

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

CMM: PRINCIPIOS Y PRÁCTICA

José Ramón Maureira Martínez

Profesor Guía: **Dr. Broderick Crawford Labrín**

Carrera: **Ingeniería Civil en Informática**

Julio, 2013

Resumen

Capability Maturity Model for Software es un modelo para evaluar la madurez del proceso de Software de una organización y para identificar las prácticas claves que se requieren para aumentar la madurez de su proceso. Este documento presenta un estudio y análisis del modelo. Además, presenta una propuesta de mejora del Proceso de Software en una empresa de desarrollo de Software. El sistema de trabajo empleado fue la realización de una encuesta, la cual fue ejecutada por medio del cuestionario de madurez del Software Engineering Institute. La metodología de trabajo empleada en la evaluación, se encuentra basada en el método CMM-Based Appraisal for Internal Process Improvement.

Palabras Claves: Capability Maturity Model for Software, cuestionario de madurez, CMM-Based Appraisal for Internal Process.

Abstract

Capability Maturity Model for Software is a model to evaluate the maturity of the Software process in an organization and to identify the key practices that are required to increase the maturity of its process. This document presents a study and analysis of the model. Also, it presents a proposal of improvement of the software process in a software development company. The work system employed was the carrying out of a survey, which was executed by a maturity questionnaire of the Software Engineering Institute. The work methodology used in the evaluation is found in the CMM-Based Appraisal for Internal Process Improvement method.

Key Words: Capability Maturity Model for Software, maturity questionnaire, CMM-Based Appraisal for Internal Process.

Índice

Resumen	i
Abstract	i
Lista de Figuras	iv
Lista de Tablas	v
1 Introducción	1
2 Definición de Objetivos	2
2.1 Objetivo General	2
2.2 Objetivos Específicos	2
3 Estado del Arte	3
4 Capability Maturity Model for Software	3
4.1 Conceptos Previos	4
4.2 Definición	5
4.3 Niveles de Madurez	6
4.3.1 Nivel 1: Inicial.....	6
4.3.2 Nivel 2: Repetible.....	7
4.3.3 Nivel 3: Definido.....	8
4.3.4 Nivel 4: Administrado.....	8
4.3.5 Nivel 5: Optimizando.....	9
5 Áreas Claves de Proceso	10
5.1 Nivel 1	12
5.2 Nivel 2	12
5.2.1 Administración de Requerimientos.....	12
5.2.2 Planificación del Proyecto de Software.....	13
5.2.3 Seguimiento y Control del Proyecto de Software.....	13
5.2.4 Administración de Subcontratos.....	14
5.2.5 Aseguramiento de la Calidad del Software.....	14
5.2.6 Administración de la Configuración del Software.....	15
Nivel 3	15
5.2.7 Administración Integrada de Software.....	16
5.2.8 Coordinación Intergrupala.....	16
5.2.9 Enfoque en el Proceso de la Organización.....	16
5.2.10 Definición del Proceso de la Organización.....	17
5.2.11 Programa de Capacitación.....	17
5.2.12 Ingeniería de Productos de Software.....	17
5.2.13 Revisiones de Pares.....	18
5.3 Nivel 4	18
5.3.1 Administración Cuantitativa del Proceso.....	18
5.3.2 Administración de la Calidad del Software.....	19

5.4	Nivel 5	19
5.4.1	Administración del Cambio de Procesos	19
5.4.2	Prevención de Defectos	20
5.4.3	Administración de Cambios Tecnológicos	20
6	Estructura del SW-CMM	21
6.1	Características Comunes	22
6.2	Prácticas Claves	23
6.3	Metas.....	23
7	Métodos de Evaluación	24
7.1	Introducción.....	24
7.2	Modelo IDEAL.....	25
7.3	Métodos de Evaluación	25
7.3.1	Método CBA IPI	26
7.3.2	Método SCE.....	30
8	Evaluación de una Empresa de Desarrollo de Software.....	36
8.1	Aspectos Generales.....	36
8.2	Etapas de la evaluación.	37
8.2.1	Planear y preparar la evaluación.	37
8.2.2	Conducir la evaluación.	37
8.3	Evaluación de los Niveles de Madurez.....	38
8.3.1	Herramientas utilizadas por la organización.	38
8.3.2	Cumplimiento Nivel 2	39
8.3.3	Cumplimiento Nivel 3.	42
8.3.4	Cumplimiento Nivel 4.	45
8.3.5	Cumplimiento Nivel 5.	46
9	Propuesta de Mejoramiento	48
9.1	Definición del proceso de desarrollo de software	48
9.1.1	Especificación de requerimientos.....	48
9.1.2	Análisis	50
9.1.3	Diseño.....	53
9.1.4	Construcción.	54
9.1.5	Pruebas.....	54
9.2	Aseguramiento de la calidad del software.....	56
9.3	Administración de la configuración del software.....	57
9.4	Capacitaciones	58
10	Conclusiones.....	60
11	Referencias	62
Anexos		65
A: Cuestionario de Madurez.....		65

Lista de Figuras

Figura 4.1: Niveles de Madurez de CMM.	6
Figura 6.1: Estructura de CMM.	21
Figura 7.1: Actividades de evaluación de los métodos basados en el CAF	27
Figura 7.2: Fases de implementación de una evaluación	28
Figura 9.1: Diagrama de actividades de la especificación de requerimientos.	49
Figura 9.2: Diagrama de actividades del análisis.	51
Figura 9.3: Diagrama de actividades del diseño.	53
Figura 9.4: Diagrama de actividades del desarrollo.	54
Figura 9.5: Diagrama de actividades de las pruebas.	55

Lista de Tablas

Tabla 5.1: División de las ACP	11
Tabla 6.1: CMM en Números	21
Tabla 7.1: Actividades del SCE.....	32
Tabla 8.1: Cuadro resumen de las actividades en las que participan los encuestados.	37

1. Introducción

El presente estudio explica y detalla una propuesta de mejoramiento del proceso de software, basada en el Modelo de Madurez de la Capacidad para el desarrollo de Software [1] (Capability Maturity Model for Software, SW-CMM) [2]. SW-CMM es un modelo para evaluar la madurez del proceso de software de una organización y para identificar las prácticas claves que se requieren para aumentar la madurez de su proceso [3].

Como primer paso en la investigación, se realizó el estudio y análisis del modelo SW-CMM. Luego, se realizó un estudio de sus métodos de evaluación, es decir, qué pasos se deben seguir al momento de evaluar una organización, con el fin de determinar el nivel de madurez de sus procesos [4]. Posteriormente, los conceptos adquiridos, tanto los de CMM como los de sus métodos de evaluación fueron aplicados a un caso real, en el cual se evaluó una organización determinada.

La metodología de trabajo empleada en la evaluación, se encuentra basada en el método CMM-Based Appraisal for Internal Process Improvement (CBA IPI) [5]. El sistema de trabajo empleado fue en primera instancia la realización de una encuesta, la cual fue ejecutada por medio del cuestionario de madurez [6] del Software Engineering Institute (SEI). Posteriormente, esta encuesta fue enriquecida por medio de entrevistas a gerentes, jefes de proyecto y desarrolladores de la organización y complementada con la revisión de la documentación asociada a los proyectos. Como resultado de la evaluación, se presentan las debilidades y fortalezas identificadas, y se describe una propuesta de mejoramiento destinada a satisfacer de manera completa, los requerimientos definidos por el modelo CMM, con el fin de alcanzar el CMM Nivel 2 [7].

Los siguientes capítulos explican en una primera parte, los conceptos necesarios para llegar a entender qué es el CMM. Se detalla su estructura, cómo está organizado, y las distintas áreas claves de proceso que lo componen. Posteriormente, se describe la metodología de trabajo empleada y los resultados obtenidos, luego de haber efectuado la evaluación del proceso de software. Luego, se presenta una propuesta de mejora [8], según los parámetros definidos por el CMM. La propuesta presentada tiene por objetivo lograr el Nivel 2 de CMM.

2. Definición de Objetivos

A continuación se presentará el objetivo general y los objetivos específicos.

1.1 Objetivo General

Realización de un análisis y una propuesta de mejora del Proceso de Software en una empresa de desarrollo de software.

1.2 Objetivos Específicos

- Estudio y análisis del modelo SW-CMM y sus métodos de evaluación.
- Evaluación de una organización (empresa de desarrollo de software).
- Elaboración de una propuesta de mejoramiento para alcanzar mejores niveles de madurez.

3. Estado del Arte

En 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. Ahí es cuando se crea el Instituto de Ingeniería del Software (SEI), fundado por el Departamento de Defensa Americano y la Universidad Carnegie Mellon. El 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)”. Luego SEI publica la versión 1.0 del Modelo de Madurez de las Capacidades para el Software (SW-CMM). Desde ese entonces esta versión fue evolucionando. [1]

En 1993 Paulk junto a otros autores [4] realizó un documento que proporciona las prácticas clave que corresponden a cada nivel de madurez del SW-CMM. Es una elaboración de lo que se entiende por madurez en cada nivel del CMM y una guía que se puede utilizar para la mejora de proceso de software, las evaluaciones del proceso de software y las evaluaciones de las capacidades de software.

El año 2001 se realizó el “CMM-Based Appraisal for Internal Process Improvement” (CBA IPI) [5], el cual es un resumen de alto nivel del CBA-IPI V1.2 y es una actualización del CBA-IPI V1.1. Proporciona una breve historia de los métodos de evaluación del SEI, y establece una evaluación en el contexto del enfoque IDEALSM a la mejora del proceso de software. CBA-IPI es una herramienta de diagnóstico que apoya, permite y fomenta el compromiso de la organización para la mejora de procesos.

Un proyecto que llevó a cabo un modelo sobre una empresa, fue el “Programa para el mejoramiento del proceso de software en el BBVA Banco Bhif”, en el año 2002. [9]

En 2005 se llevó a cabo el “Desarrollo de un modelo de madurez para valorar la gestión de proyectos en las organizaciones” [10]. El grupo de Investigación en Gestión y Evaluación de Programas y Proyectos de la Universidad del Valle en Cali-Colombia ha desarrollado un modelo de madurez construido a partir del cuerpo del conocimiento propuesto por el Project Management Institute y enriquecido con aproximaciones adicionales que reconocen y miden el contexto en el cual se desarrollan los proyectos en las organizaciones.

Un reciente estudio realizado el año 2014 [22] examinó el entorno de desarrollo de software en pequeñas y medianas empresas. Examinó los principios de funcionamiento del Modelo de Madurez de las Capacidades para el desarrollo del Software, con el fin de establecer los parámetros críticos que deben ser considerados para la adopción exitosa del SW-CMM.

4 Capability Maturity Model for Software

4.1 Conceptos Previos

Según [1] el SW-CMM es: “un modelo para evaluar la madurez del proceso de software de una organización y para identificar las prácticas claves que se requieren para aumentar la madurez de su proceso”. Para entender mejor esta definición, a continuación se analizarán los términos claves que emplea esta definición [9]:

- **Proceso de software:** ¿Qué se entiende por proceso de software? Según [1] el proceso de software es: “un conjunto de actividades, métodos, prácticas, y transformaciones que se usan para desarrollar y mantener el software y sus productos asociados”. Roger Pressman [2] señala que el proceso es un marco de trabajo de las tareas que se requieren para construir software de alta calidad.
- **Madurez:** La madurez es el ámbito o contexto en el cual el proceso de software es explícitamente definido, administrado, medido, controlado y efectivo. Así un proceso maduro debiera ser bien entendido y consistentemente implementado en toda la organización y además estar en un estado de mejoramiento continuo.

El objetivo de SW-CMM es lograr un proceso maduro de software [10], es decir, lograr un proceso de software que esté definido, documentado, que sea administrado (controlado y medido), que sea consistentemente implementado, que sea bien entendido (vía capacitación), que sea apoyado por la alta gerencia (estableciendo la infraestructura y roles necesarios para institucionalizarlo) y además que puede ser mejorado, encontrándose en un estado de mejoramiento continuo, en el cual el proceso es continuamente evaluado y son incorporadas las mejoras necesarias, destinadas a resolver sus debilidades.

Otro de los conceptos a los que hace referencia SW-CMM, es el de capacidad. La capacidad del proceso, describe el rango de resultados esperados que pueden ser logrados siguiendo un proceso de software. La capacidad del proceso proporciona una manera de predecir los resultados más probablemente esperados en el próximo proyecto de software implementado. Se dirá entonces que un aumento en la madurez del proceso implica un aumento en su capacidad, ya que al tener un proceso administrado, definido y practicado, se estará más cerca de lograr los objetivos establecidos, en cuanto a planificación, costo y funcionalidad objetivo.

La mejora del proceso debe estar sustentada por una infraestructura y una cultura que soporte los métodos, prácticas y procesos, para que estos sean la manera real de hacer negocios. La infraestructura es el marco de trabajo de una organización, incluyendo estructuras organizacionales, políticas, estándares, métodos, entrenamiento y herramientas que apoyen el desempeño. Al proceso mediante el cual la organización crea la infraestructura y la cultura necesaria para soportar los métodos, prácticas y procesos, se le denomina institucionalización del proceso.

4.2 Definición

Aunque los jefes de proyecto e ingenieros de software saben a menudo sus problemas en gran detalle, pueden existir discrepancias al momento de decidir qué mejoras son prioridad. Sin una estrategia organizada para la mejora, es difícil lograr un acuerdo general entre la dirección general y el personal, en cuáles son las actividades prioritarias de mejora. Para lograr resultados duraderos en los procesos, es necesario diseñar un camino evolutivo que aumente la madurez del proceso de software de una organización en fases. El SW-CMM ordena estas fases, para que las mejoras adquiridas en cada fase, proporcionen el soporte necesario para construir las mejoras emprendidas en la próxima fase. Así, una mejora basada en el SW-CMM, proporciona una guía para el mejoramiento continuo del proceso [11].

El SW-CMM fue diseñado para guiar a las organizaciones de software, en la selección de las estrategias de mejora del proceso [12], basado en el estado de madurez actual del proceso, determinando e identificando los puntos críticos de calidad del software y mejora del proceso. Enfocándose en un conjunto limitado de actividades y trabajando para lograrlos, una organización puede mejorar su proceso de software, permitiendo mejoras continuas y duraderas en la capacidad del proceso de la organización.

La estructura del SW-CMM está basada en principios de calidad del producto que han existido durante los últimos sesenta años. Estos principios han sido adaptados por el SEI y transformados en un modelo de madurez, que establece una base para la administración e ingeniería del proyecto, para el control cuantitativo del proceso del software, el cual es la base para la mejora continua del proceso.

La estructura del proceso de madurez [20] está pensada para su uso con una metodología de evaluación y un sistema de gestión. La evaluación permite determinar el nivel de madurez de la organización, mientras que el sistema de gestión establece una estructura para implantar las acciones prioritarias de mejoramiento [3].

4.3 Niveles de Madurez

El SW-CMM organiza sus prácticas de definición y mejora del proceso, en cinco etapas, llamadas niveles de madurez. Estos niveles de madurez, definen una escala para medir la madurez del proceso y para evaluar su capacidad. Los niveles también ayudan a una organización a priorizar sus esfuerzos de mejora.

Un nivel de madurez es una plataforma evolutiva bien definida, para lograr un proceso maduro de software. Cada nivel de madurez constituye un hito para la mejora continua del proceso. Cada nivel comprende un conjunto de prácticas claves que cuando son cumplidas logran un componente importante del proceso de software [19].

