la respuesta del sistema, si bien tiempos mayores son recomendables, su uso dependerá del tiempo total asignado a esta prueba.

5.1.4.3 Proceso “Automatizar prueba integrada”

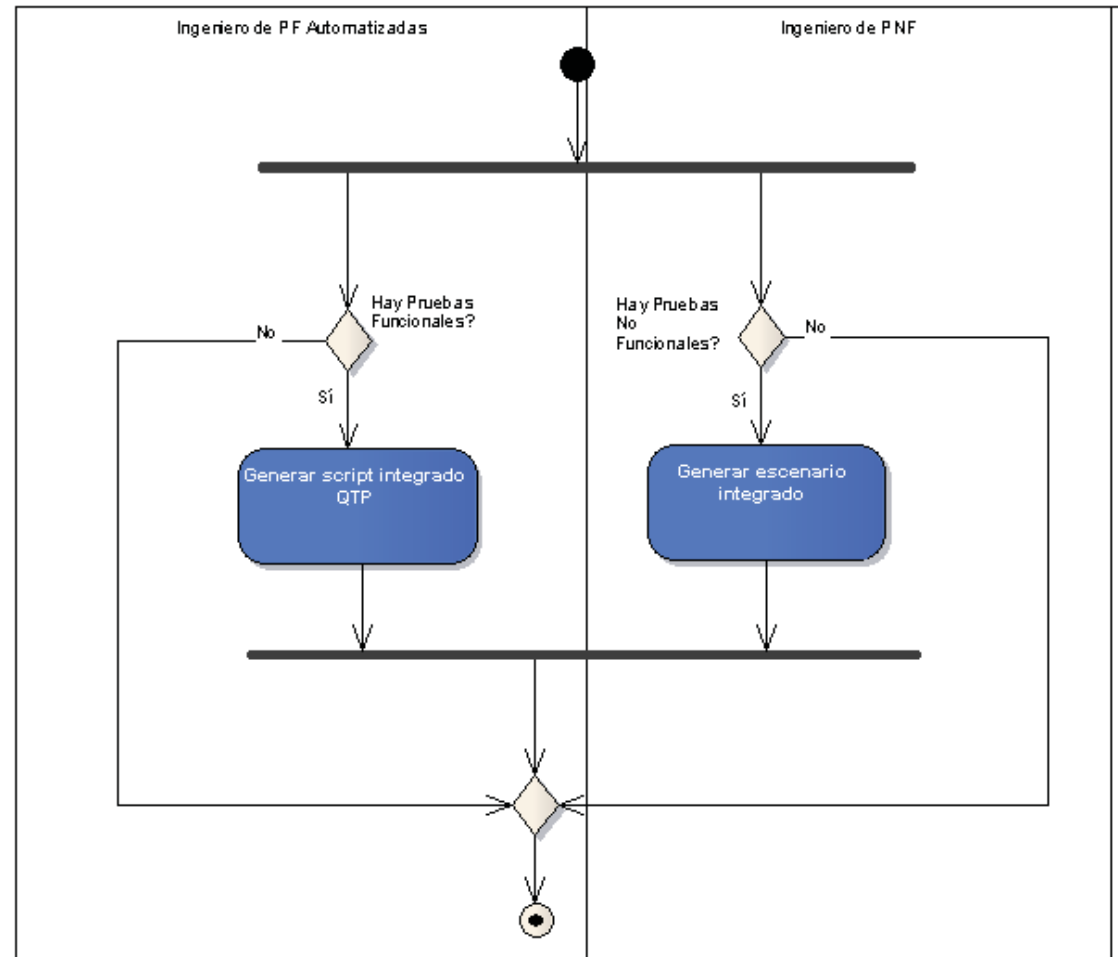


Figura 5.12: Proceso “Automatizar prueba integrada”

Proceso	Generar script integrado QTP
Descripción	<p>En este punto ya se han generado las "Acciones QTP" para las pruebas unitarias, entonces:</p> <ul style="list-style-type: none"> - Se crea un script en el cual se referencia todas aquellas acciones que componen la "Prueba Integrada" siguiendo el flujo definido. - Se declara una planilla de datos por acción, la cual deberá ser llenada.

Proceso	Generar escenario integrado
Descripción	Se genera en Load Runner Controller, el escenario integrado definiendo todos los Grupos de Usuarios que se haya contemplado en la prueba y usando los Script ya generados en VUgen para las Pruebas Unitarias.

5.1.4.4 Proceso "Generar LB datos"

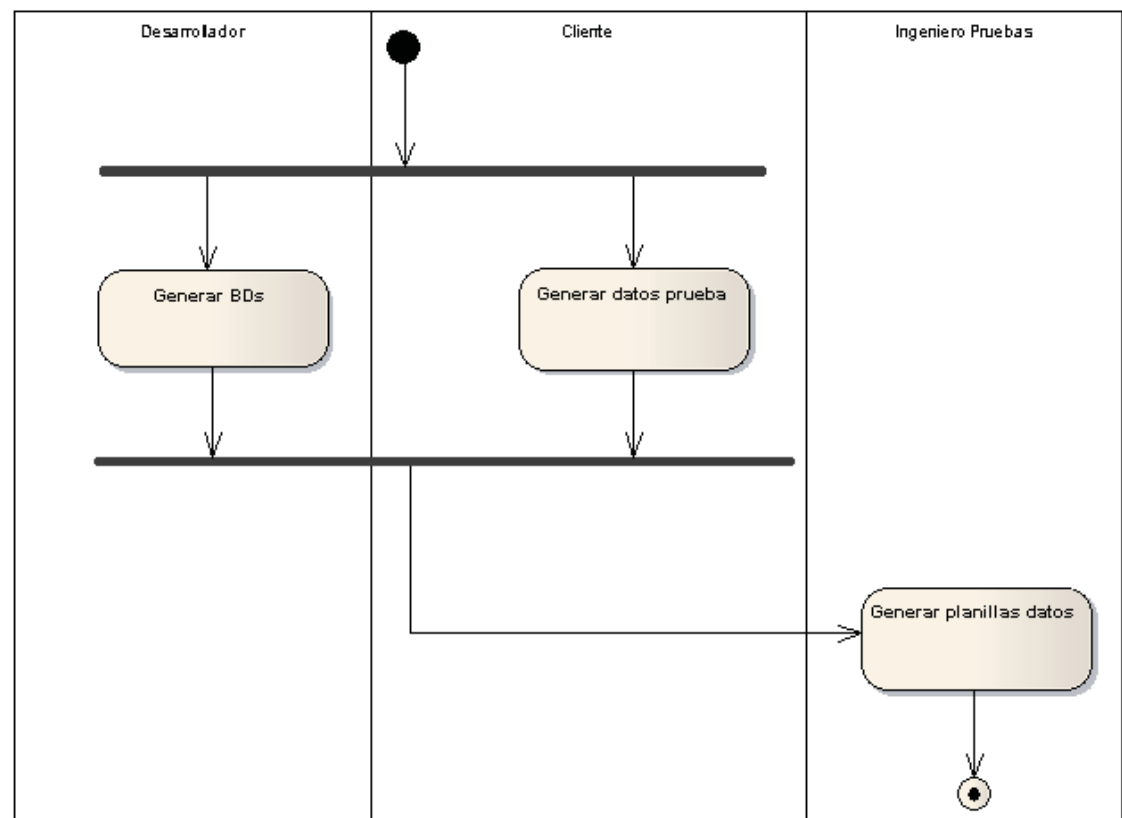


Figura 5.13: Proceso "Generar LB datos"

Proceso	Generar BDs
Descripción	Esta generación depende del tipo de pruebas y se deben considerar los siguientes aspectos: <ul style="list-style-type: none">- Las pruebas no funcionales requerirán un tamaño representativo del aplicativo en régimen permanente.- Se debe velar por la confidencialidad de los datos.- Los procedimientos de Respaldo y Restauración deben existir y estar probados antes del inicio de las pruebas, esto es necesario debido a que muchas veces se necesitan realizar las pruebas más de una vez usando la misma Release del SW. Ejemplo: Se hicieron pruebas no funcionales en las cuales se detectaron problemas en el HW, los cuales fueron resueltos y se requiere probar de nuevo.

Proceso	Generar datos prueba
Descripción	Generación a través de consultas SQL o Recolección de Datos en la cantidad necesaria para realizar las pruebas.

Proceso	Generar planillas datos
Descripción	Usando las Planillas Excel Referenciales adjuntadas a cada prueba durante las actividades "Especificar Necesidad Datos PU" y "Especificar Necesidad Datos PI", generar las Planilla Excel para las pruebas que serán realizadas durante el ciclo en curso.

5.1.4.5 Proceso “Ejecutar ciclo pruebas”

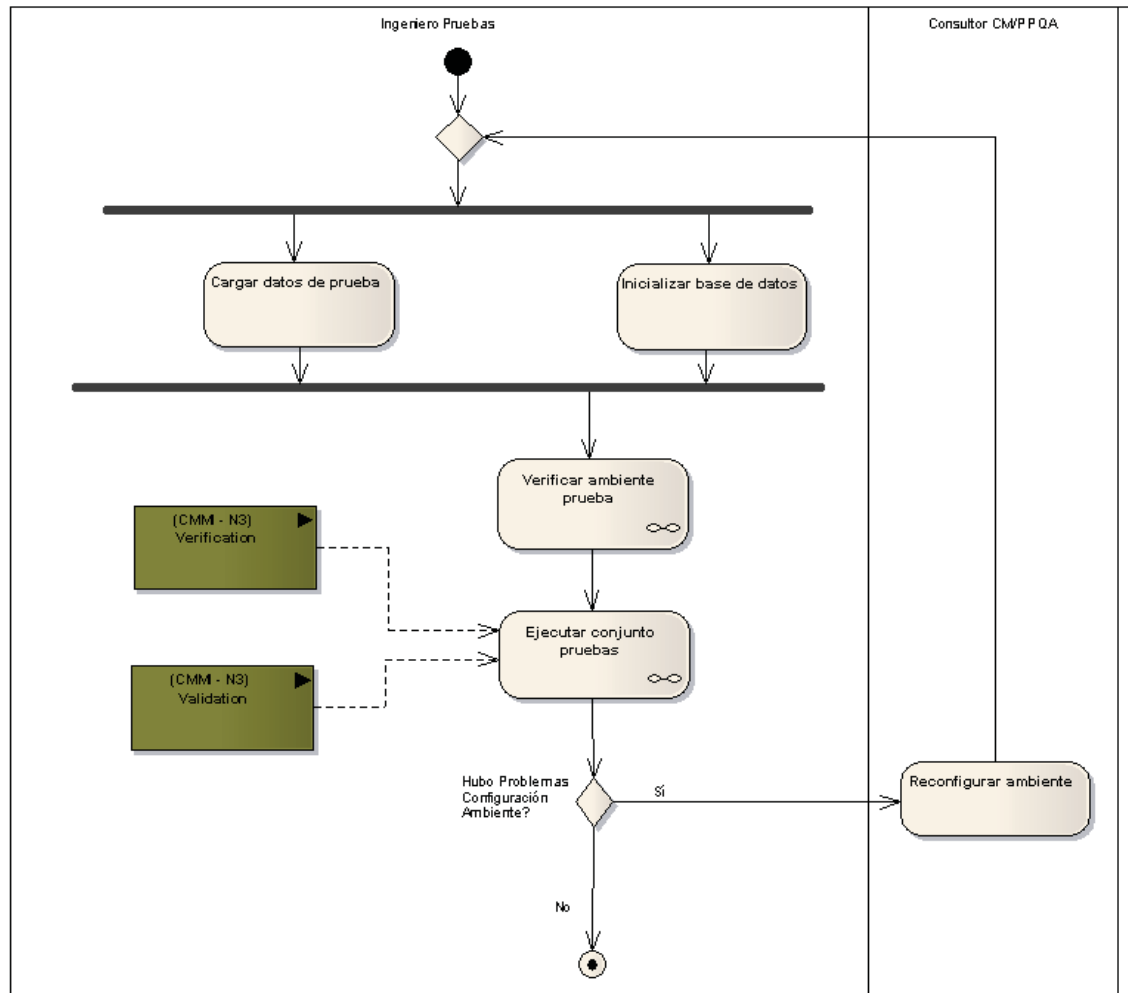


Figura 5.14: Proceso “Ejecutar ciclo pruebas”

Proceso	Cargar datos de prueba
Descripción	Cargar las Planillas de Datos de Prueba generadas en "Generar Planillas Datos" en las "Test Instance" del ciclo.

Proceso	Inicializar base de datos
Descripción	Inicializar la(s) base(s) de datos es setear esta(s) misma(s) en la condición inicial para que comience el Ciclo correspondiente, lo que implica: <ul style="list-style-type: none">- Datos Necesarios y consistentes con los datos que se usarán en las pruebas.- Para el caso no funcional, además se requiere que la(s) Base(s) de Datos tenga(n) un tamaño representativo al que tendría en el estado permanente en producción.

Proceso	Ejecutar conjunto pruebas
Descripción	Se ejecuta cada prueba dentro del Conjunto de Pruebas ya sea manual o automáticamente. El primer Conjunto de Pruebas a ejecutar debería ser algún tipo de "Pruebas Trazadoras". En el caso de Pruebas No Funcionales, es altamente recomendable que las Pruebas de Carga y Rendimiento Unitarias se ejecuten al menos 3 veces, para tener una confiabilidad mínima en la medición de tiempos y comportamiento del sistema.

5.1.4.6 Proceso “Verificar ambiente prueba”

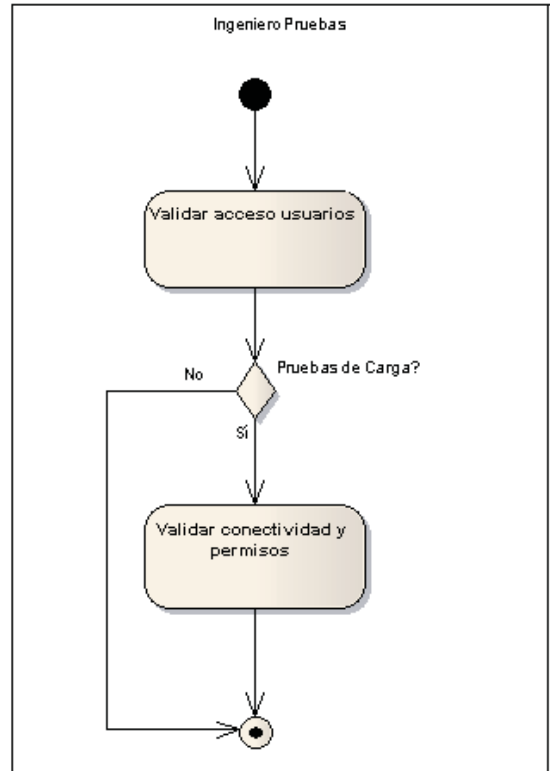


Figura 5.15: Proceso “Verificar ambiente prueba”

Proceso	Validar acceso usuarios
Descripción	Validar que sólo las personas responsables de las pruebas tengan acceso al sistema.

Proceso	Validar conectividad y permisos
Descripción	<p>En el tab Run del LR Controller validar:</p> <ul style="list-style-type: none"> - Conectividad del Controller a Generadores de Carga (Icono Load Generators). - Conectividad y permisos de monitoreo desde el Controller a los servidores del aplicativo y de BDs. (Al agregar las métricas de Windows Resources).

5.1.4.7 Proceso “Ejecutar conjunto de pruebas”

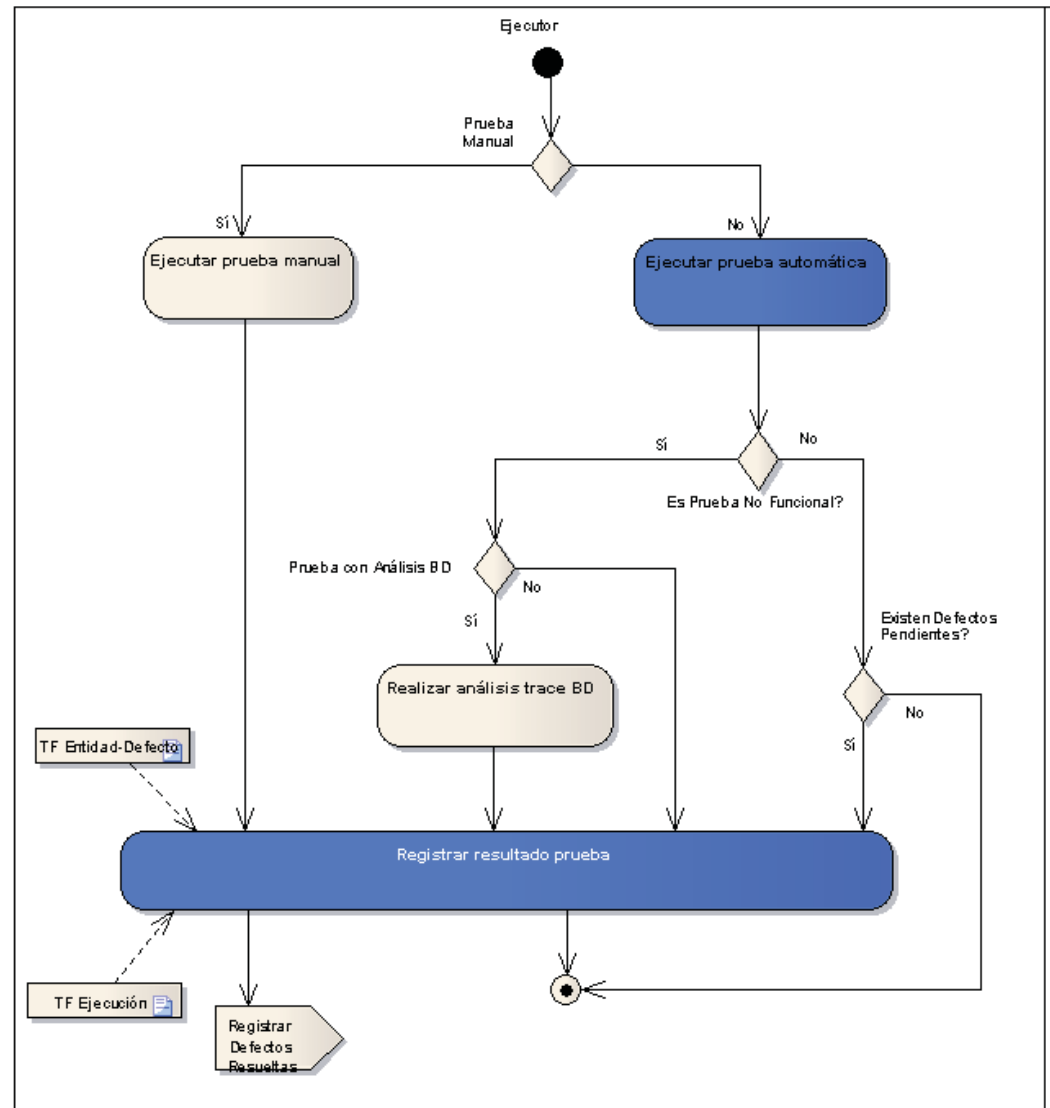


Figura 5.16: Proceso “Ejecutar conjunto de pruebas”

Proceso	Ejecutar prueba automática
Descripción	Para el Caso de Pruebas Funcionales se debe setear en QTP que registre automáticamente cualquier error como defecto.

Proceso	Registrar resultado prueba (*)
Descripción	<p>Una vez que el Analista/Ejecutor ejecuta las pruebas diseñadas, debe ir dejando registro de dichas pruebas en la planilla de “TF Ejecución” (Ver ANEXO1), la cual irá dejando la evidencia de la ejecución.</p> <p>En el caso que la prueba obtenga estado NOK, se debe registrar en la planilla "TF Entidad-Defecto" (Ver ANEXO 1), todos los defectos encontrados en la prueba.</p> <p>Al momento de ejecutar, basta con que un paso no sea el esperado para que el caso de prueba total obtenga el estado NOK, sin embargo, que un paso sea NOK no implica necesariamente que no se puedan seguir testeando los demás pasos.</p>

5.1.5 Proceso “Administrar proceso de pruebas” PARTE 3

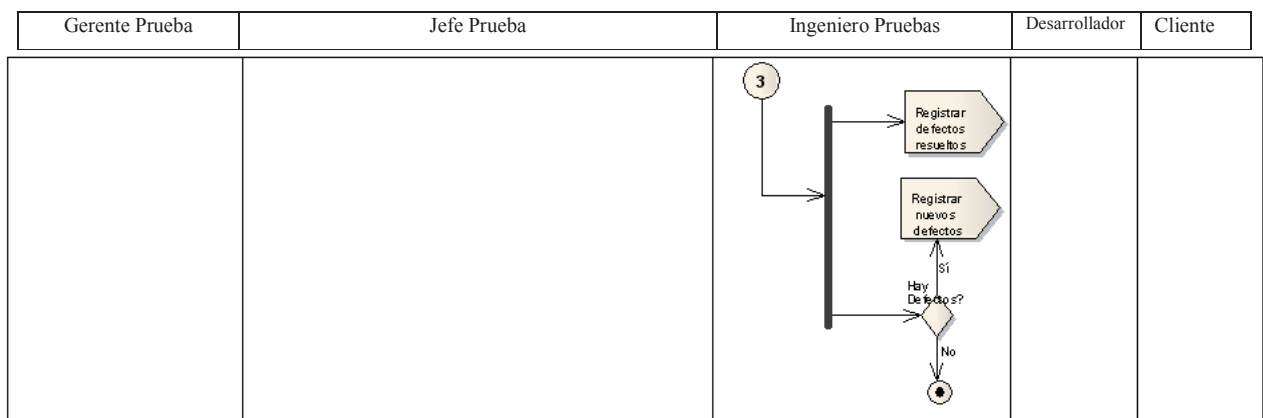


Figura 5.17: Proceso “Administrar proceso de pruebas” PARTE 3

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.6 Proceso: “Gestionar defectos”

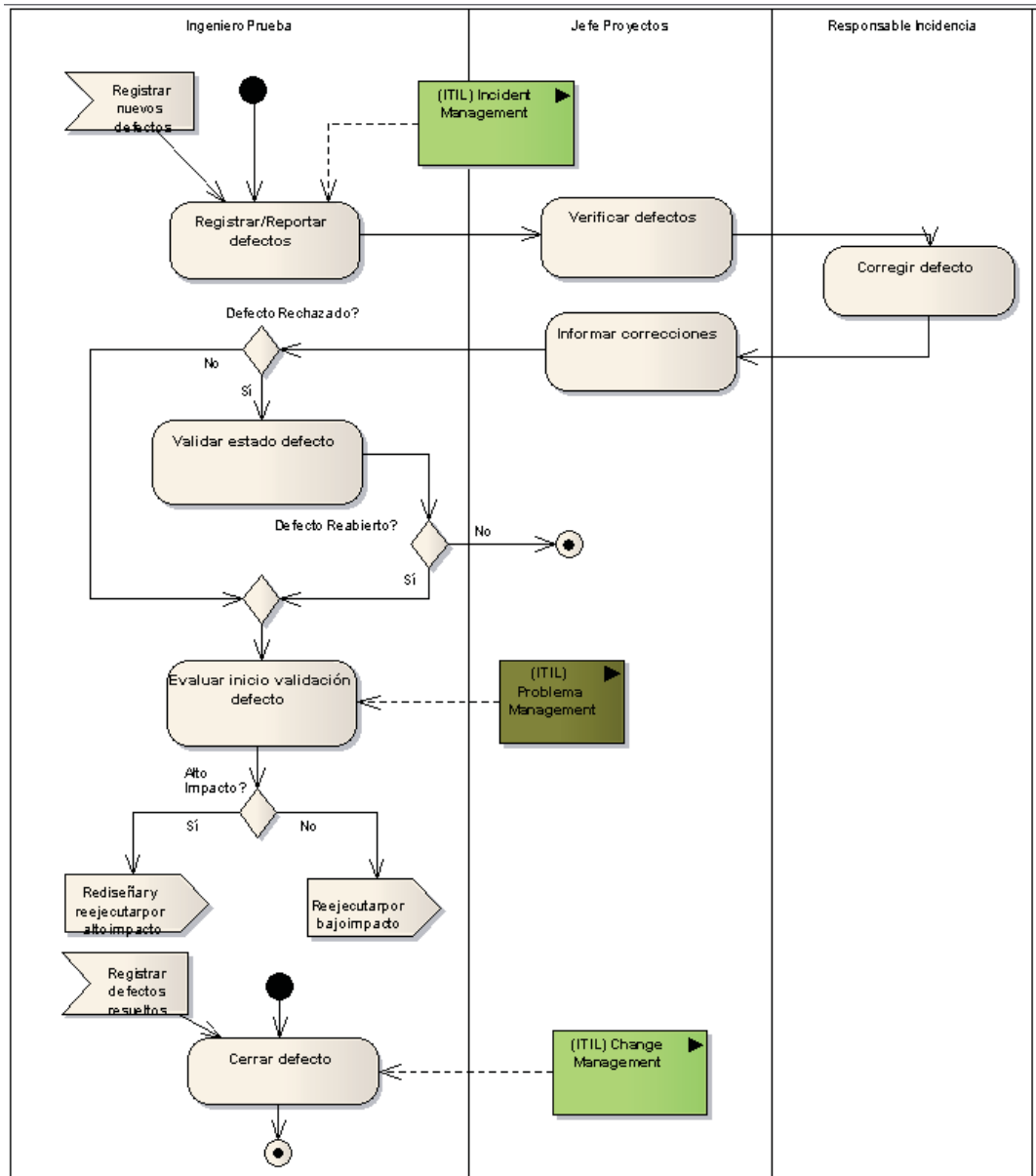


Figura 5.18: Proceso “Gestionar defectos”

Proceso	Registrar/reportar defectos
Descripción	<p>- Ingresar los atributos del defecto, es importante el ingreso del campo Área Calidad (CM, PPQA, TF, TNF) con el fin de cada área pueda posteriormente identificar que defectos debe gestionar. También no se debe identificar al "Responsable de Solución" que puede ser el Desarrollador, Responsable HW, Responsable SW, Responsable Redes.</p> <p>- Vincular las entidades que gatillan el defecto, recuerde que pueden ser las siguientes:</p> <ul style="list-style-type: none"> • Requerimiento • Prueba • Conjunto de Pruebas • Instancia de Pruebas • Ejecución de Pruebas • Paso de Pruebas (En el caso que se esté ejecutando una prueba, lo más recomendable vincular al paso el defecto y adjuntar la evidencia a este último) • A otro defecto (Por deducción) <p>Para que el defecto sea procesado como tal, se lo debe dejar en estado "Open".</p> <p>Reportes de Defectos</p> <p>- CM y PPQA, en Customize->; Excel Report Generator->; CM-PPQA Matriz No Conformidades.</p>

Proceso	Corregir defecto
Descripción	<p>El Jefe de Proyecto o el equipo de desarrollo evalúan si corresponde a un defecto, y de serlo inicia su corrección.</p> <p>Si el defecto no se acepta, se debe setear en estado "Rejected".</p> <p>Si el defecto es aceptado, en el momento de resolverlo se debe setear el estado "Fixed".</p>

Proceso	Validar estado defecto
Descripción	<p>Validar si el estado del defecto rechazado aplica o no. Si aplica, poner el defecto en estado "Closed", en caso contrario en "Reopen" y enviárselo al Desarrollador y/o Jefe de Proyecto.</p>

Proceso	Evaluar inicio validación defecto
Descripción	En función del impacto se decide si modificar el diseño o simplemente reagendar la prueba, el cual es realizado una vez que el Ciclo actual se encuentra terminado: <ul style="list-style-type: none">- Alto impacto => Revisar Diseño y reejecutar algunos "Conjuntos de Pruebas"- Bajo Impacto => Reejecutar algunos "Conjuntos de Pruebas"

Proceso	Cerrar defecto
Descripción	En el caso que un defecto haya sido resuelto, setear el estado "Closed", esto debería ser hecho en el ciclo que ocurre.

5.1.7 Proceso: “Realizar auditoría CM/PPQA”

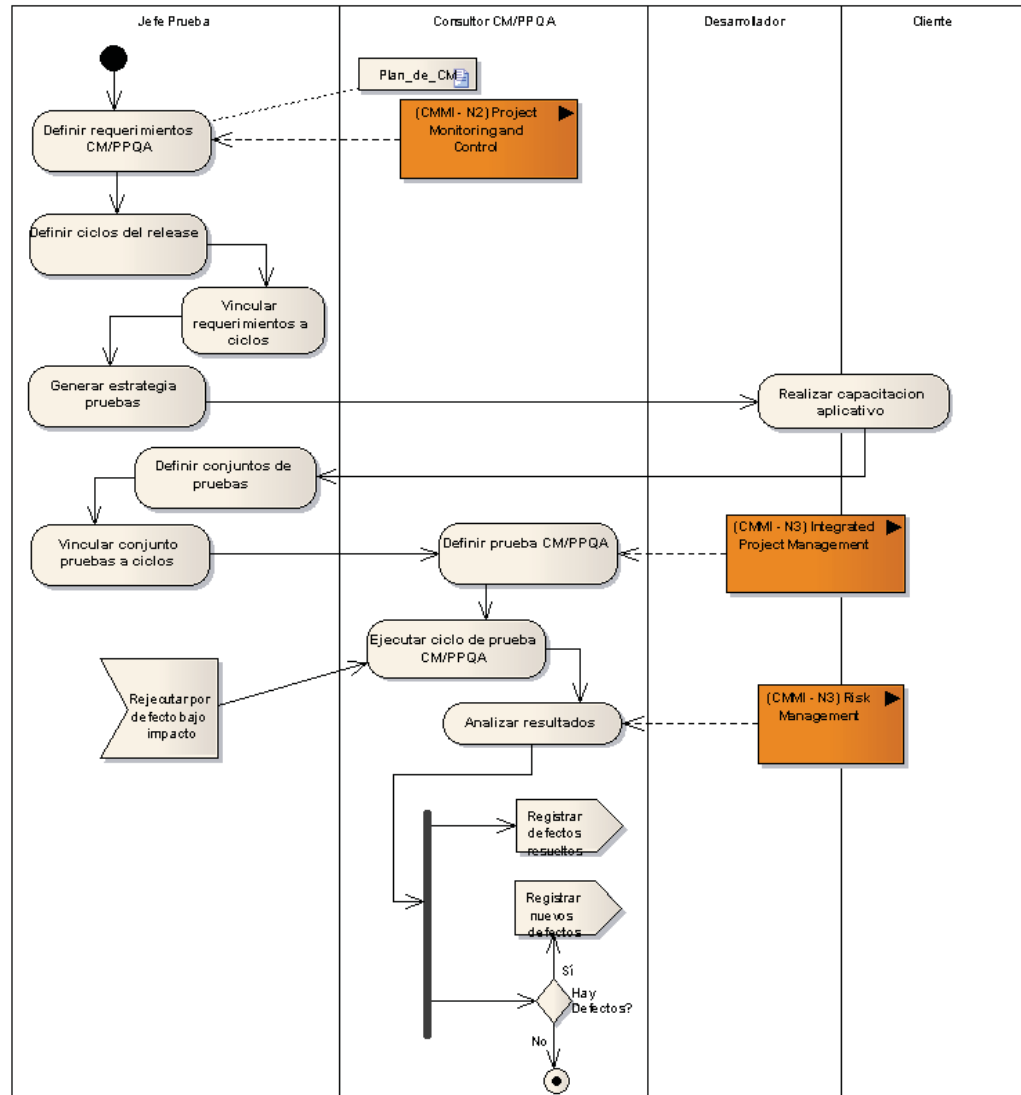


Figura 5.19: Proceso “Realizar auditoría CM/PPQA”

Proceso	Definir requerimientos CM/PPQA (*)
Descripción	<p>La planilla de apoyo "Plan de CM" se encuentra en el ANEXO 1. Dicha planilla apoya a la planificación de perfilamientos y definición de la biblioteca de configuración de los ítems.</p> <p>Propone una estructura base, especificando que debiese tener cada una de ellas.</p> <p>Se definen los requerimientos de CM/PPQA detallando los atributos mencionados a continuación, si es necesario se crean carpetas que agrupen los requerimientos por temas.</p> <ol style="list-style-type: none"> 1. Nombre 2. Especificar Tipo de Requerimiento como de Negocios (CM o PPQA) 3. Descripción 4. Autor 5. Prioridad en la implementación: <ol style="list-style-type: none"> 1-Baja 2-Media 3-Alta 4-Muy Alta 6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)

Proceso	Definir ciclos del release
Descripción	<p>Definir los "Ciclos del Release" indicando los "Conjunto de Pruebas" preliminares que los conforman, por ejemplo el primer "Ciclo de Pruebas" debiera incluir siempre un conjunto de "Pruebas Trazadoras" con el fin de establecer si el ambiente está operativo o no.</p> <p>En QC es posible realizar Attachments de información del Ciclo, como podría ser la planificación del mismo. Pero aparentemente estos elementos no son versionados, por lo que quizá es más recomendable sólo adjuntar Links a Subversion.</p>

Proceso	Vincular requerimientos a ciclos
Descripción	<p>Se debe vincular los requerimientos a al menos un "Ciclo de Prueba" con el fin de:</p> <ul style="list-style-type: none"> - Determinar en cualquier instante si los requerimientos están siendo cubiertos o no - Generar esbozo de pruebas en función de los requerimientos asociados a un ciclo. <p>Un requerimiento puede ser vinculado a un Release, en este caso, todos los "Ciclos de Prueba" dentro del Release serán vinculados, y eso implica que debe haber un "Conjunto de Pruebas" al menos dentro de cada Ciclo que valida el requerimiento.</p> <p>Por el momento, no se realizan Pruebas de Regresión al 100% en todos los ciclos por motivos de</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	recursos. Así la cobertura de prueba en un ciclo queda definida por la liberación de funcionalidades nuevas y por los defectos ocurridos en el ciclo anterior (en el caso que exista), por lo tanto al principio del proyecto, sólo se vincularán los requerimientos asociados a funcionalidades nuevas al ciclo en que se prueban, y con posterioridad dependiendo de los defectos se vincularán requerimientos a ciclos que contengan Pruebas de Regresión.
--	---

Proceso	Generar estrategia pruebas
Descripción	<p>Este documento debe ser un compendio del estado del proyecto hasta este momento, considerando:</p> <ul style="list-style-type: none"> - Roles y personas - Requerimientos de Pruebas - Ciclos de Pruebas - Protocolo de Suspensión de las Pruebas - Conclusiones de Estudio Factibilidad de Automatización. - Referencias a los documentos de Arquitectura de HW/SW. - Formato Plan de Pruebas o Operatoria al respecto - Formato Informe de Resultados de Pruebas. - Manejo de Defectos. - Severidad de Defectos. - Riesgos <p>En el caso de CM-PPQA se incluyen los checklists que se utilizarán en la Revisión.</p>

Proceso	Realizar capacitación aplicativo
Descripción	Realizar Capacitación del Aplicativo, idealmente se debiera hacer una Capacitación para Pruebas Funcionales y No Funcionales, pero podría haber casos donde tenga que ser realizada en sesiones separadas.