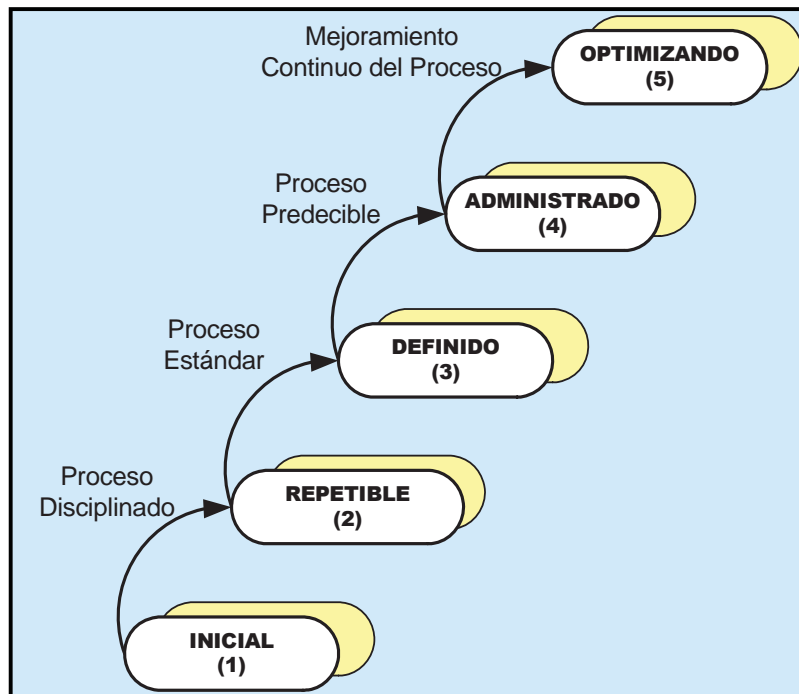


Figura 4.1: Niveles de Madurez de SW-CMM.

4.3.1 Nivel 1: Inicial

En este nivel la capacidad del proceso es imprevisible, no se cumplen los procedimientos establecidos, la planificación de los proyectos y los costos estimados. El desempeño del proyecto depende de la capacidad del equipo de trabajo y varía con sus habilidades natas, conocimiento y motivaciones.

El proceso en este nivel es clasificado como ad hoc o caótico, ya que el proceso es improvisado: no existe control de cambios y el proceso no es controlado, ni medido. Típicamente los administradores se abocan en resolver “crisis inmediatas”, dejando de lado el proceso, centrándose en las tareas de codificación y testing.

En este nivel existen pocos procesos de software estables, y el desempeño sólo puede predecirse por la capacidad del grupo de trabajo, en lugar de contar con un proceso organizacional. Todas las organizaciones por defecto se encuentran en este nivel.

4.3.2 Nivel 2: Repetible

En este nivel, se establecen políticas para administrar los proyectos y procedimientos para verificar que esas políticas son implementadas y seguidas.

Este nivel se define como repetible, por el concepto de repetir el éxito logrado en proyectos anteriores, en nuevos proyectos de similares características. Al implementar las prácticas claves de este nivel la organización gana en experiencia y además le permite conocer sus procesos, con cual se realizan planes y estimaciones más realistas (basadas en la experiencia) que le permiten lograr sus objetivos de calidad, costo y planificación. Los proyectos entonces, implementan procesos efectivos, que están bien definidos, documentados, practicados, entrenados y medidos, pero que pueden ser mejorados.

Los plazos, costos y recursos estimados en los proyectos son más reales. Para lograr esto, es necesario, llevar un recuento en cada proyecto acerca de costos, tiempo y funcionalidad (esto puede estar sustentado por una base de datos del proceso). Si se presentara el caso, los problemas para cumplir estos compromisos son detectados en su momento, ya que se cuenta con prácticas de supervisión y control, tanto del proyecto como del proceso.

La capacidad del proceso de software de este nivel puede resumirse como “disciplinada”, porque planea y controla si el proyecto se mantiene estable y pueden repetirse los éxitos más tempranamente. Se puede decir que el proyecto de software está bajo control por un sistema de administración eficaz, siguiendo planes realistas, basado en el desempeño de proyectos anteriores.

Según [3] las actividades necesarias para evolucionar al siguiente nivel son:

- Establecer un grupo de proceso, que se dedique por completo al mejoramiento continuo del proceso.
- Definir una arquitectura para el proceso de desarrollo de software.
- Introducir una familia de métodos y tecnologías de ingeniería de software.

4.3.3 Nivel 3: Definido

La característica principal de este nivel es que se cuenta con un *proceso de software estándar* para desarrollar y mantener el software. Este proceso se encuentra definido y documentado y es implementado a lo largo de toda la organización en distintos proyectos.

Algunos aspectos claves de este nivel son [1]:

- Se hace uso de prácticas de ingeniería de software para estandarizar el proceso de software definido.
- Un grupo es responsable por las actividades del proceso de software.
- Un programa de entrenamiento es implementado para asegurar que el equipo de trabajo, tengan el conocimiento y las habilidades requeridas para cumplir sus roles asignados.
- El proceso de software estándar de la organización se “customiza” para cada proyecto, para así determinar el proceso de software definido del proyecto. Este contendrá información sobre los productos de trabajo en cada etapa de desarrollo, sus entradas, estándares y procedimientos utilizados para desarrollar el trabajo, mecanismos de control y sus salidas.

La capacidad del proceso de software de este nivel puede resumirse como “consistente y estándar” porque las actividades de ingeniería y administración son estables y repetibles. Dentro de las líneas del producto son verificadas y están bajo control: la planificación, el costo y la calidad del software [18]. Esta capacidad del proceso está basada en un entendimiento común de la organización, de las actividades, roles, y responsabilidades del proceso de software.

Según [3] las actividades necesarias para evolucionar al siguiente nivel son:

- Establecer un conjunto básico de métricas de procesos [13], para identificar parámetros de calidad y costo de cada paso del proceso.
- Establecer una base de datos del proceso de software y destinar los recursos necesarios para administrarla y mantenerla.
- Disponer de los recursos suficientes para reunir y mantener los datos del proceso y para asesorar a los miembros de los proyectos para que los usen.
- Evaluar la calidad relativa de cada producto e informar a la gerencia acerca de los objetivos de calidad que no han sido alcanzados.

4.3.4 Nivel 4: Administrado

En el nivel administrado, la organización fija metas de calidad cuantitativas para el proceso y los productos de software. La productividad y la calidad son medidas para las actividades importantes del proceso en todos los proyectos, como parte de un programa de control organizacional.

Una base de datos se utiliza para almacenar y analizar los datos disponibles de los procesos definidos de los proyectos. Los procesos de software son instrumentados con medidas bien definidas y consistentes. Estas medidas, establecen la base cuantitativa para evaluar el proceso de software de los proyectos y sus productos. Los proyectos logran el control sobre sus productos y procesos estrechando el margen de variación en su rendimiento, para caer dentro de límites cuantitativos aceptables.

La capacidad del proceso de software de este nivel puede resumirse “cuantificable y predecible”, porque los procesos y productos, son medidos y operan dentro de límites cuantificables. Este nivel de capacidad, permite a una organización predecir las tendencias en el proceso y calidad del producto, dentro de límites definidos. Cuando estos límites se exceden, se toman las acciones necesarias para corregir la situación.

Según [3] las actividades necesarias para evolucionar al siguiente nivel son:

- Soportar la obtención automática de datos del proceso de software.
- Usar estos datos para analizar y modificar el proceso para prevenir problemas y mejorar la eficiencia.

4.3.5 Nivel 5: Optimizando

En el nivel optimizando, la organización entera se enfoca en una continua mejora del proceso. La organización tiene los medios para identificar las debilidades y fortalezas del proceso, con la meta de prevenir la ocurrencia de fallas.

Los equipos de trabajo de Nivel 5 analizan los defectos para determinar sus causas. Se evalúan los procesos de software para prevenir que ocurran tipos de errores conocidos, las lecciones aprendidas son aplicadas en otros proyectos.

En este nivel, los gerentes son capaces de estimar y después dar un seguimiento cuantitativo al impacto y a la efectividad del cambio. A partir de la eficiencia del proceso, es posible generar reportes de costo - beneficio de nuevas tecnologías o proponer cambios al proceso estándar de la organización.

La capacidad del proceso de software de este nivel puede resumirse como un “mejoramiento continuo”, porque las organizaciones de Nivel 5 están continuamente esforzándose para mejorar el rango de su capacidad del proceso, mejorando así, el rendimiento del proceso de sus proyectos. La mejora ocurre por los avances incrementales en el proceso existente y las innovaciones al usar las nuevas tecnologías y métodos.

4. Áreas Claves de Proceso

Un área clave de proceso (ACP), es un conjunto de actividades relacionadas que, cuando se realizan colectivamente, logran un conjunto de metas consideradas importantes para aumentar la capacidad del proceso. Cada nivel de madurez (a excepción del Nivel 1 o Inicial) se compone de un conjunto de ACP, que indican las áreas en las que se debería enfocar la organización para mejorar su proceso de software.

Si bien las ACP residen en un nivel de madurez específico, éstas deben ser practicadas o tenerse en cuenta independiente del nivel de madurez de la organización [3]. Cabe considerar que las organizaciones aunque practiquen o presten atención a todas las áreas claves de proceso, ellas deben centrar su atención en las ACP próximas a su nivel de madurez, ya que a través de su institucionalización podrán aumentar su nivel de madurez y tendrán el soporte necesario para centrar sus esfuerzos en ACP de niveles superiores.

Las ACP según [4] se pueden clasificar en tres categorías, independiente de su nivel de madurez:

- Procesos de administración: Contiene las actividades de administración de proyectos.
- Procesos organizacionales: Contiene las responsabilidades de la organización por el proceso de software.
- Procesos ingenieriles: Contiene las actividades propias de la ingeniería de software, tales como análisis de requerimientos, diseño, codificación y prueba, que se realizan en todos los niveles pero que evolucionan según el nivel de madurez en que se encuentran.

La tabla 5.1 muestra cómo se dividen las ACP según la clasificación antes descrita.

Categorías de proceso	Administrativos	Organizacionales	Ingenieriles
Niveles			
5 Optimizando		Administración de Cambios Tecnológicos	
	Administración del Cambio de Procesos		Prevención de Defectos
4 Administrado	Administración Cuantitativa del Proceso		Administración de la Calidad del Software
3 Definido	<ul style="list-style-type: none"> ▪ Administración Integrada de Software. ▪ Coordinación Intergrupala. 	<ul style="list-style-type: none"> ▪ Enfoque en el Proceso de la Organización. ▪ Definición del Proceso de la Organización. ▪ Programa de Capacitación. 	<ul style="list-style-type: none"> ▪ Ingeniería de Productos de Software. ▪ Revisiones de Pares.
2 Repetible	<ul style="list-style-type: none"> ▪ Administración de Requerimientos. ▪ Planificación del Proyecto de Software. ▪ Seguimiento y Control del Proyecto de Software. ▪ Administración de Subcontratos de Software. ▪ Aseguramiento de la Calidad del Software. ▪ Administración de la Configuración del Software. 		
1 Inicial	Proceso Ad Hoc		

Tabla 5.1: División de las ACP

Cada ACP tiene un conjunto de metas a ser logradas. Para cumplir con una ACP, deben ser satisfechas todas sus metas establecidas. Cuando las metas de una ACP son cumplidas continuamente por los proyectos, puede decirse que la organización ha institucionalizado la capacidad del proceso caracterizada por la ACP. Asimismo, para lograr un nivel de madurez, deben satisfacerse todas las ACP asociadas al nivel (con la excepción del Nivel 2, en el cual se puede obviar la ACP: “Administración de Subcontratos de Software” si es que la organización no mantiene subcontratos con organizaciones externas).

El adjetivo “clave” implica que hay áreas del proceso (y procesos) que no son claves para lograr un nivel de madurez. El SW-CMM no describe en detalle todas las áreas del proceso que están involucradas en el desarrollo y mantención del software, sino que se centra en ciertas áreas definidas como determinantes para aumentar la capacidad del proceso y que tienen como fin lograr un proceso maduro de software.

A continuación se detallan las ACP, ordenadas según el nivel de madurez al cual pertenecen:

4.4 Nivel 1

En este nivel no existen ACP, ya que las organizaciones no implementan ninguna práctica que les permita controlar y mejorar su proceso de software. El éxito del proyecto radica casi exclusivamente en la capacidad del equipo de trabajo que opera en la organización.

4.5 Nivel 2

Las ACP del nivel 2, apuntan a establecer los controles básicos que permitan implantar una eficaz y eficiente administración de proyectos. Las ACP de nivel 2, se detallan a continuación.

4.5.1 Administración de Requerimientos

Su propósito es establecer un entendimiento común entre el cliente y el proyecto de software, acerca de los requerimientos del cliente que serán abordados por el proyecto de software [4]. Este acuerdo es la base para estimar, planificar, ejecutar y seguir las actividades del proyecto de software.

Las tareas principales de la administración de requerimientos son:

- Asegurar que los requisitos del sistema asignados al software estén documentados y sean controlados.
- Efectuar el control de cambios de los requerimientos del sistema. Toda vez que se cambien los requerimientos del sistema, se deben ajustar la planificación, los productos intermedios y las actividades relacionadas.

Sus metas son [4]:

- Controlar los requerimientos del sistema asignados al software con el fin de establecer una línea base para su uso en la administración e ingeniería de software.
- La planificación, las actividades y los productos del software, son mantenidos consistentes con los requerimientos del sistema asignados al software.

4.5.2 Planificación del Proyecto de Software

Su propósito es establecer planes razonables para realizar las tareas de administración e ingeniería de software [4].

La planificación del proyecto de software involucra:

- Desarrollar estimaciones de esfuerzo, costo y tamaño, para los productos y recursos requeridos.
- Establecer los acuerdos y compromisos necesarios.
- Identificar y evaluar los riesgos asociados al proyecto.
- Definir el plan del proyecto de software.

Sus metas son [4]:

- Documentar las estimaciones, para ser usadas en la planificación y seguimiento del proyecto.
- Planificar y documentar las actividades y compromisos del proyecto.
- Los grupos y las personas involucradas aprueban sus compromisos relacionados al proyecto.

4.5.3 Seguimiento y Control del Proyecto de Software

Su propósito es proveer una adecuada visión del avance real del proyecto, de forma que la administración pueda tomar acciones efectivas cuando el rendimiento del proyecto se desvíe de la planificación definida [4]

El seguimiento y control del proyecto, involucra seguir y revisar los logros y resultados del proyecto, en contraste a las estimaciones, compromisos y planes definidos, y ajustar el plan de acuerdo a los logros y resultados reales.

Sus metas son [4]:

- El rendimiento y los resultados obtenidos son seguidos y contrastados con el plan del proyecto.
- Se toman y administran acciones correctivas cuando los resultados y el rendimiento obtenidos se desvían significativamente del plan del proyecto.
- Los cambios en los compromisos adquiridos son acordados entre los grupos y personas afectadas.

4.5.4 Administración de Subcontratos

Su propósito de la administración de subcontratos es seleccionar subcontratistas de software calificados y administrarlos efectivamente [4].

Las tareas que involucra esta ACP son:

- Selección del subcontratista sobre la base de su capacidad para desarrollar el trabajo solicitado. Esto se realiza de acuerdo a un procedimiento documentado.
- Establecer compromisos con el subcontratista (por ejemplo: plan de trabajo y control de cambios). Los cambios a los subcontratos son hechos con el acuerdo de ambas partes.
- Efectuar el seguimiento y control del desempeño y los resultados del subcontratista.

Sus metas son [4]:

- Seleccionar subcontratistas de software calificados.
- El contratante y el subcontratista están de acuerdo en sus compromisos.
- El contratante y el subcontratista mantienen una comunicación continua.
- El contratante realiza un seguimiento de los resultados del subcontratista, en relación a los compromisos establecidos.

4.5.5 Aseguramiento de la Calidad del Software

Su propósito es brindar a la administración una visibilidad adecuada del proceso de software y los productos que están siendo construidos.

El aseguramiento de la calidad del software involucra revisar y auditar los productos generados y las actividades realizadas, a fin de asegurar que ellos cumplan con los estándares y procedimientos definidos, entregando a la gerencia los resultados de esas revisiones o auditorías.

El grupo SQA prepara un plan de aseguramiento de calidad para el proyecto, el cual es la base de las actividades a realizar por el grupo. Además, participa en la preparación y revisión del plan de desarrollo del software y de los estándares y procedimientos a ser utilizados en el proyecto.

El grupo SQA revisa los productos de trabajo en cada etapa del ciclo de vida del software y verifica su conformidad con los estándares y procedimientos definidos. Cuando existen desviaciones respecto de las metas a cumplir, estas son documentadas y administradas de acuerdo a un procedimiento documentado.