Proceso	Definir conjuntos de pruebas
Descripción	<p>Se definen los conjuntos de prueba, ver ejemplos de estos conjuntos en Diagrama Conjunto_Prueba.</p> <p>En QC, Test Lab:</p> <ul style="list-style-type: none"> - Crear carpeta(s) que contienen "Conjuntos de Pruebas". - Crear cada "Conjunto de Pruebas" en la carpeta que corresponda, e ingresar las "Pruebas" que lo componen usando el tab "Execution Grid". El orden de ejecución se define en el tab "Execution Flow".

Proceso	Vincular conjunto pruebas a ciclos
Descripción	En QC, Test Lab, las carpetas que contienen "Conjuntos de Pruebas" son linkeadas a ciclos. Para esto click derecho sobre la carpeta y "Assign to Cycle".

Proceso	Analizar resultados
Descripción	Esta tarea consiste en determinar si los defectos reportados durante las pruebas son tales. - En el caso que el defecto sea tal, actualizar su estado a "Open". - En el caso que el defecto no aplique, se puede borrar o dejar su estado en "Closed", esto último sólo si es de algún aporte mantener el registro.

El ciclo de los defectos o incidentes que se explican en los Modelos anteriores es el siguiente:

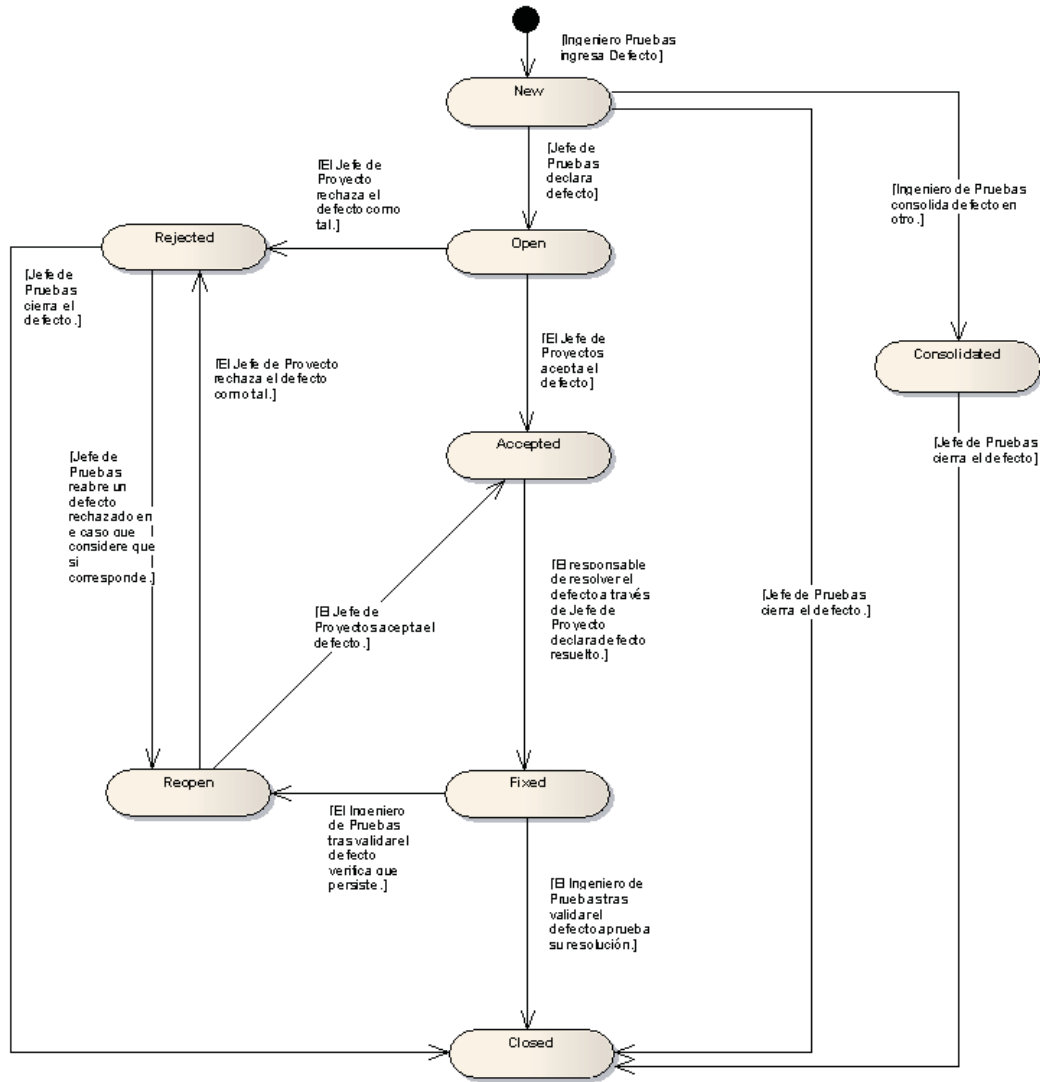


Figura 5.20: “Ciclo de defectos”

5.1.8 Proceso: “Controlar proyecto”

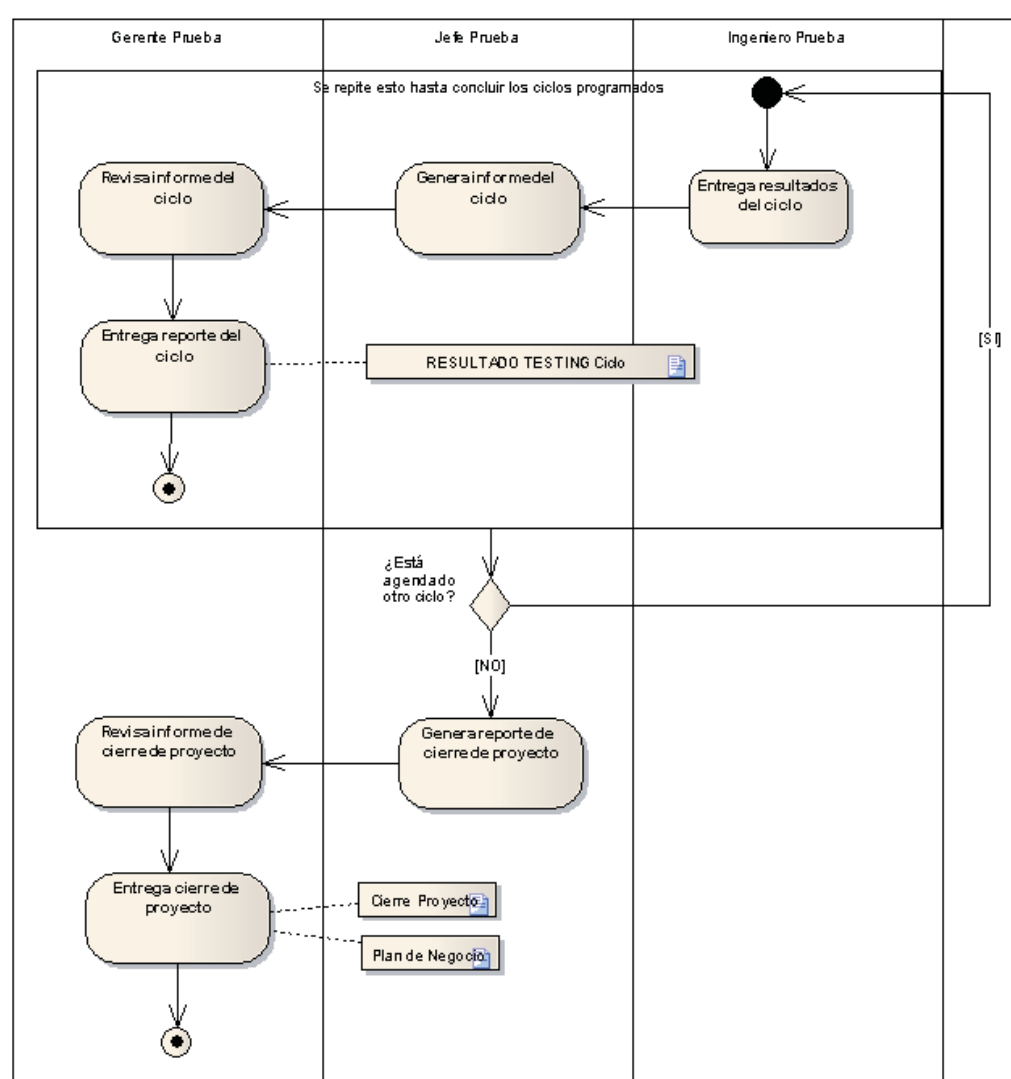


Figura 5.21: Proceso “Controlar proyecto”

Proceso	Entregar resultados del ciclo
Descripción	Una vez ejecutadas las pruebas el Ingeniero de Pruebas entrega los resultados de la ejecución y de los defectos encontrados.
Proceso	Generar informe del ciclo
Descripción	Con la entrega de los resultados el Jefe de Proyecto construye el Informe del Ciclo y lo entrega al Gerente de Pruebas para su revisión
Proceso	Revisar informe del ciclo
Descripción	Cuando el Gerente de proyecto revisa el informe lo libera al equipo de Proyecto para su análisis y revisión en conjunto al equipo de Testing.
Proceso	Entrega reporte del ciclo (*)
Descripción	El Reporte que se genera se basa en la planilla de apoyo “RESULTADO TESTING Ciclo” ANEXO 1. Dicho informe se basa esencialmente en la entrega de los resultados en comparación a las políticas de paso a producción planteadas en la planilla de apoyo “Plan de Pruebas”.
Proceso	Generar reporte de cierre de proyecto (*)
Descripción	En el caso que se terminen de ejecutar todos los ciclos programados o solicitados por emergencia es la hora que el Jefe de Proyecto Genere el reporte de Cierre de Proyecto utilizando la planilla de Apoyo “Cierre Proyecto” y “Plan de Negocio” del ANEXO 1. Luego de generado el reporte, es entregado al Gerente de Proyecto.
Proceso	Revisar informe de cierre de proyecto
Descripción	El Gerente de Proyecto revisa el Informe de Cierre de Proyecto y el Plan de Negocio entregado por el Jefe de Proyecto con el fin de evaluar las diferencias entre lo programado y lo real.
Proceso	Entrega cierre de proyecto
Descripción	Una vez realizada la revisión por el Gerente de Proyecto los entrega y presenta al equipo de proyecto, dejando en conocimiento la desviación u optimización en el caso que exista en el Plan de Negocio junto a los resultados finales de la intervención del Testing.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.2 Resumen de resultados obtenido (Métricas)

A partir de las métricas definidas inicialmente se procede a evaluar cada una de ellas con el fin de poder analizar cómo se comportó la aplicación de la Metodología aplicada al proyecto.

Nº	Medición	Medida	Métrica
1	Se medirá el cumplimiento del esfuerzo establecido para el Testing del producto.	HH	Esfuerzo Estimado v/s Esfuerzo Real
2	Se medirá el cumplimiento del costo establecido para el Testing del proyecto.	UF	Costo Estimado v/s Costo Real
3	Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.	%	% Establecido v/s % Obtenido

- 1) Métrica 1: Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto.

En base al Modelo de Servicio se obtienen los siguientes datos:

Días	HH	
2	13	Jefe de Área
14	112	Jefe de Proyecto
27	218	Externos
43	343	Total Esfuerzos

Según los datos obtenidos del Plan de Negocio se tiene lo siguiente:

Mes 2	Mes 3	Mes 4	
Ago-09	Sep-09	Oct-09	
5.00 hh	11.00 hh	6.50 hh	
90.00 hh	140.00 hh	101.00 hh	
			Total HH planificadas
95.00 hh	151.00 hh	107.50 hh	353.50 hh

Es decir, se planificaron 10 horas más de las inicialmente estimadas.

Lo que realmente se gasta en HH son las siguientes:

Mes 2	Mes 3	Mes 4	
Ago-09	Sep-09	Oct-09	
22.00 hh	24.80 hh	58.00 hh	
72.00 hh	214.70 hh	99.00 hh	
		34.00 hh	
	40.00 hh	9.00 hh	
			Total HH Reales
94.00 hh	279.50 hh	200.00 hh	622.50 hh

Existe una diferencia de 269 HH según la estimación inicial y lo real.

- 2) Métrica 2: Se medirá el cumplimiento del costo establecido para el Testing del proyecto.

En base al Modelo de servicio se obtienen los siguientes datos:

UF	
14	Costo Jefe de Área
107	Costo Jefe de Proyecto
194	Costo Externos
315	Total Costo Proyecto

Según los datos obtenidos del Plan de Negocio, se obtiene lo siguiente:

APROBADO TOTAL COSTO SERVICIO TF	446.33 UF
REAL TOTAL COSTO SERVICIO TF	541.06 UF
DIFERENCIA	-94.73 UF

3) Métrica 3: Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.

Los criterios de aceptación establecidos para este proyecto fueron los siguientes (ver planilla aplicada Plan de Pruebas) son:

- 0% de los Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad INVALIDANTE.
- A lo más, 5% de Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad ALTA.
- A lo más, 10% de Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad Media o Baja.
- A lo menos, 85% de Casos de Prueba Integrados efectivamente probados con estado OK.

Los resultados obtenidos en el ciclo 1 ejecutado son los siguientes:

En base a la necesidad de acortar el ciclo de Testing, finalmente se probó un 62% de las pruebas inicialmente planificadas; es decir, 241 CPI de las 387 CPI planificadas.

El resultado de la ejecución de los 241 CPI arroja los siguientes resultados:

Failed	Passed	Total General	
84	157	241	CPI
35%	65%	100%	Porcentajes

Existen 146 CPI con estado N/A, estos casos son los que NO se ejecutaron en el primer ciclo.

De lo anterior se desprende que el aplicativo tiene un **Nivel de Madurez del 65%** sobre los casos de pruebas **ejecutados**, con los siguientes defectos encontrados:

Cuenta de Criticidad	Criticidad				Total general
	1-Baja	2-Media	3-Alta	4-Invalidante	
Total	14	23	62	7	106
Porcentaje	35%		58%	7%	100%

Se recomienda realizar un CICLO 2 de pruebas, para aclarar si las incidencias encontradas en el ciclo anterior, han sido corregidas o corresponden a errores en la instalación del software, en el ambiente de Testing utilizado. Además no se cumplieron los criterios establecidos en el Plan de Pruebas para el paso a producción.

Los resultados obtenidos en el ciclo 2 ejecutado se describen a continuación:

Se indica la estrategia usada en el ciclo 2 de Certificación del proyecto que considera la redefinición del total de pruebas definidas en el Plan de Pruebas inicialmente aprobado, en base a los cambios solicitados por el proyecto (inicialmente en el ciclo 2 se habían definido 194 CPI a ejecutar).

En base a la nueva estrategia que apunta a minimizar los riesgos de no completar las pruebas del ciclo 1, tal como estaban planificadas, en el ciclo 2 se definió la ejecución de 215 Casos de Prueba que estaban compuestos de la siguiente manera:

Failed	Passed	Total general	
65	150	215	CPI
30%	70%	100%	Porcentajes

De lo anterior se desprende que el aplicativo tiene un **Nivel de Madurez del 70%** sobre el total de casos de pruebas **ejecutados**, con la siguiente distribución de los defectos encontrados:

Cuenta de Criticidad	Criticidad				
	1-Baja	2-Media	3-Alta	4-Invalidante	Total general
Total general	2	27	52	9	90
Porcentajes	32%		58%	10%	100%

Se recomienda realizar un CICLO 3 de pruebas, ya que el Nivel de Madurez obtenido no es suficiente para el Paso a Producción del aplicativo. Además no se cumplieron los criterios establecidos en el Plan de Pruebas para el paso a producción.

Pese a lo anterior, el equipo de proyecto define pasar igualmente el aplicativo a producción con los antecedentes de cuáles son los errores detectados para así declararlos a los clientes finales y realizar seguimiento de su resolución.

Los puntos importantes que se rescatan de la entrevista final con la persona que dirigió la aplicación de la Metodología fueron los siguientes:

- 1) Si en el Modelo se tiene claridad de los costos asociados a cada recurso se puede lograr con más fineza el cálculo de la métrica final.
- 2) Las tasas de rendimiento incorporadas en el Modelo de Servicio distaron mucho de lo real; la tasa de rendimiento de los diseños no fue 5 CPI por hora, lo real se acercó al 2.5 por hora.
- 3) Las tasas de rendimiento incorporadas en el Modelo de Servicio de la ejecución de las pruebas no fue 5 CPI por hora, sino que fue de 3 CPI por hora.
- 4) Además es importante mencionar que los costos utilizados en el Modelo Inicial son distintos a los que finalmente se obtuvieron en el Plan de Negocio. Dichos costos fueron finalmente de 1.01 UF HH del Jefe de Proyecto y de 0.88 UF HH para los Externos, Analistas y Ejecutores.
- 5) A pesar de lo antes expuesto, el proceso fue limpio y claro de aplicar. Se siguiere generar una BD de conocimiento de proyectos con el fin de ir refinando cada vez más la estimación.

En base a las observaciones entregadas anteriormente se realiza el ejercicio de ajustar el Modelo de servicio inicial, obteniendo los resultados que se muestran en la Figura 5.22.

Variables					
Universo de Pruebas	2024	CPI	Cantidad de Gestión LB CM	10	Cantidad
Cobertura	85	Universo Funcionalidades	HH invertidas Gestión LB en TF	1	HH
	58	Funcionalidades a cubrir			
	89	% sobre universo	Tiempo Total de Proyecto	0	Meses
Profundidad	20	% sobre universo		0	Semanas
				0	Días hábiles
Cantidad de Ciclos	405	CPI Ciclo 1 (100%)	Costo total del proyecto	1106	UF
	202	CPI Ciclo 2 (50%)	Costo máximo Propuesta TF	332	UF
Diseños	405	CPI			
Ciclo Extra	202	CPI Ciclo N (50%)	N° de Entregas de SW	1	Entregables de SW
HH Mensuales	160	Horas mes			
HH trabajo Diario Interno	8	HH Dia			
HH trabajo Diario Externo	8	HH Dia			
Jefe de Área			Costos Jefe de Área	161	UF Mes
Cantidad	1			40,3	UF Semana
% Dedicación	5,00%	%		8,05	UF Dia Hábil
Horas mensuales	0,08	HH Mensuales		1,01	UF HH
Costo Mensual					
Costo HH					
Jefe de Proyecto			Costos Jefe de Proyecto	141	UF Mes
Cantidad	1			35,3	UF Semana
% Dedicación	100,00%	Real 50%		7,05	UF Dia Hábil
Horas mensuales	1,6	HH Mensuales		0,88	UF HH
N° Analistas/Ejecutores			Costos HH Externos	141	UF Mes
Cantidad	1			35,2	UF Semana
% Dedicación	100,00%	%		7,04	UF Dia Hábil
Horas mensuales	160	HH Mensuales		0,88	UF HH
Tasa de Rendimiento Diseño	2,5	CPI por HH por Recurso			
	20	CPI por HH trabajo Diario Externo			
Tasa de Rendimiento Ejecución	3	CPI por HH por Recurso			
	24	CPI por HH trabajo Diario Externo			

Figura 5.22: Ajuste del modelo en las variables definidas

En función de los nuevos ajustes se obtienen los costos que se muestran en la Figura 5.23.

Costos Estimación Proyecto			
Cantidad de Pruebas	Días	HH	
	56	452	Total Esfuerzo Proyecto
	2	19	Jefe de Área
	24	189	Jefe de Proyecto
	49	389	Externos
		UF	
		20	Costo Jefe de Área
		167	Costo Jefe de Proyecto
		342	Costo Externos
		528	Total Costo Proyecto

Figura 5.23: Costos por ajuste

De lo anterior se desprende la tabla comparativa (Figura 5.24):

Iteración	Métrica 1: Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto			Métrica 2: Se medirá el cumplimiento del costo establecido para el Testing del proyecto.		
	Planificado	Real	% Desviación	Planificado	Real	% Desviación
Modelo de Servicio Inicial	353.50 HH	622.50 HH	43.2	446.33 UF	541.06 UF	17.5
Modelo de Servicio Ajustado	597 HH	622.50 HH	4.1	528 UF	541.06 UF	2.4

Figura 5.24: Tabla comparativa de las métricas con y sin ajustes

Como se puede ver en la tabla anterior (Figura 5.24), el Modelo se puede ajustar sólo si se precisan los datos de entrada del Modelo de Servicio. Mientras más exactos, más es posible acercarse al real esperado del proyecto.

La métrica de Apego a la Metodología se calcula a continuación:

A	B	C	D	E	F	G	H	I	J	K	L		
Estructura	Plantilla	Propuesta Metodológica	Importancia en Metodología	Apego	Aplicada	Apego por Aplicada	Distribución Importancia	Refinada	Multipl. (Distribución * Refinado)	Apego por Fase	Porcentaje por Estructura		
0. Entradas	Informe de Diseño proveedor(Extracción).doc	CMMI	3	4	12	1	12	0,40	NO	0,9	0,36	0. Entradas	
	PlanificacionSeguimientoSemanal.mpp	CMMI	3	4	12	1	12	0,40	SI	1	0,40		
	Visión y Alcance.doc	CMMI	3	2	6	1	6	0,20	NO	0,9	0,18		0,94
1. Análisis	Matriz_de_Evaluacion_de_Propuestas.xls	CMMI	3	2	6	0	0	0,00	NO	0,9	0,00	1. Análisis	
	Estimación Modelo de Servicios.xls	Cruce	5	4	20	1	20	0,63	SI	1	0,63		
	LB_Requerimientos_Calidad.xls	ITIL	2	3	6	1	6	0,19	NO	0,9	0,17		0,79
2. Diseño	Definición CPI.xls	ITIL	2	4	8	1	8	0,31	NO	0,9	0,28	2. Diseño	
	TF Diseño.xls	ITIL	2	3	6	1	6	0,23	SI	1	0,23		
	Plan_de_CM.doc	CMMI	3	1	3	1	3	0,12	NO	0,9	0,10		
	Plan_de_Pruebas.doc	CMMI	3	3	9	1	9	0,35	NO	0,9	0,31		0,92
3. Ejecución	TF Ejecución.xls	Cruce	5	3	15	1	15	1,00	SI	1	1,00	1,00	3. Ejecución
4. Gestionar Defectos	TF Entidad-Defecto.xls	Cruce	5	3	15	1	15	1,00	SI	1	1,00	1,00	4. Gestionar Defectos
5. Informes	Plan de Negocio.xls	Cruce	5	4	20	1	20	0,50	SI	1	0,50	5. Informes	
	Cierre Proyecto.xlsx	Cruce	5	2	10	1	10	0,25	NO	0,9	0,23		
	RESULTADO TESTING Ciclo.doc	Cruce	5	2	10	1	10	0,25	NO	0,9	0,23		0,95

Figura 5.25: Resultados del apego a la metodología

Métrica de Porcentaje de Apego a la Metodología obtenido = 93%

6 Conclusiones

La Metodología Propuesta se basa fundamentalmente en la necesidad que impone el mercado en entregar servicios de calidad a bajo costo y en los tiempos comprometidos utilizando estándares de mejores prácticas y modelos de servicios que han existido desde hace ya varios años.

Al realizar el cruce de los procesos de CMMI versus las mejores prácticas de ITIL sobre uno de los procesos más importantes a la hora de validar la calidad de un software, se obtiene una Metodología eficiente y consistente de aplicar a cualquier tipo de proyecto de desarrollo. Además la expertise de trabajar en distintos tipos de proyectos en el proceso escogido (Validación y Verificación), apoyó en forma significativa al momento de crear las plantillas de apoyo que ayudan a concretizar la aplicación de la propuesta metodológica a un proyecto real.

El análisis de las métricas en este documento se realiza en función de los dos ciclos ejecutados en el proyecto de Testing, las métricas de Esfuerzo y Costos se realizan en base a los antecedentes entregados por el Jefe de Proyecto de Testing Funcional, en función de que el ciclo debe terminar en la fecha replanificada.

En el cálculo de métricas se puede ver que las variables cuantitativas son posibles de ajustar en función de la precisión y realismo con que se analicen; por ejemplo, las tasas de rendimiento de diseño y ejecución, así como también los costos relacionados al proyecto.

El cálculo de la métrica de Apego a la Metodología ayuda a determinar que en la aplicación de dicha métrica, se establece el grado de utilización efectiva en relación al objetivo de la Metodología. Sin embargo, se pueden recoger mejoras sustanciales en cuanto a la entrevista al jefe de proyecto sobre como fue el proceso de aplicación, ya que él indica que en sí la Metodología le sirvió para mantener el control y orden de los pasos a seguir, además de aumentar la percepción de calidad de servicio por parte del cliente, ya que se había trabajado en otros proyectos con este cliente, sin embargo esta vez las reglas para una mejor entrega del servicio fueron entregadas inicialmente, apoyando el proceso de desarrollo de software en lo que respecta a formalidad en los documentos de diseño y estimación de esfuerzo y planificación más acertada para el desarrollo, ya que consideraron la generación de los documentos necesarios en los momentos reales del ciclo de vida del proyecto y no al final como se estila realizar.

Se propone como trabajo futuro modelar y generar una Base de Conocimiento para registrar los tipos de proyectos y sus mejores ajustes planificados y reales con el fin de mejorar cada vez más el Modelo del Servicio.

También se planteó la iniciativa de llevar a programa la plantilla de Modelado de Servicio, con el fin de que pueda ser accesada desde internet al momento que se desea realizar la estimación de costo o esfuerzo involucrado en un servicio de Testing funcional.

7 Bibliografía

- 1) Axentia (2006), White Paper, Una Introducción a CMMI, Transforming Your IT Organization. (Ver ANEXO 2 entregado en el Informe de Avance), 2006.
- 2) Fernando Sánchez, (2008), CMMI vs. ITIL: dos enfoques complementarios, gerente de Grupo Delaware, <http://www.mkm-pi.com/mkmpi.php?article1817>, Jueves 6 de marzo de 2008.
- 3) Juan Palacio, (2006), Sinopsis de los Modelos SW-CMM y CMMI, <http://www.navegapolis.net/jpalacio@navegapolis.net>, 1.0 Abril – 2006.
- 4) Juan Raggio Pérez, (2004), Universidad Politécnica de Madrid, Facultad de Informática, Estudios de Doctorado, DESARROLLO DE PROCESOS DE GESTIÓN DE SERVICIOS DE EXPLOTACIÓN SIGUIENDO EL MODELO CMMI, 2004.
- 5) Lorena León – Gabriela Puglla, (2008), Métricas de Proceso y proyecto de software, Universidad Técnica particular de Loja. <http://www.slideshare.net/loreknelamorena/mtricas-de-proceso-y-proyecto-de-software>, 2008.
- 6) Mariana Isela Jaramillo González (2005), Utilización de Estándares ITIL para lograr el Nivel 3 de CMMI en una Organización. Universidad Autónoma del Estado de México, <http://docente.uco.mx/juancont/documentos/cap02/6.pdf>, 24 Octubre 2005.
- 7) OGC (2007), ITIL Managing IT Services. Best Practice for Application Management, http://www.ogc.gov.uk/guidance_itol.asp, 2007.
- 8) PMBOK (2004), [PMBOK](#) Tercera Versión en Español [Project Management Institute](#). «Capítulo 1», *Guía de los Fundamentos de la Dirección de Proyectos*, 3ª edición. ISBN 1-930699-73-5. http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos, 2004.
- 9) Raúl Suárez O. y Felipe Donoso Jaurés, (2006), METODOLOGÍA ITIL, Descripción, Funcionamiento y Aplicaciones, PÍA RAMÍREZ BRAVO - FELIPE DONOSO JAURÉS, Santiago, Julio de 2006.
- 10) Serge Thorn Geneva (2007), ITIL and CMMI synergies. <http://sergethorn.blogspot.com/2007/05/itil-and-cmmi-synergies.html>, 23 Mayo 2007.