Sus metas son [4]:

- Planificar las actividades del aseguramiento de la calidad del software.
- Verificar objetivamente que los productos y actividades satisfacen los requisitos y estándares aplicables.
- Los grupos y personas afectadas son informadas sobre las actividades del aseguramiento de calidad y de los resultados obtenidos.
- Problemas de no conformidad que no puedan ser resueltos dentro del proyecto son abordados por la gerencia.

4.5.6 Administración de la Configuración del Software

Su propósito es establecer y mantener la integridad de los productos de software del proyecto a través del ciclo de vida del proyecto de software [4]. Involucra identificar la configuración del software, es decir, una selección de productos de trabajo y sus descripciones, en puntos dados de tiempo, controlando los cambios a la configuración, y manteniendo la integridad y auditoriabilidad de la configuración a través del ciclo de vida del software.

Algunos productos de trabajo de software, tales como el plan de desarrollo del software, no están bajo la administración de la configuración del software, pero de igual forma se identifican y definen, para asegurar que el proyecto prosiga de una manera disciplinada.

Uno de los hitos importantes de la administración de la configuración del software, es establecer un repositorio de líneas base de software¹, que almacena todas las líneas base de software definidas para el proyecto.

Sus metas son [4]:

- Planificar las actividades de administración de la configuración.
- Identificar, controlar y mantener disponibles productos de trabajo seleccionados.
- Controlar los cambios a los productos de trabajo seleccionados.
- Los grupos e individuos afectados son informados del estado y contenido de las líneas base de software.

Nivel 3

Las ACP de nivel 3 hacen referencia a aspectos organizacionales y de proyectos, de manera tal que la organización posea la infraestructura necesaria para institucionalizar en todos sus proyectos los procesos de ingeniería y de administración que el modelo propone a este nivel.

¹ Líneas base de software: una especificación o producto que se ha revisado y acordado formalmente, y que después de esto sirve como la base para el desarrollo posterior, y que sólo puede modificarse a través de los procedimientos de control de cambios formales [Gue00].

La descripción de cada una de las ACP de nivel 3 se detalla a continuación:

4.5.7 Administración Integrada de Software

Su propósito es integrar las actividades de ingeniería de software y de administración en un proceso definido y coherente, que es adaptado del proceso de software estándar de la organización y elementos de proceso relacionados, los que están descritos en la *Definición del Proceso de la Organización* [4]. Esto involucra lo establecido en las áreas claves del proceso de Nivel 2: *Planificación del Proyecto de Software* y *Seguimiento y Control del Proyecto de Software*.

Las metas de esta área clave de proceso son [4]:

- El proceso de software definido del proyecto es una versión adaptada del proceso de software estándar de la organización.
- El proyecto es planificado y administrado de acuerdo con el proceso de software definido del proyecto.

4.5.8 Coordinación Intergrupala

Su propósito es establecer los medios para que el grupo de ingeniería de software participe activamente con los otros grupos de ingeniería, de modo que el proyecto mejore su habilidad de satisfacer las necesidades del cliente [4].

Las metas de esta área clave de proceso son [4]:

- Acordar los requerimientos del cliente por todos los grupos afectados.
- Los compromisos entre los grupos de ingeniería son acordados por los grupos afectados.
- Los grupos de ingeniería identifican, siguen y resuelven los problemas entre grupos.

4.5.9 Enfoque en el Proceso de la Organización

Su propósito es establecer la responsabilidad organizacional por las actividades del proceso de software que mejoren la capacidad organizacional del proceso [4]. Esto implica mantener y desarrollar el entendimiento de los procesos (cómo se realiza el proceso de software) a nivel organizacional y a nivel de los proyectos, así como también coordinar las actividades para evaluar, desarrollar, mantener y mejorar estos procesos. La organización establece compromisos de largo plazo y provee de los recursos necesarios para coordinar el desarrollo y mantención de los procesos de software a lo largo de los proyectos actuales y futuros, para tal efecto existe un grupo de ingeniería de software, que se encarga de realizar estas actividades.

Las metas de esta área clave de proceso son [4]:

- Coordinar las actividades de desarrollo y mejoría del proceso de software a través de toda la organización.
- Identificar las fortalezas y debilidades de los procesos de software utilizados en relación con un proceso estándar.
- Planificar las actividades de desarrollo y mejoría del proceso de la organización.

4.5.10 Definición del Proceso de la Organización

Su propósito es desarrollar y mantener un conjunto de activos del proceso de software que mejoran el rendimiento del proceso a través de los proyectos y que provean las bases para beneficios acumulativos y perdurables para la organización. Esto implica desarrollar y mantener el proceso de software estándar de la organización, así como sus elementos, que incluyen por ejemplo, la descripción del modelo de ciclo de vida del software.

Las metas de esta área clave de proceso son [4]:

- Desarrollar y mantener un proceso de software estándar para la organización.
- Recolectar, revisar y poner a disposición la información relacionada con el uso del proceso de software estándar de la organización por parte de los proyectos de software.

4.5.11 Programa de Capacitación

Su propósito es desarrollar las habilidades y conocimientos necesarios en las personas, para que ellos puedan cumplir sus roles en forma eficaz y efectiva [4].

La capacitación es una responsabilidad organizacional, pero los proyectos de software deben identificar las habilidades necesitadas y proporcionar la capacitación necesaria cuando las necesidades del proyecto son únicas.

Las metas de esta área clave de proceso son [4]:

- Planificar las actividades de capacitación.
- Proveer la capacitación necesaria para desarrollar las habilidades y conocimientos requeridos, para ejecutar los roles técnicos y de administración del software.
- Los individuos del grupo de ingeniería de software y otros grupos relacionados con el software reciben la capacitación necesaria para ejecutar sus roles.

4.5.12 Ingeniería de Productos de Software

Su propósito es ejecutar consistentemente un proceso de ingeniería bien definido que integre todas las actividades de ingeniería de software, para producir efectiva y eficientemente productos de software correctos y consistentes. La *Ingeniería de Productos de Software* describe las actividades técnicas del proyecto, como el análisis de requerimientos, el diseño, la codificación y el testing del software.

Las metas de esta área clave de proceso son [4]:

- Definir, integrar y ejecutar consistentemente las actividades de ingeniería de software para producir el software.
- Mantener la consistencia entre los productos de trabajo de software.

5.2.13 Revisiones de Pares

En una revisión normal, un producto de trabajo de software o un conjunto de productos de trabajo se presentan a los gerentes, clientes, usuarios finales u otros individuos interesados para que comenten o aprueben dichos productos. Las revisiones típicamente ocurren al final de una tarea.

En una revisión de pares, un producto de trabajo de software o un conjunto de productos de trabajo se presentan a los “colegas” (personal vinculado con el análisis y desarrollo del producto), para que identifiquen defectos. Esto produce que los defectos de los productos de trabajo sean detectados más tempranamente, permitiendo una mayor productividad y productos de más alta calidad.

Un efecto colateral importante de esta práctica, es desarrollar una mejor comprensión de los productos de trabajo de software y los defectos que pueden ser prevenidos.

Las metas de esta área clave de proceso son [4]:

- Planificar las actividades de revisión de pares.
- Identificar y remover los defectos de los productos de trabajo de software.

4.6 Nivel 4

Las áreas clave de proceso del nivel 4, apuntan a establecer una comprensión cuantitativa del proceso y de los productos de software que se construyen. Las descripciones de cada una de las áreas claves de proceso de nivel 4 se detallan a continuación:

4.6.1 Administración Cuantitativa del Proceso

Su propósito es controlar cuantitativamente el desempeño del proceso en el proyecto de software. El desempeño del proceso de software representa los resultados reales logrados al seguir un proceso de software. El énfasis está en identificar las causas especiales de variación dentro del proceso y corregir las circunstancias que dieron lugar a la variación, para lograr las metas definidas.

Las metas de esta área clave de proceso son [4]:

- Planificar las actividades de administración cuantitativa.
- Controlar cuantitativamente el rendimiento del proceso de software de los proyectos.
- Conocer la madurez del proceso de software de la organización, cuantitativamente.

4.6.2 Administración de la Calidad del Software

Su propósito es desarrollar una comprensión cuantitativa de la calidad de los productos de software del proyecto y lograr metas de calidad específicas.

Esto implica establecer metas de calidad cuantitativas para los productos de software basadas en las necesidades de la organización, del cliente y de los usuarios finales. Para que estas metas puedan ser logradas, la organización establece estrategias y planes, y el proyecto ajusta específicamente su proceso de software definido para lograr las metas de calidad definidas.

Las prácticas de *Administración de la Calidad del Software* están construidas sobre las prácticas de las áreas claves de proceso *Administración Integrada de Software* e *Ingeniería de Productos de Software* que establecen e implementan el proceso de software definido del proyecto y el área clave de proceso *Administración Cuantitativa del Proceso* que establece una comprensión cuantitativa de la habilidad de los procesos de software definidos de los proyectos para lograr los resultados deseados.

Las metas de esta área clave de proceso son [4]:

- Planificar las actividades de administración de calidad del software.
- Definir metas medibles y con prioridades para la calidad de los productos de software.
- Cuantificar y administrar el avance real para lograr las metas de calidad para los productos de software.

4.7 Nivel 5

Las áreas clave del proceso del nivel 5, cubren los problemas que la organización y los proyectos deben dirigir, para implementar la mejora continua y medible del proceso de software. Las descripciones de cada una de las áreas claves de proceso de nivel 5 se detallan a continuación:

4.7.1 Administración del Cambio de Procesos

Su propósito es mejorar los procesos de software usados en la organización, con el intento de mejorar la calidad del software, aumentar la productividad, y acortar el tiempo de desarrollo del producto. Esto implica definir metas para el mejoramiento del proceso, con el apoyo de la alta gerencia; implica identificar, evaluar e implementar mejoras al proceso de software estándar de la organización en forma continua, proactiva y sistemática. Se deben establecer programas de incentivos y entrenamiento para permitir y promover la participación de todos en las actividades de mejoramiento del proceso. Se realizan planes pilotos para validar los cambios propuestos al proceso, los que son revisados y adecuados apropiadamente al proceso de software estándar y a los proyectos.

Las metas de esta área clave de proceso son [4]:

- Planear la mejora continua del proceso de software.
- Toda la organización participa en la mejora del proceso de software.
- Mejorar continuamente el proceso de software tanto de la organización como del proyecto.

4.7.2 Prevención de Defectos

Su propósito es identificar las causas de los defectos y prevenir que se repitan. El proyecto de software analiza los defectos, identifica sus causas, y cambia su proceso definido de software, como se describe en la *Administración Integrada de Software*. Las lecciones aprendidas son tomadas por la *Administración del Cambio de Procesos*, para que éstas sean institucionalizadas a nivel organizacional.

Las metas de esta área clave de proceso son [4]:

- Planear las actividades de prevención de defectos.
- Buscar e identificar las causas comunes de los defectos.
- Priorizar y eliminar las causas comunes de los defectos.

4.7.3 Administración de Cambios Tecnológicos

Su propósito es identificar las nuevas tecnologías (es decir, herramientas, métodos, y procesos) que sean beneficiosas y transferirlas a la organización de una manera ordenada, como se describe en la *Administración del Cambio de Procesos*. La organización establece un grupo que trabaja en los proyectos de software para introducir y evaluar nuevas tecnologías y administrar los cambios de las tecnologías en uso.

Las metas de esta área clave de proceso son [4]:

- Planear las actividades de administración de cambios tecnológicos.
- Evaluar nuevas tecnologías para determinar su efecto en calidad y productividad.
- Incorporar nuevas tecnologías apropiadas a la organización como prácticas normales.

5. Estructura del SW-CMM

Cada nivel de madurez (a excepción del nivel 1) se compone de áreas claves de proceso (ACP). Una ACP está compuesta por prácticas claves, que describen la infraestructura o actividades que se deben realizar para satisfacer una determinada ACP. A su vez, las prácticas claves se encuentran agrupadas en *Características Comunes*, las cuales buscan la implementación o la institucionalización de las ACP.

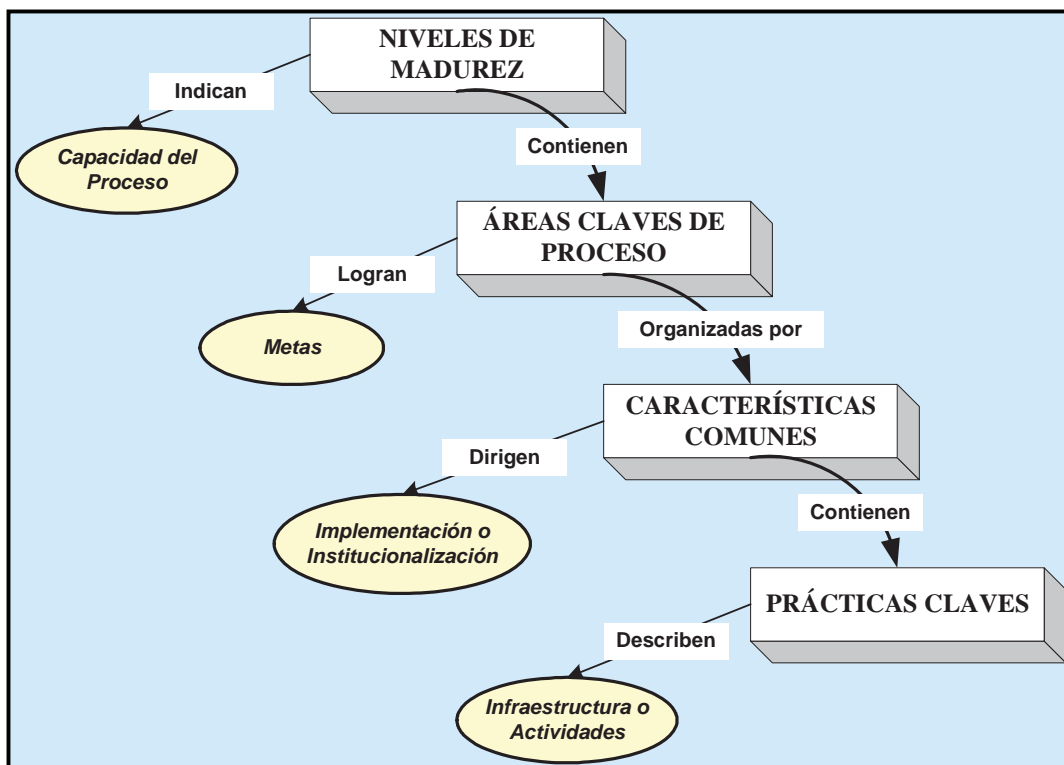


Figura 6.1: Estructura de SW-CMM.

La siguiente tabla resume en números el SW-CMM.

Niveles de Madurez	5
Áreas Claves de Proceso	18 (0 a 7 por nivel)
Metas	2 a 4 por ACP
Prácticas Claves	316, organizadas en 5 Características Comunes

Tabla 6.1: SW-CMM en Números

4.8 Características Comunes

Las características comunes son atributos que indican si una implementación o una institucionalización de un área clave de proceso es eficaz, repetible y duradera.

Las características comunes son las siguientes:

- 1. Compromisos para el desarrollo:** describe las acciones que la organización debe realizar para asegurar que el proceso sea establecido y pueda perdurar. Típicamente involucra establecer políticas organizacionales y el compromiso de la alta gerencia.
- 2. Habilidades para el desarrollo:** describe las precondiciones que deben existir en el proyecto u organización para implementar un área clave de proceso en forma competente. Típicamente involucra contar con los recursos y financiamiento necesarios, definir una estructura organizacional y realizar capacitaciones.
- 3. Actividades a realizar:** describe los roles y procedimientos necesarios para implementar una área clave de proceso. Involucra establecer planes y procedimientos, desarrollar trabajos, hacer un seguimiento de las actividades y realizar acciones correctivas si fuera necesario.
- 4. Medición y análisis:** describe la necesidad de medir y analizar el estado de las actividades de un área clave de proceso. Típicamente incluye ejemplos de medidas que se pueden tomar para determinar el estado y efectividad de las actividades desarrolladas.
- 5. Verificación de la implementación:** describe los pasos para asegurar que las actividades a realizar estén de acuerdo al proceso establecido. Típicamente involucra efectuar revisiones y auditorías por parte de la administración y del grupo de aseguramiento de la calidad del software (SQA).