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**PROPUESTA METODOLÓGICA PARA LA MEJORA DEL
PROCESO DE VERIFICACIÓN Y VALIDACIÓN EN LA
INDUSTRIA DE SERVICIOS TI EN BASE A MODELOS CMMI
CON ITIL**

NATALIA KARIN CARPIO ARNAIZ

TESIS DE GRADO
MAGÍSTER EN INGENIERÍA INFORMÁTICA

Diciembre 2009

Pontificia Universidad Católica de Valparaíso
Facultad de Ingeniería
Escuela de Ingeniería Informática

**PROPUESTA METODOLÓGICA PARA LA MEJORA DEL
PROCESO DE VERIFICACIÓN Y VALIDACIÓN EN LA
INDUSTRIA DE SERVICIOS TI EN BASE A MODELOS CMMI
CON ITIL**

NATALIA KARIN CARPIO ARNAIZ

Profesor Guía: Silvana Roncagliolo de la Horra

Programa: Magíster en Ingeniería Informática

Diciembre 2009

Resumen

Cada vez más, a los departamentos de TI se les exigen niveles de calidad y servicio directamente proporcionales a los niveles de calidad y servicio que se exigen a las distintas líneas de negocio de las organizaciones. Esto está provocando un afán por encontrar métodos y técnicas que permitan desarrollar y mantener software de alta calidad, así como implantar y mantener infraestructura con un alto grado de disponibilidad.

Se presenta una propuesta metodológica creada en base a los procesos de Verificación y Validación de CMMI pareados con los Procesos/Funciones de ITIL. En la metodología se exponen las prácticas específicas y se detalla el proceso de cómo implantar la metodología que apoya y mejora los niveles de calidad en el testing funcional de aplicaciones.

A partir de la metodología propuesta se presenta la evaluación de su aplicación a un caso real, definiendo como punto de partida las métricas que se consideran en dicha evaluación, generando mejoras a la propuesta en sí.

Abstract

Everyday, TI Departments are demanded with quality and service levels directly proportional to the quality and service levels demanded to the different business lines of the organizations. This is causing an urge to find methods and techniques that allow to develop and maintain high quality software, such as to implant and keep the infrastructure with a high level of availability.

A methodological proposal is shown, created based on the CMMI verification and validation processes matched against the ITIL processes/functions. In the methodology specific practices are exposed and the quality levels are improved in the functional testing of the applications.

From the proposed methodology, the evaluation of its application to a real case is showed, defining as a start point the metrics considered in that evaluation, generating improvements to the proposal itself.

Índice

1	Introducción.....	9
2	Objetivos	10
3	Estado del Arte	11
3.1	ITIL v2.0.....	11
3.1.1	<i>Descripción (PRamirez-FDonoso, 2006)</i>	<i>11</i>
3.1.2	<i>Objetivos de ITIL (PRamirez-FDonoso, 2006).....</i>	<i>12</i>
3.2	CMMI.....	13
3.2.1	<i>Historia y evolución (JPalacio, 2006)</i>	<i>13</i>
3.2.2	<i>Los Cinco Niveles de Madurez (Axentia, 2006).....</i>	<i>13</i>
3.2.3	<i>Arquitectura del Modelo (Axentia, 2006)</i>	<i>15</i>
3.3	CMMI e ITIL (JRaggio, 2004)	17
3.4	Análisis del Estado del Arte.....	18
4	Solución Propuesta.....	21
4.1	Conocimiento de los modelos.....	21
4.2	Proceso de cruce de los modelos	21
4.3	Definición de métricas para la evaluación final	25
4.3.1	<i>Tiempo</i>	<i>26</i>
4.3.2	<i>Costo.....</i>	<i>26</i>
4.3.3	<i>Alcance</i>	<i>26</i>
5	Implantación de la metodología propuesta y aplicación de plantillas de apoyo	32
5.1	Proceso para aplicar la metodología	32
5.1.1	<i>Proceso “Definir proyecto”</i>	<i>34</i>
5.1.2	<i>Proceso “Administrar externo”.....</i>	<i>37</i>
5.1.3	<i>Proceso “Administrar proceso de pruebas” PARTE 1.....</i>	<i>40</i>
5.1.3.1	<i>Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”.....</i>	<i>45</i>
5.1.3.2	<i>Proceso “Estimar grado automatización y nivel diagnóstico”.....</i>	<i>46</i>
5.1.3.3	<i>Proceso “Definir requerimientos no funcionales”</i>	<i>51</i>
5.1.4	<i>Proceso “Administrar proceso de pruebas” PARTE 2.....</i>	<i>53</i>
5.1.4.1	<i>Proceso “Definir prueba”</i>	<i>57</i>
5.1.4.2	<i>Proceso “Automatizar prueba unitaria”.....</i>	<i>61</i>
5.1.4.3	<i>Proceso “Automatizar prueba integrada”</i>	<i>63</i>
5.1.4.4	<i>Proceso “Generar LB datos”.....</i>	<i>64</i>
5.1.4.5	<i>Proceso “Ejecutar ciclo pruebas”</i>	<i>66</i>
5.1.4.6	<i>Proceso “Verificar ambiente prueba”</i>	<i>68</i>
5.1.4.7	<i>Proceso “Ejecutar conjunto de pruebas”</i>	<i>69</i>
5.1.5	<i>Proceso “Administrar proceso de pruebas” PARTE 3.....</i>	<i>70</i>

5.1.6	Proceso: “Gestionar defectos”.....	71
5.1.7	Proceso: “Realizar auditoría CM/PPQA”	74
5.1.8	Proceso: “Controlar proyecto”.....	79
5.2	Resumen de resultados obtenido (Métricas).....	81
6	Conclusiones	89
7	Bibliografía	90

Tabla de Ilustraciones

Figura 3.01: Modelo de Arquitectura CMMI.....	15
Figura 3.02: Niveles de Madurez CMMI.....	16
Figura 3.03: Representación escalonada. Por Nivele de Madurez CMMI.....	17
Figura 4.01: Tabla de mapeo CMMI e ITIL.....	22
Figura 4.02: Metodología propuesta.....	24
Figura 4.03: Triángulo de restricciones.....	26
Figura 4.04: Métricas definidas para validación de la implantación de metodología propuesta.....	28
Figura 4.05: Métrica “Importancia Metodológica”.....	28
Figura 4.06: Métrica “Propuesta Metodológica”.....	29
Figura 4.07: Tabla “Cálculo de métricas”.....	29
Figura 4.08: Tabla “Apego de la metodología propuesta”.....	30
Figura 5.01: Identificación planillas de apoyo.....	32
Figura 5.02: Proceso “Definir proyecto”.....	34
Figura 5.03: Proceso “Administrar externo”.....	37
Figura 5.04: Proceso “Administrar proceso de pruebas” PARTE 1.....	40
Figura 5.05: Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”.....	45
Figura 5.06: Proceso “Estimar grado automatización y nivel diagnóstico”.....	46
Figura 5.07: Proceso “Habilitar PC para captura”.....	50
Figura 5.08: Proceso “Definir requerimientos no funcionales”.....	51
Figura 5.09: Proceso “Administrar proceso de pruebas” PARTE 2.....	53
Figura 5.10: Proceso “Definir prueba”.....	57
Figura 5.11: Proceso “Automatizar prueba unitaria”.....	61
Figura 5.12: Proceso “Automatizar prueba integrada”.....	63
Figura 5.13: Proceso “Generar LB datos”.....	64
Figura 5.14: Proceso “Ejecutar ciclo pruebas”.....	66
Figura 5.15: Proceso “Verificar ambiente prueba”.....	68
Figura 5.16: Proceso “Ejecutar conjunto de pruebas”.....	69
Figura 5.17: Proceso “Administrar proceso de pruebas” PARTE 3.....	70
Figura 5.18: Proceso “Gestionar defectos”.....	71
Figura 5.19: Proceso “Realizar auditoría CM/PPQA”.....	74
Figura 5.20: “Ciclo de defectos”.....	78
Figura 5.21: Proceso “Controlar proyecto”.....	79
Figura 5.22: Ajuste del modelo en las variables definidas.....	86
Figura 5.23: Costos por ajuste.....	87
Figura 5.24: Tabla comparativa de las métricas con y sin ajustes.....	87
Figura 5.25: Resultados del apego a la metodología.....	88

Glosario

CPU: Caso de Prueba Unitario, corresponde a la descripción de una prueba unitaria que posee un resultado y puede ser constituyente de un Caso de Prueba Integrado (CPI).

CPI: Caso de Prueba Integrado, corresponde a la descripción de aquella prueba cuyo objetivo es validar una funcionalidad o una operación del aplicativo. Está constituida por más de un CPU.

Ciclo: Es una ventana de tiempo definida en la planificación que corresponde a un grupo de iteraciones de pruebas con sus respectivos datos para cada versión liberada.

Iteración: Es una instancia de pruebas para una o un grupo de funcionalidades, interfaces y/o reportes que podría repetirse dentro de un ciclo, esto dependiendo del grado de aprobación, en caso de cumplir con todos los criterios de aceptación para la primera iteración, no es necesario seguir con las siguientes.

Pruebas con Error: Corresponde a aquellas pruebas que luego de ser ejecutadas presentan algún tipo de falla que implica que el estado de ejecución de éstas no sea óptimo.

Pruebas de Aceptación de Usuario: Son aquellas diseñadas y ejecutadas por el usuario para demostrar que la aplicación satisface los requerimientos. En inglés: User Acceptance Testing.

Pruebas de Nuevas Funcionalidades: Corresponde a aquellas pruebas que apuntan a validar aquella operatoria del sistema que no ha sido certificada en ninguna etapa de pruebas anterior.

Pruebas de Regresión: Es un "Conjunto de Pruebas" realizado para identificar "defectos" que causan "Regresión". Si el desarrollador resuelve un defecto, se deben realizar pruebas en los alrededores para verificar que no se haya impactado el resto del código, a esto se le llama regresión. En inglés: Regression Test.

Pruebas Trazadoras: Es un "Conjunto de Pruebas" corridas sobre una cierta "Release" para asegurar que es suficientemente estable para continuar con el "Ciclo" de pruebas activo. Estas pruebas usualmente son un subconjunto del conjunto completo de pruebas, preferiblemente de tipo automatizado, que involucran cada parte del sistema al menos en una forma superficial. Unas buenas pruebas trazadoras también mantienen el sistema corriendo por un lapso de tiempo suficiente como para que se manifiesten problemas "gruesos" de confiabilidad y disponibilidad. En inglés: Smoke Test o Sanity Test.

Requerimientos de Negocio: Corresponde a aquel requerimiento que es generado por el cliente y que representa alguna funcionalidad del sistema a validar. En este caso, el responsable de dicho requerimiento es el equipo de Gestión de Proyecto.

Requerimientos de Prueba Funcional: Corresponde a aquel requerimiento generado a partir de un Requerimiento de Negocio, representa el alcance de las pruebas que se realizarán para abordar la validación del requerimiento de negocio respectivo. Este tipo de requerimiento es de responsabilidad del equipo de Calidad.

1 Introducción

Cada vez más, a los departamentos de TI se les exige niveles de calidad y servicio directamente proporcionales a los niveles de calidad y servicio que se exige a las distintas líneas de negocio de las organizaciones.

Esto está provocando un afán por encontrar métodos y técnicas que permitan desarrollar y mantener software de alta calidad, así como implantar y mantener infraestructuras con un alto grado de disponibilidad.

En los últimos años han aparecido dos líneas metodológicas que están consiguiendo un gran protagonismo por su carácter práctico y por los buenos resultados obtenidos: CMMI e ITIL. La primera se centra en los procesos asociados al desarrollo de software, mientras que la segunda enfatiza en el enfoque a servicios (FSanchez, 2008).

Según revisión en la Web, no existe actualmente una Propuesta tan específica como ésta, sólo se presentan cruces a nivel completo de Metodología CMMI y Mejores Prácticas de ITIL, todas en un marco genérico.

Esta Propuesta apunta específicamente al cruce y presenta una Metodología en la implantación del proceso de Verificación y Validación de CMMI basándose en la Mejores Prácticas de ITIL.

El trabajo presenta una guía clara de cómo enfrentar un proyecto de Testing tanto desde su estimación pasando por el diseño, ejecución, concluyendo por el análisis de los resultados, cierre y revisión del cierre de proyecto en lo que respecta a su Plan de negocio inicial.

Además de presentar la Metodología, se presentan plantillas de apoyo para dicho proceso y se aplica a un proyecto real con el fin de poder refinar la Propuesta y evaluar los resultados en base al apego obtenido al aplicar dicha Metodología, resultando de este trabajo una Propuesta validada y refinada desde su concepción.

2 Objetivos

El objetivo propuesto en este proyecto es:

- Generar una propuesta Metodológica de apoyo para la implantación de Testing en la Industria de TI en base al proceso de CMMI Verificación y Validación con las Mejores Prácticas de ITIL, comprobando su aplicabilidad con un caso real.

Lo anterior se realizará a través de la persecución de los siguientes objetivos específicos:

- Proponer una Metodología de apoyo para la implantación de Testing en la Industria de TI en base al proceso de CMMI Verificación y Validación, cruzándolo con las Mejores Prácticas de ITIL.
- Generar los documentos de apoyo (plantillas) para la aplicación de la Metodología Propuesta, especificando su utilización en cada uno de los procesos que se definen en la Metodología que apoya al proceso de Verificación y Validación de CMMI.
- Generar métricas para evaluar la correcta implementación de la Metodología Propuesta.
- Aplicar la Metodología a un caso real (Proyecto Gestor de Oportunidades), desde su fase de evaluación hasta la ejecución del Testing, para así evaluar la efectividad en la aplicación de la Metodología en base a la definición de métricas para su evaluación.
- Evaluar la aplicación de la Metodología en base a las métricas propuestas.

3 Estado del Arte

Al iniciar el estado del arte es importante presentar los conceptos, así como los orígenes de la metodología y las mejores prácticas, lo que se encuentra en las siguientes secciones.

3.1 ITIL v2.0

3.1.1 Descripción (PRamirez-FDonoso, 2006)

Information Technology Infrastructure Library (ITIL), es una metodología que se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones en sus Sistemas de Información y Tecnologías de Información. Esta metodología fue desarrollada a petición del Gobierno del Reino Unido a finales de los 80 y recoge las mejores prácticas en la gestión de los Sistemas de Información. Desde entonces se ha ido extendiendo su uso en la empresa privada, tanto multinacional como PYME, llegando a ser considerado un estándar de facto para la gestión de esta área de la empresa.

En un entorno donde los periodos de disponibilidad de los servicios son cada vez más amplios, donde las exigencias del cliente son cada vez más elevadas, donde los cambios en los negocios son cada vez más rápidos, es muy importante que los Sistemas de Información estén adecuadamente organizados y alineados con la estrategia del negocio. Igualmente ITIL, ofrece toda una serie de definiciones de conceptos típicos de los Sistemas de Información para garantizar que todos sus conocedores hablen de lo mismo, reduciendo así los tiempos y riesgos por malas interpretaciones.

ITIL, es un set de documentos donde se describen los procesos requeridos para la gestión eficiente y efectiva de los Servicios de Tecnologías de Información dentro de una organización. Son un conjunto de mejores prácticas y estándares en procesos para hacer más eficiente el diseño y administración de las infraestructuras de datos dentro de la organización. Es un “marco de trabajo” (framework) para la Administración de Procesos de TI.

Basándose en el principio de mejora continua, ITIL fue madurado y de los 40 libros originales se redefinieron; se agruparon y eliminaron redundancias y actualmente se presenta en 7 libros:

- 1) Service Support
- 2) Service Delivery
- 3) Security Management
- 4) Application Management
- 5) The Infrastructure Management
- 6) Planning to Implement Service Management
- 7) The Business Perspective

Esta metodología se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos que cubren las actividades más importantes de las organizaciones. Garantizando así los niveles de servicio establecidos entre la organización y sus clientes.

El objetivo de ITIL es diseminar las mejores prácticas en la Gestión de Servicios de Tecnologías de Información. Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios IT, al mismo tiempo de garantizar los requerimientos de la información en cuanto a seguridad, mantienen e incrementan sus niveles de fiabilidad, consistencia y calidad.

ITIL brinda una descripción detallada de un número de prácticas importantes en TI, a través de una amplia lista de verificación, tareas, procedimientos y responsabilidades que pueden adaptarse a cualquier organización. ITIL describe una aproximación sistemática y profesional a la Gestión de Servicios TI, haciendo énfasis en la importancia clave de cumplir con los requerimientos del negocio respetando los costos acordados.

3.1.2 Objetivos de ITIL (PRamirez-FDonoso, 2006)

El objetivo que persigue ITIL es diseminar las mejores prácticas en la gestión de servicios de Tecnologías de Información de forma sistemática y coherente. El planteo principal se basa en la calidad de servicio y el desarrollo eficaz y eficiente de los procesos.

Esta metodología está especialmente desarrollada para reducir los costos de provisión y soporte de los servicios de TI, al mismo tiempo que se garantizan los requerimientos de la información en cuanto a seguridad manteniendo e incrementando sus niveles de fiabilidad, consistencia y calidad.

La filosofía ITIL adopta la gestión de procesos y considera que, para lograr los objetivos claves de la Administración de Servicios estos procesos deberían ser usados por las personas y las herramientas efectiva, eficiente y económicamente en el desarrollo de la alta calidad y la innovación de los servicios de TI alineados con los procesos de negocio.

Los estándares ITIL exigen un replanteamiento del área tecnológica y la definición de los elementos y procesos "críticos" dentro de la empresa.

3.2 CMMI

3.2.1 Historia y evolución (JPalacio, 2006)

1984 El Congreso del Gobierno Americano aprobó la creación de un organismo de investigación para el desarrollo de modelos de mejora para los problemas en el desarrollo de los sistemas de software, y evaluar la capacidad de respuesta y fiabilidad de las compañías que suministran software al Departamento de Defensa.

Creación del SEI (Instituto de Ingeniería del Software), fundado por el Departamento de Defensa Americano y la Universidad Carnegie Mellon.

1985 SEI empieza a trabajar en un marco de madurez de procesos que permita evaluar a las empresas productoras de software.

La investigación evoluciona hacia el “Modelo de Madurez de las Capacidades (CMM)”.

1991 En agosto SEI publica la versión 1.0 del Modelo de Madurez de las Capacidades para el Software (SWCMM, Capability Maturity Model for Software).

1993 SEI publica la versión 1.1 de SW-CMM.

1997 Publicación de la versión 1.2.

2000 SW-CMM fue integrado y relevado por el nuevo modelo CMMI.

2006 Se publica CMMI –DEV, V1.2, CMMI for Development, Version 1.2.

2007 Se publica CMMI-ACQ, V1.2, CMMI for Acquisition, Version 1.2, *Improving processes for acquiring better products and services*.

La propuesta metodológica que se expone a continuación, se realiza sobre CMMI-ACQ, ya que las empresas de desarrollo pueden contar con externalización de servicios a la hora de desarrollos de software.

3.2.2 Los Cinco Niveles de Madurez (Axentia, 2006)

En general, los niveles de madurez suelen explicarse en orden creciente; se tomará aquí una dirección distinta y se explicarán exactamente al revés: desde el nivel cinco al nivel uno.

Imagine por un momento que está en una organización de nivel cinco. En este tipo de organizaciones, los procesos son analizados para eliminar las causas comunes de variación, o sea, aquellas que tienen que ver con la misma naturaleza del proceso, no atribuibles a causas externas. Las variaciones en las salidas de los procesos son al azar, pero se encuentran controladas estadísticamente (se puede predecir los resultados de los procesos con cierto nivel de confiabilidad).

Para poder llegar a este nivel, la organización debió primero haber eliminado las causas especiales de variación, aquellas que tienen que ver con causas externas, como por ejemplo falta de entrenamiento del personal, problemas con las herramientas, etc. Este tipo de causas no son aleatorias: Si se examinan los resultados se podrán ver las tendencias que claramente indican que las variaciones tienen un origen concreto. En una organización de nivel cuatro, entonces, las causas especiales de variación son identificadas y eliminadas.

Para poder llegar a identificar causas de variación se necesita tener un proceso estándar: difícil sería poner bajo control estadístico un proceso que no se encuentre mínimamente formalizado.

Así se llega al nivel tres, en el cual los proyectos emplean un proceso productivo adaptado del proceso estándar de la organización. Las actividades técnicas y de gestión son realizadas de acuerdo a políticas, procesos y procedimientos formalizados en algún tipo de estándar organizacional profundamente arraigado en la cultura. La gente está entrenada y dispone de recursos para poder hacer su trabajo. También hay una infraestructura básica (personal, herramientas, etc.) para definir y mejorar el proceso productivo.

Pero para poder llegar a esta situación es necesario pasar por una etapa previa: difícilmente se puede introducir en una organización prácticas estándar relacionadas con la ingeniería del producto (análisis, diseño, etc.) si no se ofrece un contexto en donde ellas puedan ser correctamente ejecutadas. Ese es el foco del nivel dos: poner en orden las prácticas relacionadas con el manejo elemental de los proyectos.

En el nivel dos, los proyectos de la organización siguen algún tipo de proceso para realizar las actividades relacionadas con la gestión del proyecto (planificación, control), para administrar los requerimientos y las configuraciones, y para medir y analizar la calidad de los productos y el desempeño de los procesos. También hay prácticas de aseguramiento de la calidad que permiten garantizar que cada proyecto sigue sus propios estándares.

Y así se llega al nivel uno: La situación aquí es caótica. No existen procesos (no al menos en el sentido del modelo) y la *performance* de los proyectos depende profundamente de la buena voluntad y la capacidad de la gente.

3.2.3 Arquitectura del Modelo (Axentia, 2006)

En la representación por niveles (ver Figura 3.01), cada nivel de madurez contiene varias áreas de proceso, las que a su vez quedan definidas por uno o varios objetivos específicos y un objetivo genérico. Cada uno de ellos tiene vinculado un conjunto de prácticas, llamadas específicas y genéricas respectivamente.

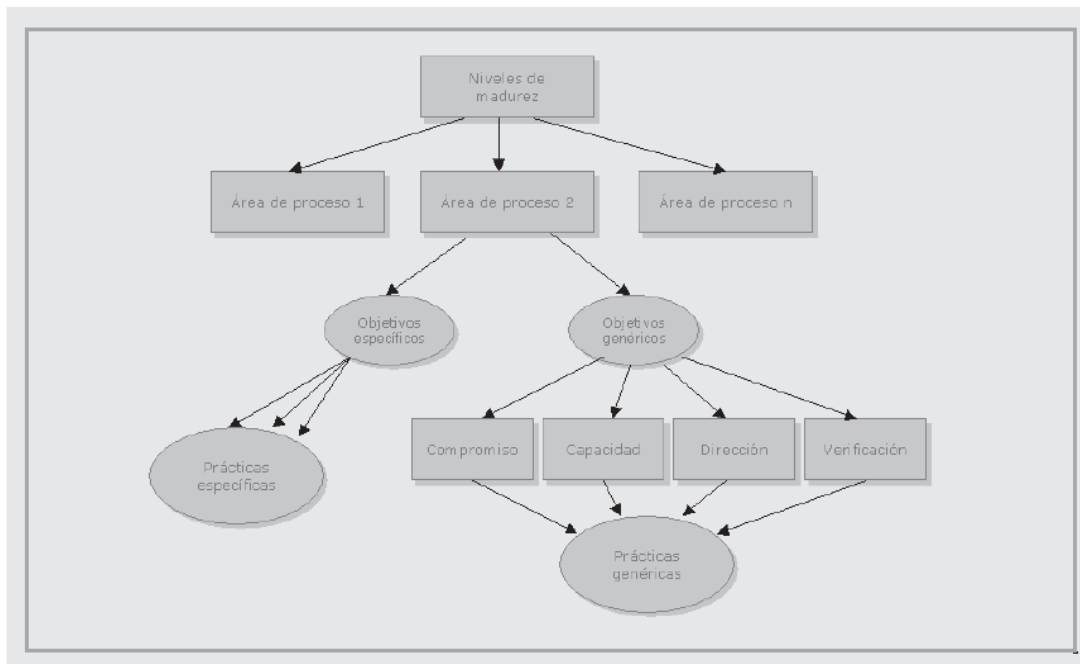


Figura 3.01: Modelo de Arquitectura CMMI

CMMI ofrece una representación sistemática y estructurada para abordar el modelo basado en un proceso de mejora que permite avanzar una etapa a la vez. El logro de cada etapa asegura que el proceso se ha establecido, y cuenta con la infraestructura adecuada que permite avanzar a la siguiente etapa.

Las áreas de proceso del modelo están organizadas en 5 niveles de madurez que representan, cada una, alguna conjetura respecto del proceso de mejora, lo que se muestra en la Figura 3.02.

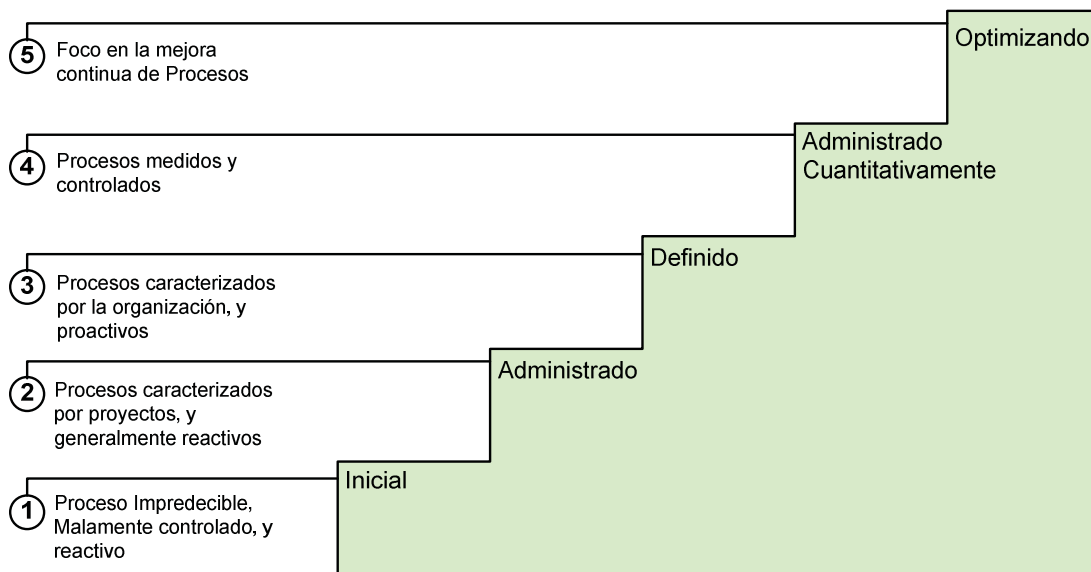


Figura 3.02: Niveles de Madurez CMMI

Esta representación sugiere un camino para la mejora de acuerdo al nivel de madurez de la organización, desde el nivel “Inicial” al nivel “Optimizando”.

El logro de cada nivel de madurez garantiza que un nuevo estado de mejora ha sentado las bases para el siguiente nivel, permitiendo así una mejora incremental.

La representación por niveles de madurez es una buena opción para mejorar un conjunto específico de procesos en cada etapa. Este orden se ha determinado a través de más de una década de investigación y experiencia.

Las áreas de proceso que contiene cada nivel de madurez se muestran en la Figura 3.03.

Nivel	Foco	Áreas de Procesos	Calidad
5 Optimizado	Proceso de Mejoramiento Continuo	Innovación organizacional y Desarrollo Análisis Causal y Resolución	Productividad
4 Administrado Cuantitativamente	Administración Cuantitativa	Ejecución proceso organizacional Administración cuantitativa del proyecto	
3 Definido	Estandarización del Proceso	Desarrollo de Requerimientos Solución Técnica Integración del producto Verificación Validación Foco en proceso organizacional Definición del proceso organizacional + IPPD Entrenamiento organizacional Administración del proyecto integrado + IPPD Administración de Riesgo Análisis de Decisión y Resolución	
2 Administrado	Administración de Proyecto Básico	Administración de Requerimientos Planificación del Proyecto Monitoreo y Control del Proyecto Administración de Acuerdo con Proveedor Medición y Análisis Aseguramiento de la Calidad de Proceso y Producto Administración de la Configuración	Riesgo
1 Inicial			Reelaboración

Figura 3.03: Representación escalonada. Por Nivele de Madurez CMMI

3.3 CMMI e ITIL (JRaggio, 2004)

Como se ha comentado anteriormente CMMI es una colección de buenas prácticas alineadas con el concepto de modelos de madurez. Debido a la herencia de SW/CMM y tal y como se ha visto en la descripción de las áreas de procesos, objetivos y las prácticas, está muy orientado hacia desarrollo y mantenimiento de software.

Si bien CMMI es la evolución e integración de un conjunto de modelos de madurez, el desarrollo y mantenimiento de software, fijan en gran medida sus líneas de actuación. En el otro punto se encuentra ITIL, modelo totalmente orientado a los servicios de mantenimiento y gestión de las operaciones.

ITIL y CMMI son dos modelos de madurez claramente distintos pero no mutuamente excluyentes. Las principales diferencias se deben a que CMMI está focalizado en la madurez de los procesos software a través de una actividad de mejora continua e ITIL se centra en el entendimiento y desarrollo de todas las áreas referentes a la infraestructura de TI, además de incorporar el ciclo de vida del hardware dentro de una organización.

3.4 Análisis del Estado del Arte

El término “nivel de calidad” es aplicable directamente a software o aplicaciones que se desarrollan, pero este término es demasiado subjetivo ya que depende de la definición y/u objetivos de cada empresa que desea implantar este “nivel de calidad”. Actualmente las empresas de TI han involucrado el concepto pero en base a criterios y/o definiciones que nacen por experiencias propias.

Al hacer una revisión en la Web, se pueden encontrar diversos estudios y propuestas de solución en lo que respecta a mejoras al “nivel de calidad” de los productos, pero la mayoría de ellos se enfoca a soluciones generales que condicionan el testing a mejoras en los procesos iniciales de construcción del producto.

La gran mayoría de los análisis y estudios existentes, se basa fundamentalmente en resaltar la sinergia que existe entre CMMI e ITIL, proponiendo distintos puntos de vista a nivel de cómo fusionar o complementar éstos en función de aprovechar ambos en la aplicación de la industria para conseguir mejores “niveles de servicio”.

En general ITIL se puede combinar con CMMI para cubrir todo, pero no aborda el desarrollo de los sistemas de gestión de la calidad. Además, no está orientado a los procesos de desarrollo de software y su uso depende en gran medida de la interpretación. Mientras CMMI es el estándar de calidad de fábrica para los procesos de desarrollo de software, ITIL para muchos es la herramienta de elección para las operaciones y la parte de infraestructura de TI, especialmente para servicios de TI (SThorn 2007).

ITIL y CMMI se aplican de mejor manera a diferentes partes de la organización de TI:

- El uso de CMMI en el desarrollo de aplicaciones
- El uso de CMMI en las TIC. Específicamente en los proyectos de infraestructura.
- Uso de ITIL en las operaciones de TI y servicios

Dentro de las propuestas existentes con un poco más de detalle a nivel de cruces entre la metodología y las mejores prácticas, se encuentra un modelo en el que las mejores prácticas de ITIL ayudan a lograr las metas requeridas para obtener un nivel particular de CMMI, por ejemplo nivel 3 (MJaramillo, 2005).

El modelo apunta a generalizar, a través de los niveles de ITIL como apoyar a CMMI para llegar al nivel 3.

1) Nivel de Servicio de Soporte:

Se lleva a cabo la administración de cada proceso planeado en la primera etapa de CMMI permitiendo así la interacción con las partes involucradas y/o afectadas, alcanzando una eficiente comunicación de usuario y proveedor,

los cuales adquieren un compromiso con el plan de trabajo ya establecido y por último llevan a cabo un mantenimiento continuo del plan.

Este nivel de ITIL también ayuda al Monitoreo y Control del Proyecto, es decir, logra el entendimiento de los procesos para tomar decisiones correctivas adecuadas.

2) Nivel de Entrega de Servicios:

Describe los procesos necesarios de entrega con respecto a la calidad y costo lo cual permite llevar a cabo una eficiente Administración de Proveedores (Etapa de CMMI), la cual analiza todo tipo de adquisición, los contratos, etc.

Por lo que participa en la Administración del Proyecto para su Desarrollo Integral (Etapa de CMMI), donde se considera el grado de involucramiento de quienes administran el proyecto.

3) Nivel de Administración de Seguridad:

Mediante este nivel se logra proteger la tecnología de información de la compañía motivo que contribuye a mejorar la “Administración de Riesgos”, que lleva a cabo CMMI, etapa que persigue evitar los peligros que impidan lograr los objetivos del proyecto.

4) Nivel de Perspectiva del Negocio:

Es indispensable para CMMI ya que mediante este nivel el gerente debe conocer el impacto de la tecnología de información y las medidas que deberá tomar para protegerse de situaciones adversas logrando así la Integración del Equipo de Trabajo, etapa de CMMI que pretende que todas las partes involucradas compartan sus habilidades y experiencias a fin de enriquecer el trabajo de la organización trayendo como resultado la calidad de cada proceso que desarrolla.

5) Nivel de Administración de la Infraestructura de TIC:

Mediante este nivel se logra tener un control de calidad con respecto a todos los servicios de comunicación que proporciona la organización facilitando así la Administración Integral de Proveedores que lleva a cabo CMMI evaluando de esta forma sus procesos y productos y haciendo los ajustes en la relación y en sus contratos evitando así comprar productos o servicios obsoletos para la organización.

6) Nivel de Administración de las Aplicaciones:

Determina las soluciones para cubrir las necesidades de los usuarios de acuerdo a los recursos y presupuesto con los que se cuenta logrando así brindar información importante a la Administración Cuantitativa del Proyecto (Etapa de CMMI) la cual considera el tiempo, mano de obra necesaria, y los costos de los procesos definidos por la organización cumpliendo de esta forma los objetivos establecidos en calidad y eficiencia.

7) Nivel de Planeación para la Administración de Servicios:

Ayuda a las organizaciones a identificar sus fortalezas y debilidades generando así que CMMI logre definir aún mejor sus procesos para un progreso en su madurez.

A partir de los análisis antes expuestos, se desprende que en su mayoría, todo apunta a refundir la metodología y sus mejores prácticas a nivel general, sin embargo es necesario contar con propuestas específicas sobre los procesos de CMMI versus ITIL, con el fin de entregar propuestas metodológicas que apoyen el desarrollo de un proceso específico sin ánimo de estandarizar el desarrollo del proceso, sino más bien orientar en cómo y cuándo aplicar las mejores prácticas en un proceso determinado en base a experiencias vividas.

Por lo anterior se desea proponer una Metodología de apoyo en la implantación del proceso de Testing de software y/o aplicativos en base a los Modelos antes mencionados.

Se analizaron los puntos de semejanza entre dichos Modelos, sin olvidar la diferencia entre sus enfoques, para así identificar los procesos de apoyo específicos al Testing. Se propone una Metodología clara y precisa para el levantamiento y mejora de los procesos existentes, con el fin de solucionar la subjetividad que existe en la implantación del proceso de pruebas, así como también en los procesos de apoyo para entrega de los servicios relacionados al Testing.

La Metodología Propuesta, puede ser aplicada tanto en departamentos de TI que pertenezcan a empresas pequeñas, que no pueden optar a obtener una certificación de los Modelos base, o bien en departamentos de TI que a pesar de contar con presupuesto para optar por estas certificaciones no cuentan con la experiencia en la implantación de servicios de Testing.

El cruce se realizó en base a las prácticas concretas y se utilizó como inicio el análisis de ITIL (PRamirez-FDonoso, 2006), ya que el enfoque a servicios debe ser la base de la Metodología que se desea generar. Sin embargo como el enfoque a nivel de procesos apunta a Verificación y Validación (Testing), se consideró sólo el detalle de los procesos que tienen paridad; es decir, Problem Management (Administración de Problemas) y Application Management (Administración de Aplicaciones).

El proyecto presenta la Metodología creada en base a los procesos de Verificación y Validación de CMMI pareado con los procesos/funciones de ITIL.

Los procesos más importantes que se incluyen en el flujo fueron descritos. El Modelo propuesto tiene inmerso los procesos de apoyo tanto de CMMI como de ITIL (JPalacio, 2006) (Axentia, 2006) (JRaggio, 2004) (OGC, 2007).

4 Solución Propuesta

4.1 Conocimiento de los modelos

Para comenzar se despliega el alcance de la solución propuesta en base a los antecedentes entregados en el Estado del Arte, con el propósito de establecer los puntos en los cuales estos Modelos son semejantes y así poder orientar el análisis y generación de la Metodología de trabajo para el proceso de Testing de Software aplicativos.

Una vez que existe claridad en la definición de los Modelos se comienza con el pareo.

4.2 Proceso de cruce de los modelos

A partir de los 7 libros existentes en ITIL se analizarán:

- Service Support
- Service Delivery
- Application Management

Se realiza un análisis de cada uno de ellos con el fin de encontrar semejanzas a nivel de prácticas concretas con CMMI. La selección de estos libros es en base a conocimiento experto por la certificación obtenida en ITIL. Como la orientación es sobre implantar Testing, se sabe de antemano que estos libros son los que apoyan de manera más directa dicha implantación.

Por su parte el Modelo CMMI cuenta con 22 procesos que incluyen en forma estricta las mejores prácticas a aplicar a nivel de desarrollo de software.

Analizando entonces a nivel de prácticas concretas se pueden mapear algunos procesos de CMMI a ITIL en lo que respecta a soporte y gestión de proyectos.

A continuación se presenta un mapeo de los procesos/funciones de los Modelos CMMI e ITIL. Estos Modelos están enfocados a diferentes áreas, por lo cual guardan escasa relación, pero existen relaciones entre ambos Modelos a nivel de prácticas concretas, aunque pueden generalizarse algunos procesos de soporte y gestión de proyectos.

El Nivel al cual pertenece el Proceso de CMMI en el Modelo está descrito en la tabla como “N+el número del nivel”. La Figura 4.01 muestra los procesos de CMMI pareados a los Procesos/Funciones de ITIL.

Procesos/Funciones ITIL	Áreas de Procesos CMMI
Configuration Management	Configuration Management (N2)
Change Management	Configuration Management (N2)
Release Management	Configuration Management (N2)
Incident Management	
Problem Management	Verification (N3), Causal Analysis & Resolution (N5)
Capacity Management, Availability Management, Continuity Management	Risk Management (N3)
Application Management	Requirements Management (N2), Requirements Development (N3), Technical Solution (N3), Product Integration (N3), Verification (N3), Validation (N3), Integrated Project Management (N3)
Project Management	Project Planning (N2), Project Monitoring and Control (N2), Supplier Agreement Management (N2)

Figura 4.01: Tabla de mapeo CMMI e ITIL

Como se muestra en la Figura 4.01, la base del análisis es ITIL, ya que el enfoque a servicios debe ser la base de la Metodología que se desea generar. Sin embargo como el enfoque a nivel de procesos apunta a Verificación y Validación (Testing), se considerarán sólo el detalle de los procesos que tienen paridad en la tabla anterior; es decir, Problem Management (Administración de Problemas) y Application Management (Administración de Aplicaciones).

No se debe perder de vista que los procesos de Verificación y Validación están soportados en el nivel 3 de CMMI, por lo cual es necesario contar con el apoyo de algunos procesos del nivel anterior (nivel 2) para poder satisfacer en forma correcta las necesidades de la implantación de los procesos de Verificación y Validación del nivel 3.

Los procesos de apoyo para Servicios (ITIL) serán los siguientes:

- Configuration Management (Administración de la Configuración)
- Change Management (Administración del Cambio)
- Release Management (Administración de Release)
- Incident Management (Administración de Incidentes)
- Project Management (Administración de Proyectos)

Los procesos de apoyo para Desarrollo de software (CMMI) serán los siguientes:

- Configuration Management (Administración de la Configuración) → N2
- Project Planning (Planificación de Proyecto) → N2
- Project Monitoring and Control (Monitoreo y Control de Proyecto) → N2
- Supplier Agreement Management (Administración de Proveedores) → N2
- Requirements Management (Administración de Requerimientos) → N2
- Requirement Development (Requerimientos de Desarrollo) → N3
- Technical Solution (Solución Técnica) → N3
- Product Integration (Integración de Producto) → N3
- Integrated Project Management (Administración de Integración de Proyecto) → N3
- Risk Management (Administración de Riesgos) → N3

Es importante mencionar que cuando se habla de procesos de apoyo se entiende que al momento de detallar la Metodología a generar, se nombraran las mejores prácticas y los procesos listados, con el fin de que se tenga claridad de cómo ellos interactúan en el proceso de implantación; sin embargo no se entregará detalle de cómo se deben implantar, a diferencia de los procesos específicos de Verificación y Validación.

La Metodología obtenida es presentada en la Implementación de la misma.

La Figura 4.02 presenta la estructura definida para el Modelo Propuesto y se incluyen los proceso tanto del cruce como los de apoyo de CMMI e ITIL. Cabe destacar que el Modelo Propuesto fue generado en Enterprise Architect - UML CASE Tool - Desktop, Professional and Corporate editions, Version 7.0.

La simbología de los colores utilizados en el Modelo se encuentra en la Figura 4.02 en la parte superior derecha.

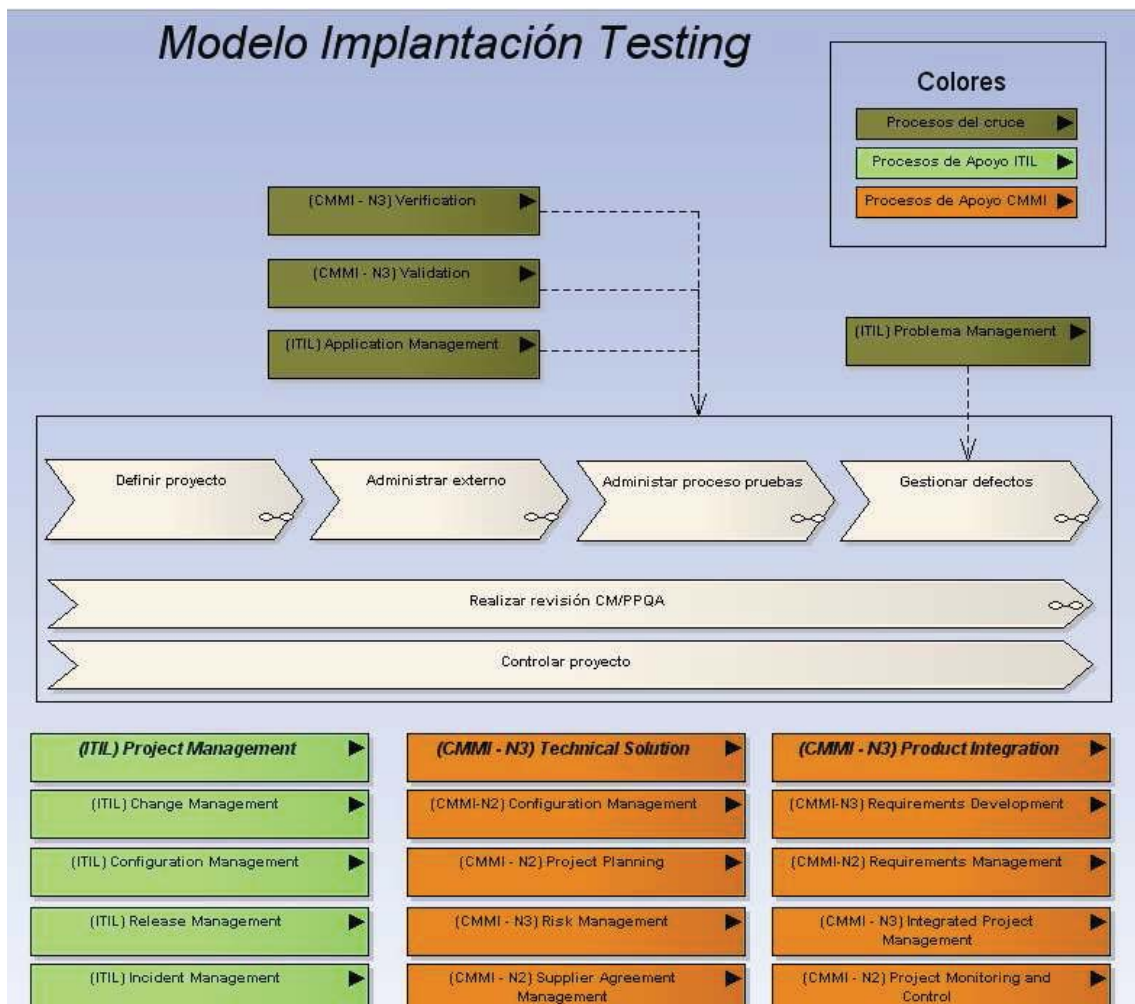


Figura 4.02: Metodología propuesta

En la Metodología se exponen las prácticas específicas y se detalla el proceso de cómo implantar la Metodología. El texto de la Metodología fue apoyado con un Diagrama de Flujo, el cual indica paso a paso como implantar el proceso de Testing en una empresa cualquiera.

A partir de la Metodología Propuesta se presenta la evaluación de su aplicación a un caso real, definiendo como punto de partida las métricas que se considerarán en dicha evaluación.

Al aplicar la Metodología al caso real, se refinaron algunas de las plantillas inicialmente generadas como apoyo.

El proceso de implementación consiguió como objetivo refinar y generalizar aún más los documentos de apoyo, ya que se amplió el espectro inicialmente pensado para tales documentos.

4.3 Definición de métricas para la evaluación final

A continuación se presentan los conceptos utilizados para la definición de métricas:

- Existen cuatro medidas para cuantificar: Caracterizar, Evaluar, Predecir y Mejorar.
 - 1) Medida: Valor asignado a un atributo de una entidad mediante una medición. Por ejemplo: 35.000 líneas de código.
 - 2) Medición: Es el acto de determinar una medida. Por ejemplo: Silvana será la encargada de medir las LDC de cada módulo del sistema.
 - 3) Métrica: Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Incluye el método de medición. Por ejemplo: La productividad de este proyecto fue de 500 líneas/personas-mes (LDC/pm).
 - 4) Indicador: Es una métrica o combinación de métricas que proporcionan una visión profunda del proceso de software. Por ejemplo: la productividad media de nuestra empresa es de 500 (LDC/pm).

Las métricas ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. El proceso para intentar mejorarlo y el producto para intentar aumentar su calidad (LLeón-GPuglla, 2008).

- Por otro lado es importante saber que muchas veces el gerente de proyecto, se encuentra en la disyuntiva de dar o no el peso correcto a cada una de las variables que afectan el desarrollo de un proyecto en particular.

Existen restricciones tradicionales, como en cualquier empresa, los proyectos necesitan ser ejecutados y entregados bajo ciertas restricciones. Tradicionalmente, estas restricciones han sido alcance, tiempo y costo. Esto también se conoce como el Triángulo de la Gestión de Proyectos, donde cada lado representa una restricción. Un lado del triángulo no puede ser modificado sin impactar a los otros. Un refinamiento posterior de las restricciones separa la calidad del producto del alcance, y hace de la **calidad** una cuarta restricción.

- La restricción de tiempo se refiere a la cantidad de tiempo disponible para completar un proyecto.
- La restricción de costo se refiere a la cantidad presupuestada para el proyecto.
- La restricción de alcance se refiere a lo que se debe hacer para producir el resultado final del proyecto.

Estas tres restricciones son frecuentemente competidoras entre ellas: incrementar el alcance típicamente aumenta el tiempo y el costo, una restricción fuerte de tiempo puede significar un incremento en costos y una reducción en los alcances, así como también un presupuesto limitado puede traducirse en un incremento en tiempo y una reducción de los alcances.

La disciplina de la gestión de proyectos consiste en proporcionar las herramientas y técnicas que permiten al equipo de proyecto (no solamente al gerente del proyecto) organizar su trabajo para cumplir con todas esas restricciones.

4.3.1 Tiempo

El tiempo se descompone para propósitos analíticos en el tiempo requerido para completar los componentes del proyecto que es, a su vez, descompuesto en el tiempo requerido para completar cada tarea que contribuye a la finalización de cada componente. Cuando se realizan tareas utilizando gestión de proyectos, es importante partir el trabajo en pedazos menores para que sean fáciles de seguir.

4.3.2 Costo

El costo de desarrollar un proyecto depende de múltiples variables incluyendo costos de mano de obra, costos de materiales, administración de riesgo, infraestructura (edificios, máquinas, etc.), equipo y utilidades. Cuando se contrata a un consultor independiente para un proyecto, el costo típicamente será determinado por la tarifa de la empresa consultora multiplicada por un estimado del avance del proyecto.

4.3.3 Alcance

Requerimientos especificados para el resultado final. La definición global de lo que se supone que el proyecto debe alcanzar y una descripción específica de lo que el resultado final debe ser o debe realizar. Un componente principal del alcance es la *calidad* del producto final. La cantidad de tiempo dedicado a las tareas individuales determina la *calidad* global del proyecto. Algunas tareas pueden requerir una cantidad dada de tiempo para ser completadas adecuadamente, pero con más tiempo podrían ser completadas excepcionalmente. A lo largo de un proyecto grande, la *calidad* puede tener un impacto muy significativo en el tiempo y en el costo (o viceversa) (PMBOK, 2004).

La Figura 4.03 muestra las restricciones anteriormente descritas. Éstas se pueden representar en un triángulo, el cual aumenta o disminuye sus ángulos dependiendo de la importancia que se le aplique a cada uno de sus vértices (restricciones). Lo anterior refleja que en el caso que se le asigne mayor o menor importancia a uno de sus ángulos, éstos impactarán en la importancia de los demás vértices.

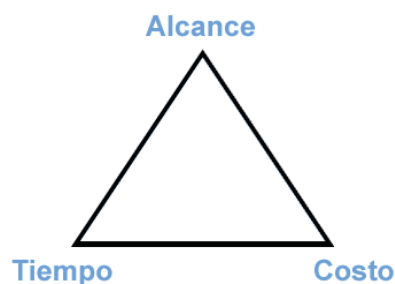


Figura 4.03: Triángulo de restricciones

A partir de los puntos antes descritos; Métricas y Triángulo de restricciones, se definen las siguientes métricas para la medición del éxito en la aplicación de la Metodología Propuesta a nivel de proyecto de Testing:

- El tiempo siempre será la variable menos movable con respecto a las restricciones, ya que todo proyecto tiene como límite una fecha, ya sea por necesidades del cliente o por imposición política (frecuente en empresas públicas).
- Una vez que el tiempo está establecido es importante saber el presupuesto con el que el cliente cuenta, ya que muchas veces es la segunda variable impuesta a nivel de requerimientos. El presupuesto puede estar dado mensual, semanal o incluso diario, dependiendo del tipo de proyecto.
- Finalmente queda el alcance, en el cual el cliente indica su óptimo con respecto al requerimiento en general. A parte de los costos que involucra el desarrollo del proyecto (específicamente lo que se evaluará en la Metodología), se debe considerar la calidad asociada al proceso de desarrollo, el cual también debe ser considerado en esta restricción.

Para el Testing de un aplicativo, teóricamente es prudente considerar a lo menos el 30% del Universo de pruebas posibles; es decir, si se estiman 100 pruebas de negocio para el aplicativo, lo prudente es testear al menos 30 de dichas combinatorias de pruebas de negocio. Más adelante se explicará la forma de estimar el Universo para luego indicar el cálculo del 30% en base a pruebas de negocio base y específicas.

Del 30% que será testeado, se procederá a establecer en el Plan de Pruebas, las políticas para la aceptación del aplicativo; es decir, se indicarán los porcentajes de aceptación para considerar efectivo el esfuerzo aplicado en el Testing.

Por experiencias en proyectos de Testing, lo mínimo recomendable como política de aceptación del 30% testeado será un 80% de aceptación SIN incidentes invalidantes. Muchas veces este porcentaje no es alcanzado, por malas prácticas en el desarrollo del software, así como también por la inexperiencia en el proceso de testeo.

Lo anterior indica que en el ejemplo de las 100 pruebas como Universo estimado, 24 pruebas (de las 30 seleccionadas) deben ser exitosas y no deben existir pruebas con invalidantes al momento de la ejecución. Por lo anteriormente expuesto, entonces se declaran 3 métricas para la evaluación de la implantación de la métrica, las cuales se listan en la Figura 4.04.

N°	Medición	Medida	Métrica
1	Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto.	HH	Esfuerzo Estimado v/s Esfuerzo Real
2	Se medirá el cumplimiento del costo establecido para el Testing del proyecto.	UF	Costo Estimado v/s Costo Real
3	Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.	%	% Establecido v/s % Obtenido

Figura 4.04: Métricas definidas para validación de la implantación de metodología propuesta

Las métricas obtenidas deberán tender en su mayoría a cero una vez calculado su diferencial. Luego de calculado su diferencial, éstos deberán ser analizados por el administrador del proyecto, quien tendrá la noción del proyecto general con respecto a los diferenciales obtenidos. Lo anterior, siempre con la mirada de "un traje de novia con excelencia en calidad al costo estimado, no sirve de nada si es entregado después de la fecha establecida".

Para medir la aplicación de la Metodología Propuesta se sugiere la siguiente métrica:

- 1) Porcentaje de Apego a la Metodología: Esta métrica tiene relación con identificar en que porcentaje se ha aplicado la Metodología Propuesta al concluir el Proyecto de Testing.

Las medidas utilizadas en el cálculo de la métrica se desprenden de la expertise obtenida en el desarrollo de la Propuesta Metodológica, así como también en la aplicación consecutiva de éstas en proyectos diversos para mejorarla.

La Figura 4.05 se refiere a la importancia de la plantilla diseñada para el apoyo al proceso de implantación. Esta importancia está dada por la experiencia en el proceso, pensando siempre en el cumplimiento del objetivo propuesto: apego a la Metodología.

Importancia Metodológica	
Muy Importante	4
Importante	3
Media	2
Baja	1

Figura 4.05: Métrica "Importancia Metodológica"

La Figura 4.06 entrega la relación de la planilla al proceso de implantación bajo la visión de Modelo diseñado del proceso de Testing; es decir, si la planilla apoya al proceso de CMMI, ITIL o es parte del apoyo a un proceso del cruce realizado entre ambos Modelos.

Propuesta Metodológica	
Cruce	5
CMMI	3
ITIL	2

Figura 4.06: Métrica “Propuesta Metodológica”

Una vez definidas las medidas se define la forma en la cual se calculará el apego a la Metodología. A continuación (Figura 4.07) se presenta la tabla con los campos ya asignados para el cálculo de la métrica de apego.

Columna A	Columna B	Columna C		Columna D	Columna E
Estructura	Plantilla	Propuesta Metodológica		Importancia en Metodología	Apego
0. Entradas	Informe de Diseño proveedor(Extracción).doc	CMMI	3	4	12
	PlanificaciónSeguimientoSemanal.mpp	CMMI	3	4	12
	Visión y Alcance.doc	CMMI	3	2	6
1. Análisis	Matriz_de_Evaluacion_de_Propuestas.xls	CMMI	3	2	6
	Estimación Modelo de Servicios.xls	Cruce	5	4	20
	LB_Requerimientos_Calidad.xls	ITIL	2	3	6
2. Diseño	Definición CPL.xls	ITIL	2	4	8
	TF Diseño.xls	ITIL	2	3	6
	Plan_de_CM.doc	CMMI	3	1	3
	Plan_de_Pruebas.doc	CMMI	3	3	9
3. Ejecución	TF Ejecución.xls	Cruce	5	3	15
4. Gestionar Defectos	TF Entidad-Defecto.xls	Cruce	5	3	15
5. Informes	Plan de Negocio.xls	Cruce	5	4	20
	Cierre Proyecto.xlsx	Cruce	5	2	10
	RESULTADO TESTING Ciclo.doc	Cruce	5	2	10

Figura 4.07: Tabla “Cálculo de métricas”

- 1) La Columna A se refiere a la estructura dada para organizar las planilla diseñadas. Concuerdan con el ciclo del proyecto que se testea.
- 2) La Columna B identifica cada una de las plantillas diseñadas para el proceso de implantación del Testing.
- 3) La Columna C clasifica cada planilla diseñada según la medida explicada anteriormente (Propuesta Metodológica).
- 4) La Columna D clasifica cada planilla diseñada según la medida explicada anteriormente (Importancia en Metodología).
- 5) La Columna E es la multiplicación de la Propuesta Metodológica (Columna C) por la Importancia Metodológica (Columna D).
- 6) En la Figura 4.08 se explica la segunda parte de la Métrica en base al apego de la Metodología Propuesta, la cual considera principalmente la aplicación de las métricas propuestas (Figura 4.05 y 4.06).

Columna B	Columna F	Columna G	Columna H	Columna I	Columna J	Columna K	Columna L
Plantilla	Aplicada	Apego por Aplicada	Distribución Importancia	Refinada	Multiplica (Distribución * Refinado)	Apego por Fase	Porcentaje por Estructura
Informe de Diseño proveedor(Extracción).doc							0. Entradas
PlanificacionSeguimientoSemanal.mpp							
Visión y Alcance.doc							
Matriz_de_Evaluacion_de_Propuestas.xls							1. Análisis
Estimación Modelo de Servicios.xls							
LB_Requerimientos_Calidad.xls							
Definición CPI.xls							2. Diseño
TF Diseño.xls							
Plan_de_CM.doc							
Plan_de_Pruebas.doc							3. Ejecución
TF Ejecución.xls							
TF Entidad-Defecto.xls							4. Gestionar Defectos
Plan de Negocio.xls							5. Informes
Cierre Proyecto.xlsx							
RESULTADO TESTING Ciclo.doc							

Figura 4.08: Tabla “Apego de la metodología propuesta”

- 1) La Columna F debe ser llenada con 1 si la planilla fue aplicada en el proceso de Testing o con uno 0 en el caso de que no se haya aplicado.
- 2) La Columna G multiplica la Columna E (Apego) por la Columna F (Aplicada) con el fin de obtener el apego real en la Metodología aplicada por cada una de las plantillas.
- 3) La Columna H es la distribución de la importancia que se calcula considerando Columna G (Apego por Aplicada) dividido por la sumatoria de cada plantilla que compone una estructura (Columna A).
- 4) La Columna I se debe llenar con un 1 para representar que SI se ajustará la planilla propuesta, al momento de su aplicación en lo que respecta a mejoras para su posterior utilización, o bien con un 0,9 en el caso que NO haya sufrido modificaciones la planilla propuesta al momento de su aplicación en lo que respecta a mejoras para su posterior utilización.
- 5) La Columna J es la multiplicación de Columna H (Distribución Importancia) por Columna I (Refinada) para así obtener la medida final en la siguiente columna.
- 6) La Columna K es la sumatoria de toda la columna J de cada plantilla existente por Estructura (Columna A). Esta medición entrega el apego a la Metodología por Estructura definida.
- 7) La Columna L es el recordatorio de la composición por Estructura.

El porcentaje de Apego a la Metodología se calcula sumando todas las medidas obtenidas por estructura (Columna K) dividida por 6 (cantidad de estructuras: Entradas, Análisis, Diseño, Ejecución, Gestionar Defectos e Informes).

Es importante mencionar que se realizó el ejercicio considerando el mínimo recomendable en la utilización de las plantillas en la implementación de la Metodología, llegando a la conclusión de que al aplicar la métrica de Apego a la Metodología se puede observar que el escenario peor es a lo menos un 80% de apego a la Metodología como porcentaje general, ya que sin este porcentaje no se estaría consiguiendo el objetivo de aportar en el proceso de implementación del Testing.

5 Implantación de la metodología propuesta y aplicación de plantillas de apoyo

A partir de la Metodología Propuesta se definen las plantillas de apoyo por proceso en el siguiente orden, considerando siempre el apoyo de los procesos del cruce, procesos de apoyo ITIL y procesos de apoyo CMMI. Es importante mencionar que se creó un nuevo proceso de Control de Proyecto que permitirá ir analizando las métricas propuestas anteriormente.

Estas plantillas apoyan el proceso de Testing (Verificación y Validación).

1. Definir Proyecto
2. Administrar Externo
3. Administrar Proceso Pruebas
4. Gestionar Defectos
5. Realizar Revisión CM/PPQA
6. Controlar Proyecto

Las planillas de apoyo se verán reflejadas como documentos asociados al proceso específico según se muestra en la Figura 5.01.

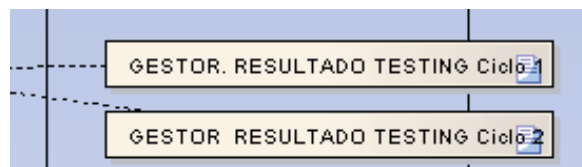


Figura 5.01: Identificación planillas de apoyo

Es importante indicar que el documento contiene el nombre de la planilla de apoyo que se encuentra en los anexos. Además la línea punteada relaciona el documento al proceso que lo utiliza.

5.1 Proceso para aplicar la metodología

La generación de esta Metodología nace de la necesidad y experiencia de establecer un método en el trabajo de la estimación y aplicación del Testing Funcional y No Funcional en un Área de Calidad de una EMPRESA.

Históricamente se sabe que un proyecto nunca es igual a otro, siempre existen diferencias significativas en el desarrollo de éstos, por lo que para la aplicación de la Metodología inicialmente se consideró incorporar como prerrequisitos mínimos los entregables que aplican como entrega por parte del equipo de proyecto. Por lo

anteriormente indicado se hacía cada vez más necesario establecer patrones de semejanza en la aplicación de un método para levantar un “proyecto” de Testing sobre un proyecto de desarrollo de software.

A medida que se analiza la generación de la Metodología se comienzan a exigir como Área, entregables mínimos a considerar en un proyecto de desarrollo; como por ejemplo, documentos de especificación de requerimientos, documentos de diseño, y documentos formales de aceptación de los documentos antes indicados, con el fin de establecer líneas base claras a la hora de estimar y planificar el proyecto de Testing.

Una vez generada la Metodología, se escogió un proyecto real, de una de las áreas con las cuales se llevaban muchos trabajos históricos, lo que ayudó, en gran medida, a contar con los documentos mínimos necesarios.

Al aplicar esta Metodología, se podrá apreciar que muchas veces los documentos mínimos no son considerados formalmente en un proyecto de desarrollo. Esto es una muy mala práctica, ya que no se puede establecer un Método sobre un proyecto que viene mal “engendrado”. A pesar de lo anteriormente expuesto, las plantillas generadas servirán de apoyo para las etapas de Requisitos, Análisis y Diseño de un proyecto de desarrollo de software.

El proceso de aplicación comienza con la exigencia de los documentos mínimos y la definición (estimación y planificación) del proyecto de Testing que se apegará a las condiciones del proyecto de desarrollo.

Se establecieron Métricas que se consideran evaluar al término de la aplicación de la Metodología, con el fin de mostrar el real aporte de su aplicación.

Las Plantillas aplicadas, son presentadas en el ANEXO 1. A medida que se utilizan en la Metodología serán referenciadas. Algunas de ellas se detallarán, con el fin de describir las transformaciones que han sufrido en el proceso de aplicación al proyecto real.

Es importante mencionar que además se incorporaron algunos documentos más que no estaban considerados inicialmente como plantillas, pero que en el proceso se pudieron percibir como gran apoyo.

Se recuerda que los procesos en azul tienen directa relación con la posible utilización de la Metodología para procesos apoyados con herramientas existentes en el mercado; en este caso, específicamente con Mercury de HP.

Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.1 Proceso “Definir proyecto”

En la Figura 5.02 se puede ver que no existen plantillas de apoyo al proceso. En su completitud este proceso apoya en la generación de un nuevo proyecto en el caso de que se utilice una herramienta de Prueba.