Las prácticas claves de las *Actividades a realizar* describen qué se debe implementar para establecer la capacidad del proceso. Las otras prácticas forman las bases a través de las cuales una organización puede institucionalizar las prácticas descritas en las *Actividades a realizar*. Las actividades a realizar por los proyectos o la organización proveen las listas más largas de prácticas claves, porque describen la implementación actual de un área clave de proceso. Las prácticas claves que están en las otras características comunes son igualmente importantes, sin embargo ellas describen qué es lo que se necesita hacer para apoyar e institucionalizar un área clave de proceso.

4.9 Prácticas Claves

Cada área clave del proceso se describe en términos de prácticas claves, que contribuyen a satisfacer sus metas. Las prácticas claves describen la infraestructura y actividades que contribuyen a la implementación eficaz y la institucionalización de las áreas claves de proceso.

El propósito de las prácticas claves es proveer una descripción de los principios esenciales de un proceso de software eficaz. Estos principios deben ser interpretados por cada organización de acuerdo a su cultura y experiencia de sus gerentes y personal técnico. El modelo describe las prácticas que se deben implementar, pero no cómo implementarlas, aunque de igual manera entrega referencias de cómo llevar a cabo este proceso.

4.10 Metas

Las metas resumen las prácticas claves de un área clave de proceso y pueden ser usadas para determinar si una organización o proyecto ha implementado eficazmente una área clave de proceso.

Las metas abarcan el alcance, los límites y el propósito de cada área clave de proceso. Al adaptar las prácticas claves de un área clave de proceso a un proyecto específico, las metas se pueden usar para determinar si la adaptación es una interpretación “razonable” de las prácticas claves. Similarmente cuando se evalúan formas alternativas de implementar un área clave de proceso, las metas se pueden usar para determinar si las prácticas alternativas satisfacen los propósitos del área clave de proceso.

6. Métodos de Evaluación

4.11 Introducción

Antes de comenzar a describir los métodos de evaluación utilizados por el SW-CMM, se dará una breve introducción del contexto en el cual se encuentra inmerso el proceso de evaluación.

Para comenzar, nos haremos la pregunta: ¿por qué o para qué evaluar?, diremos que se evalúa para medir el estado actual de los procesos de desarrollo de software de la organización, de manera tal, de conocer sus fortalezas, debilidades y riesgos. El proceso de evaluación se enmarca en el contexto de un plan de mejora, que permita mejorar nuestros procesos con el fin de lograr objetivos como el aumento de la calidad y la productividad del software.

El proceso de evaluación es una parte integral del plan de mejoramiento de los procesos, que nos permite conocer el estado actual de los procesos con el fin de medir y localizar los problemas que permitan asignar los recursos adecuados a aquellas áreas de mejoras más urgentes, donde la inversión será más efectiva.

La evaluación de los procesos de desarrollo puede tener los siguientes objetivos, dependiendo de las necesidades de la empresa o los proyectos participantes:

- Identificar los puntos fuertes y las debilidades para iniciar un programa de mejoramiento.
- Constatar el progreso alcanzado por las iniciativas de mejoramiento de procesos en curso (habitualmente iniciadas como producto de una evaluación anterior).
- Demostrar a una tercera parte (clientes, socios potenciales, organización) la madurez de procesos de la organización (como medio para aumentar las posibilidades de nuevos negocios).
- Verificar la implementación de los procesos de desarrollo de software con respecto a un marco de trabajo.

La evaluación es el resultado final de un proceso que comienza cuando la gerencia comprende la incidencia que tiene el proceso de desarrollo en la producción de productos de mejor calidad. Este entendimiento motiva a la gerencia para decidir el establecimiento de una iniciativa de mejoramiento de procesos. Esto implica el compromiso para asignar los recursos adecuados y mantener los objetivos durante el tiempo necesario para completar la iniciativa.

Si la gerencia no está convencida de la importancia de un proceso maduro, existe un elevado riesgo que la iniciativa no perdure en el largo plazo.

4.12 Modelo IDEAL

El proyecto de mejoramiento de los procesos de la organización, debe estar sustentado por un modelo que describa las fases, actividades y recursos necesarios para llevar a cabo un mejoramiento de los procesos de software exitoso. El modelo sugerido por el SEI para estos fines es llamado IDEAL.

IDEAL es un proceso cíclico de mejoramiento de los procesos, el cual proporciona un conjunto de actividades coherentes para sustentar la adopción de las prácticas recomendadas por el SW-CMM. Se podría decir que IDEAL es un modelo de ciclo de vida para el mejoramiento de los procesos de software basado en el SW-CMM.

IDEAL se compone de cinco fases, éstas son:

- **I** (Initiating). Iniciar: Establecer la infraestructura para un esfuerzo exitoso de mejoramiento.
- **D** (Diagnosing). Diagnosticar: Determinar donde se encuentra, relativo a donde se desea llegar.
- **E** (Establishing). Planificar: Planificar las actividades necesarias para alcanzar el estado deseado.
- **A** (Acting). Ejecutar: Implementar el plan de trabajo definido.
- **L** (Learning). Aprender: Aprender de la experiencia y mejorar las habilidades para adoptar nuevas tecnologías en el futuro.

Para una mayor descripción del modelo IDEAL, ver Anexo 1

4.13 Métodos de Evaluación

Existen dos categorías distintas de evaluaciones: las valoraciones y las evaluaciones del proceso de software.

Las valoraciones del proceso de software son usadas para determinar el estado del proceso de software, identificando sus fortalezas y debilidades, que permitan determinar las áreas prioritarias de mejora, para obtener el soporte organizacional necesario, para emprender el proyecto de mejora de los procesos de software.

Las evaluaciones del proceso de software son usadas para identificar qué contratistas están más calificados para realizar un proyecto de software o para monitorear el estado actual de las actividades del subcontratista en la organización.

La diferencia entre las valoraciones y evaluaciones del proceso de software es principalmente por el uso de sus resultados. En las valoraciones, los resultados son la base para el plan de mejora de los procesos. En las evaluaciones, es la guía para desarrollar un perfil de riesgos. En la selección de contratistas, éste perfil de riesgos, permite contar con un criterio de selección más fundamentado y sólido. En la supervisión del subcontrato, el perfil de riesgos puede ser usado para motivar al contratista a entregar un mejor servicio.

4.13.1 Método CBA IPI

El CBA IPI [5] fue desarrollado por el SEI en 1994 y es una herramienta de diagnóstico que permite a las organizaciones poder determinar las fortalezas y debilidades de sus procesos de desarrollo de software.

Las metas a nivel general, de este método son:

- Dar soporte, permitir y alentar el compromiso de la organización para la mejora del proceso.
- Entregar un cuadro exacto de las debilidades y fortalezas del proceso, usando a SW-CMM como referencia e identificando las ACP a mejorar.

El enfoque de CBA IPI es unir y entrenar un grupo de evaluación bajo el mando de un líder y conducir una serie de actividades con la gente clave de la organización para entender sus problemas, preocupaciones e ideas de mejoramiento.

El CBA IPI se basa en los siguientes principios:

- Se usa el SW-CMM como modelo de referencia.
- Involucrar a la alta gerencia para que sea el auspiciador del proyecto.
- La evaluación se basa en las metas y necesidades del negocio.
- Acercar la evaluación a una colaboración entre el grupo de evaluación y el personal de la organización.

Consta de tres etapas:

- Planear y preparar la evaluación.
- Conducir la evaluación.
- Reportar los resultados.

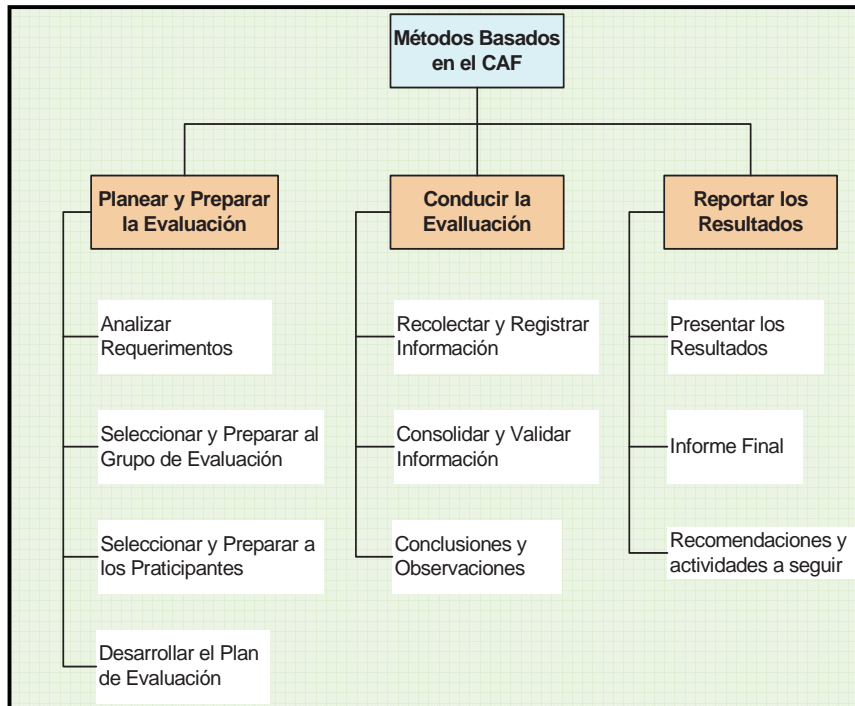


Figura 7.1: Actividades de evaluación de los métodos basados en el CAF

1. Planear y preparar la evaluación.

Esta etapa incluye lo siguiente:

Identificar el enfoque de la evaluación: El propósito de esta etapa es desarrollar un entendimiento común entre el líder de la evaluación y el auspiciador del proyecto, en las metas a lograr, enfoque, límites, roles, responsabilidades, resultados y obtener los compromisos necesarios para el cumplimiento de estos objetivos.

Desarrollar el plan de evaluación: El plan se desarrolla basándose en las metas de evaluación definidas por el auspiciador. El equipo estudia la organización a ser evaluada, se definen los proyectos y las personas a ser entrevistadas y también se definen los documentos a ser revisados.

Preparar y capacitar al grupo de evaluación: Los miembros del grupo de evaluación son capacitados en SW-CMM y en CBA IPI.

Preparar a los participantes: El propósito de esta actividad es asegurarse que los participantes entiendan el proceso de evaluación y tengan claras sus expectativas del proceso.

Realizar un cuestionario: A través del cuestionario de madurez del SEI [6] se recolecta información sobre la capacidad del proceso de software de la organización. El cuestionario le permite al grupo de evaluación tener una primera aproximación acerca del estado de las actividades desarrolladas.

Preparar las actividades en terreno: Hay ciertas actividades que se deben cumplir para realizar la evaluación en terreno. Estas son:

- Examinar las respuestas del cuestionario.
- Revisión inicial de documentos: Un conjunto inicial de documentos sobre el proceso de desarrollo de software de la organización son revisados por el grupo de evaluación para encontrar áreas adicionales de investigación, entender los ciclos de vida usados por la organización y “mapear” los datos recopilados al modelo SW-CMM.
- Realizar las preguntas de las entrevistas a ser efectuadas.

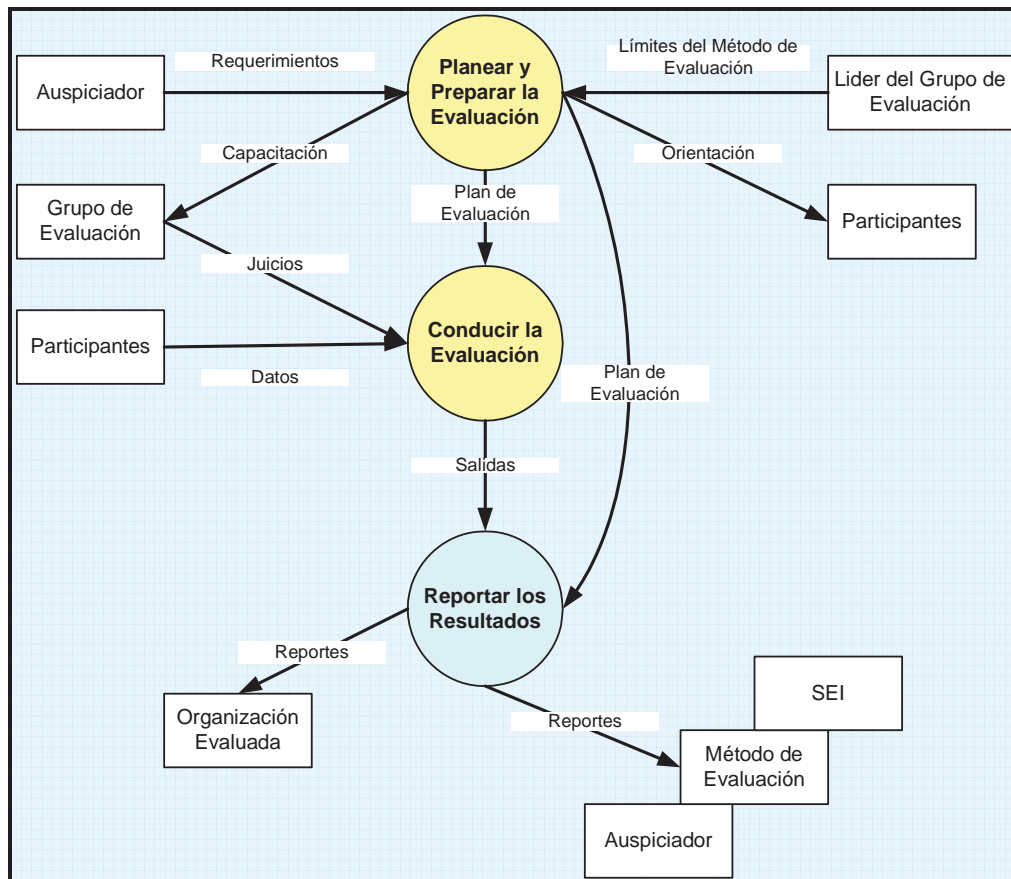


Figura 7.2: Fases de implementación de una evaluación

2. Conducir la evaluación.

A continuación se describen las actividades que se realizan en terreno, mientras dura la evaluación:

Entrevistas.

El propósito de las entrevistas es:

- Identificar las áreas que la gente cree que se pueden mejorar en la organización.
- Entender cómo el trabajo es realizado.
- Entender los procesos que se usan.
- Entender las relaciones entre los procesos y los proyectos.
- Asegurarse que la evaluación está de acuerdo al enfoque definido.

Las personas a entrevistar son:

- Jefes de Proyecto: Son entrevistados individualmente. Los temas a cubrir son administración de proyectos y procesos que son usados.
- Supervisores: Son entrevistados en grupos. Sirve para entender cómo se realiza el trabajo e identificar las áreas que presentan problemas o cuales áreas se creen que presentan problemas y las mejoras que creen que se deben hacer.
- Áreas representativas: Sirve para obtener la visión de esta área de la organización sobre las mejoras que creen que se deben implementar para mejorar el proceso actualmente implementado. Las áreas representativas incluyen personal vinculado con las actividades de ingeniería de software, requerimientos, diseño, programación, prueba e integración.

Integración de la información.

El propósito de esta actividad es resumir y consolidar la información recopilada y categorizarla según las ACPs.

Debe haber consenso en el grupo de evaluación en la validación de los datos y que existe suficiente información sobre las áreas que están siendo investigadas. Es responsabilidad del grupo de evaluación el obtener la suficiente información para abarcar toda la organización, los ciclos de desarrollo de software utilizados y las ACP definidas dentro del enfoque de la evaluación.

Preparar la presentación de las conclusiones.

Las conclusiones se obtienen de las entrevistas realizadas. Las evaluaciones finales no son consideradas hasta después de la presentación de las conclusiones.

Presentar las conclusiones.

El propósito de esta actividad es obtener el feedback de los participantes acerca de las conclusiones realizadas.

Consolidar, evaluar y preparar las conclusiones finales.