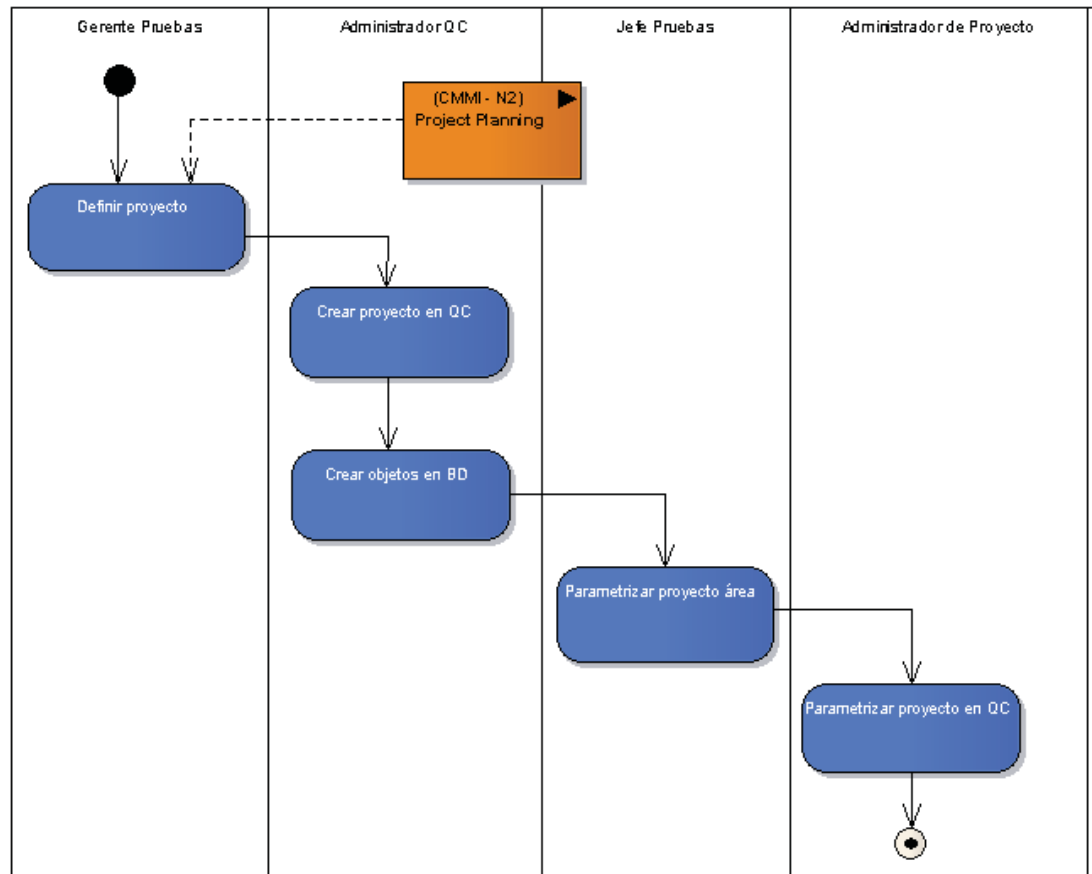


Figura 5.02: Proceso “Definir proyecto”

Proceso	Definir proyecto
Descripción	<p>Una vez que ha sido aprobado por el cliente la propuesta de servicio:</p> <ul style="list-style-type: none"> - Se define el nombre del proyecto y a qué línea de negocios pertenece. - Definición de las áreas de calidad involucradas y los Jefe de Área/Prueba involucrados. - Se debe definir las contrapartes válidas (Jefes de Proyectos) a las cuales se les derivarán los defectos y si accederán QC. - Se define la política de respaldo y Planes de Pruebas - Checklists serán versionados en Subversion (herramienta de versionamiento) indicando el repositorio. - Se definen los aplicativos y las versiones para cada uno. - Se entrega la información anterior al "Administrador QC" para crear el proyecto en QC.

Proceso	Crear proyecto en QC
Descripción	<p>Se realizan las tareas asociadas a la creación del proyecto en QC, se ajusta una copia de la plantilla Domain: Base y Project: Proyecto basándose en la especificación entregada por el Jefe de Pruebas.</p> <ul style="list-style-type: none"> - Conectarse como Administrador de QC a QC Project Administration - Crear un proyecto dentro del dominio correspondiente. (QC Project Administration- Site Projects). Elegir la opción 'Create a project by copying data from an existing project' - Seleccionar el proyecto plantilla mencionado Domain: Base y Project: Proyecto - Chequear Customization, Requirements, Tests, Test Sets, Public Favorite Views. - Especificar el servidor de BD donde residirá el proyecto. - Asignar el Administrador del Proyecto (Se asume un mismo Administrador para todos los proyectos) - Activar el proyecto (Chequear Activate project) - Dado que los proyectos se guardarán distribuidamente en la BD y en directorios del S.O. no chequear 'Store project's repository in the database' - Dado que no se usará versionamiento, debido a que éste sólo cubre las pruebas y no el resto de las entidades, no chequear 'Create a Version Control database' - Chequear por conectividad a la BD usando el botón Ping. - Crear los usuarios que aún no existen y hayan sido especificados en la Definición del Proyecto .(QC Project Administration - Site Users) - Conectarse a Quality Center como el Administrador de Proyecto al nuevo proyecto. - Crear los aplicativos y releases. Renombrar la carpeta raíz Releases con el nombre del proyecto. - Informar al administrador de la BD, que se generó una nueva BD, con el fin de actualizar el job de respaldo.

Proceso	Crear objetos en BD
Descripción	<p>Crear los objetos cuyos scripts están en el tab Files usando el usuario td, previa especificación de la BD en ellos.</p> <p>Crear sólo las funciones get_domainname y get_projectname pues éstas ya fueron creadas en la BD de Administración de QC.</p>

Proceso	Parametrizar proyecto área
Descripción	<p>En esta tarea se especifica las parametrizaciones propias de cada Área de Calidad:</p> <ul style="list-style-type: none"> - Definir usuarios del área que participarán en el proyecto. - Se identifica si el proyecto necesita customizaciones especiales. Estas pueden implicar: <ol style="list-style-type: none"> (1) Creación de grupos usuarios especiales (2) Creación de nuevos campos para las entidades (3) Creación o mantención de Listas de Lookup.

Proceso	Parametrizar proyecto en QC
Descripción	<p>Una vez que el proyecto ha sido creado se realiza su parametrización:</p> <ul style="list-style-type: none"> - Crear los grupos de usuarios necesarios (QC Project Customization - Groups). - Asignar los usuarios al proyecto (QC Site Administration - Site Users - User Projects) - Si QC se versiona en Subversion y no se está usando sólo la cuenta de Subversion Default, agregar los mappings entre usuarios QC y Subversion. Nótese que si el versionamiento ya estaba activo, debe ser desacoplado, cambiado y vuelto a acoplar. - Realizar las customizaciones (QC Project Customization)

5.1.2 Proceso “Administrar externo”

El proceso “Administrar Externo” se basa fundamentalmente en la búsqueda de externos para el apoyo del Testing. En el caso que sea necesaria la utilización de externos es importante tener realizada la estimación del proyecto es lo que respecta al proceso de Testing.

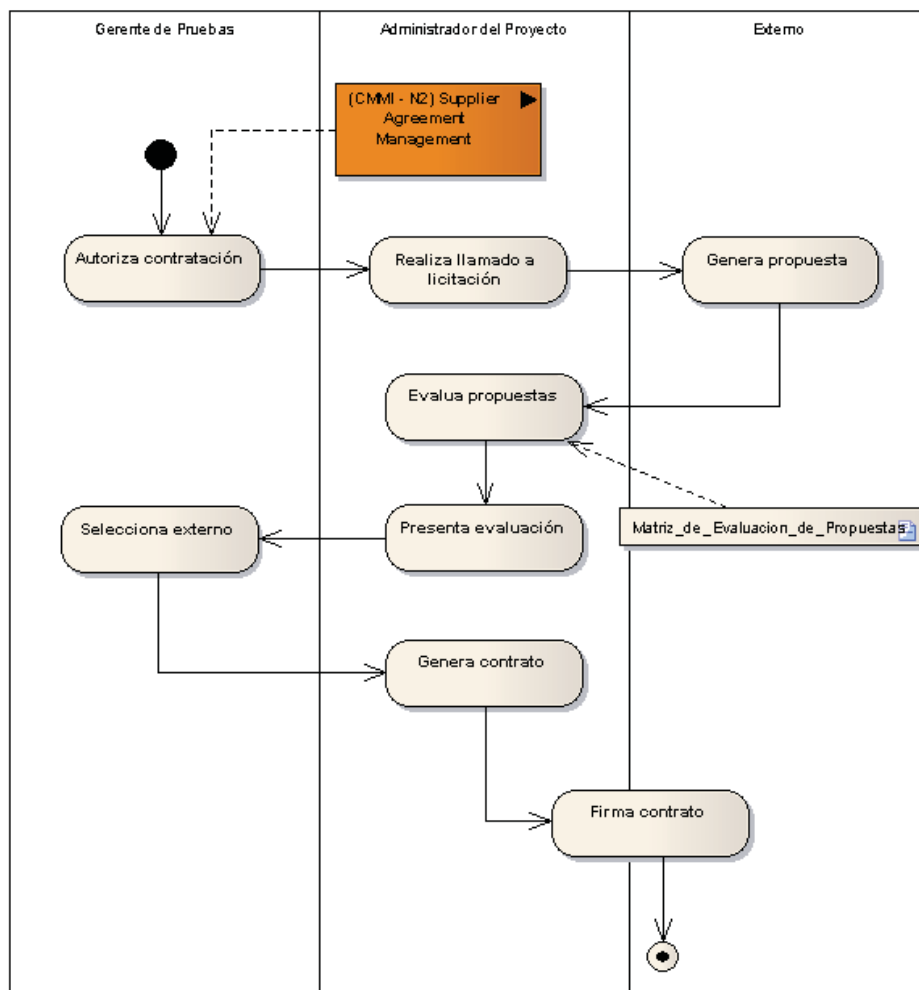


Figura 5.03: Proceso “Administrar externo”

Proceso	Autoriza contratación
Descripción	Una vez que el equipo de Gestión de Proyecto autoriza la utilización de recursos externos. El equipo de proyecto debe tener presente que en estos casos el costo aumenta, ya que contratar una “buena” empresa de Testing involucra experiencia, eficacia y eficiencia en el proceso.

Proceso	Realiza llamado a licitación
Descripción	El equipo de Proyecto entrega la potestad al equipo de Testing en el llamado a licitación, pudiendo ésta ser de carácter público o reservado, dependiendo de la estrategia de empresa por la que se rija el equipo de Testing. Es importante mencionar que un buen llamado a licitación consiste en precisar el servicio que se solicita, tanto en tiempo, costo, calidad y por sobre todo en lo que respecta a los rendimientos tanto del diseño de los casos como en la ejecución de ellos. Otro punto importante a considerar es explicar en detalle la definición de los conceptos base para poder realizar el cálculo del Testing, que en este caso sería la definición de Caso de Prueba Integrado, Casos de Prueba Unitarios o Casos de Negocio. Las definiciones antes mencionadas están especificadas en el Glosario de este documento.

Proceso	Genera propuesta
Descripción	En base al llamado de licitación, el externo genera su propuesta de servicio, en la cual debe especificar claramente el cumplimiento de plazos, restricciones del servicio en caso de necesidades de horas extras y el costo asociado.

Proceso	Evalúa propuestas (*)
Descripción	Una vez que los proveedores de servicio entregan sus propuestas, el equipo de Testing debe comenzar con la evaluación. Es en este punto donde aparece nuestra primera Plantilla de apoyo (ver ANEXO 1 “Matriz_de_Evaluación_Propuestas”).

Proceso	Presentar evaluación
Descripción	Una vez aplicada la matriz de evaluación, es necesario presentar los resultados al equipo de Proyecto y realizar las recomendaciones respecto a dicha evaluación.

Proceso	Selecciona externo
Descripción	El equipo de Proyecto entonces está en condiciones de realizar la selección del externo para que participe en el proceso de Testing, pudiendo ser ésta el diseño y/o ejecución de las Pruebas.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

Proceso	Generar contrato
Descripción	<p>El equipo de Testing está en condiciones de generar el contrato de servicio con el externo.</p> <p>Es importante recomendar que dicho contrato esté orientado a entregables de trabajo relacionados con el proceso del Testing en una primera vista, pero que en segunda línea se especifique claramente la cantidad de HH relacionadas al servicio.</p> <p>Se debe tener especial cuidado en las excepciones que se acuerden y en el aviso y satisfacción del servicio en lo que respecta a horas de trabajo extraordinario.</p>
Proceso	Firma contrato
Descripción	La firma del contrato debe ser entre equipo de Testing y externo, ya que las exigencias con respecto a respuesta en la entrega de servicio son de responsabilidad del equipo de Testing.

5.1.3 Proceso “Administrar proceso de pruebas” PARTE 1

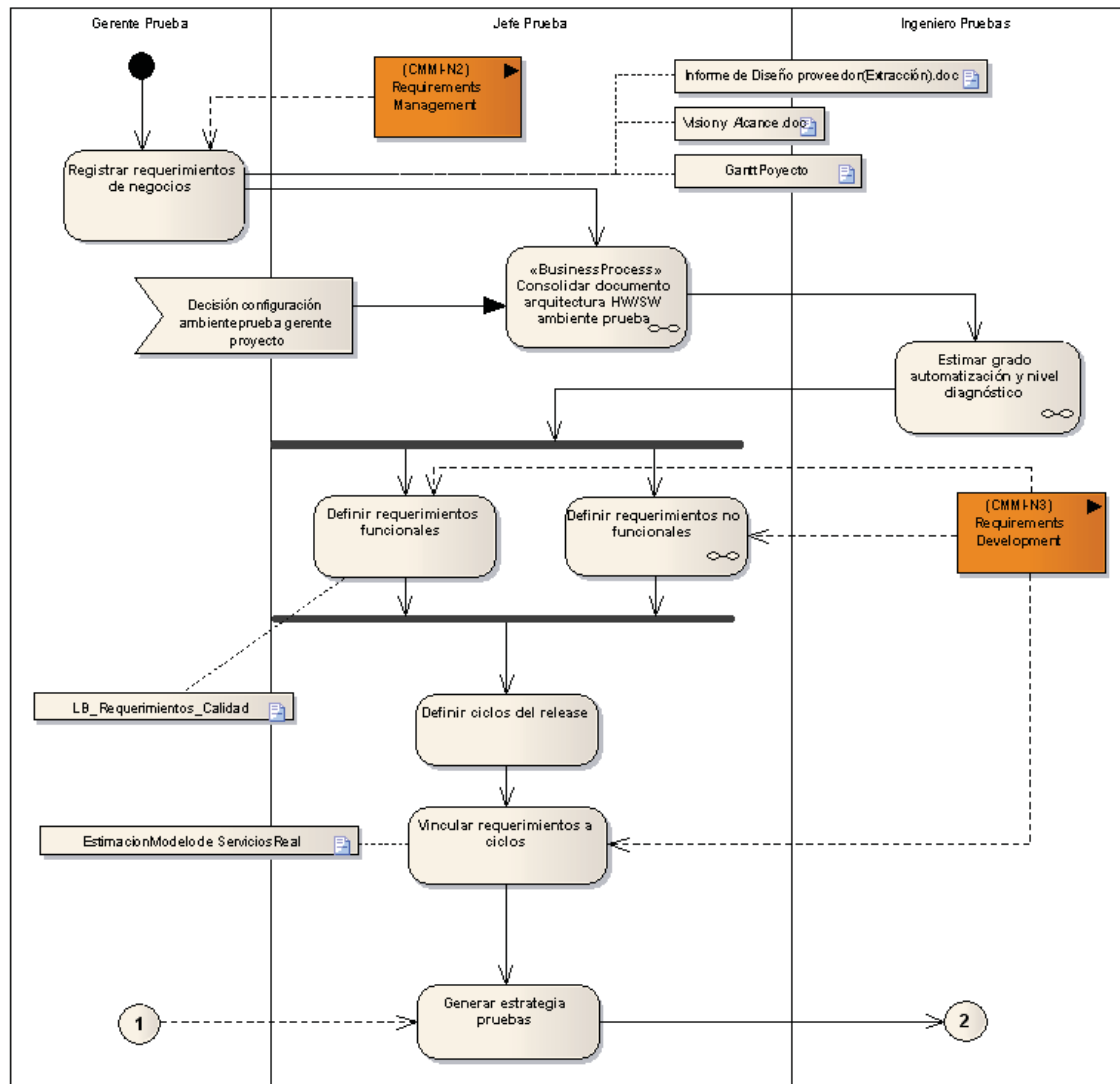


Figura 5.04: Proceso “Administrar proceso de pruebas” PARTE 1

Proceso	Registrar requerimientos de negocios (*)
Descripción	<p>El primer documento que debe aprobar el equipo de proyecto al desarrollador del aplicativo a testear es el de Visión y Alcance, el cual debe especificar claramente lo que se va a desarrollar. Siempre en términos generales en lo que respecta al aplicativo. En la plantilla de apoyo ANEXO 1 “Visión y Alcance” se explica el contenido esperado como mínimo en dicho documento.</p> <p>Luego que el documento de Visión y Alcance está acordado y aprobado por las partes es hora de que el desarrollador comience con el análisis para continuar con el diseño, otro de los input importantes a la hora de analizar qué testear.</p> <p>El proveedor de desarrollo debe entregar al equipo de proyecto, el documento de diseño del aplicativo que se va a testear. A partir de este documento el equipo de Testing debe ser capaz de entender y reflejar en la planilla de requerimientos, todos los requisitos que el aplicativo debe contener.</p> <p>El Informe de diseño debe contener todo lo indicado en la plantilla de apoyo ANEXO 1 “Informe Diseño del Proveedor”.</p> <p>Una vez aprobado el documento de Diseño, el Proveedor ya está en condiciones de declarar tiempos claros y precisos en una Carta Gantt, sobre la cual el equipo de proyecto y el equipo de Testing deben comenzar a analizar los instantes en los cuales el equipo de Testing podrá intervenir, así como también las posibles restricciones de tiempo y costo, con el fin de poder dilucidar cuál será el alcance de la claridad del Testing.</p> <p>La plantilla siguiente muestra una planificación tipo que puede ser utilizada como referencia para analizar las intervenciones del equipo de Testing. Ver ANEXO 1 “Planificación_Seguimiento_Semanal”.</p> <p>A nivel de equipo de Testing, a partir de los 3 documentos antes mencionados, debe ser capaz de definir los requerimientos del aplicativo, por lo anterior se deben realizar los siguientes pasos:</p> <p>Definir el requerimiento como carpeta, especificando:</p> <ol style="list-style-type: none"> 1. Nombre 2. Tipo de Requerimiento como de Negocios. 3. Descripción 4. Autor 5. Prioridad en la implementación:

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<p>1-Baja 2-Media 3-Alta 4-Muy Alta</p> <p>6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)</p> <p>Nota: Es posible agregar Attachments a los Requerimientos también.</p> <p>En el caso de Pruebas No Funcionales es recomendable que los requerimientos de negocios tengan la forma de SLAs estándares.</p> <p>NOTA: LOS REQUERIMIENTOS DE NEGOCIO SON DE ALTO NIVEL, Y SU ANALISIS PUEDE INVOLUCRAR SU DESCOMPOSICION EN REQUERIMIENTOS FUNCIONALES, DESEMPEÑO, SEGURIDAD, ETC.</p>
--	--

Proceso	Consolidar documento arquitectura HW/SW ambiente prueba
Descripción	<p>Si bien el área de calidad no es el encargado de generar este documento en forma íntegra si debiera consolidarlo para sí, previa ejecución de:</p> <ul style="list-style-type: none"> - Recolectar las especificaciones de Arquitectura que generen otras áreas de la Empresa, el Cliente o Terceros. - Generar la especificación del dispositivo GUI, su cantidad, ubicación física y conectividad. <p>El documento debe contener todas las componentes detalladas en el Diagrama "Ambiente_Prueba". Dado que los ambientes para Pruebas Funcionales y No Funcionales son potencialmente distintos se recomienda elaborar un documento para cada uno.</p>

Proceso	Estimar grado automatización y nivel diagnóstico
Descripción	<p>En esta tarea se estima que porcentajes de las Pruebas serán realizadas automática y manualmente. Dicha estimación se realiza basándose en:</p> <ul style="list-style-type: none"> - Diseños Preliminares del Sistema. Por ejemplo si el "Dispositivo GUI" es una PALM y no hay emulador confiable, las Pruebas funcionales deberán ser realizadas manualmente, pero puede que de todos modos sea factible realizar las Pruebas no funcionales. - Pruebas sobre el Aplicativo en el caso que esté disponible usando QTP y LR para las Pruebas funcionales y no funcionales respectivamente. <p>En el caso No Funcional si no es factible realizar las Pruebas automáticamente, probablemente la opción manual sea demasiado cara.</p>

Proceso	Definir requerimientos funcionales (*)
Descripción	<p>La planilla de apoyo "LB_Requerimientos_Calidad" en ANEXO 1 entrega el detalle del llenado para obtener el Universo estimado.</p> <p>Esta plantilla debe estar aprobada por el equipo de proyecto y debe sumar todos los casos de Prueba relacionados a los requerimientos con estado Declarado.</p> <p>Se definen los requerimientos funcionales detallando los atributos mencionados a continuación, si es necesario se crean carpetas que agrupen los requerimientos por temas.</p> <ol style="list-style-type: none"> 1. Nombre 2. Especificar Tipo de Requerimiento (Negocios, Funcional, No Funcional) 3. Descripción 4. Autor 5. Prioridad en la implementación: <ol style="list-style-type: none"> 1-Baja 2-Media 3-Alta 4-Muy Alta 6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)

Proceso	Definir requerimientos No Funcionales
Descripción	<p>Rendimiento, en los Requerimientos de Negocios debería existir la definición de SLA de tiempo de respuesta vinculado a un conjunto de transacciones, sin embargo se dan tres posibles situaciones:</p> <ul style="list-style-type: none"> - El conjunto de transacciones es adecuado para el tiempo asignado a Pruebas de Carga en el proyecto. Usualmente el tiempo alcanza para considerar entre 10 y 30 transacciones. - El conjunto de transacciones es muy amplio y no es factible realizar todas las mediciones en términos económicos. - El conjunto de transacciones ha sido enunciado en forma general y no se han identificado las transacciones.

Proceso	Definir ciclos del release
Descripción	Definir los "Ciclos del Release" indicando los "Conjunto de Pruebas" preliminares que los conforman, por ejemplo el primer "Ciclo de Pruebas" debiera incluir siempre un conjunto de "Pruebas Trazadoras" con el fin de establecer si el ambiente está operativo o no.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

Proceso	Vincular requerimientos a ciclos (*)
Descripción	<p>Esta es la instancia para utilizar la planilla de Apoyo “Estimación Modelo de Servicios” del ANEXO 1. Es importante mencionar que esta plantilla no se podrá utilizar si no está cerrada la planilla de apoyo de LB Requerimientos Calidad, ya que ésta tiene el input más importante para la estimación del servicio.</p> <p>En la plantilla de apoyo además se definen las fases y ciclos de Prueba así como también los recursos relacionados.</p> <p>Se deben vincular los requerimientos a al menos un "Ciclo de Prueba" con el fin de:</p> <ul style="list-style-type: none"> - Determinar en cualquier instante si los requerimientos están siendo cubiertos o no - Generar esbozo de Pruebas en función de los requerimientos asociados a un ciclo. <p>Un requerimiento puede ser vinculado a un Release, en este caso, todos los "Ciclos de Prueba" dentro del Release serán vinculados, y eso implica que debe haber un "Conjunto de Pruebas" al menos dentro de cada Ciclo que valida el requerimiento.</p> <p>La cobertura de Prueba en un ciclo queda definida por la liberación de funcionalidades nuevas y por los defectos ocurridos en el ciclo anterior (en el caso que exista), por lo tanto al principio del proyecto, sólo se vincularán los requerimientos asociados a funcionalidades nuevas al ciclo en que se prueban, y con posterioridad dependiendo de los defectos se vincularán requerimientos a ciclos que contengan Pruebas de Regresión.</p>

Proceso	Generar estrategia pruebas
Descripción	<p>Este documento debe ser un compendio del estado del proyecto hasta este momento, considerando:</p> <ul style="list-style-type: none"> - Roles y personas - Requerimientos de Pruebas - Ciclos de Pruebas - Protocolo de Suspensión de las Pruebas - Conclusiones de Estudio Factibilidad de Automatización. - Referencias a los documentos de Arquitectura de HW/SW. - Formato Plan de Pruebas o Operatoria al respecto - Formato Informe de Resultados de Pruebas. - Manejo de Defectos. - Severidad de Defectos. - Riesgos <p>En el caso de CM-PPQA se incluyen los checklists que se utilizarán en la Revisión. Para extraerlos de QC: Tools->; Excel Report Generator->; Checklist CM-PPQA->; Generate Excel Report->; (Ingresar como parámetro el nombre de la prueba que contiene el Checklist).</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.3.1 Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”

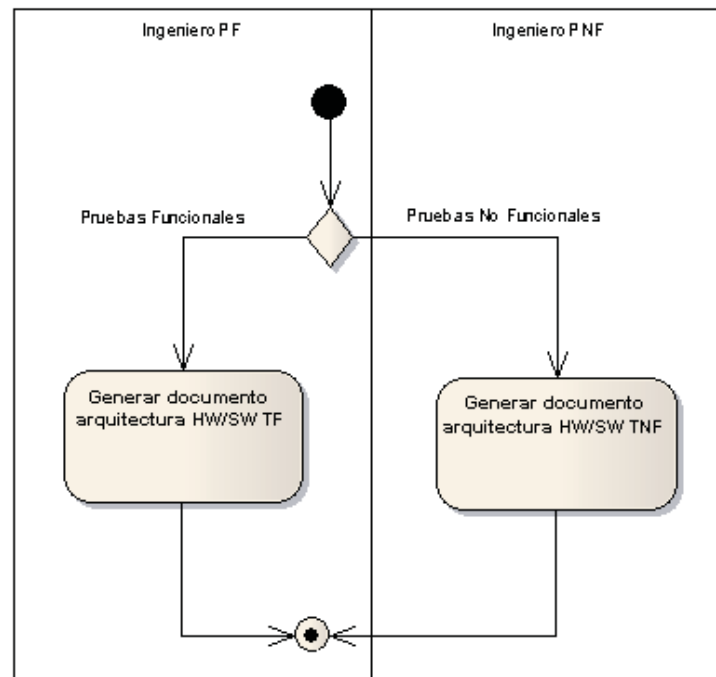


Figura 5.05: Proceso “Consolidar documento arquitectura HW/SW ambiente prueba”

5.1.3.2 Proceso “Estimar grado automatización y nivel diagnóstico”

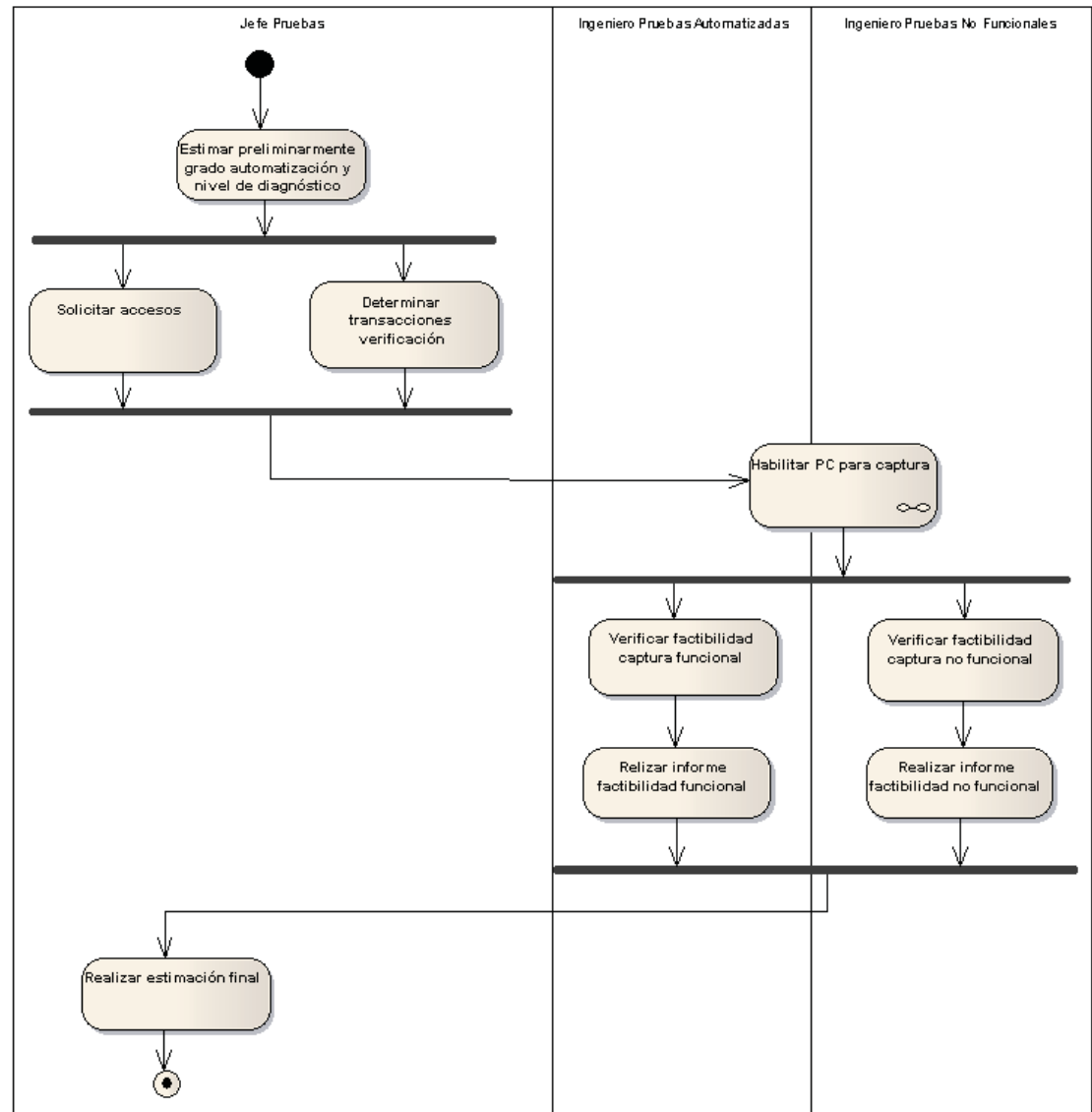


Figura 5.06: Proceso “Estimar grado automatización y nivel diagnóstico”

Proceso	Estimar preliminarmente grado automatización y nivel de diagnóstico
Descripción	<p>El análisis preliminar en esta tarea aborda el Grado de Automatización y el Nivel de Diagnóstico de la siguiente forma:</p> <p>Grado de Automatización</p> <p>En el caso que se disponga del aplicativo, se identifican todas aquellos módulos del aplicativo que no son automatizables con Mercury tanto desde el punto de vista funcional como no funcional, candidatos serían:</p> <ul style="list-style-type: none"> - General: (a) Uso de algoritmos de reconocimiento de imágenes, sonidos u otro distinto a eventos de teclado y mouse. (b) Dispositivos GUI no emulables en PC o notebooks. - Caso Funcional: (a) Reportes (b) Procesos Batch (c) Dispositivos GUI con mala emulación en PC. - Caso No Funcional: (a) Procesos Batch (b) Protocolos a nivel de GUI inexistentes en Mercury o no adquiridos. <p>Nivel de Diagnóstico</p> <p>Tanto en las Pruebas Funcionales como en las No Funcionales existe el mismo concepto, la diferencia es que en el último caso es siempre parte del servicio, en cambio en el caso funcional no.</p> <ul style="list-style-type: none"> - Caso Funcional: Si el servicio es Graybox, se deben verificar los resultados en la Base de Datos. Se debe evaluar si existen impedimentos para acceder la BD como Modelo no documentado. - Caso No Funcional: En este caso, si bien se espera un nivel Whitebox, no es siempre posible, debido a que pueda que no exista el SW para monitorear cada uno de los servidores aplicativos o en el caso que sea Mercury Diagnostics, no se esté dispuesto a hacer la inversión. <p>Si no se disponen del aplicativo se deberá hacer supuestos sobre las transacciones para las cuales no es tan claro si pueden ser automatizadas o no.</p>

Proceso	Solicitar accesos
Descripción	<p>Se debe solicitar:</p> <ul style="list-style-type: none"> - Acceso al aplicativo - Conectividad a Servidor QC en la Organización para QTP en el caso de pruebas funcionales.