El grupo de evaluación integra la información adicional obtenida durante las presentaciones, continuaciones de entrevistas y revisión de documentos. Cuando el grupo logra una completa cobertura de SW-CMM, la organización y el ciclo de vida del software, se puede comenzar a evaluar el cumplimiento de cada meta de las ACP involucradas en la evaluación.

Por cada meta, el grupo revisa todas las debilidades y se pregunta: ¿es esta debilidad lo suficientemente influyente como para tener un impacto negativo en la meta? si es así la meta no es cumplida.

Una ACP puede ser satisfecha, insatisfecha, no aplicable o no evaluada. Para cumplir una ACP todas sus metas deben ser satisfechas. Para satisfacer un nivel de madurez todas sus ACPs y la de los niveles inferiores deben ser satisfechas.

Las conclusiones generales son elaboradas por el grupo de evaluación y son presentadas al auspiciador: las fortalezas y debilidades observadas por cada ACP, las evaluaciones de cada ACP y el nivel de madurez si se desea.

3. Reportar los resultados.

Presentar las conclusiones finales.

Los resultados son presentados al auspiciador del proyecto. Durante la presentación, el presentador se asegura que se entendieron los descubrimientos y las ACPs que se evaluaron.

Actividades a seguir.

Para aprovechar la evaluación, la organización debe implementar una infraestructura de mejora. Esta incluye: auspicio de la gerencia para el plan de mejoramiento, establecer un grupo de procesos de ingeniería de software y capacitar a la organización en SW-CMM.

Como actividad final el grupo de evaluación documenta los resultados finales de la evaluación y realiza las recomendaciones necesarias para ocupar de buena manera los resultados obtenidos.

4.13.2 Método SCE

El Software Capability Evaluation (SCE) [8] es una evaluación que se realiza para determinar los riesgos asociados al efectuar un subcontrato de software. También es un método usado para seleccionar subcontratistas, que es desarrollado por un grupo externo a la organización, liderado por un evaluador autorizado del SEI [15]. El grupo de evaluación revisa los requerimientos propuestos por quién desea realizar la evaluación, y determina los posibles riesgos del subcontrato en base al SW-CMM.

SCE solo indica los riesgos asociados al contratar una organización, basándose en una propuesta definida. SCE no es un método adecuado para determinar la capacidad de proceso de la organización. El estudio del proceso de la organización solo está enfocado en

determinar si la organización a contratar tiene la suficiente habilidad para realizar un trabajo específico. La determinación del nivel de madurez para una propuesta específica es solo útil para definir los posibles riesgos involucrados.

SCE no es adecuado para determinar la capacidad del proceso, porque está limitado a los requerimientos específicos de la propuesta. Está enfocado en determinar qué está malo, no en qué mejoras pueden ser hechas.

Los resultados del SCE son consistentes con el CBA IPI [5] en áreas de investigación similares, ya que ambos métodos están basados en el CMM Appraisal Framework (CAF).

A continuación se describe el método usado por el SCE para seleccionar un subcontratista, las otras formas de ocupar SCE no se incluyen en esta investigación.

Etapas del SCE.

Al igual que CBA IPI, las etapas del SCE son:

- Planificar y preparar la evaluación.
- Conducir la evaluación.
- Reportar los resultados.

A continuación se presenta el cuadro resumen de las actividades del SCE.

Fases	Actividades y Salidas
Planificar y preparar la evaluación	<p>La organización a evaluar:</p> <ul style="list-style-type: none"> • Determina los atributos de los productos a obtener. • Determina la capacidad de proceso más apropiada según las metas del negocio (capacidad de proceso objetivo). • Selecciona y entrena al grupo de evaluación. <p>Salida: Los objetivos y los requerimientos de la evaluación son definidos.</p> <p>El grupo de evaluación:</p> <ul style="list-style-type: none"> • Identifica las áreas de desarrollo software donde se carece de experiencia (indicando los riesgos potenciales). • Define el enfoque de la evaluación. <p>Salida: el enfoque de la evaluación es definido y los preparativos de alto nivel para la evaluación son completados.</p> <p>El grupo de evaluación:</p> <ul style="list-style-type: none"> • Selecciona los proyectos a evaluar. • Prepara temas específicos a evaluar. <p>Analiza los instrumentos de recolección de datos.</p>
Conducir la evaluación	<p>El grupo de evaluación:</p> <ul style="list-style-type: none"> • Investiga cada tópico planeado en terreno. • Conduce las actividades de recolección de datos a través de entrevistas, revisión de documentos y presentaciones. • Consolida la información recolectada y valida las observaciones. • Determina las debilidades, fortalezas y las actividades de mejora. <p>Salida: El proceso de recolección de información está consolidado y los descubrimientos encontrados son determinados.</p>
Reportar los resultados	<p>El grupo de evaluación:</p> <ul style="list-style-type: none"> • Presenta y entrega los resultados. • Produce el informe final para el auspiciador. • Hace las recomendaciones del uso de los resultados. <p>Salida: los resultados de la evaluación son documentados.</p>

Tabla 7.1: Actividades del SCE

1. Planear y Preparar la evaluación.

Durante esta fase, la planificación considera:

- Capacitar al personal.
- Coordinar la visita del grupo SCE y responder por la información de la organización.
- Programar las actividades de la evaluación en el contexto en que se usa el método, en este caso selección de subcontratistas.

Como resultado de esta planificación, la organización compromete los recursos necesarios para conducir la evaluación. Además, se debe determinar el rol del SCE y planificar cómo se va a usar.

Una vez que la decisión de usar SCE fue hecha, la organización determina la capacidad de proceso necesaria para cumplir sus objetivos de negocio. Esto se consigue analizando los atributos que se desearía que tuviera el producto y entonces determinar la capacidad de proceso más apropiada para construir dicho producto.

La organización también selecciona al grupo SCE. Este grupo está conformado por personal con experiencia que ha completado el curso de evaluador SCE. Estos pueden ser de la empresa o externos a ella.

El grupo SCE define el enfoque de la investigación para todas las organizaciones a ser evaluadas. El enfoque de SCE consiste en cuales componentes de SW-CMM serán investigados dentro de la capacidad objetivo de proceso.

Los atributos del producto a ser construido son comparados con los atributos de los productos desarrollados por los contratistas. En esta comparación se identifican las áreas en las cuales las organizaciones carecen de experiencia, indicando posibles riesgos involucrados.

Las potenciales áreas de riesgo identificadas en cada organización evaluada son consolidadas a través de todas las organizaciones evaluadas. Basado en los puntos débiles de las organizaciones evaluadas, el grupo de evaluación determina los componentes a evaluar dentro de cada área clave de proceso, en el marco de la capacidad objetivo de proceso. Estas áreas serán investigadas en todas las organizaciones de desarrollo a ser evaluadas.

Una vez que el equipo ha identificado qué evaluar, se prepara para la visita en terreno. El grupo SCE selecciona los proyectos a evaluar y las personas a entrevistar. Para esto, el equipo selecciona áreas temáticas dentro del enfoque de la evaluación. Los temas se refieren a prácticas de trabajo observables y se usan para investigar el proceso de implementación que corresponde a las áreas de proceso identificadas para la investigación. Un tópico define un área de contenido específico que será investigado durante la evaluación. Por ejemplo, un tema puede ser “investigar si la organización tiene procedimientos estándar para el proceso de control de cambio de la administración de configuración del software”.

Luego de seleccionar los temas de evaluación, el equipo crea una estrategia para conducir las entrevistas y revisar los documentos. El equipo luego trabaja de cerca con el coordinador técnico de la organización a evaluar, para programar entrevistas y pedir información para revisar.

Cuando la etapa de Planificación y Preparación de la evaluación se termina, el grupo SCE estará listo para realizar la visita a terreno. El equipo habrá determinado qué temas se investigarán (y a qué nivel), con quiénes necesitan hablar, qué preguntas iniciales deben

realizar durante la entrevista y qué documentos revisarán primero. La organización a ser evaluada habrá preparado las instalaciones necesarias para el equipo SCE, habrá pedido la documentación, y se habrá asegurado que los entrevistados estén disponibles.

2. Conduciendo la fase de evaluación.

El propósito de esta fase es investigar los tópicos asociados con cada área clave de proceso a evaluar, identificando las fortalezas, debilidades y las actividades de mejora de estas áreas. Aunque el propósito es simple, esta es una de las actividades más complicadas de la evaluación.

Para tener éxito en esta fase el equipo de evaluación, necesita tener una buena relación con el personal de la organización a evaluar. Para estos fines, el equipo de evaluación debe mantener un alto nivel de profesionalismo (por ejemplo: cumplir la programación establecida), para así obtener credibilidad por parte de la organización que le permita aumentar el nivel de cooperación de ésta.

Esta fase comienza con la recolección de información. La recolección de información tiene dos componentes básicos: la investigación de los tópicos seleccionados y tomar decisión de qué información recolectar. Estos componentes se aplican iterativamente hasta que todos los tópicos a evaluar sean entendidos.

El grupo de evaluación utiliza mecanismos complementarios para la investigación de los tópicos a evaluar, como lo son la: revisión de documentos, entrevistas y presentaciones. Instrumentos como el cuestionario de madurez [6], son usados como fuente de información durante la fase de preparación y planificación de la evaluación.

La revisión de documentos provee una evidencia objetiva del proceso que está siendo utilizado. Los procesos interesantes para la evaluación, son aquellos procesos que entregan como resultado un producto, sea este intermedio o final.

Las entrevistas entregan profundidad de cómo el proceso es implementado en la práctica y muestran el conocimiento que el personal tiene del proceso. Hay dos tipos de evaluaciones realizadas en el SCE: entrevistas exploratorias y de consolidación. Durante las entrevistas exploratorias las preguntas y respuestas revelan el actual proceso implementado y guían al grupo de evaluación en el soporte de documentación. Las entrevistas de consolidación se enfocan en validar los descubrimientos realizados.

Las presentaciones son una fuente más de información, estas sirven para obtener retroalimentación del personal de la organización acerca de los hallazgos encontrados por el grupo de evaluación.

Usar múltiples mecanismos de recolección de datos, ayuda a mejorar la calidad de la información. Por ejemplo, los cuestionarios son usados para recoger la información inicial, de una manera sencilla y económica. La información obtenida es usada para guiar las entrevistas las que pueden ser rápidamente reenfocadas según el área de interés. La revisión de documentos es usada para complementar las entrevistas, esto sirve para eliminar errores

de interpretación propios de las entrevistas. Más adelante, las presentaciones iniciales del análisis de la información recolectada [21], dan la posibilidad a la organización de corregir en errores incurridos. En resumen, la información recolectada a través de múltiples medios es mucho mejor que la información recolectada por un solo mecanismo.

Los miembros del grupo registran la información recolectada. Estas observaciones son consolidadas a través de reuniones, en las cuales se discute si se dispone de la suficiente información como para emitir un juicio acerca de los tópicos evaluados. Si se determina que no existe la suficiente información, se planifican nuevas reuniones y una nueva ronda de entrevistas y revisión de documentos.

Una vez que se tiene la suficiente información sobre los tópicos evaluados, se identifican las fortalezas, debilidades y actividades de mejora asociadas a cada ACP evaluada.

Al término de esta fase, el equipo de evaluación se encuentra en condiciones de producir el informe final.

3. Reporte de resultados finales.

La fase de reporte completa el SCE. Durante esta fase, el equipo de SCE documenta los resultados de la investigación y los entrega a su auspiciador y a la organización evaluada. Seguido a esto, el equipo (o individuos del equipo) apoya al auspiciador en seguir las actividades que incorporen el uso de los descubrimientos.

Debido a la importancia de los descubrimientos en la mejora del proceso, se debe hacer un esfuerzo para entregar retroalimentación de una manera completa y oportuna. El resumen de los descubrimientos, incluirá todos los descubrimientos determinados por el grupo de evaluación y describirá qué uso se le pueden dar a estos resultados.

Un reporte final es completado por los miembros del grupo de evaluación después de la visita a terreno y se convierte en la base para todas las actividades de evaluación que se realizarán en el futuro. El reporte final puede incluir recomendaciones si el auspiciador lo desea. Por lo general, las recomendaciones se hacen seguido de las actividades y se documentan por separado. También se generan reportes para el SEI [16] y para quien desarrolla el método para mejoramientos futuros del proceso de evaluación.

Los miembros del equipo SCE y ciertamente el líder del equipo pueden esperar participar en las actividades posteriores de mejora. El conocimiento que han ganado durante la evaluación es valioso para el auspiciador, al decidir qué acciones tomar basado en los resultados del SCE. En la selección de subcontratistas, el equipo puede apoyar al auspiciador en la identificación de riesgos relacionados a los descubrimientos específicos realizados. En un proceso de monitoreo de subcontratos, éste apoyo puede incluir asistir al auspiciador en producir conjuntamente un plan de mejoramiento con la organización distribuidora o en realizar un plan de incentivos. En una evaluación interna, el apoyo puede estar en hacer recomendaciones participando en equipos de acción u otras formas de planificación de mejoramiento e implementación.

7. Evaluación de una Empresa de Desarrollo de Software

La evaluación realizada se basó en el método CBA IPI.

4.14 Aspectos Generales.

A continuación se describen ciertos aspectos generales definidos antes de realizar la evaluación:

Metas y objetivos: El objetivo de la evaluación es analizar el estado actual del proceso de software utilizado por la organización, respecto del SW-CMM, con miras a lograr una futura certificación. La evaluación abarca los cinco niveles de madurez, excluyendo la ACP: “*Administración de Subcontratos de Software*”, ya que la organización no cuenta con subcontratación de proyectos de software a terceros.

Plan de evaluación: Para la evaluación se definieron dos proyectos a ser evaluados. Ambos proyectos utilizan la misma plataforma de desarrollo, la cual es soportada íntegramente por la herramienta CASE desarrollada por la organización, la cual se explicará más adelante.

Cuestionario de madurez: El cuestionario utilizado en la evaluación fue el cuestionario de madurez del SEI [6]. Este cuestionario contiene 125 preguntas, las cuales cubren las 18 ACP del SW-CMM para el Software (6 a 8 preguntas por cada ACP). Las preguntas del cuestionario son del tipo Si / No y se dividen en claves y generales. Las preguntas claves se relacionan con las metas a ser logradas en cada ACP, mientras que las preguntas generales se relacionan con las características comunes de cada ACP (compromisos, habilidades, actividades).

Participantes: Las personas encuestadas fueron los equipos de trabajo de ambos proyectos, contabilizando un total de 13 personas. Entre estas personas se cuentan: el gerente general, el gerente de software, el gerente de investigación y desarrollo, jefes de proyecto y desarrolladores.

Las actividades a las que se dedican las personas encuestadas, se resumen en el siguiente cuadro:

Actividad	Número de Personas
Análisis de requerimientos	7
Diseño	10
Codificación	9
Pruebas e integración	10
Aseguramiento de la Calidad del software	3
Administración de software	2
Mejoramiento del proceso de software	6

Tabla 8.1: Cuadro resumen de las actividades en las que participan los encuestados.

La experiencia de las personas encuestadas en la organización, va desde los 9 y 10 años (gerentes), hasta 1 año (desarrolladores). La media de experiencia del Proyecto 1 es de 5 años y del Proyecto 2 es de 1 año, con la excepción de su jefe de proyecto (10 años de experiencia).

4.15 Etapas de la evaluación.

4.15.1 Planear y preparar la evaluación.

Esta etapa incluye las siguientes actividades:

- *Definir el enfoque de la evaluación:* Aquí se definen los objetivos y metas esperadas de la evaluación, el enfoque de la evaluación, sus límites, responsabilidades y resultados esperados, además de obtener los compromisos necesarios para el cumplimiento de estos objetivos.
- *Desarrollar el plan de evaluación:* El plan de la evaluación, se desarrolla basándose en las metas definidas anteriormente. En esta etapa: se estudia la organización a ser evaluada, se definen los proyectos y las personas a ser entrevistadas. También se definen los documentos que deben ser analizados.
- *Realizar la encuesta de madurez:* El cuestionario de madurez es entregado a las personas a ser encuestadas [14].

4.15.2 Conducir la evaluación.

En esta etapa se analizaron las respuestas obtenidas del cuestionario de madurez. Paralelamente a esto, se comenzó la revisión inicial de documentos, para conocer la organización evaluada y tener un primer acercamiento al proceso productivo de software de la organización.

Luego, se realizó entrevistas a la mayoría de las personas encuestadas, principalmente los gerentes y el personal con mayor experiencia de la organización, ya que estas personas mostraron en sus respuestas un mayor grado de conocimiento de las prácticas administrativas y de ingeniería realizadas en la organización.

4.16 Evaluación de los Niveles de Madurez.

A continuación se describe la evaluación realizada. Como primera parte se describirán las herramientas utilizadas por la organización para el desarrollo y administración de sus proyectos.

4.16.1 Herramientas utilizadas por la organización.

Antes de describir el análisis de la evaluación, se describirán las herramientas utilizadas por la organización, para el desarrollo y administración de sus proyectos. Se realiza esta explicación para entender el análisis posteriormente expuesto.

Herramienta Case: Z3.

La herramienta Z3 provee un ambiente de modelamiento gráfico, basado en las especificaciones UML, para la automatización de la generación de código y documentación. Z3 apoya todas las etapas del proceso productivo de software; desde el análisis de requerimientos hasta la construcción y documentación de aplicaciones. Z3 está totalmente basado en las especificaciones UML, permite modelar en forma gráfica: diagramas de actividades (workflow), modelos navegacionales (web), modelos de clases, entre otros; todo integrado en un ambiente común de desarrollo.

Z3 se encarga de administrar la persistencia, el despliegue y la concurrencia de los componentes que proveen los diferentes modeladores y generadores, los que se suministran como bibliotecas acoplables a la herramienta. Al separar los modeladores de los generadores de código, es posible utilizar los mismos modelos para producir sistemas bajo diferentes plataformas y lenguajes de programación, así Z3 permite la generación de código para: Delphi, Java/J2EE, Visual Basic, otros; y bases de datos como: MS-SQL Server, Oracle, Access.

Los proyectos evaluados utilizan Z3 como repositorio y guía para su proceso de desarrollo. Cabe mencionar que los proyectos evaluados son desarrollados bajo la plataforma J2EE²(Java 2 Enterprise Edition).

² J2EE es un conjunto de especificaciones de APIs Java para la construcción de aplicaciones empresariales. Para mas información visitar <http://java.sun.com/j2ee>

Sitio web de documentación.

Los proyectos evaluados disponen de un sitio web, en el cual se almacena toda la documentación relevante asociada al proyecto. Los documentos se encuentran a disposición de todos los integrantes del proyecto: tanto del equipo de desarrollo, como del grupo de trabajo del cliente. Cuando se efectúa la publicación o modificación de cualquier documento en el sitio, todos los usuarios son informados vía e-mail. Es responsabilidad de cada individuo leer los contenidos de los documentos publicados. A cada documento publicado se le puede adjuntar observaciones o comentarios. También el sitio permite llevar un historial de versiones para un mismo documento.

Entre los documentos almacenados en el sitio se cuentan:

- Acuerdos, compromisos.
- Minutas de reunión.
- Minutas técnicas (presentan la descripción y solución de una problemática técnica).
- Documentos de Planificación.
- Pruebas (planillas de registro de errores), entre otros.

CVS (Concurrent Versión System).

CVS³ es una herramienta que automatiza el control de versiones de los códigos fuente de los distintos productos de un proyecto.

A continuación se detalla por nivel de madurez, el cumplimiento de las áreas clave de proceso.

4.16.2 Cumplimiento Nivel 2

Administración de Requerimientos.

Para la administración de requerimientos, se manifiesta el cumplimiento de las metas, aunque falta formalización en el proceso de análisis de requerimientos. Respecto al establecimiento de políticas que respalden esta práctica, la organización no ha definido políticas escritas, aunque los proyectos siguen una metodología de trabajo definida y utilizan a Z3 como repositorio y guía de su proceso de desarrollo.

Respecto a la capacitación de las personas encargadas de administrar los requerimientos, no se entrena a éstas, su capacitación está dada por su formación profesional y su experiencia en proyectos de desarrollo. Esta capacitación también está fuertemente ligada al conocimiento de la tecnología de desarrollo utilizada por el proyecto, la cual en algunos casos es deficiente.

³ Más información se puede obtener en: <http://www.cvshome.org>

Respecto a las mediciones, solo se lleva un control del esfuerzo incurrido en realizar nuevos requerimientos y re-trabajo, ya que estos esfuerzos están asociados al cobro de mantenciones.

Las actividades de administración de requerimientos, no están sujetas a la revisión de un área de aseguramiento de calidad. El control de estas actividades es realizado por el jefe de proyecto y el gerente, en forma periódica.

Planificación del Proyecto de Software.

Las metas de esta área clave de proceso son cumplidas parcialmente.

El modelo define que las estimaciones de tamaño, tiempo y costo, deben ser documentadas, esto se cumple parcialmente, ya que no existen estimaciones de costos. Las estimaciones de tamaño de los productos se realizan eventualmente. Lo que sí se documenta son las estimaciones de esfuerzo, en todo momento del proyecto.

Por otro lado, el plan de desarrollo de software especifica las actividades a ser realizadas y los compromisos establecidos, aunque se observa inconsistencia en la manera de llevar este registro (en un proyecto se lleva una carta gantt, que se actualiza semanalmente; en el otro proyecto el registro de actividades se mantiene en las minutas de reunión).

Todos los individuos y grupos afectados conocen los compromisos relacionados al proyecto y están de acuerdo con éstos. Las personas asumen y son informadas de sus compromisos por medio de reuniones. También los compromisos son documentados y almacenados en el sitio web del proyecto, quedando a disposición de todos los afectados.

No existe una política organizacional escrita para la planificación de los proyectos de software.

Las actividades de planificación de proyecto están sujetas a revisión por parte del jefe de proyecto en forma periódica.

Seguimiento y Control del Proyecto de Software.

Esta área clave de proceso presenta un alto porcentaje de respuestas afirmativas (71,4%). Las metas son cumplidas casi en forma completa.

Los resultados reales del proyecto son comparados con los estimados, en forma periódica solo en términos de esfuerzos. Cuando los resultados reales se desvían de la planificación establecida, se toman acciones correctivas: asignando más personal al proyecto o discutiendo los problemas en las reuniones de control de proyecto.

Los cambios en los compromisos son acordados por los individuos afectados, sean éstos del equipo de trabajo de la organización o del cliente.

La organización no cuenta con una política escrita para el seguimiento y control de los proyectos.

Se afirma por parte de los encuestados que existe un responsable por los productos de trabajo y por las actividades del proyecto. Esta persona es el jefe de proyectos.

No existen medidas asociadas a determinar el estado de las actividades de esta práctica.

Las actividades de seguimiento y control son revisadas periódicamente por la alta gerencia.

Aseguramiento de la Calidad del Software.

La organización no cuenta con un grupo de aseguramiento de calidad del software, ni tampoco con un plan de aseguramiento de la calidad. Las actividades de aseguramiento de la calidad que se realizan son: el testing de productos y revisiones de documentos, principalmente cuando éstos deben ser entregados al cliente.

Las actividades de testing son planificadas y las realiza una persona independiente al equipo de desarrollo. Eventualmente esta actividad no es ejecutada eficazmente (cuando existen actividades de mayor prioridad en los proyectos). Los resultados de las revisiones son informados a las personas correspondientes y son seguidos hasta su finalización.

La organización no cuenta con procedimientos, estándares o políticas que definan la forma en que se deberían realizar las actividades de ingeniería y administración de proyectos. En contraste a esto, la organización cuenta con su propia herramienta CASE (Z3), diseñada y desarrollada íntegramente por ellos, en la cual se encuentra definido (en parte) la metodología de trabajo que deben seguir los analistas y desarrolladores de los proyectos. Otro factor importante de destacar, es la generación automática de código que permite la herramienta, el cual permite reducir los esfuerzos de programación y aumentar la calidad de los productos resultantes.

Los aspectos de no cumplimiento que no están resueltos en los proyectos, son tratados por la alta gerencia, aunque sin el respaldo de una política SQA, ya que ésta no se ha definido a nivel organizacional.

Administración de la Configuración del Software.

Esta área clave de proceso presenta un 12,5% de respuestas afirmativas, lo cual implica que la organización no ha implementado la administración de la configuración en sus proyectos.

La organización no dispone ni de una política SCM, ni de planes SCM para los proyectos. Actualmente lo que se realiza, es el control de versiones de algunos ítems de configuración (no formalmente establecidos). Los ítems de configuración sobre los cuales existe un control de versiones son: documentación asociada a proyecto (acuerdos, compromisos, validaciones, etc.) y los códigos fuente.

El control de versiones del código fuente está automatizado por la herramienta CVS (Concurrent Version System). Los distintos modelos de datos realizados en Z3 (modelos de clases, diagramas navegacionales, diagramas de actividades y otros) son respaldados semanalmente, al igual que las bases de datos. No existe un control de versiones para estos ítems de configuración.

4.16.3 Cumplimiento Nivel 3.

Enfoque en el Proceso de la Organización.

En la organización existe una constante preocupación por el mejoramiento del proceso de desarrollo de software.

Las actividades para desarrollar y mejorar los procesos de software, se encuentran coordinadas a través de la organización, por medio del departamento de Investigación y Desarrollo. Este departamento cumple la función de materializar las “ideas” de mejoramiento propuestas por medio de la construcción de herramientas. Es política organizacional, la concepción, diseño y construcción de sus propias herramientas de desarrollo de software, el ejemplo más concreto de esto, es la construcción de su propia herramienta Z3CASE.

La gerencia participa y auspicia las actividades de mejoramiento de los procesos, dispone de recursos y diseña planes de largo plazo para apoyar tales actividades.

El proceso de software implementado en los proyectos se evalúa periódicamente por las gerencias de la organización. Estas evaluaciones no son formales ni documentadas, se realizan por medio de reuniones.

La organización no sigue un plan escrito para desarrollar e implementar su proceso de software.

No se cuenta con medidas que indiquen el estado de las actividades de desarrollo y mejoramiento.

Las actividades de desarrollo y mejoramiento del proceso de software de la organización son revisadas periódicamente por la gerencia general.

Definición del Proceso de la Organización.

La organización no ha desarrollado, ni mantenido un proceso de software estándar a nivel organizacional. Tampoco cuenta con los elementos relacionados al proceso, como son: las descripciones de los ciclos de vida a utilizar, las guías y criterios de adaptación del proceso, la base de datos del proceso organizacional y la librería de documentación asociada al proceso de software.

A pesar de lo anterior, Z3CASE define la metodología de desarrollo diseñada por la organización para el desarrollo de sus proyectos.

Programa de Capacitación.

Las capacitaciones en la organización no son planificadas y son insuficientes. Muchos de los encuestados señalan que existiendo una adecuada capacitación en ciertas tecnologías, se lograría un mejor desempeño y se evitarían muchos problemas.

No existen encargados de impartir capacitaciones. Las capacitaciones en la organización se realizan mediante un “auto estudio guiado”, en el cual los gerentes asignan la resolución de problemas al personal.

No existe una política escrita organizacional, para satisfacer las necesidades de capacitación. Sin embargo son identificadas las necesidades de capacitación para los proyectos, esto es parte de las responsabilidades de la gerencia y del equipo de elaboración de propuestas.

Se han elaborado materiales (documentos Word y presentaciones Power Point) de capacitación, pero éstos son insuficientes.

No se han elaborado procedimientos documentados de eximición para las capacitaciones requeridas. Las eximiciones de las capacitaciones son tarea de los gerentes, ellos determinan cuándo las personas tienen el grado de conocimiento y/o habilidad necesaria para desempeñar sus tareas.

No se mantienen registros de las capacitaciones realizadas.

La evaluación del programa de capacitación no es formal. La evaluación se realiza a través de conversaciones entre los participantes y la dirección. Se preguntan opiniones o sugerencias y se intenta mejorar la próxima vez (estos datos no son registrados).

Administración Integrada de Software.

Esta área clave de proceso presenta 0% de respuestas afirmativas. Debido a que el propósito de la administración integrada de software es: “integrar las actividades de administración e ingeniería de software en un proceso coherente y definido que es adaptado del proceso de software estándar de la organización y elementos de proceso relacionado”.

Como se mencionó anteriormente, la organización no cuenta con un proceso organizacional estándar de software, por lo que no cumple con ninguna de las prácticas claves definidas por esta ACP, ya que todas las prácticas claves definidas en esta ACP tienen como requisito las prácticas claves descritas en la ACP: “Definición del proceso de la organización”.

Ingeniería de Productos de Software.

Z3CASE es la herramienta que se utiliza para ejecutar las tareas de ingeniería de software. Como se mencionó anteriormente, esta herramienta sirve como repositorio de los distintos productos de trabajo (diagramas y modelos de datos) generados en las actividades de ingeniería de software: análisis de requerimientos (casos de uso), diseño (diagrama de actividades, modelo de clases, modelo navegacional, entre otros), construcción (códigos fuente y base de datos), pruebas e integración.

La organización utiliza un proceso desarrollo iterativo- incremental, en el cual se desarrollan módulos de software, los que posteriormente son integrados.

Otro aspecto a destacar en ésta área, es que se reutiliza diseño y código, los que se representan en componentes del negocio (según la tecnología de desarrollo definida por la organización). Los componentes genéricos existentes, son cargados en Z3CASE en forma de bibliotecas. Más que el análisis, el diseño es la etapa considerada más importante dentro del proceso de desarrollo de software en la organización.

El gerente de Investigación y Desarrollo también cumple el cargo de “desarrollador corporativo”, el cual tiene la función de “analizar” los distintos proyectos en ejecución para diseñar o desarrollar, productos o herramientas “genéricas”, para ahorrar esfuerzo y ganar en modularidad y reusabilidad.

En lo que respecta al testing, se realiza: testing unitario, de integración y alfa y beta. Las actividades de testing son planificadas.

Eventualmente son diseñados casos de pruebas a ser realizados. El testing también se realiza en la etapa de diseño, Z3 tiene métodos para rutar los diagramas de actividades. En la etapa de construcción, se prueba la funcionalidad general del sistema.

Los errores o defectos encontrados en revisiones internas y externas son documentados y almacenados en el sitio web del proyecto.

No existen mediciones asociadas al testing (número de errores encontrados, solucionados, esfuerzo empleado, etc.).

El testing es realizado por una persona independiente al equipo de desarrollo, encargada además de la documentación del proyecto.

Coordinación Intergrupala.

El propósito de la coordinación intergrupala es: *“establecer los mecanismos para que el grupo de ingeniería de software participe activamente con los otros grupos de ingeniería de modo que el proyecto mejore su habilidad de satisfacer las necesidades de los clientes efectiva y eficientemente”*.

En la organización no existen grupos de ingeniería interdisciplinarios, solo existe el grupo de ingeniería del proyecto, por lo que las prácticas definidas en esta área clave de proceso deben ser adaptadas.

La coordinación entre los miembros del grupo de ingeniería es óptima, principalmente porque se efectúan reuniones de control en forma periódica, en las cuales cada miembro aprueba sus compromisos establecidos en el plan del proyecto.

La definición de requerimientos con el cliente (como ya se mencionó en secciones anteriores) no se realiza bajo un proceso formalmente establecido.

Revisión de Pares.

Revisión de pares no es práctica formalmente establecida. Se efectúan revisiones de pares cuando la jefatura del proyecto lo estima conveniente, éstas no son planificadas, ni tampoco existe una política organizacional que respalde esta actividad.

No existe capacitación en los objetivos, principios y métodos de revisión de pares. Revisión de pares en la organización no es práctica establecida, es eventualmente realizada.

No existe un procedimiento documentado a nivel organizacional que defina cómo debe ser ejecutada esta práctica.

Cuando se realizan revisiones de pares, los defectos identificados son removidos, su eliminación se convierte en parte de las actividades del proyecto. Los defectos encontrados no son documentados y su seguimiento tampoco.

Según los encuestados, se realizan revisiones de pares: al código de los componentes reusables. En los proyectos, al diseño.