Proceso	Determinar transacciones verificación
Descripción	<p>En base a los antecedentes que se disponga, se debe definir que transacciones serán las que se usarán para validar si es posible automatizar ya sea funcional o no funcionalmente. Se debe tener en cuenta al menos:</p> <p>Generales</p> <ul style="list-style-type: none"> - Debería haber al menos 1 transacción por arquitectura de SW distinta a nivel de "Dispositivo GUI" <p>Pruebas No Funcionales</p> <ul style="list-style-type: none"> - Informarse si algún SW encripta o comprime a nivel de "Dispositivo GUI" y si fuera así, solicitar los algoritmos al desarrollador (siempre y cuando sean indispensables para variabilizar los scripts VUGEN posteriormente). - Determinar si la aplicación es mono o multiusuario por dispositivo GUI.

Proceso	Habilitar PC para captura
Descripción	<p>Se contemplan 2 situaciones:</p> <ul style="list-style-type: none"> - Aplicativo se encuentra disponible a través de la red de la organización, en ese caso se utilizarán los computadores destinados para ese fin en dicha organización. - En el caso que no esté disponible en la red organizacional, se usará una máquina virtual que tenga QTP y LR listos para ser instalados.

Proceso	Verificar factibilidad captura funcional
Descripción	<p>La verificación consiste en que para cada transacción de verificación se ejecuta:</p> <ul style="list-style-type: none"> - Identificar los Add-ins que podrían necesitar en base a la documentación o referencias de la GUI. - Generar un script para cada una de las transacciones - Reejecutar el script. - Comprobar que se puedan incorporar puntos de verificación. - Comprobar que se puede variabilizar la entrada. <p>Una vez realizadas las tareas anteriores complete la planilla de Factibilidad QTP.</p>

Proceso	Verificar factibilidad captura No Funcional
Descripción	<p>La verificación consiste en que para cada transacción de verificación se ejecuta:</p> <ul style="list-style-type: none">- Generar un script que implique login, ejecutar transacción y logout.- Detectar correlación y resolverla en el caso que la hubiere.- Determinar si los mensajes son parametrizables.- Reejecutar el script.- Si es posible realizar una prueba con más de un usuario a la vez sobre el mismo "Dispositivo GUI". <p>Nota: Para resolución de problemas ver tab Files.</p>

Proceso	Realizar estimación final
Descripción	<p>Tanto para el caso funcional como no funcional, se consolida la estimación inicial con los resultados obtenidos en la verificación con el aplicativo.</p> <p>El resultado de esta estimación puede producir RFC de un impacto no despreciable en el presupuesto.</p>

5.1.3.2.1 Proceso “Habilitar PC para captura”

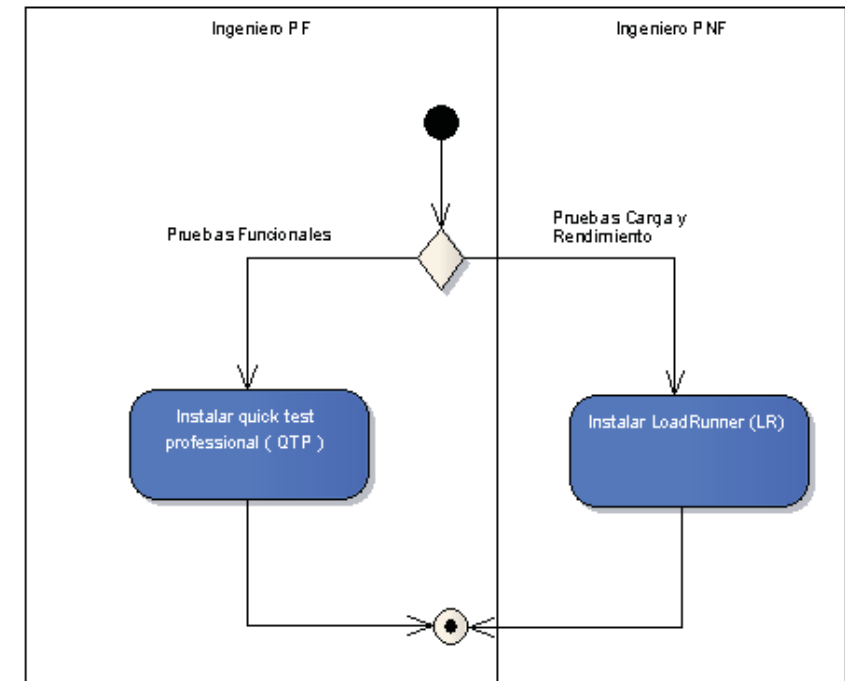


Figura 5.07: Proceso “Habilitar PC para captura”

5.1.3.3 Proceso “Definir requerimientos no funcionales”

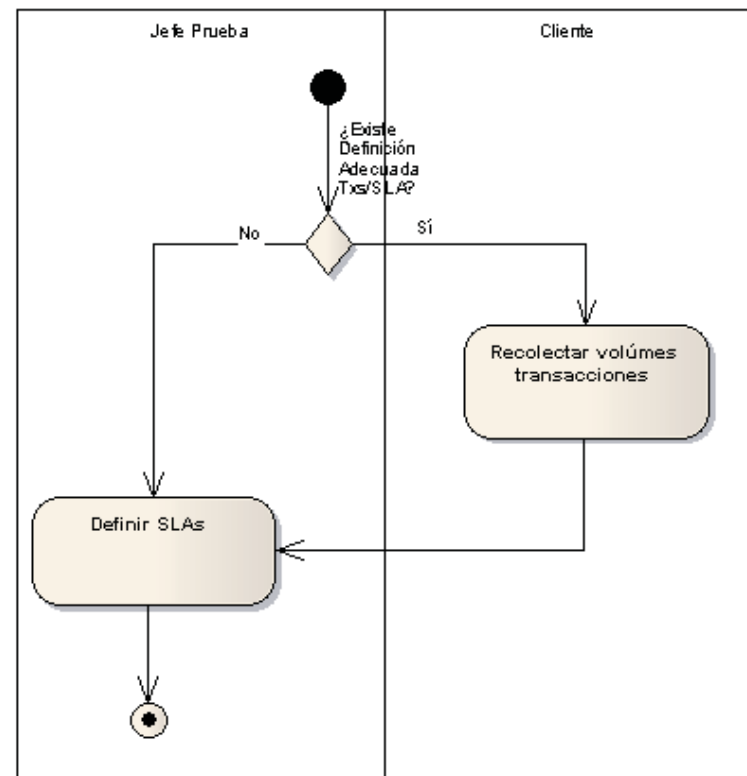


Figura 5.08: Proceso “Definir requerimientos no funcionales”

Proceso	Definir SLAs
Descripción	<ul style="list-style-type: none"> - Definir una carpeta con nombre "Pruebas de Carga y Rendimiento". - En el caso que exista información histórica o estimaciones, agregar attachment tipo URL apuntando a la ubicación de una planilla Excel "Volúmenes de Transacciones" en subversion. - Definir para cada transacción seleccionada un Requerimiento de Testing detallando su SLA, para el cual se debe especificar que acción del usuario gatilla la transacción que va a ser medida (por ejemplo click sobre el botón grabar).

Proceso	Recolectar volúmenes transacciones
Descripción	<p>Recolectar información histórica o estimar en un documento llamado "Volúmenes Transacciones", la siguiente información:</p> <ul style="list-style-type: none"> - Número de usuarios del sistema - % Crecimiento Anual Usuarios del Sistema - % Concurrencia de usuarios máxima - Una hoja de cálculo de transacciones cuyas columnas sean: Id Transacción, Nombre Transacción, Cantidad Transacciones Promedio x Tramo Horario i ($i=1, \dots, n$), % Crecimiento Anual. Ojala se pudieran descomponer la cantidad de transacciones por perfil - Una hoja de cálculo de transacciones cuyas columnas sean: Id Transacción, Nombre Transacción, Cantidad Transacciones Peak x Tramo Horario i ($i=1, \dots, n$), % Crecimiento Anual. Ojala se pudieran descomponer la cantidad de transacciones por perfil.

5.1.4 Proceso “Administrar proceso de pruebas” PARTE 2

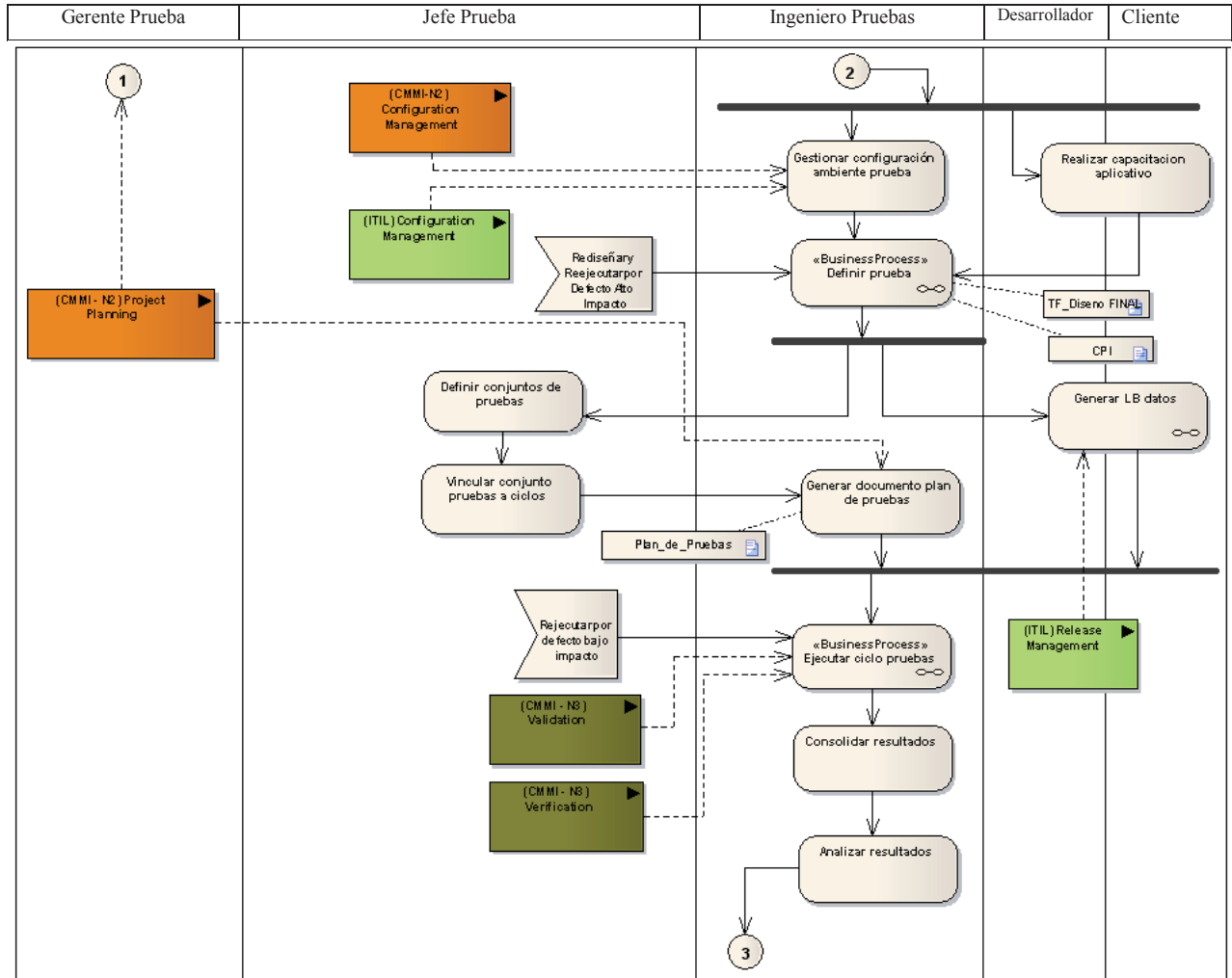


Figura 5.09: Proceso “Administrar proceso de pruebas” PARTE 2

Proceso	Gestionar configuración ambiente prueba
Descripción	<p>Con el objetivo de realizar las Pruebas, además de disponer del "Ambiente de Pruebas" (se debe definir de quien es la responsabilidad de entregar dicho ambiente de Pruebas), Calidad debe adaptar el ambiente para usar Mercury:</p> <ul style="list-style-type: none"> - Verificar que todas las componentes del Ambiente de pruebas que no son responsabilidad de Calidad, están disponibles. - Habilitar Laboratorio(s) Prueba(s): Disponer de "Dispositivos GUI", Instalar SW Calidad, Instalar SW Cliente. (Ver manual de instalación en Tab Files). - Generar los RFC para las conectividades de enlace con los "Servidores de Calidad" y "Dispositivo_GUI". - Generar los RFC para las configuraciones o instalaciones de "SW Calidad" en los "Servidores Aplicativo". - Enviar los RFCs a los responsables. - Verificar si los RFCs fueron correctamente ejecutados. - Verificar Normas de Seguridad impartidas por el Oficial de Seguridad.(Ver archivos adjuntos)

Proceso	Realizar capacitación aplicativo
Descripción	Realizar Capacitación del Aplicativo, idealmente se debiera hacer una Capacitación para Pruebas Funcionales y No Funcionales, pero podría haber casos donde tenga que ser realizada en sesiones separadas.

Proceso	Definir prueba (*)
Descripción	<p>La primera plantilla de apoyo que se utiliza en este proceso es "Definición CPI" ANEXO 1, en la cual se definen todas las combinatorias posibles en base a los criterios especificados en la planilla de apoyo.</p> <p>Una vez seleccionadas las Pruebas a diseñar, se debe preparar la planilla de apoyo "TF Diseño" ANEXO 1, en la cual se debe comenzar a identificar los ID de Prueba y relacionarlos a las Pruebas seleccionadas de la planilla anterior.</p> <p>Al mismo tiempo en este proceso se debe proceder con lo siguiente:</p> <p>Este proceso se basa en:</p> <ul style="list-style-type: none"> - Definición de Requerimientos de Pruebas: De los cuales se derivarán las Pruebas o "carpetas de Pruebas", las cuales serán refinadas con la documentación de Diseño que se tenga del aplicativo. QC tiene la opción de derivar automáticamente de los "Requerimientos de Pruebas" una jerarquía preliminar de Pruebas, la cual debe ser refinada. - La Capacitación Funcional y con otros entregables (sería ideal contar con las grabaciones de

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<p>ejecuciones sobre la GUI del aplicativo) se tiene la información necesaria para detallar los "Pasos" de cada Prueba y generar los "Scripts de Prueba" en el caso de automatización con "Load Runner".</p> <p>y contempla la realización de las siguientes tareas:</p> <ul style="list-style-type: none"> - Definir Estructura Directorios -Definir Pruebas Unitarias - Detallar Pruebas Unitarias - Definir Pruebas Integradas - Generar Scripts - Especificar Necesidad Datos - Detallar Pruebas Integradas <p>Es muy importante tener en mente que en cada ciclo puede ser necesario reejecutar algunas de las anteriores.</p>
--	--

Proceso	Definir conjuntos de pruebas
Descripción	<p>Se definen los conjuntos de prueba, ver ejemplos de estos conjuntos en Diagrama Conjunto_Prueba.</p> <p>En QC, Test Lab:</p> <ul style="list-style-type: none"> - Crear carpeta(s) que contienen "Conjuntos de Pruebas". - Crear cada "Conjunto de Pruebas" en la carpeta que corresponda, e ingresar las "Pruebas" que lo componen usando el tab "Execution Grid". El orden de ejecución se define en el tab "Execution Flow".

Proceso	Vincular Conjunto Pruebas a Ciclos
Descripción	En QC, Test Lab, las carpetas que contienen "Conjuntos de Pruebas" son linkeadas a ciclos. Para esto hacer click derecho sobre la carpeta y "Assign to Cycle".

Proceso	Generar documento plan de pruebas (*)
Descripción	<p>Con todos los procesos anteriores concluidos, y con todas las plantillas de apoyo concretadas, se debe dar pie a la generación del Plan de Pruebas, el cual contiene información de resumen de las decisiones y acuerdos tomados; es decir, es dejar reflejado el acuerdo para dar comienzo formula al proceso de diseño de las Pruebas.</p> <p>Plantilla de apoyo "Plan de Pruebas" ANEXO 1.</p>

Proceso	Generar LB datos
Descripción	Con el fin de realizar las Pruebas es necesario llevar al sistema a un determinado nivel de datos y generar los datos suficientes a utilizar en los escenarios de Prueba, esto implica :

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	<ul style="list-style-type: none"> - Generar BDs, las cuales deben ser parte de la LB, la cual debería ser administrada por CM (ver qué pasa en los clientes externos). - Generar Datos en la cantidad necesaria para realizar las Pruebas
--	--

Proceso	Consolidar resultados
Descripción	<p>Esta tarea consiste en determinar si los defectos reportados durante las Pruebas son reducibles a un grupo menor o son preexistentes.</p> <p>En el caso que un grupo de defectos pueda ser reemplazado por un solo defecto:</p> <ul style="list-style-type: none"> - Elegir un defecto del grupo como representativo. - Linkear el resto de los defectos al seleccionado en el paso 1. - Dejar los defectos linkeados en estado "Consolidated".

Proceso	Analizar resultados
Descripción	<p>Esta tarea consiste en determinar si los defectos reportados durante las Pruebas son tales.</p> <ul style="list-style-type: none"> - En el caso que el defecto sea tal, actualizar su estado a "Open". - En el caso que el defecto no aplique, se puede borrar o dejar su estado en "Closed", esto último sólo si es de algún aporte mantener el registro.

5.1.4.1 Proceso “Definir prueba”

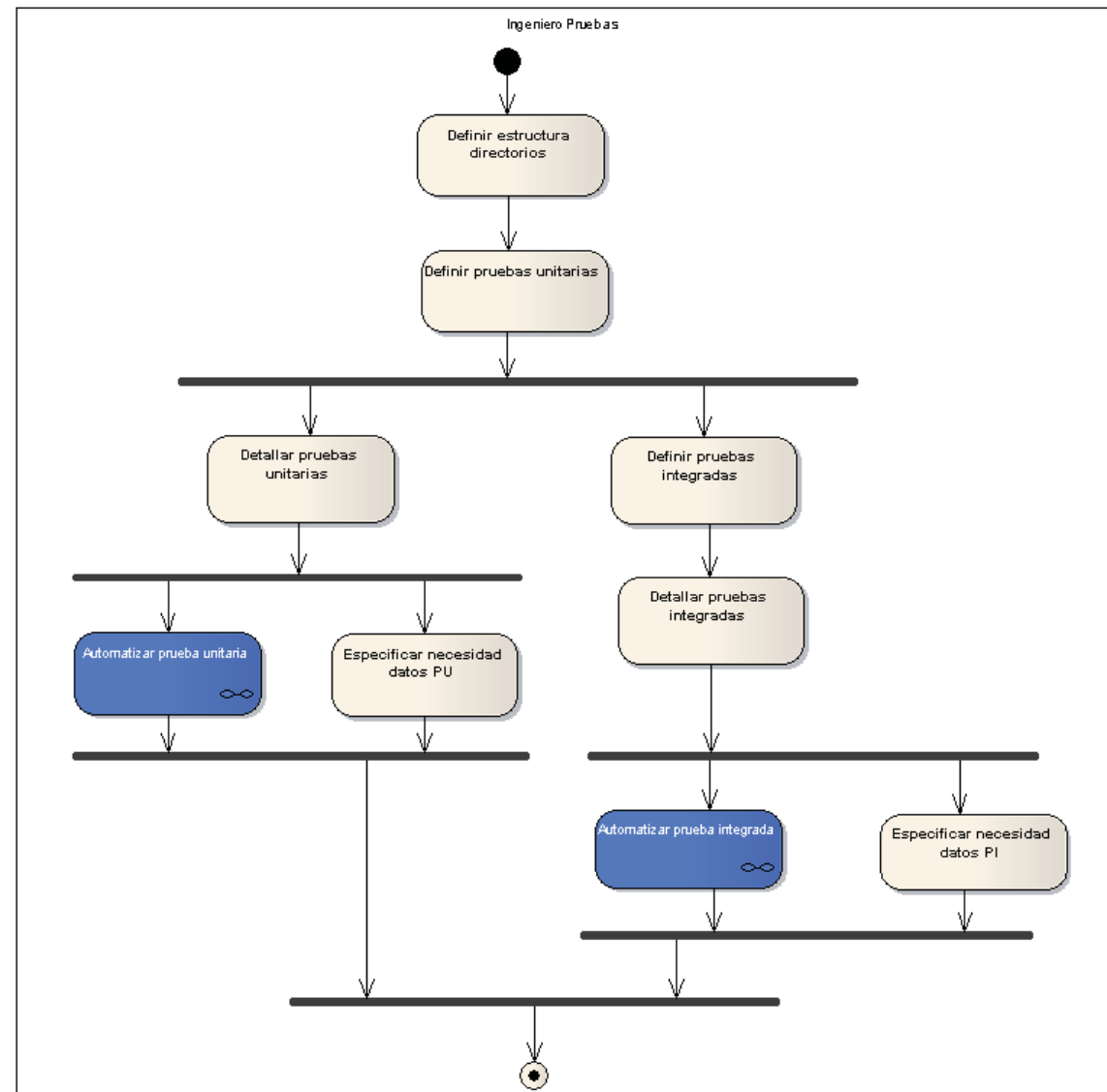


Figura 5.10: Proceso “Definir prueba”

Proceso	Definir estructura directorios
Descripción	<p>Crear los directorios que se necesitan para almacenar los scripts de pruebas, repositorios de objetos y bibliotecas de funciones tanto de QTP como LR. Se recomienda crear en Subversion:</p> <ul style="list-style-type: none"> - Crear directorio de Scripts, se sugiere que el nombre de cada script se un prefijo de a lo más 3 caracteres alfabéticos concatenados con el ID QC de la Pruebas en formato '99999'. - Crear directorio de Repositorios de Objetos (Sólo para uso de QTP, es decir Pruebas Funcionales) - Crear directorio de Bibliotecas de Funciones (Sólo para uso de QTP, es decir Pruebas Funcionales) - Crear directorio de Escenarios de Recuperación <p>Los scripts deben ser actualizados en QC cuando se quiera ejecutarlos y en otro momento no corresponderán necesariamente a la LB.</p> <p>NOTA GENERAL: EL ORDEN EN ESTE DIAGRAMA DE PROCESOS NO IMPLICA QUE UNA TAREA NO SE PUEDA VOLVER A REALIZAR EN UN MOMENTO POSTERIOR CON FINES DE REFINAR UNA DEFINICION YA EXISTENTE, TENIENDO EN MENTE NO OLVIDAR GESTIONAR LA LB, SI CORRESPONDE.</p>

Proceso	Definir pruebas unitarias
Descripción	<p>Una Prueba unitaria en el caso de Pruebas funcionales corresponderá a una unidad funcional indivisible y posiblemente derivable de un requerimiento de Pruebas (si éstos han sido detallados adecuadamente).</p> <p>A este nivel basta con definir la Prueba Unitaria, pero no indicar sus pasos.</p>

Proceso	Detallar pruebas unitarias
Descripción	<p>Esta tarea siempre se realiza para las Pruebas manuales y se evalúa su realización para el caso automático dependiendo del tiempo y del servicio otorgado. Por ejemplo, cuando no se realiza la ejecución por parte de la Organización, si se deberían documentar.</p> <p>Se ingresa la Prueba a QC en Test Plan ->; Design Steps, de acuerdo a la especificación entregada en el Diagrama "Diseño Prueba" para la "Prueba_Unitaria", "Paso_Documentado", "Escenario_Prueba_Unitaria" y "Parametro_Prueba_Unitaria" y "Regla".</p>

Proceso	Especificar necesidad datos PU
Descripción	<p>- Confeccionar documento que indique cual es la necesidad de datos en función de los parámetros que se identifiquen.</p> <p>- Especificar un atributo que permita particionar los escenarios entre los que consideran datos existentes en la Base de Datos y datos generados.</p> <p>- Generar y adjuntar a la Prueba en QC la Planilla Referencial de Datos siguiendo el formato de nombre D999999_01.xls donde los primeros 6 dígitos corresponden al ID generado en QC para la Prueba (Test Plan->; Seleccionar Test->; Tab Attachments). La planilla referencial Excel tiene una hoja con nombre "Global", cuya primera fila contiene los nombres de los parámetros identificados en el punto 1 y tiene tantas filas como escenarios se vayan a probar en la Prueba. Como regla general en el caso manual, debería haber una línea por escenario para hacer más simple la descripción de los pasos y el trabajo del Tester. En el caso automático puede haber más de un escenario manejado por el script.</p> <p>Observación: La planilla referencial Excel es usado como base para las planillas que contendrán los datos reales que se usarán para ejecutar las Pruebas, estas últimas planillas deben ser adjuntadas a los "Test Instance".</p>

Proceso	Definir pruebas integradas
Descripción	<p>Combina 2 o más pruebas unitarias dentro de una sola prueba</p> <p>Esta tarea siempre se realiza para las pruebas manuales y se evalúa su realización para el caso automático dependiendo del tiempo y del servicio otorgado. Por ejemplo, cuando no se realiza la ejecución por parte de organización, si se deberían documentar. (EN EVALUACIÓN DURANTE EL PILOTO)</p>

Proceso	Detallar pruebas integradas
Descripción	Se ingresa la Prueba a QC en Test Plan ->; Design Steps, de acuerdo a la especificación entregada en el Diagrama "Diseño Prueba" para la "Prueba_Integrada".

Proceso	Especificar necesidad datos PI
Descripción	<p>- Se debe especificar un atributo que permita particionar los escenarios entre los que consideran datos existentes en la Base de Datos y datos generados.</p> <p>- Generar y adjuntar a la prueba en QC la Planilla Referencial de Datos siguiendo el formato de nombre D999999_01.xls donde los primeros 6 dígitos corresponden al ID generado en QC para la prueba (Test Plan->; Seleccionar Test->; Tab Attachments). La planilla referencial Excel tiene una hoja de nombre Global para la Prueba unitaria guía y tantas hojas como Pruebas unitarias dependan de la guía en una relación uno es a muchos. Como ejemplo considere el caso de un Maestro-Detalle, el maestro es la Prueba unitaria guía y el detalle es la Prueba unitaria asociada, esta última tendrá una hoja propia.</p> <p>Observación: La planilla referencial Excel es usada como base para las planillas que contendrán los datos reales que se usarán para ejecutar las Pruebas, estas últimas planillas deben ser adjuntadas a los "Test Instance".</p>

5.1.4.2 Proceso “Automatizar prueba unitaria”

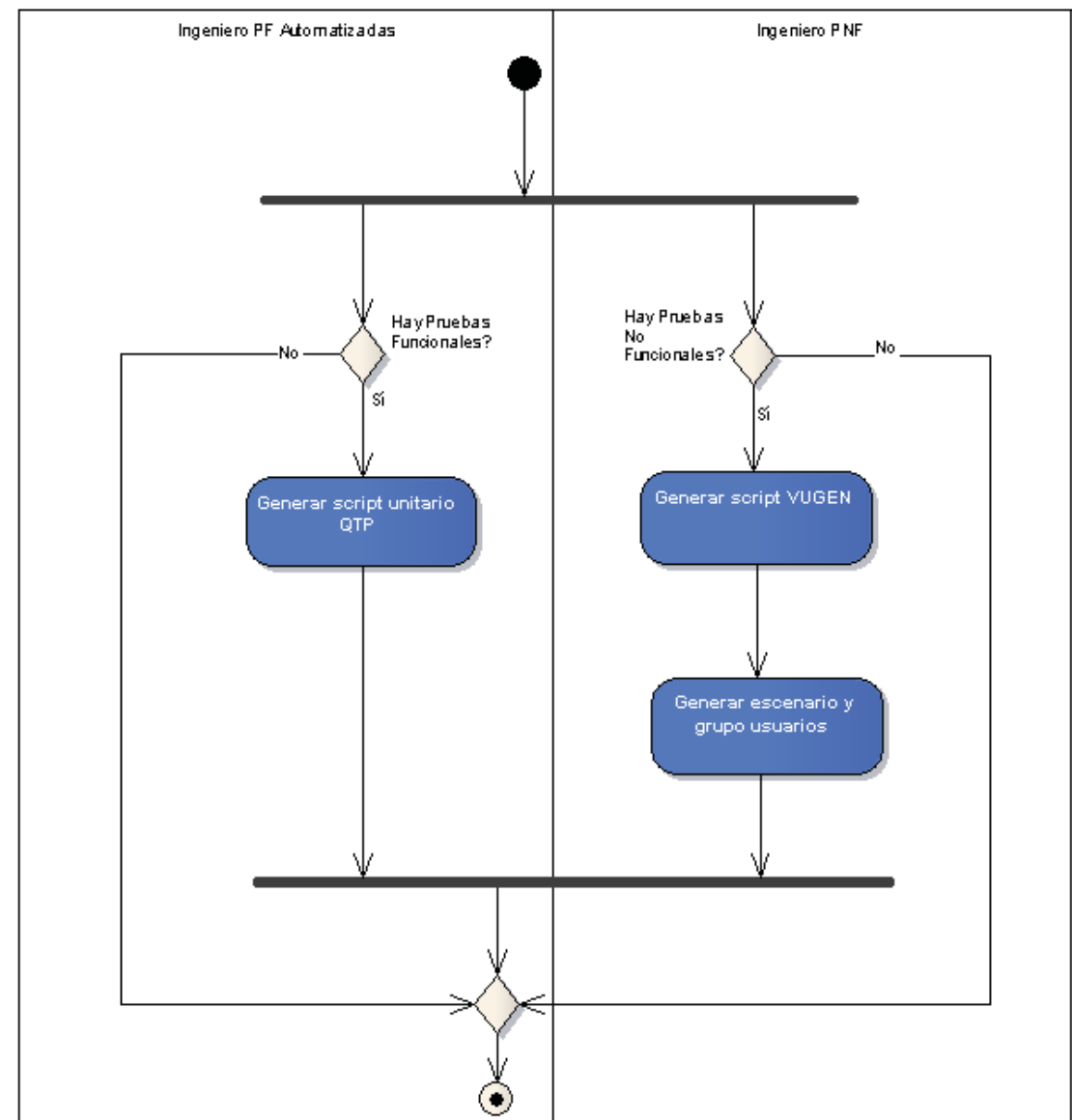


Figura 5.11: Proceso “Automatizar prueba unitaria”

Proceso	Generar script unitario QTP
Descripción	<ul style="list-style-type: none"> - Se graba un "Acción QTP" dentro del script, siguiendo los pasos (recordar que en QTP las validaciones de cualquier tipo son pasos) y parametrización detallada para la prueba en QC. Debido a que puede haber varios flujos de pasos (al menos el caso positivo y el negativo) probablemente se deba intervenir el script en su vista "Keyword-View" o "Script-View". - La "Acción QTP" debe informar siempre a través de Report.ReportEvent si la prueba fue exitosa o falló. - Al final el script se almacena en el directorio que corresponde para que sea versionado. - Se recomienda medir el tiempo de respuesta, pero sólo con el fin de tener una referencia, pues debido a que ni el Ambiente de Pruebas ni las Bases de Datos en él son representativos del Ambiente Productivo en régimen permanente.