4.16.4 Cumplimiento Nivel 4.

Este nivel es el que presenta el más bajo nivel de respuestas afirmativas 0%, para ambas áreas claves de procesos, esto se debe a que éstas áreas claves de proceso no están implementadas en la organización.

Administración Cuantitativa del Proceso.

A nivel organizacional no existe una administración cuantitativa de los proyectos. Son muy pocas las métricas definidas en los proyectos [17], éstas están principalmente ligadas al control de cambios de requerimientos.

Para los proyectos no existe: un plan de administración cuantitativa, ni una política, ni procedimientos que indiquen cómo se deben planificar, ejecutar y evaluar las actividades de la administración cuantitativa del proceso.

Administración de la Calidad del Software.

A nivel organizacional no existe una administración de la calidad del software. La única actividad de la administración de la calidad del software que se planifica es el testing de productos entregables.

Para los proyectos no existe: un plan de administración de la calidad del software, ni una política, ni procedimientos que indiquen cómo se debe planificar, ejecutar y evaluar las actividades de la administración de la calidad del software.

4.16.5 Cumplimiento Nivel 5.

Prevención de Defectos.

La prevención de defectos es una práctica que no está implantada formalmente en la organización.

En las reuniones de control de proyecto se realiza análisis de causas comunes de defectos (no formal, ni documentado). Las causas comunes de defectos, generalmente son tratadas y eliminadas a través de una redefinición del proceso o a través de del desarrollo de una nueva herramienta.

Administración de Cambios en Tecnología.

En la organización se tiene una constante preocupación por el mejoramiento continuo de los procesos. La incorporación y evaluación de nuevas tecnologías al proceso, es tarea del gerente de investigación y desarrollo.

No realiza una evaluación de productividad y calidad para nuevas tecnologías incorporadas al proceso, esta evaluación la realiza el directorio de la empresa por medio de reuniones. Por política organizacional se tiene definido que cualquier incorporación de nueva tecnología al proceso, debe ser apoyada por una herramienta que soporte los métodos y actividades definidas por esa tecnología.

La gerencia general participa y auspicia (compromete recursos) para las actividades de administración de cambios tecnológicos.

No existe un procesamiento de datos para la selección de nuevas tecnologías.

Administración de Cambios de Proceso.

El mejoramiento continuo es una constante preocupación de la organización. Constantemente se están evaluando los procesos de desarrollo de software. Lo que intenta realizar es automatizar la mayor cantidad de tareas posibles para obtener un proceso de mayor calidad y uniforme.

Las reuniones de directorio para evaluar los procesos implementados son eventuales, principalmente como respuesta a eventos.

En el anexo, se presenta un gráfico resumen del cumplimiento de los niveles de madurez, según la encuesta realizada.

Por lo estudiado y analizado, se clasifica a la organización en el nivel inicial, ya que no cumple con la totalidad de las áreas claves de nivel 2, presentando su mayor debilidad en la las áreas clave de proceso: administración de la configuración y el aseguramiento de la calidad del software.

8. Propuesta de Mejoramiento

Realizada la evaluación, en el siguiente capítulo se describe y detalla una propuesta de mejoramiento, destinada a cubrir todas las debilidades detectadas, con el fin de alcanzar SW-CMM Nivel 2.

La siguiente planificación tiene por objetivo definir un proceso de desarrollo de software de manera tal que especifique las actividades a ser desarrolladas, todo lo anterior complementado con la incorporación de las áreas claves de proceso de SW-CMM Nivel 2.

La siguiente descripción de actividades no representa un paradigma de desarrollo, sino la definición de las etapas a ser consideradas al momento de afrontar un proyecto de software.

Cabe mencionar que para la definición del proceso de desarrollo, se tomó a UML como lenguaje gráfico de modelado de procesos, por ser éste el lenguaje utilizado por la organización evaluada.

4.17 Definición del proceso de desarrollo de software

Según lo estudiado e investigado, se describen las siguientes fases como parte del proceso de desarrollo de software:

1. Especificación de requerimientos.
2. Análisis.
3. Diseño.
4. Construcción.
5. Pruebas e integración.
6. Instalación.

A continuación se describen y explican cada una de las siguientes fases indicando sus actividades y procesos involucrados. También serán descritos los documentos (salidas) involucrados en cada actividad.

Los métodos y procesos para ejecutar las actividades definidas en las siguientes etapas, no forman parte de los objetivos del presente estudio.

4.17.1 Especificación de requerimientos.

La especificación de requerimientos marca el inicio del proceso de desarrollo de software. El objetivo de esta etapa es elaborar un documento que describa apropiadamente todos los requerimientos del cliente.

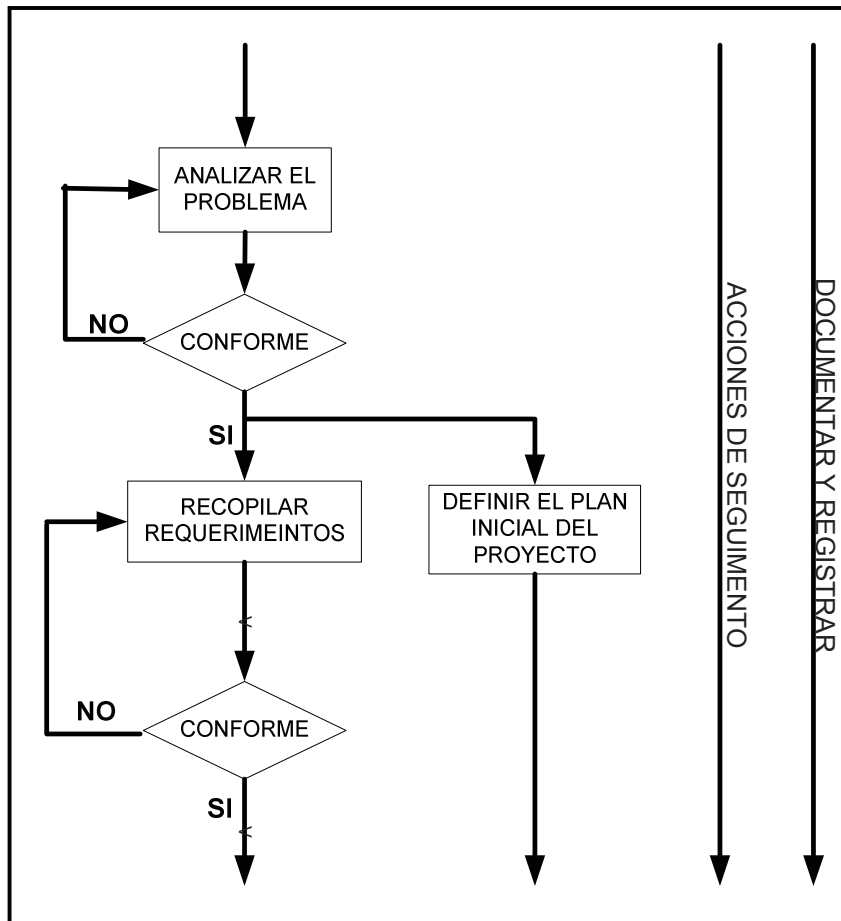


Figura 9.1: Diagrama de actividades de la especificación de requerimientos.

La figura 9.1 muestra las actividades a seguir al momento de llevar a cabo la especificación de requerimientos.

De esta etapa se obtienen los siguientes productos:

1. Resumen del problema
2. Resumen solución tentativa
3. Lista inicial de riesgos: identificación y evaluación de los posibles riesgos.
4. Definir el Plan inicial del proyecto.
5. Modelo inicial del negocio.

Luego del análisis del problema, se comienza a recopilar sistemáticamente los requerimientos del sistema. Esta etapa tiene como salida los siguientes productos:

1. Listado de requerimientos.
2. Modelo inicial de caso de uso.
3. Criterios de aceptación del producto.

Los requerimientos en esta etapa deben ser revisados y aprobados, antes de pasar a la siguiente fase. Para esto se debe definir un procedimiento mediante el cual:

- Se identifiquen los requerimientos faltantes o incompletos.
- Se determine si los requerimientos son:
 - Factibles y apropiados para ser implementados.
 - Clara y apropiadamente definidos.
 - Consistentes.
 - Validables.
 - Testeables.

En esta etapa también se debe desarrollar un documento de inicio de Anteproyecto, el cual debe ser revisado y aprobado. De forma general este documento cubre:

- Alcance del trabajo.
- Metas técnicas y objetivos.
- Identificación de clientes y usuarios finales.
- Estándares impuestos.
- Responsabilidades asignadas.
- Restricciones de costo y programación.
- Restricciones de recursos.
- Otras restricciones y metas para el desarrollo y/o mantenimiento.

4.17.2 Análisis

Una vez concluida la etapa de especificación de requerimientos se efectúa el análisis. Como primera actividad de esta etapa se deben asignar las responsabilidades por los requerimientos identificados. Luego se debe refinar sistemáticamente el listado inicial de requerimientos obtenidos.

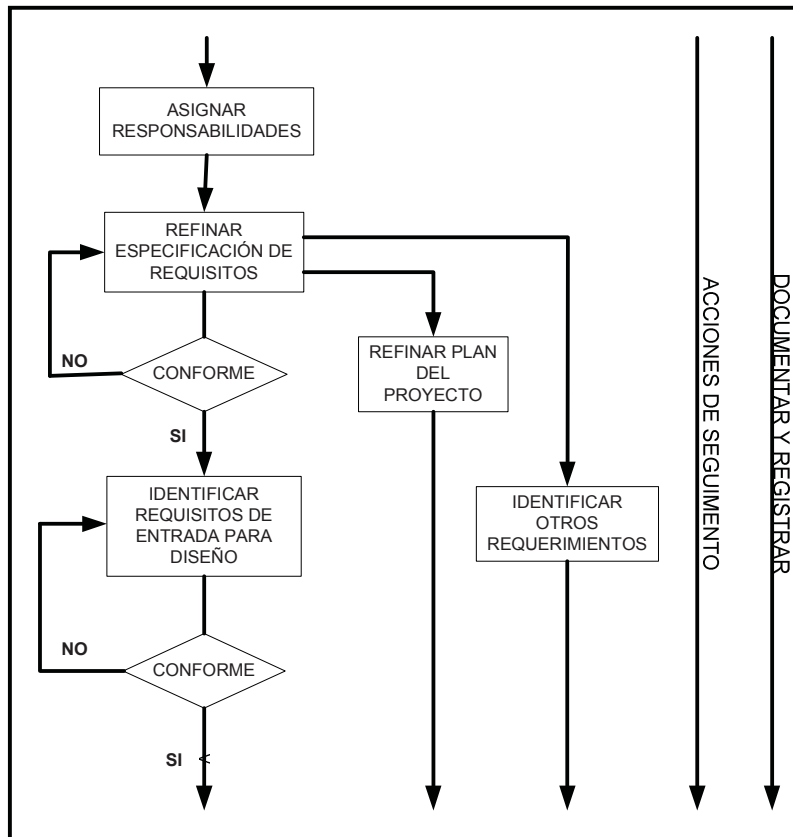


Figura 9.2: Diagrama de actividades del análisis.

La figura 9.2 muestra las actividades a seguir al momento de llevar a cabo la especificación de requerimientos.

De esta etapa se obtienen los siguientes productos:

- Planificación de la etapa de análisis.
- Prototipo del sistema.
- Modelo de casos de usos final.
- Modelo del negocio final.
- Arquitectura del sistema.
- Lista de riesgos refinada.

El refinamiento de los requerimientos del sistema incide en la planificación inicial realizada, producto de esto es que se tiene que modificar la planificación. Los cambios en los requerimientos o los nuevos requerimientos encontrados deben ser:

- Identificados.
- Evaluados (impacto en los compromisos, planes, actividades o productos).
- Evaluados en riesgo.
- Documentados.
- Realizar sus estimaciones de: tiempo, esfuerzo, recursos y tamaño de los productos si corresponde.
- Seguidos hasta su finalización.
- Asignados responsables.

El plan de desarrollo de software desarrollado en esta etapa, debe ser documentado, revisado y aprobado. El plan de desarrollo de software debe cubrir como mínimo los siguientes tópicos:

- El propósito, ámbito, metas y objetivos del proyecto de software.
- Ciclo de vida del proyecto.
- Identificación de los procedimientos, métodos y estándares de desarrollo y/o mantenimiento a utilizar.
- Productos de trabajo a ser desarrollados.
- Estimaciones de tamaño de los productos de trabajo.
- Estimaciones de esfuerzo y costo del proyecto.
- Uso estimado de recursos críticos.
- Cronograma del proyecto, incluyendo identificación de hitos y revisiones.
- Identificación y evaluación de riesgos.
- Planificación del uso de instalaciones de ingeniería y herramientas de apoyo.

Además, se deben definir procedimientos para realizar la administración de riesgos y las estimaciones de: tamaño de los productos, esfuerzo y costo, tiempos, recursos.

Cada riesgo debe ser identificado, evaluado, priorizado, documentado, identificadas sus contingencias y seguido.

El análisis debe especificar las entradas para el diseño.

4.17.3 Diseño.

Las entradas de la etapa de diseño (definidas en la fase anterior), deben ser revisadas y aprobadas. Luego, se debe realizar el modelo de clases, modelo de componentes y la interfaz de usuario.

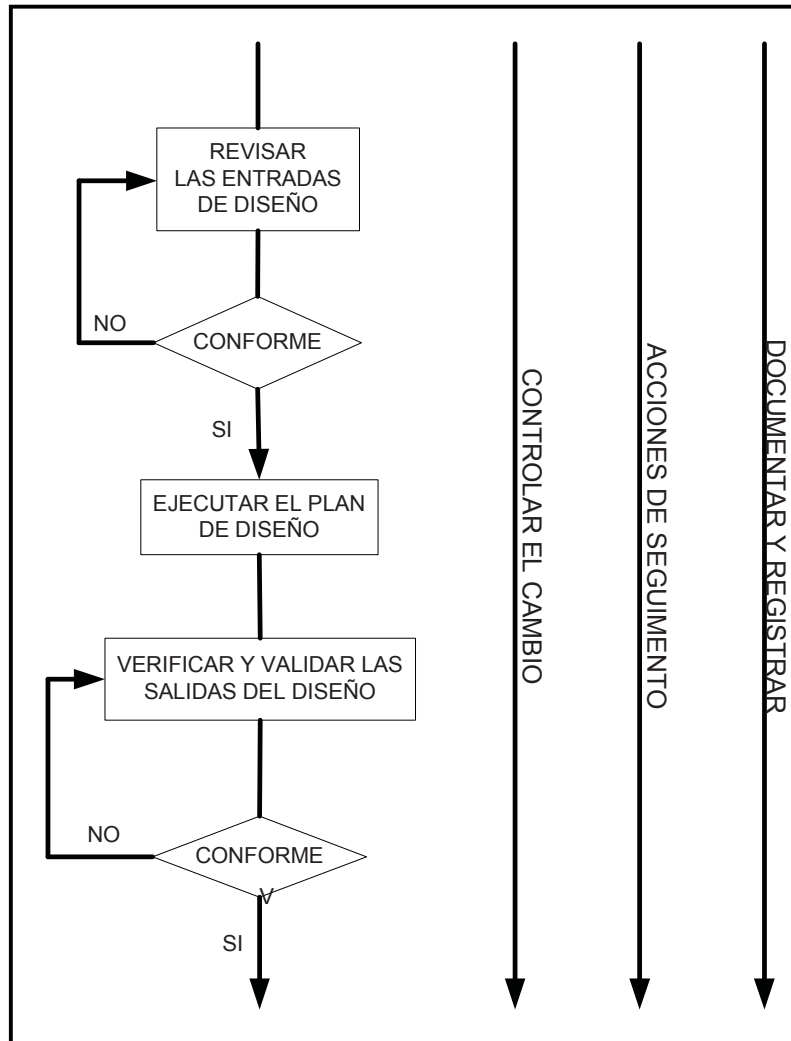


Figura 9.3: Diagrama de actividades del diseño.

4.17.4 Construcción.

En esta etapa se realiza la construcción misma del sistema. Es importante mencionar que en esta etapa se debe verificar y validar constantemente el producto, según las especificaciones definidas y también con los usuarios. Paralelo a la construcción del sistema se realiza el manual de usuario.