Proceso	Generar script VUGEN
Descripción	<ul style="list-style-type: none"> - Se graba el login y el logout dentro de dos acciones. - Para cada transacción acordada con el Cliente como parte de la prueba se graba un script con una acción y dentro de ella se marca el inicio y el término de la "Transaction" la cual mide el tiempo sujeto a SLA. - A continuación se debe depurar el script revisando: Pacing, Log, Think Time, Network - Speed Simulation (útil cuando el ancho de banda disponible es mayor al real) y detección de Correlación y cerciorándose que el script se puede re-ejecutar exitosamente. - Dependiendo del tipo de aplicativo, agregar puntos de Verificación para determinar si la transacción está retornando adecuadamente, pues hay veces que la no disponibilidad no se manifiesta en la caída de la aplicación sino en comportamiento anormal.

Proceso	Generar escenario y grupo usuarios
Descripción	En el caso de Pruebas de Carga y Rendimiento Unitarias, usando LR Controller, se declara un escenario por cada script usando un solo grupo programando tramos de 1, 10, 20, 30 y 50 usuarios respectivamente. La duración de cada tramo debería ser al menos 5 minutos con el fin de ver la estabilidad de la respuesta del sistema, si bien tiempos mayores son recomendables, su uso dependerá del tiempo total asignado a esta prueba.

5.1.4.3 Proceso “Automatizar prueba integrada”

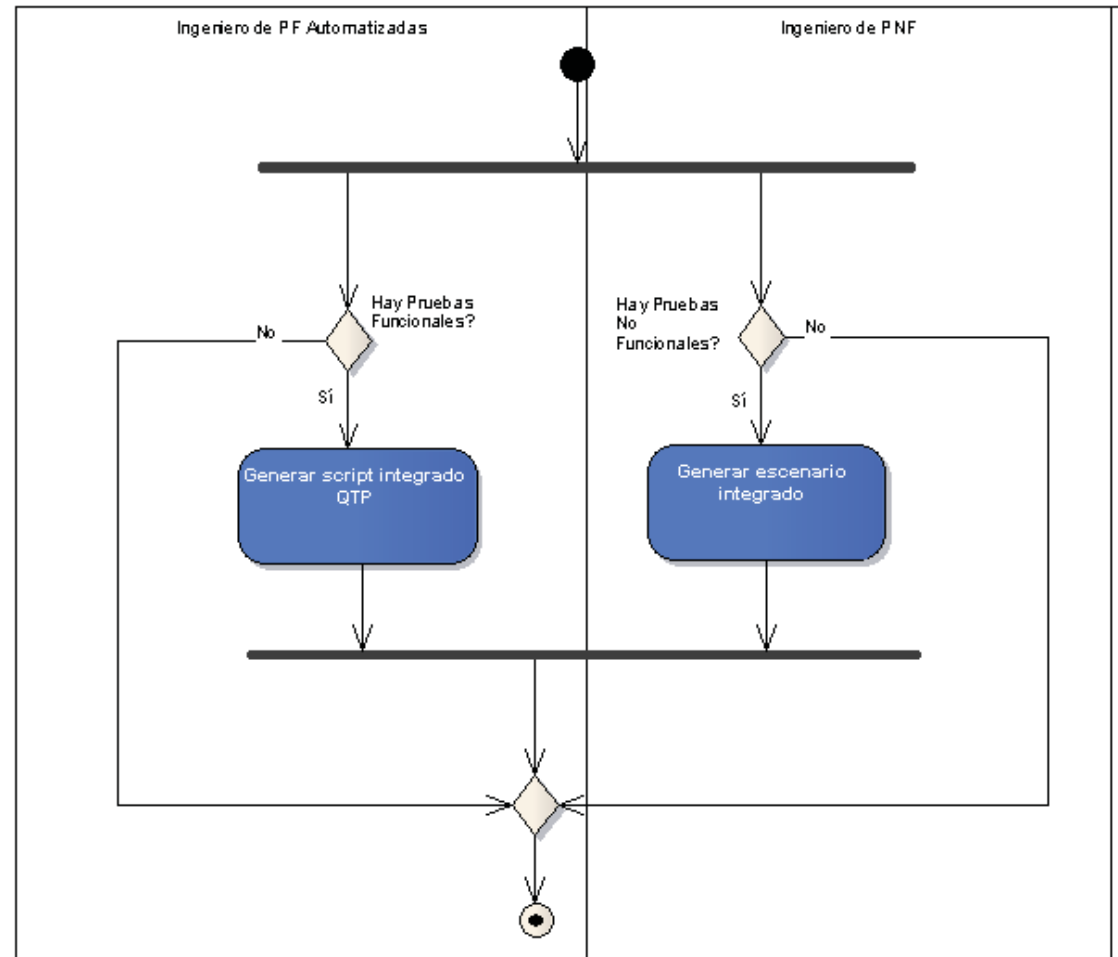


Figura 5.12: Proceso “Automatizar prueba integrada”

Proceso	Generar script integrado QTP
Descripción	En este punto ya se han generado las "Acciones QTP" para las pruebas unitarias, entonces: <ul style="list-style-type: none"> - Se crea un script en el cual se referencia todas aquellas acciones que componen la "Prueba Integrada" siguiendo el flujo definido. - Se declara una planilla de datos por acción, la cual deberá ser llenada.

Proceso	Generar escenario integrado
Descripción	Se genera en Load Runner Controller, el escenario integrado definiendo todos los Grupos de Usuarios que se haya contemplado en la prueba y usando los Script ya generados en VUgen para las Pruebas Unitarias.

5.1.4.4 Proceso "Generar LB datos"

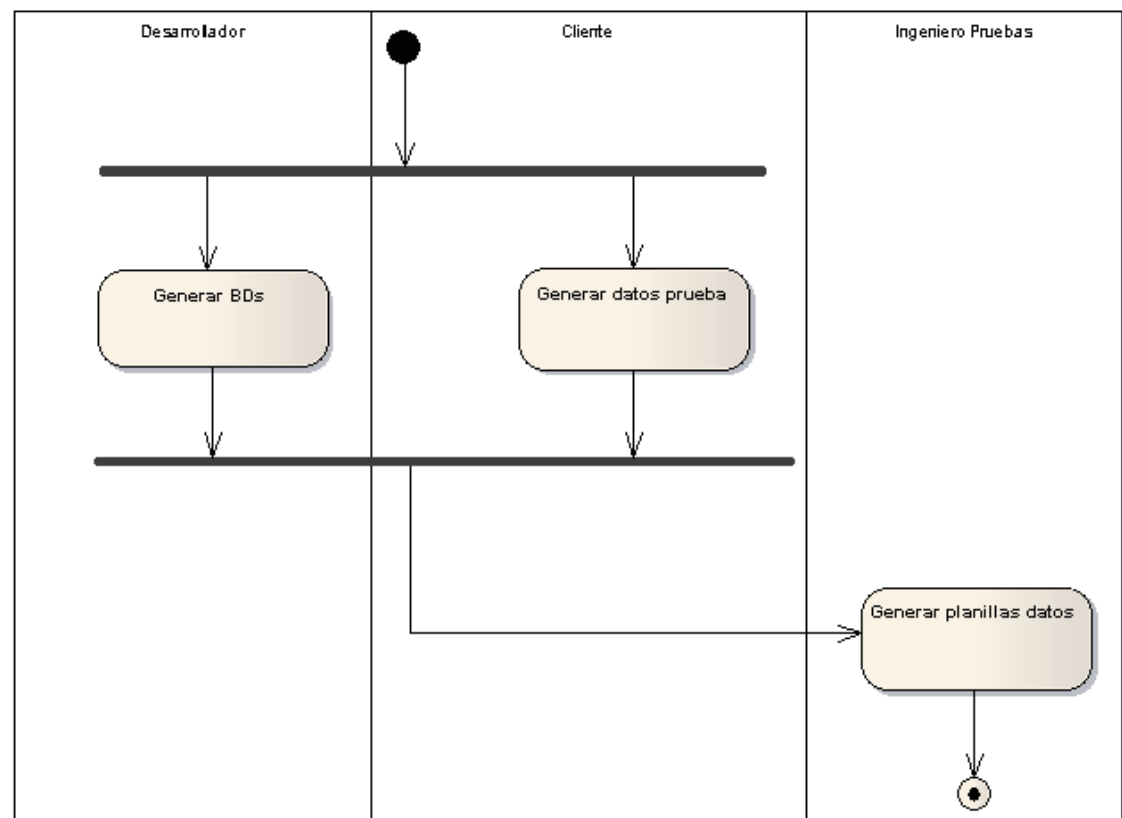


Figura 5.13: Proceso "Generar LB datos"

Proceso	Generar BDs
Descripción	Esta generación depende del tipo de pruebas y se deben considerar los siguientes aspectos: <ul style="list-style-type: none">- Las pruebas no funcionales requerirán un tamaño representativo del aplicativo en régimen permanente.- Se debe velar por la confidencialidad de los datos.- Los procedimientos de Respaldo y Restauración deben existir y estar probados antes del inicio de las pruebas, esto es necesario debido a que muchas veces se necesitan realizar las pruebas más de una vez usando la misma Release del SW. Ejemplo: Se hicieron pruebas no funcionales en las cuales se detectaron problemas en el HW, los cuales fueron resueltos y se requiere probar de nuevo.

Proceso	Generar datos prueba
Descripción	Generación a través de consultas SQL o Recolección de Datos en la cantidad necesaria para realizar las pruebas.

Proceso	Generar planillas datos
Descripción	Usando las Planillas Excel Referenciales adjuntadas a cada prueba durante las actividades "Especificar Necesidad Datos PU" y "Especificar Necesidad Datos PI", generar las Planilla Excel para las pruebas que serán realizadas durante el ciclo en curso.

5.1.4.5 Proceso “Ejecutar ciclo pruebas”

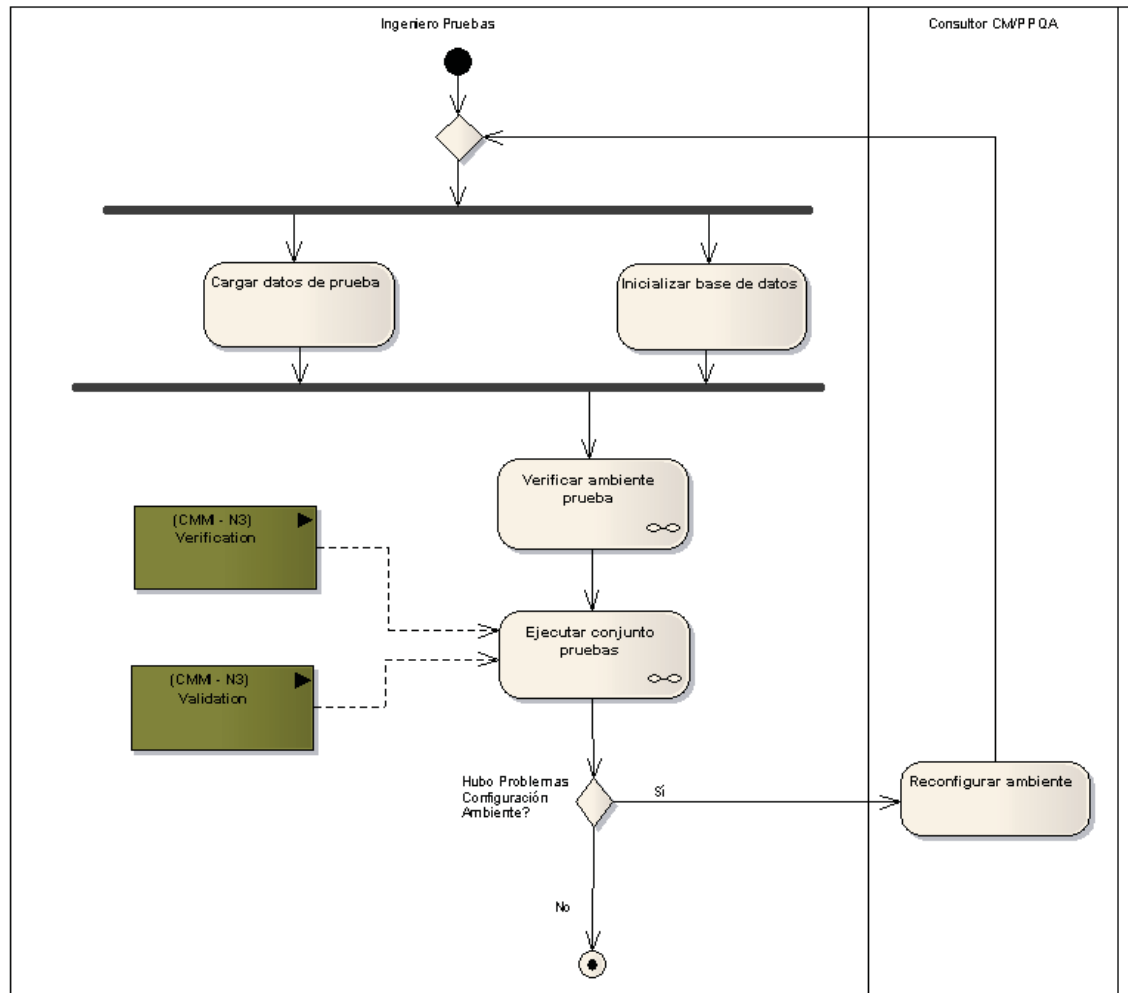


Figura 5.14: Proceso “Ejecutar ciclo pruebas”

Proceso	Cargar datos de prueba
Descripción	Cargar las Planillas de Datos de Prueba generadas en "Generar Planillas Datos" en las "Test Instance" del ciclo.

Proceso	Inicializar base de datos
Descripción	Inicializar la(s) base(s) de datos es setear esta(s) misma(s) en la condición inicial para que comience el Ciclo correspondiente, lo que implica: <ul style="list-style-type: none">- Datos Necesarios y consistentes con los datos que se usarán en las pruebas.- Para el caso no funcional, además se requiere que la(s) Base(s) de Datos tenga(n) un tamaño representativo al que tendría en el estado permanente en producción.

Proceso	Ejecutar conjunto pruebas
Descripción	Se ejecuta cada prueba dentro del Conjunto de Pruebas ya sea manual o automáticamente. El primer Conjunto de Pruebas a ejecutar debería ser algún tipo de "Pruebas Trazadoras". En el caso de Pruebas No Funcionales, es altamente recomendable que las Pruebas de Carga y Rendimiento Unitarias se ejecuten al menos 3 veces, para tener una confiabilidad mínima en la medición de tiempos y comportamiento del sistema.

5.1.4.6 Proceso “Verificar ambiente prueba”

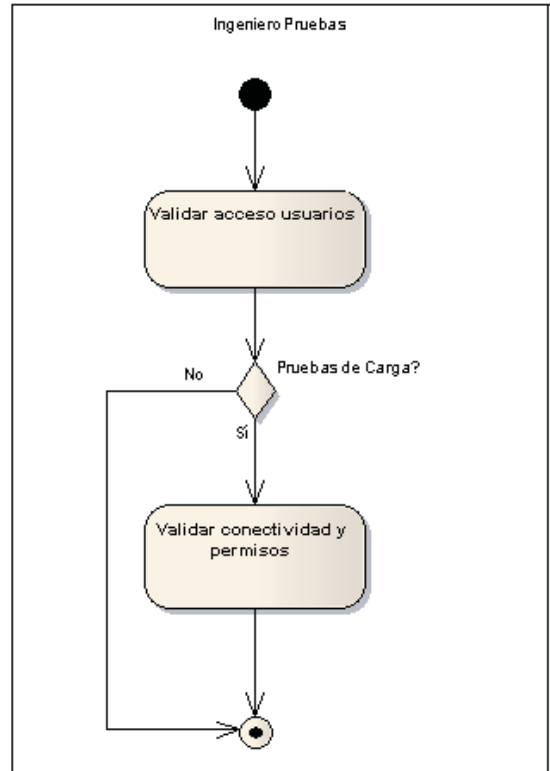


Figura 5.15: Proceso “Verificar ambiente prueba”

Proceso	Validar acceso usuarios
Descripción	Validar que sólo las personas responsables de las pruebas tengan acceso al sistema.

Proceso	Validar conectividad y permisos
Descripción	<p>En el tab Run del LR Controller validar:</p> <ul style="list-style-type: none"> - Conectividad del Controller a Generadores de Carga (Icono Load Generators). - Conectividad y permisos de monitoreo desde el Controller a los servidores del aplicativo y de BDs. (Al agregar las métricas de Windows Resources).

5.1.4.7 Proceso “Ejecutar conjunto de pruebas”

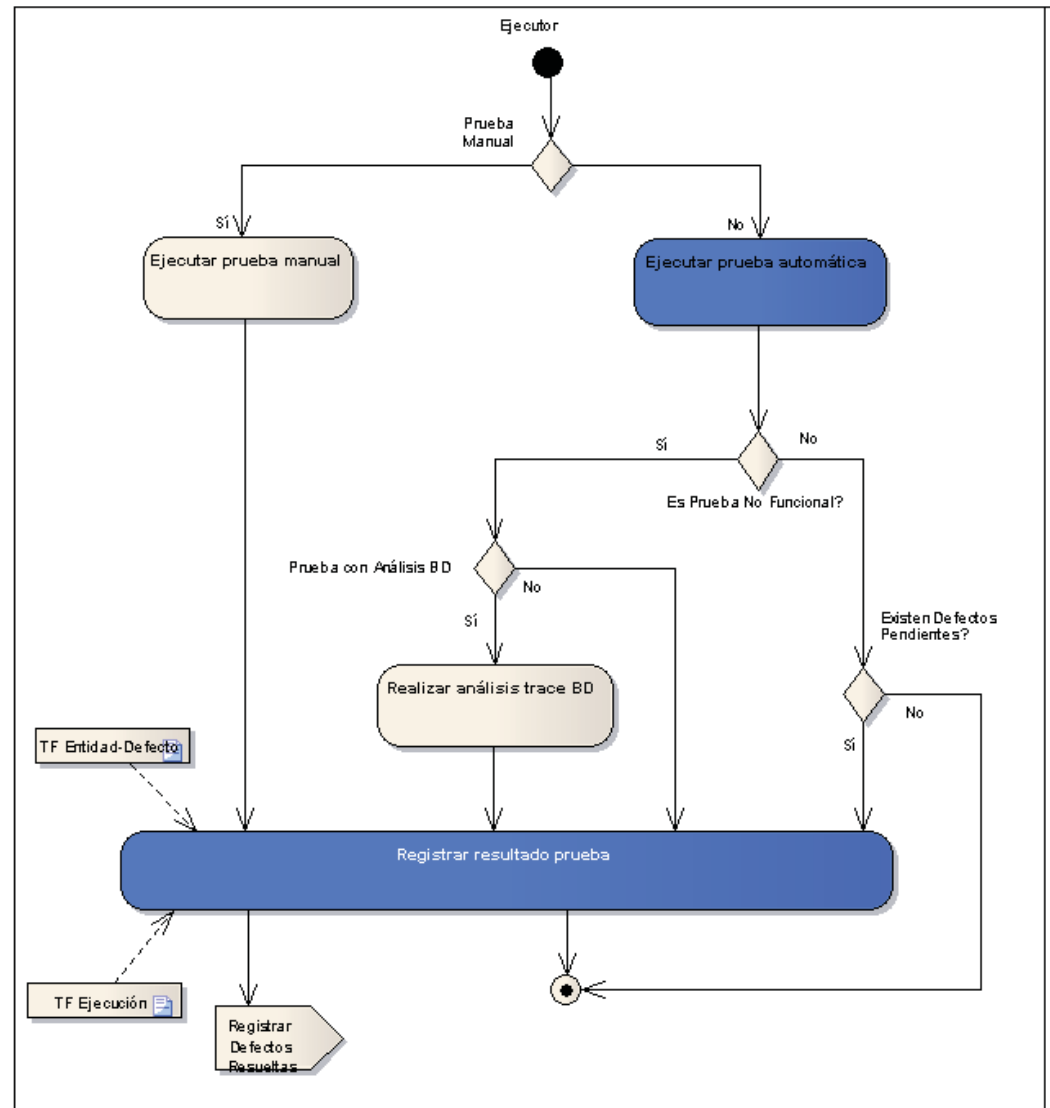


Figura 5.16: Proceso “Ejecutar conjunto de pruebas”

Proceso	Ejecutar prueba automática
Descripción	Para el Caso de Pruebas Funcionales se debe setear en QTP que registre automáticamente cualquier error como defecto.

Proceso	Registrar resultado prueba (*)
Descripción	<p>Una vez que el Analista/Ejecutor ejecuta las pruebas diseñadas, debe ir dejando registro de dichas pruebas en la planilla de “TF Ejecución” (Ver ANEXO1), la cual irá dejando la evidencia de la ejecución.</p> <p>En el caso que la prueba obtenga estado NOK, se debe registrar en la planilla "TF Entidad-Defecto" (Ver ANEXO 1), todos los defectos encontrados en la prueba.</p> <p>Al momento de ejecutar, basta con que un paso no sea el esperado para que el caso de prueba total obtenga el estado NOK, sin embargo, que un paso sea NOK no implica necesariamente que no se puedan seguir testeando los demás pasos.</p>

5.1.5 Proceso “Administrar proceso de pruebas” PARTE 3

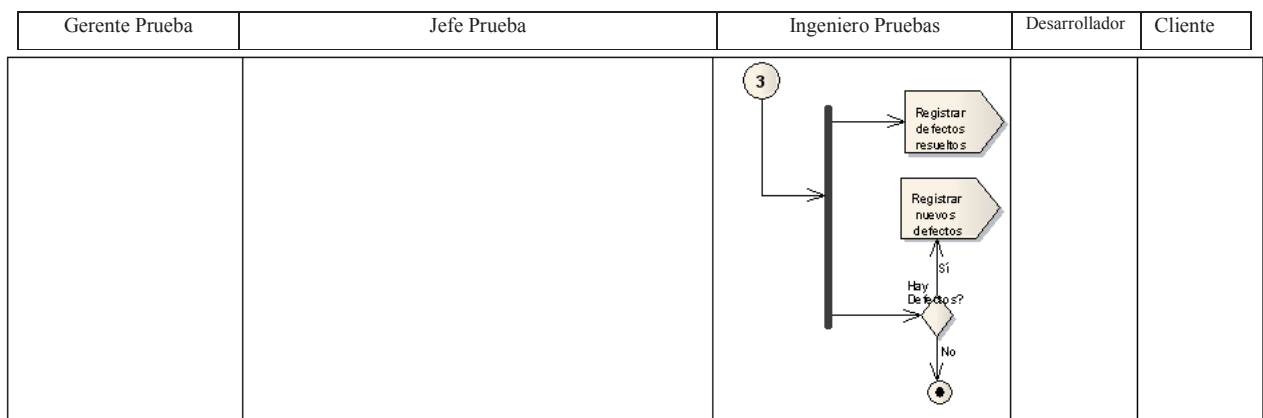


Figura 5.17: Proceso “Administrar proceso de pruebas” PARTE 3

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.1.6 Proceso: “Gestionar defectos”

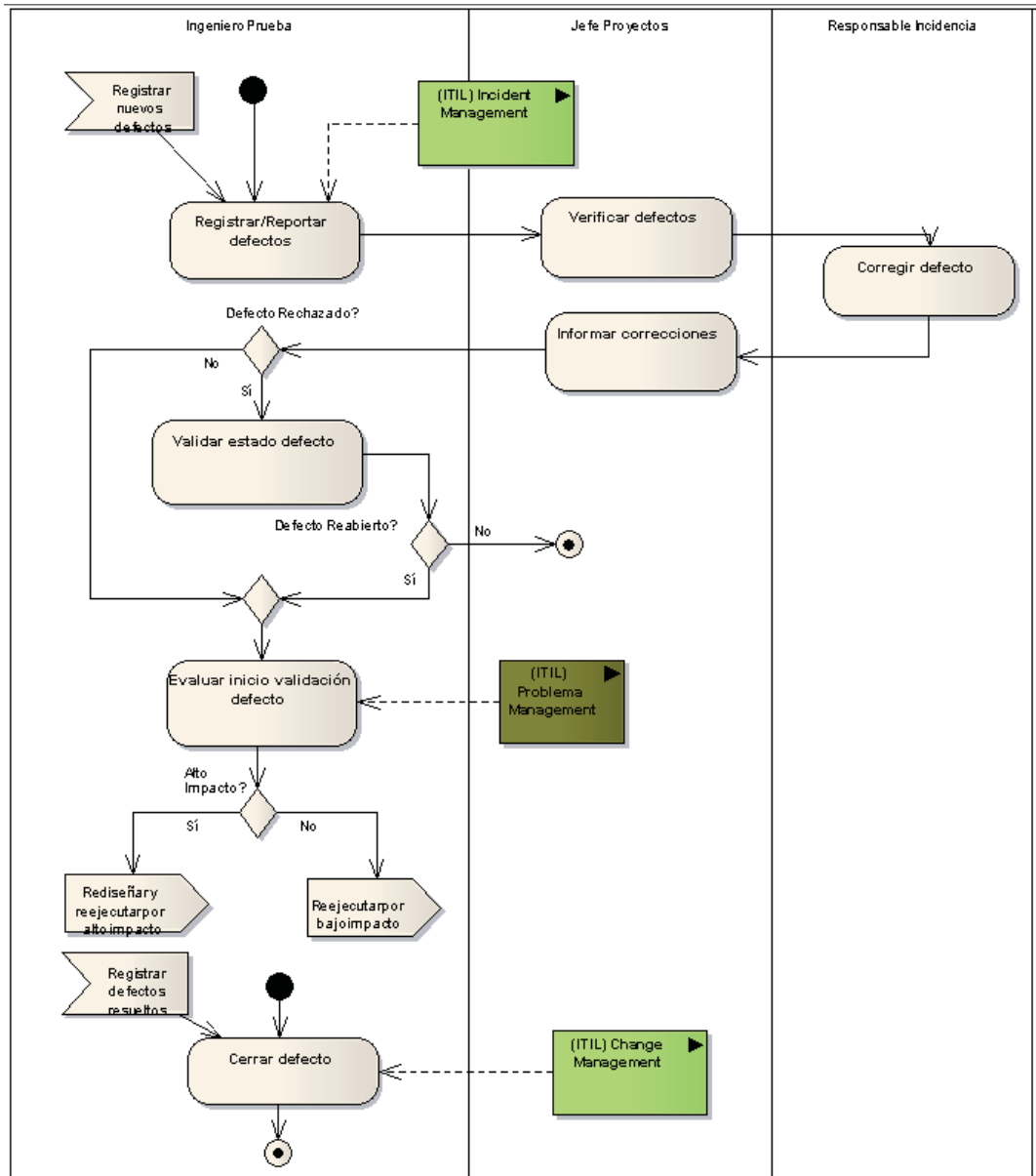


Figura 5.18: Proceso “Gestionar defectos”

Proceso	Registrar/reportar defectos
Descripción	<p>- Ingresar los atributos del defecto, es importante el ingreso del campo Área Calidad (CM, PPQA, TF, TNF) con el fin de cada área pueda posteriormente identificar que defectos debe gestionar. También no se debe identificar al "Responsable de Solución" que puede ser el Desarrollador, Responsable HW, Responsable SW, Responsable Redes.</p> <p>- Vincular las entidades que gatillan el defecto, recuerde que pueden ser las siguientes:</p> <ul style="list-style-type: none"> • Requerimiento • Prueba • Conjunto de Pruebas • Instancia de Pruebas • Ejecución de Pruebas • Paso de Pruebas (En el caso que se esté ejecutando una prueba, lo más recomendable vincular al paso el defecto y adjuntar la evidencia a este último) • A otro defecto (Por deducción) <p>Para que el defecto sea procesado como tal, se lo debe dejar en estado "Open".</p> <p>Reportes de Defectos</p> <p>- CM y PPQA, en Customize->; Excel Report Generator->; CM-PPQA Matriz No Conformidades.</p>

Proceso	Corregir defecto
Descripción	<p>El Jefe de Proyecto o el equipo de desarrollo evalúan si corresponde a un defecto, y de serlo inicia su corrección.</p> <p>Si el defecto no se acepta, se debe setear en estado "Rejected".</p> <p>Si el defecto es aceptado, en el momento de resolverlo se debe setear el estado "Fixed".</p>

Proceso	Validar estado defecto
Descripción	<p>Validar si el estado del defecto rechazado aplica o no. Si aplica, poner el defecto en estado "Closed", en caso contrario en "Reopen" y enviárselo al Desarrollador y/o Jefe de Proyecto.</p>

Proceso	Evaluar inicio validación defecto
Descripción	En función del impacto se decide si modificar el diseño o simplemente reagendar la prueba, el cual es realizado una vez que el Ciclo actual se encuentra terminado: <ul style="list-style-type: none">- Alto impacto =>; Revisar Diseño y reejecutar algunos "Conjuntos de Pruebas"- Bajo Impacto =>; Reejecutar algunos "Conjuntos de Pruebas"

Proceso	Cerrar defecto
Descripción	En el caso que un defecto haya sido resuelto, setear el estado "Closed", esto debería ser hecho en el ciclo que ocurre.

5.1.7 Proceso: “Realizar auditoría CM/PPQA”

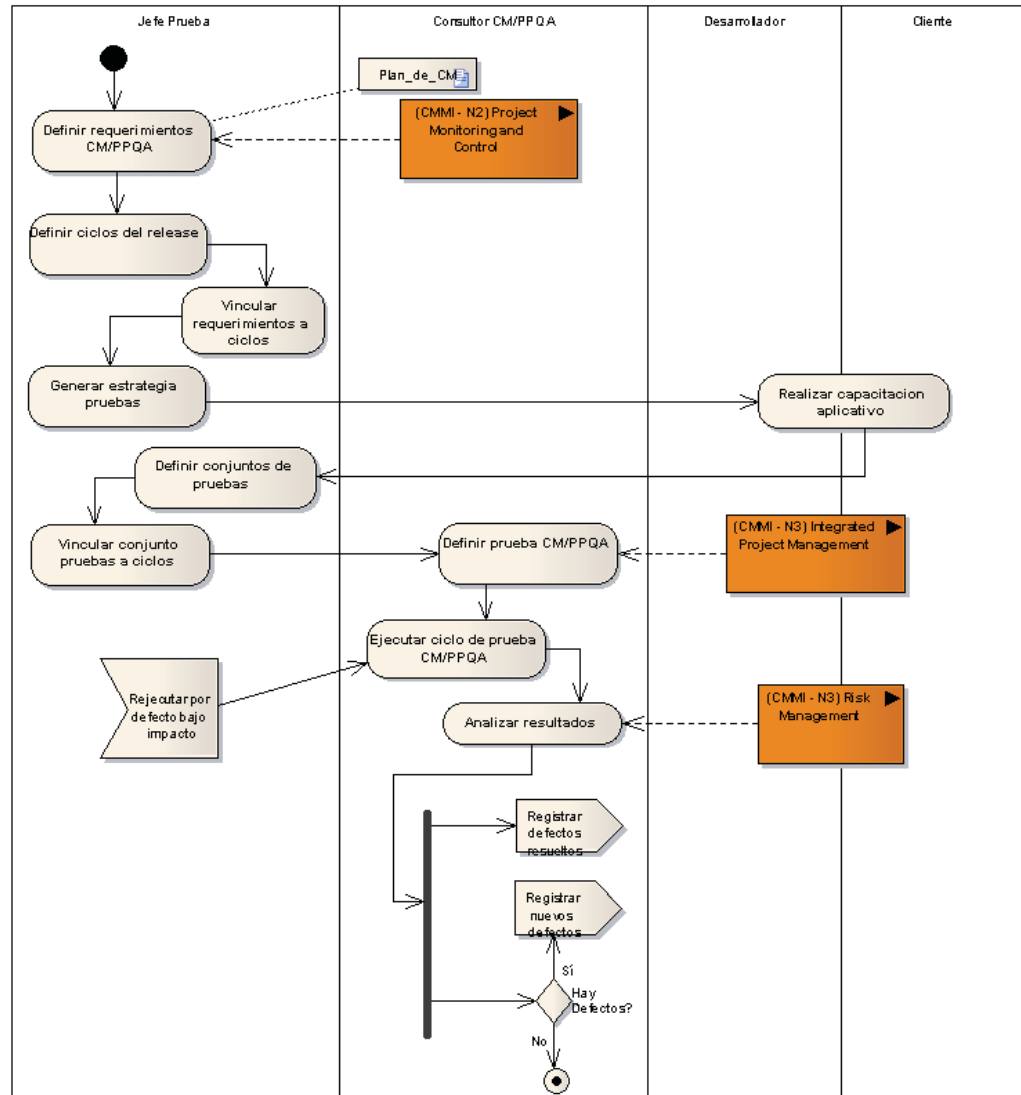


Figura 5.19: Proceso “Realizar auditoría CM/PPQA”

Proceso	Definir requerimientos CM/PPQA (*)
Descripción	<p>La planilla de apoyo "Plan de CM" se encuentra en el ANEXO 1. Dicha planilla apoya a la planificación de perfilamientos y definición de la biblioteca de configuración de los ítems.</p> <p>Propone una estructura base, especificando que debiese tener cada una de ellas.</p> <p>Se definen los requerimientos de CM/PPQA detallando los atributos mencionados a continuación, si es necesario se crean carpetas que agrupen los requerimientos por temas.</p> <ol style="list-style-type: none"> 1. Nombre 2. Especificar Tipo de Requerimiento como de Negocios (CM o PPQA) 3. Descripción 4. Autor 5. Prioridad en la implementación: <ol style="list-style-type: none"> 1-Baja 2-Media 3-Alta 4-Muy Alta 6. Producto o Módulo (Falta actividad donde se definan estos módulos) 7. Estado de revisión (Aprobado por el Cliente)

Proceso	Definir ciclos del release
Descripción	<p>Definir los "Ciclos del Release" indicando los "Conjunto de Pruebas" preliminares que los conforman, por ejemplo el primer "Ciclo de Pruebas" debiera incluir siempre un conjunto de "Pruebas Trazadoras" con el fin de establecer si el ambiente está operativo o no.</p> <p>En QC es posible realizar Attachments de información del Ciclo, como podría ser la planificación del mismo. Pero aparentemente estos elementos no son versionados, por lo que quizá es más recomendable sólo adjuntar Links a Subversion.</p>

Proceso	Vincular requerimientos a ciclos
Descripción	<p>Se debe vincular los requerimientos a al menos un "Ciclo de Prueba" con el fin de:</p> <ul style="list-style-type: none"> - Determinar en cualquier instante si los requerimientos están siendo cubiertos o no - Generar esbozo de pruebas en función de los requerimientos asociados a un ciclo. <p>Un requerimiento puede ser vinculado a un Release, en este caso, todos los "Ciclos de Prueba" dentro del Release serán vinculados, y eso implica que debe haber un "Conjunto de Pruebas" al menos dentro de cada Ciclo que valida el requerimiento.</p> <p>Por el momento, no se realizan Pruebas de Regresión al 100% en todos los ciclos por motivos de</p>

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

	recursos. Así la cobertura de prueba en un ciclo queda definida por la liberación de funcionalidades nuevas y por los defectos ocurridos en el ciclo anterior (en el caso que exista), por lo tanto al principio del proyecto, sólo se vincularán los requerimientos asociados a funcionalidades nuevas al ciclo en que se prueban, y con posterioridad dependiendo de los defectos se vincularán requerimientos a ciclos que contengan Pruebas de Regresión.
--	---

Proceso	Generar estrategia pruebas
Descripción	<p>Este documento debe ser un compendio del estado del proyecto hasta este momento, considerando:</p> <ul style="list-style-type: none"> - Roles y personas - Requerimientos de Pruebas - Ciclos de Pruebas - Protocolo de Suspensión de las Pruebas - Conclusiones de Estudio Factibilidad de Automatización. - Referencias a los documentos de Arquitectura de HW/SW. - Formato Plan de Pruebas o Operatoria al respecto - Formato Informe de Resultados de Pruebas. - Manejo de Defectos. - Severidad de Defectos. - Riesgos <p>En el caso de CM-PPQA se incluyen los checklists que se utilizarán en la Revisión.</p>

Proceso	Realizar capacitación aplicativo
Descripción	Realizar Capacitación del Aplicativo, idealmente se debiera hacer una Capacitación para Pruebas Funcionales y No Funcionales, pero podría haber casos donde tenga que ser realizada en sesiones separadas.

Proceso	Definir conjuntos de pruebas
Descripción	<p>Se definen los conjuntos de prueba, ver ejemplos de estos conjuntos en Diagrama Conjunto_Prueba.</p> <p>En QC, Test Lab:</p> <ul style="list-style-type: none"> - Crear carpeta(s) que contienen "Conjuntos de Pruebas". - Crear cada "Conjunto de Pruebas" en la carpeta que corresponda, e ingresar las "Pruebas" que lo componen usando el tab "Execution Grid". El orden de ejecución se define en el tab "Execution Flow".

Proceso	Vincular conjunto pruebas a ciclos
Descripción	En QC, Test Lab, las carpetas que contienen "Conjuntos de Pruebas" son linkeadas a ciclos. Para esto click derecho sobre la carpeta y "Assign to Cycle".

Proceso	Analizar resultados
Descripción	Esta tarea consiste en determinar si los defectos reportados durante las pruebas son tales. - En el caso que el defecto sea tal, actualizar su estado a "Open". - En el caso que el defecto no aplique, se puede borrar o dejar su estado en "Closed", esto último sólo si es de algún aporte mantener el registro.

El ciclo de los defectos o incidentes que se explican en los Modelos anteriores es el siguiente:

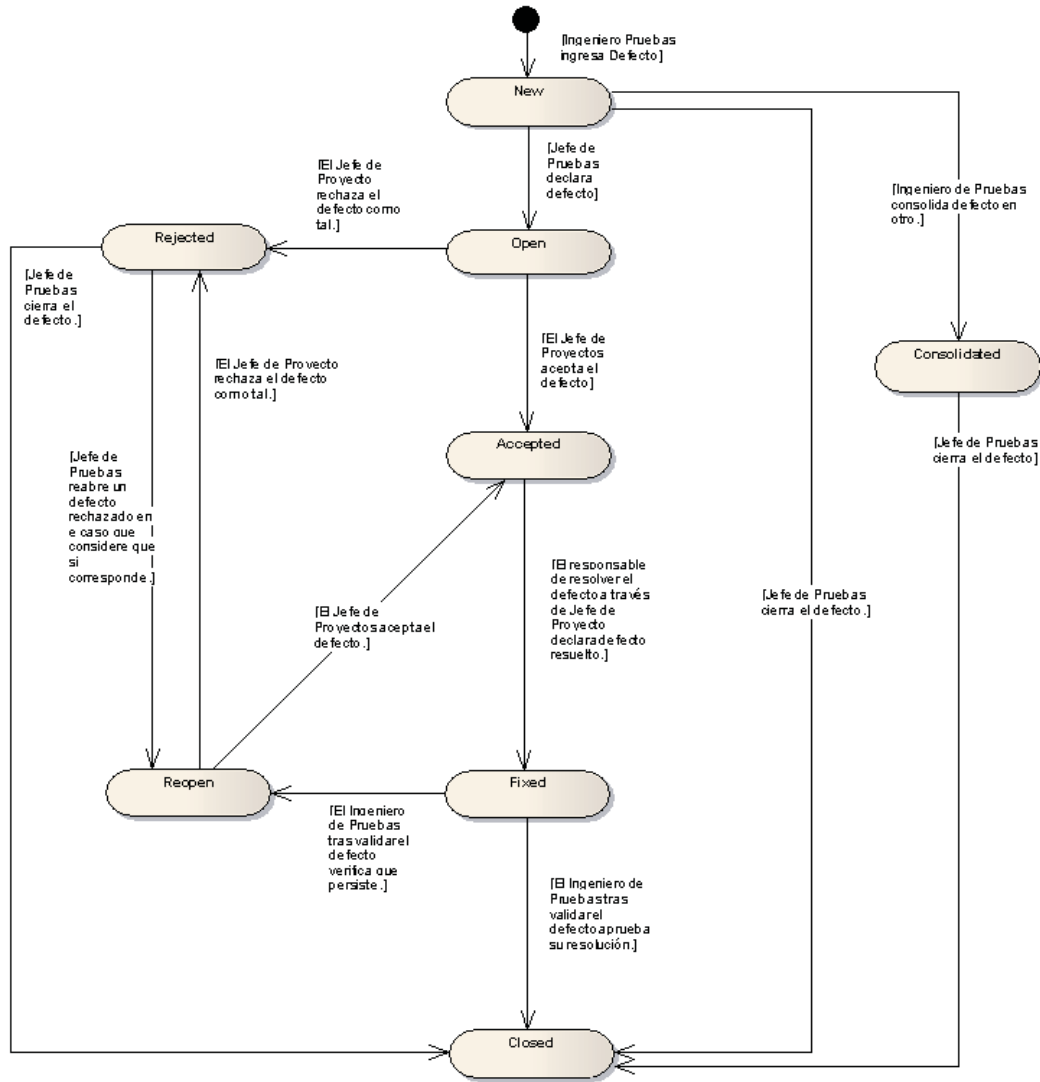


Figura 5.20: “Ciclo de defectos”

5.1.8 Proceso: “Controlar proyecto”

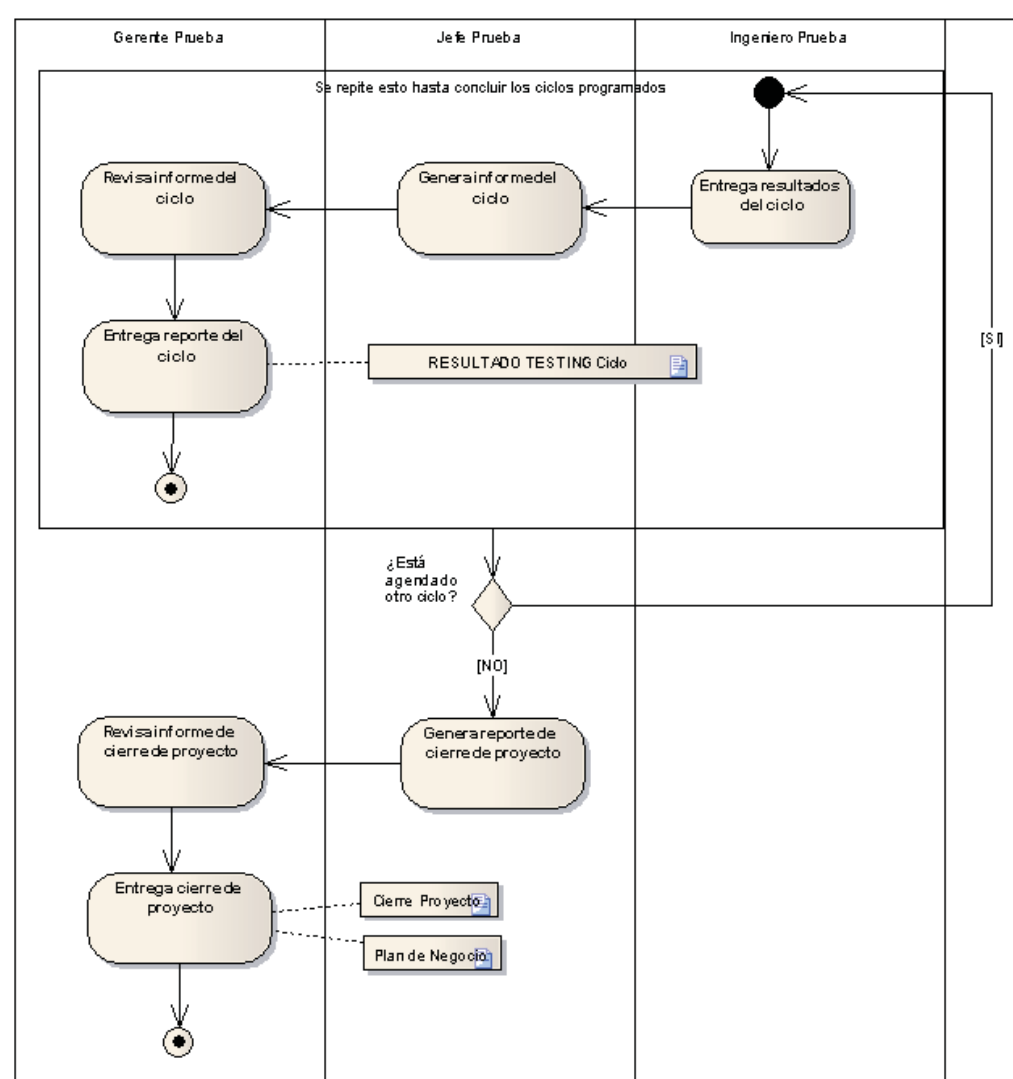


Figura 5.21: Proceso “Controlar proyecto”

Proceso	Entregar resultados del ciclo
Descripción	Una vez ejecutadas las pruebas el Ingeniero de Pruebas entrega los resultados de la ejecución y de los defectos encontrados.
Proceso	Generar informe del ciclo
Descripción	Con la entrega de los resultados el Jefe de Proyecto construye el Informe del Ciclo y lo entrega al Gerente de Pruebas para su revisión
Proceso	Revisar informe del ciclo
Descripción	Cuando el Gerente de proyecto revisa el informe lo libera al equipo de Proyecto para su análisis y revisión en conjunto al equipo de Testing.
Proceso	Entrega reporte del ciclo (*)
Descripción	El Reporte que se genera se basa en la planilla de apoyo “RESULTADO TESTING Ciclo” ANEXO 1. Dicho informe se basa esencialmente en la entrega de los resultados en comparación a las políticas de paso a producción planteadas en la planilla de apoyo “Plan de Pruebas”.
Proceso	Generar reporte de cierre de proyecto (*)
Descripción	En el caso que se terminen de ejecutar todos los ciclos programados o solicitados por emergencia es la hora que el Jefe de Proyecto Genere el reporte de Cierre de Proyecto utilizando la planilla de Apoyo “Cierre Proyecto” y “Plan de Negocio” del ANEXO 1. Luego de generado el reporte, es entregado al Gerente de Proyecto.
Proceso	Revisar informe de cierre de proyecto
Descripción	El Gerente de Proyecto revisa el Informe de Cierre de Proyecto y el Plan de Negocio entregado por el Jefe de Proyecto con el fin de evaluar las diferencias entre lo programado y lo real.
Proceso	Entrega cierre de proyecto
Descripción	Una vez realizada la revisión por el Gerente de Proyecto los entrega y presenta al equipo de proyecto, dejando en conocimiento la desviación u optimización en el caso que exista en el Plan de Negocio junto a los resultados finales de la intervención del Testing.

(*) Es importante mencionar que los procesos en los cuales se aplica alguna de las planillas generadas se destacan con gris su encabezado de descripción del proceso.

5.2 Resumen de resultados obtenido (Métricas)

A partir de las métricas definidas inicialmente se procede a evaluar cada una de ellas con el fin de poder analizar cómo se comportó la aplicación de la Metodología aplicada al proyecto.

Nº	Medición	Medida	Métrica
1	Se medirá el cumplimiento del esfuerzo establecido para el Testing del producto.	HH	Esfuerzo Estimado v/s Esfuerzo Real
2	Se medirá el cumplimiento del costo establecido para el Testing del proyecto.	UF	Costo Estimado v/s Costo Real
3	Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.	%	% Establecido v/s % Obtenido

- 1) Métrica 1: Se medirá el cumplimiento el esfuerzo establecido para el Testing del producto.

En base al Modelo de Servicio se obtienen los siguientes datos:

Días	HH	
2	13	Jefe de Área
14	112	Jefe de Proyecto
27	218	Externos
43	343	Total Esfuerzos

Según los datos obtenidos del Plan de Negocio se tiene lo siguiente:

Mes 2	Mes 3	Mes 4	
Ago-09	Sep-09	Oct-09	
5.00 hh	11.00 hh	6.50 hh	
90.00 hh	140.00 hh	101.00 hh	
			Total HH planificadas
95.00 hh	151.00 hh	107.50 hh	353.50 hh

Es decir, se planificaron 10 horas más de las inicialmente estimadas.

Lo que realmente se gasta en HH son las siguientes:

Mes 2	Mes 3	Mes 4	
Ago-09	Sep-09	Oct-09	
22.00 hh	24.80 hh	58.00 hh	
72.00 hh	214.70 hh	99.00 hh	
		34.00 hh	
	40.00 hh	9.00 hh	
			Total HH Reales
94.00 hh	279.50 hh	200.00 hh	622.50 hh

Existe una diferencia de 269 HH según la estimación inicial y lo real.

- 2) Métrica 2: Se medirá el cumplimiento del costo establecido para el Testing del proyecto.

En base al Modelo de servicio se obtienen los siguientes datos:

UF	
14	Costo Jefe de Área
107	Costo Jefe de Proyecto
194	Costo Externos
315	Total Costo Proyecto

Según los datos obtenidos del Plan de Negocio, se obtiene lo siguiente:

APROBADO TOTAL COSTO SERVICIO TF	446.33 UF
REAL TOTAL COSTO SERVICIO TF	541.06 UF
DIFERENCIA	-94.73 UF

3) Métrica 3: Se medirán las políticas de aceptación establecidas en el Plan de Pruebas del Proyecto.

Los criterios de aceptación establecidos para este proyecto fueron los siguientes (ver planilla aplicada Plan de Pruebas) son:

- 0% de los Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad INVALIDANTE.
- A lo más, 5% de Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad ALTA.
- A lo más, 10% de Casos de Prueba Integrados efectivamente probados con estado NO OK asociados a incidencias con criticidad Media o Baja.
- A lo menos, 85% de Casos de Prueba Integrados efectivamente probados con estado OK.

Los resultados obtenidos en el ciclo 1 ejecutado son los siguientes:

En base a la necesidad de acortar el ciclo de Testing, finalmente se probó un 62% de las pruebas inicialmente planificadas; es decir, 241 CPI de las 387 CPI planificadas.

El resultado de la ejecución de los 241 CPI arroja los siguientes resultados:

Failed	Passed	Total General	
84	157	241	CPI
35%	65%	100%	Porcentajes

Existen 146 CPI con estado N/A, estos casos son los que NO se ejecutaron en el primer ciclo.

De lo anterior se desprende que el aplicativo tiene un **Nivel de Madurez del 65%** sobre los casos de pruebas **ejecutados**, con los siguientes defectos encontrados:

Cuenta de Criticidad	Criticidad				Total general
	1-Baja	2-Media	3-Alta	4-Invalidante	
Total	14	23	62	7	106
Porcentaje	35%		58%	7%	100%

Se recomienda realizar un CICLO 2 de pruebas, para aclarar si las incidencias encontradas en el ciclo anterior, han sido corregidas o corresponden a errores en la instalación del software, en el ambiente de Testing utilizado. Además no se cumplieron los criterios establecidos en el Plan de Pruebas para el paso a producción.

Los resultados obtenidos en el ciclo 2 ejecutado se describen a continuación:

Se indica la estrategia usada en el ciclo 2 de Certificación del proyecto que considera la redefinición del total de pruebas definidas en el Plan de Pruebas inicialmente aprobado, en base a los cambios solicitados por el proyecto (inicialmente en el ciclo 2 se habían definido 194 CPI a ejecutar).

En base a la nueva estrategia que apunta a minimizar los riesgos de no completar las pruebas del ciclo 1, tal como estaban planificadas, en el ciclo 2 se definió la ejecución de 215 Casos de Prueba que estaban compuestos de la siguiente manera:

Failed	Passed	Total general	
65	150	215	CPI
30%	70%	100%	Porcentajes

De lo anterior se desprende que el aplicativo tiene un **Nivel de Madurez del 70%** sobre el total de casos de pruebas **ejecutados**, con la siguiente distribución de los defectos encontrados:

Cuenta de Criticidad	Criticidad				
	1-Baja	2-Media	3-Alta	4-Invalidante	Total general
Total general	2	27	52	9	90
Porcentajes	32%		58%	10%	100%

Se recomienda realizar un CICLO 3 de pruebas, ya que el Nivel de Madurez obtenido no es suficiente para el Paso a Producción del aplicativo. Además no se cumplieron los criterios establecidos en el Plan de Pruebas para el paso a producción.

Pese a lo anterior, el equipo de proyecto define pasar igualmente el aplicativo a producción con los antecedentes de cuáles son los errores detectados para así declararlos a los clientes finales y realizar seguimiento de su resolución.

Los puntos importantes que se rescatan de la entrevista final con la persona que dirigió la aplicación de la Metodología fueron los siguientes:

- 1) Si en el Modelo se tiene claridad de los costos asociados a cada recurso se puede lograr con más fineza el cálculo de la métrica final.
- 2) Las tasas de rendimiento incorporadas en el Modelo de Servicio distaron mucho de lo real; la tasa de rendimiento de los diseños no fue 5 CPI por hora, lo real se acercó al 2.5 por hora.
- 3) Las tasas de rendimiento incorporadas en el Modelo de Servicio de la ejecución de las pruebas no fue 5 CPI por hora, sino que fue de 3 CPI por hora.
- 4) Además es importante mencionar que los costos utilizados en el Modelo Inicial son distintos a los que finalmente se obtuvieron en el Plan de Negocio. Dichos costos fueron finalmente de 1.01 UF HH del Jefe de Proyecto y de 0.88 UF HH para los Externos, Analistas y Ejecutores.
- 5) A pesar de lo antes expuesto, el proceso fue limpio y claro de aplicar. Se siguiere generar una BD de conocimiento de proyectos con el fin de ir refinando cada vez más la estimación.

En base a las observaciones entregadas anteriormente se realiza el ejercicio de ajustar el Modelo de servicio inicial, obteniendo los resultados que se muestran en la Figura 5.22.

Variables					
Universo de Pruebas	2024	CPI	Cantidad de Gestión LB CM	10	Cantidad
Cobertura	85	Universo Funcionalidades	HH invertidas Gestión LB en TF	1	HH
	58	Funcionalidades a cubrir			
	89	% sobre universo	Tiempo Total de Proyecto	0	Meses
Profundidad	20	% sobre universo		0	Semanas
				0	Días hábiles
Cantidad de Ciclos	405	CPI Ciclo 1 (100%)	Costo total del proyecto	1106	UF
	202	CPI Ciclo 2 (50%)	Costo máximo Propuesta TF	332	UF
Diseños	405	CPI			
Ciclo Extra	202	CPI Ciclo N (50%)	N° de Entregas de SW	1	Entregables de SW
HH Mensuales	160	Horas mes			
HH trabajo Diario Interno	8	HH Dia			
HH trabajo Diario Externo	8	HH Dia			
Jefe de Área			Costos Jefe de Área	161	UF Mes
Cantidad	1			40,3	UF Semana
% Dedicación	5,00%	%		8,05	UF Dia Hábil
Horas mensuales	0,08	HH Mensuales		1,01	UF HH
Costo Mensual					
Costo HH					
Jefe de Proyecto			Costos Jefe de Proyecto	141	UF Mes
Cantidad	1			35,3	UF Semana
% Dedicación	100,00%	Real 50%		7,05	UF Dia Hábil
Horas mensuales	1,6	HH Mensuales		0,88	UF HH
N° Analistas/Ejecutores			Costos HH Externos	141	UF Mes
Cantidad	1			35,2	UF Semana
% Dedicación	100,00%	%		7,04	UF Dia Hábil
Horas mensuales	160	HH Mensuales		0,88	UF HH
Tasa de Rendimiento Diseño	2,5	CPI por HH por Recurso			
	20	CPI por HH trabajo Diario Externo			
Tasa de Rendimiento Ejecución	3	CPI por HH por Recurso			
	24	CPI por HH trabajo Diario Externo			

Figura 5.22: Ajuste del modelo en las variables definidas

La métrica de Apego a la Metodología se calcula a continuación:

A	B	C	D	E	F	G	H	I	J	K	L		
Estructura	Plantilla	Propuesta Metodológica	Importancia en Metodología	Apego	Aplicada	Apego por Aplicada	Distribución Importancia	Refinada	Multipl. (Distribución * Refinado)	Apego por Fase	Porcentaje por Estructura		
0. Entradas	Informe de Diseño proveedor(Extracción).doc	CMMI	3	4	12	1	12	0,40	NO	0,9	0,36	0. Entradas	
	PlanificacionSeguimientoSemanal.mpp	CMMI	3	4	12	1	12	0,40	SI	1	0,40		
	Visión y Alcance.doc	CMMI	3	2	6	1	6	0,20	NO	0,9	0,18		0,94
1. Análisis	Matriz_de_Evaluacion_de_Propuestas.xls	CMMI	3	2	6	0	0	0,00	NO	0,9	0,00	1. Análisis	
	Estimación Modelo de Servicios.xls	Cruce	5	4	20	1	20	0,63	SI	1	0,63		
	LB_Requerimientos_Calidad.xls	ITIL	2	3	6	1	6	0,19	NO	0,9	0,17		0,79
2. Diseño	Definición CPI.xls	ITIL	2	4	8	1	8	0,31	NO	0,9	0,28	2. Diseño	
	TF Diseño.xls	ITIL	2	3	6	1	6	0,23	SI	1	0,23		
	Plan_de_CM.doc	CMMI	3	1	3	1	3	0,12	NO	0,9	0,10		
	Plan_de_Pruebas.doc	CMMI	3	3	9	1	9	0,35	NO	0,9	0,31		0,92
3. Ejecución	TF Ejecución.xls	Cruce	5	3	15	1	15	1,00	SI	1	1,00	1,00	3. Ejecución
4. Gestionar Defectos	TF Entidad-Defecto.xls	Cruce	5	3	15	1	15	1,00	SI	1	1,00	1,00	4. Gestionar Defectos
5. Informes	Plan de Negocio.xls	Cruce	5	4	20	1	20	0,50	SI	1	0,50	5. Informes	
	Cierre Proyecto.xlsx	Cruce	5	2	10	1	10	0,25	NO	0,9	0,23		
	RESULTADO TESTING Ciclo.doc	Cruce	5	2	10	1	10	0,25	NO	0,9	0,23		0,95

Figura 5.25: Resultados del apego a la metodología

Métrica de Porcentaje de Apego a la Metodología obtenido = 93%

6 Conclusiones

La Metodología Propuesta se basa fundamentalmente en la necesidad que impone el mercado en entregar servicios de calidad a bajo costo y en los tiempos comprometidos utilizando estándares de mejores prácticas y modelos de servicios que han existido desde hace ya varios años.

Al realizar el cruce de los procesos de CMMI versus las mejores prácticas de ITIL sobre uno de los procesos más importantes a la hora de validar la calidad de un software, se obtiene una Metodología eficiente y consistente de aplicar a cualquier tipo de proyecto de desarrollo. Además la expertise de trabajar en distintos tipos de proyectos en el proceso escogido (Validación y Verificación), apoyó en forma significativa al momento de crear las plantillas de apoyo que ayudan a concretizar la aplicación de la propuesta metodológica a un proyecto real.

El análisis de las métricas en este documento se realiza en función de los dos ciclos ejecutados en el proyecto de Testing, las métricas de Esfuerzo y Costos se realizan en base a los antecedentes entregados por el Jefe de Proyecto de Testing Funcional, en función de que el ciclo debe terminar en la fecha replanificada.

En el cálculo de métricas se puede ver que las variables cuantitativas son posibles de ajustar en función de la precisión y realismo con que se analicen; por ejemplo, las tasas de rendimiento de diseño y ejecución, así como también los costos relacionados al proyecto.

El cálculo de la métrica de Apego a la Metodología ayuda a determinar que en la aplicación de dicha métrica, se establece el grado de utilización efectiva en relación al objetivo de la Metodología. Sin embargo, se pueden recoger mejoras sustanciales en cuanto a la entrevista al jefe de proyecto sobre como fue el proceso de aplicación, ya que él indica que en sí la Metodología le sirvió para mantener el control y orden de los pasos a seguir, además de aumentar la percepción de calidad de servicio por parte del cliente, ya que se había trabajado en otros proyectos con este cliente, sin embargo esta vez las reglas para una mejor entrega del servicio fueron entregadas inicialmente, apoyando el proceso de desarrollo de software en lo que respecta a formalidad en los documentos de diseño y estimación de esfuerzo y planificación más acertada para el desarrollo, ya que consideraron la generación de los documentos necesarios en los momentos reales del ciclo de vida del proyecto y no al final como se estila realizar.

Se propone como trabajo futuro modelar y generar una Base de Conocimiento para registrar los tipos de proyectos y sus mejores ajustes planificados y reales con el fin de mejorar cada vez más el Modelo del Servicio.

También se planteó la iniciativa de llevar a programa la plantilla de Modelado de Servicio, con el fin de que pueda ser accesada desde internet al momento que se desea realizar la estimación de costo o esfuerzo involucrado en un servicio de Testing funcional.

7 Bibliografía

- 1) Axentia (2006), White Paper, Una Introducción a CMMI, Transforming Your IT Organization. (Ver ANEXO 2 entregado en el Informe de Avance), 2006.
- 2) Fernando Sánchez, (2008), CMMI vs. ITIL: dos enfoques complementarios, gerente de Grupo Delaware, <http://www.mkm-pi.com/mkmpi.php?article1817>, Jueves 6 de marzo de 2008.
- 3) Juan Palacio, (2006), Sinopsis de los Modelos SW-CMM y CMMI, <http://www.navegapolis.net/jpalacio@navegapolis.net>, 1.0 Abril – 2006.
- 4) Juan Raggio Pérez, (2004), Universidad Politécnica de Madrid, Facultad de Informática, Estudios de Doctorado, DESARROLLO DE PROCESOS DE GESTIÓN DE SERVICIOS DE EXPLOTACIÓN SIGUIENDO EL MODELO CMMI, 2004.
- 5) Lorena León – Gabriela Puglla, (2008), Métricas de Proceso y proyecto de software, Universidad Técnica particular de Loja. <http://www.slideshare.net/loreknelamorena/mtricas-de-proceso-y-proyecto-de-software>, 2008.
- 6) Mariana Isela Jaramillo González (2005), Utilización de Estándares ITIL para lograr el Nivel 3 de CMMI en una Organización. Universidad Autónoma del Estado de México, <http://docente.uco.mx/juancont/documentos/cap02/6.pdf>, 24 Octubre 2005.
- 7) OGC (2007), ITIL Managing IT Services. Best Practice for Application Management, http://www.ogc.gov.uk/guidance_itol.asp, 2007.
- 8) PMBOK (2004), [PMBOK](#) Tercera Versión en Español [Project Management Institute](#). «Capítulo 1», *Guía de los Fundamentos de la Dirección de Proyectos*, 3ª edición. [ISBN 1-930699-73-5](#). http://es.wikipedia.org/wiki/Gesti%C3%B3n_de_proyectos, 2004.
- 9) Raúl Suárez O. y Felipe Donoso Jaurés, (2006), METODOLOGÍA ITIL, Descripción, Funcionamiento y Aplicaciones, PÍA RAMÍREZ BRAVO - FELIPE DONOSO JAURÉS, Santiago, Julio de 2006.
- 10) Serge Thom Geneva (2007), ITIL and CMMI synergies. <http://sergethorn.blogspot.com/2007/05/itil-and-cmmi-synergies.html>, 23 Mayo 2007.