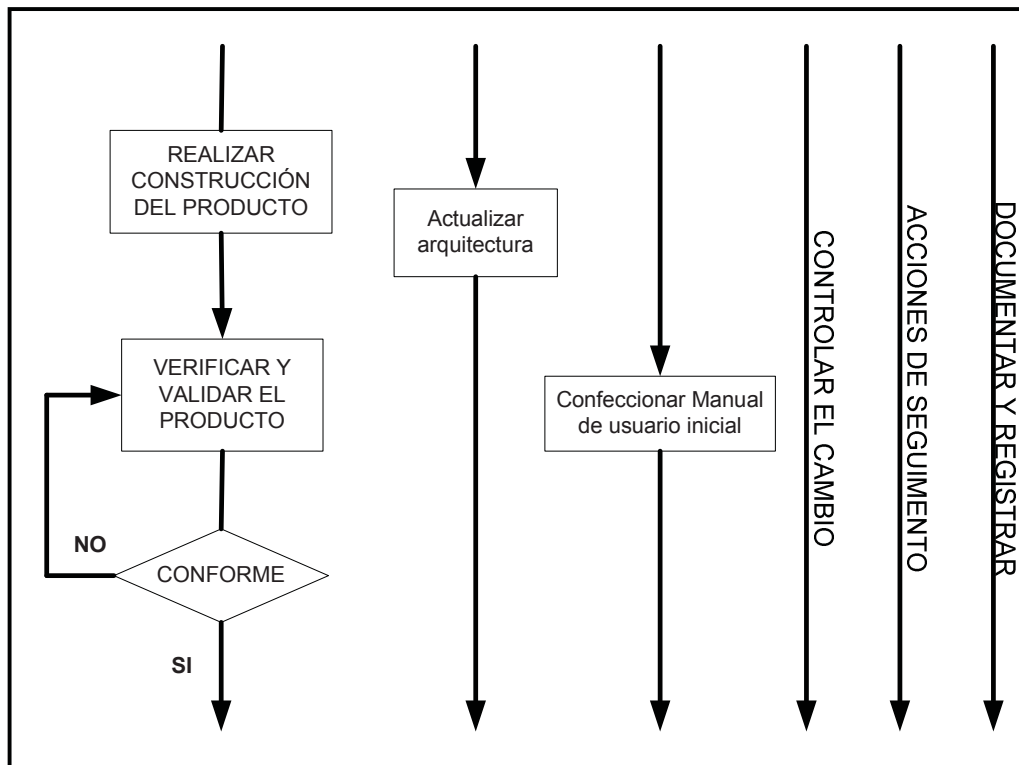


Figura 9.4: Diagrama de actividades del desarrollo.

4.17.5 Pruebas.

En esta etapa se deben desarrollar los planes de prueba, ejecutarlos y validar sus resultados.

Producto de la ejecución de las pruebas, se detectan los defectos y errores encontrados. Esta lista de defectos y errores debe ser: documentada, asignados a responsables y seguidos hasta su finalización.

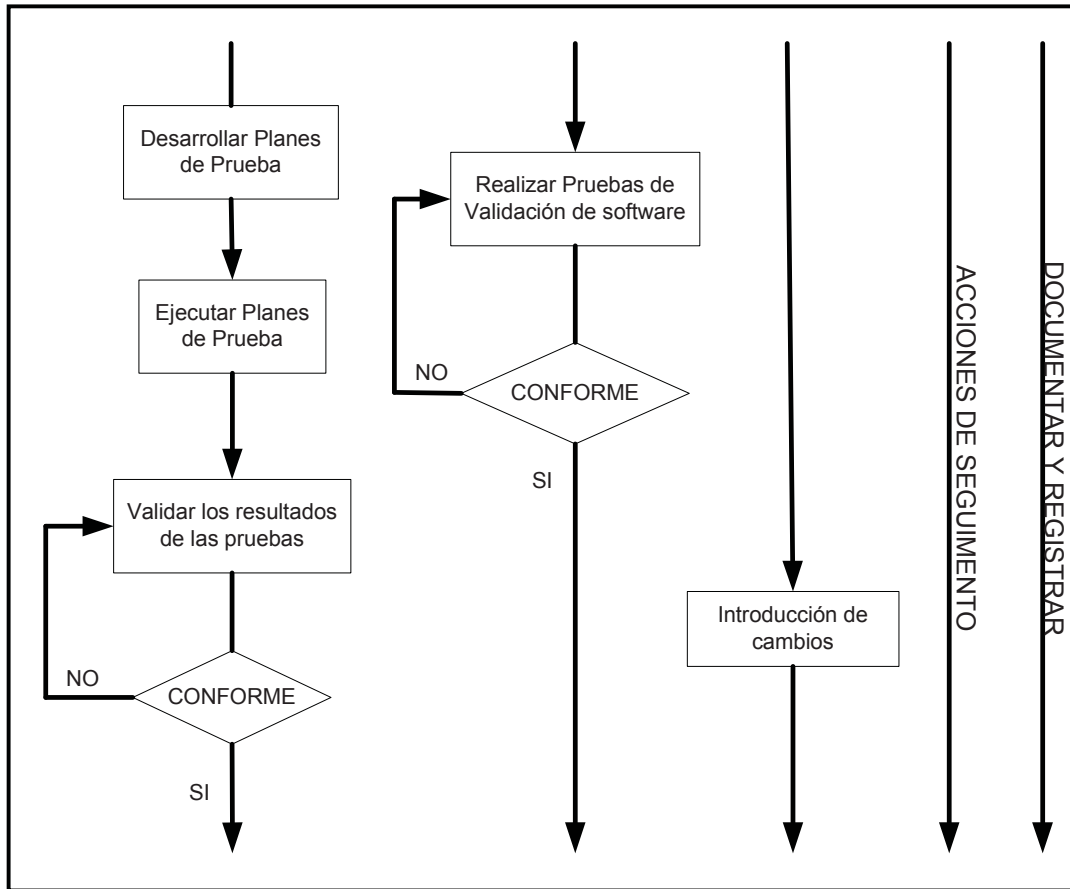


Figura 9.5: Diagrama de actividades de las pruebas.

Las etapas y actividades antes descritas deben ser seguidas y controladas. Seguidas, ya sea tanto por la gerencia como por el jefe de proyecto, además de otros responsables encargados de planes, actividades o productos.

Además de este seguimiento y control existen actividades como el aseguramiento de la calidad del software y la administración de la configuración del software, que dirigen y controlan las actividades del desarrollo del software, además de los productos que están siendo construidos.

A continuación se explica el rol de estas actividades y su descripción, en la propuesta de mejoramiento realizada.

4.18 Aseguramiento de la calidad del software.

Para implementar el aseguramiento de la calidad del software, se deben realizar las siguientes actividades:

- Definir los procedimientos y estándares que apoyan las distintas actividades definidas del proceso de desarrollo de software.
- Conformar un grupo SQA, responsable de coordinar e implementar las actividades de aseguramiento de la calidad del software en los proyectos. Este grupo debe ser independiente al equipo de desarrollo y debe contar con el apoyo de gerencia.
- Capacitar al grupo SQA en las actividades SQA.

Además, se debe preparar un plan SQA para el proyecto. Este plan debe cubrir:

- Responsabilidades y autoridad del grupo SQA.
- Recursos a ser utilizados por el grupo SQA.
- Programación de las actividades del grupo SQA del proyecto.
- La participación del grupo SQA en la definición del plan de desarrollo de software, estándares y procedimientos del proyecto.
- Evaluaciones a ser realizadas por el grupo SQA.
- Auditorías y revisiones a ser conducidas por el grupo SQA.
- Estándares y procedimientos a ser utilizados para conducir las auditorías y revisiones.
- Procedimientos para documentar y seguir hasta el cierre los problemas de no conformidad.
- Documentación a ser producida por el grupo SQA.
- Método y la frecuencia para proveer retroalimentación al grupo de ingeniería de software y a otros grupos, sobre las actividades SQA del proyecto.

Las actividades del grupo se realizan de acuerdo al plan revisado y aprobado. A continuación se nombran las actividades que deben ser realizadas por el grupo:

En la administración de requerimientos:

- Se audita la revisión de los requerimientos antes de que sean incorporados al proyecto.
- Se revisan los planes, productos, actividades, cuando los requerimientos cambian.

En la planificación del proyecto:

- Se revisan las actividades de estimación y planificación.
- Se revisan las actividades de revisión y toma de compromisos del proyecto.
- Se revisan las actividades de preparación del plan de desarrollo de software.
- Se revisan los estándares usados para preparar el plan de desarrollo de software.
- Se revisa el contenido del plan de desarrollo de software.

En el seguimiento y control del proyecto:

- Se revisan las actividades para revisión y modificación de compromisos.
- Se auditan los compromisos adquiridos o cambios en ellos.
- Se revisan las actividades de revisión del plan de desarrollo de software.
- Se revisa el contenido del plan de desarrollo de software revisado.
- Se revisan las actividades de seguimiento de los costos, programación, riesgos, restricciones técnicas, de diseño, funcionalidad y de rendimiento.
- Se revisan las actividades para conducir las revisiones planificadas técnicas y de la administración.

En la administración de la configuración:

- Se revisa y/o audita las actividades y productos de trabajo de la administración de la configuración y se informan los resultados.

4.19 Administración de la configuración del software.

Para implementar la administración de la configuración del software, se deben realizar las siguientes actividades:

- Se debe establecer un comité que tiene la autoridad para administrar las líneas base de software (Comité de control de la configuración del software - CCCS).
- Se debe establecer un grupo de administración de la configuración, responsable de coordinar e implementar las actividades de administración de la configuración en los proyectos.
- Se debe identificar los productos de software (ítems de configuración) a ser puestos bajo la administración de la configuración. Cada ítem de configuración debe tener:
 - Identificador único.
 - Descripción (características).
 - Indicar a las líneas base que pertenece.
 - Asignar su responsable.
- Se debe desarrollar un plan para la administración de la configuración del proyecto.
- Se debe establecer un sistema de librería de administración de la configuración.
- Se debe disponer de un repositorio de líneas base de software.
- Se debe proporcionar el almacenamiento y recuperación de los ítems de configuración.

- Permitir el almacenamiento, realización y recuperación de los registros de la administración de la configuración.
- Las solicitudes de cambio a las líneas base y los reportes de problemas deben ser: documentados, revisados, aprobados y seguidos, de acuerdo a un procedimiento documentado.
- Los cambios en las líneas base deben ser controlados.
- Se deben desarrollar informes estándar que documentan las actividades de administración de la configuración y los contenidos de las líneas base.

Además de las actividades descritas es necesario definir procedimientos que apoyen las actividades a ser implementadas. Todos estos deben estar debidamente documentados y se deben establecer mecanismos para su actualización.

Se deben desarrollar procedimientos para:

- Realizar el plan de desarrollo de software.
- Revisar el plan de desarrollo de software del proyecto.
- Revisar compromisos adquiridos (o cambios en ellos) con personas o grupos externos a la organización.
- Realizar revisiones formales para abordar logros y resultados del proyecto.
- Desarrollar el plan de aseguramiento de la calidad del software.
- Documentar y manejar las desviaciones identificadas en los productos y actividades revisados y/o auditados por el grupo SQA.
- Realizar el plan de administración de la configuración.
- Controlar los cambios en las líneas base.
- Liberación de productos desde la librería de líneas base.
- Registrar el estado de los ítems de configuración.
- Conducir auditorías a las líneas base de software.

4.20 Capacitaciones

El programa de capacitación está basado en la ACP: “*Programa de Capacitación*” de Nivel 3. A continuación se mencionan algunas de las capacitaciones que deben ser efectuadas:

- Capacitar al grupo SQA en las actividades de aseguramiento de la calidad.
- Capacitar al grupo de administración de la configuración en los objetivos, procedimientos y métodos para ejecutar las actividades de administración de la configuración.
- Capacitar al equipo de trabajo del proyecto en:

- SW-CMM.
- Procedimientos de estimación.
- Roles, responsabilidades, autoridad y valor del grupo SQA.
- Actividades de administración de la configuración.

Se deben definir métricas destinadas a informar el estado de las distintas actividades del proyecto.

Se deben definir y documentar las políticas que norman y definen las áreas claves de proceso de Nivel 2. Éstas son:

- Política de Administración de requerimientos.
- Política de Planificación de Proyectos de Software.
- Política de Seguimiento y Control de proyectos.
- Política de aseguramiento de la calidad del software.
- Política de Administración de la configuración del software.

Se deben definir actividades de consolidación:

- Definición de roles y responsables por las actividades del proceso.
- Herramientas y forma de comunicación entre los distintos roles definidos.

9. Conclusiones

Por medio del estudio y análisis del modelo SW-CMM y sus métodos de evaluación, se logró cumplir el objetivo principal de esta investigación, es decir, la realización del análisis y la propuesta de mejora del Proceso de Software de una empresa de desarrollo de Software. Según el trabajo realizado se pudo concluir que la organización se encuentra en el nivel 1 o inicial de SW-CMM, lo que no implica que ésta opere con un mal funcionamiento, sino que falta la implementación de ciertas áreas claves de proceso, como la Administración de Requerimientos, Aseguramiento de la Calidad y Administración de la Configuración del Software.

Dentro del área de Administración de Requerimientos se detectó la falta de formalización en el proceso de análisis de requerimientos, por tanto, esta falta fue incorporada dentro de la propuesta realizada para la mejora del proceso de software. Respecto a la capacitación de las personas encargadas de administrar los requerimientos, se observó que éstas no son entrenadas, su capacitación está dada por su formación profesional y su experiencia en proyectos de desarrollo, por lo que se propone un programa de capacitación basado en la ACP de Nivel 3.

En el área de Aseguramiento de la Calidad, se detectó que la organización no cuenta con un grupo de aseguramiento de la calidad del software, ni tampoco con un plan de aseguramiento de la calidad. Los aspectos de no cumplimiento que no están resueltos en los proyectos, son tratados por la alta gerencia, aunque sin el respaldo de una política SQA, ya que ésta no se ha definido a nivel organizacional. Por tanto, para implementar el aseguramiento de la calidad del software se propuso una serie de actividades, principalmente la preparación de un plan SQA.

Por último, en el área de Administración de la Configuración del Software, la organización no dispone de una política SCM, tampoco de planes SCM para los proyectos. Para la mejora de esta etapa se propone por ejemplo, el establecimiento de un comité que tendrá la autoridad para administrar las líneas base de software, y de un grupo de administración de la configuración.

De esta forma se fueron detectando las debilidades de cada una de las áreas y al mismo tiempo se detalló una propuesta para las falencias detectadas, para lograr una mejora del proceso de software. Todo esto gracias al cuestionario de madurez efectuado, que se encuentra organizado de forma que cubre las 18 ACP del SW-CMM. Por lo que es mucho más fácil detectar las debilidades de forma ordenada para así poder dar una propuesta de solución a ellas.

Es importante también mencionar, que si bien su objetivo es lograr alcanzar el Nivel 2 de SW-CMM, existen actividades a implementar que pueden ser apoyadas por áreas claves de proceso de Nivel 3, como lo son las actividades de capacitación (que pueden ser apoyadas por el área clave: Programa de Capacitación) y la definición del proceso de Software (que puede ser apoyada por el área clave: Definición del proceso de la organización).

Cabe destacar que la organización se encuentra en una constante evolución, incorporando nuevas tecnologías y mejorando continuamente sus procesos de desarrollo y sus actividades administrativas, con el objeto de lograr un proceso productivo de mayor calidad y automatizado. Si bien la planificación de las actividades se realizó según las directrices del SW-CMM, existen aspectos que deben ser abordados en trabajos futuros, como lo es la implementación del plan de mejoramiento. Para este fin la organización se puede apoyar en modelos tales como el modelo IDEAL.

Finalmente se puede concluir que para lograr una mejora en el Proceso del Software con el modelo SW-CMM, es fundamental la participación del personal de la empresa a ser evaluada. Para la evaluación de esta empresa participó personal que realizaba diferentes actividades, por ejemplo: análisis de requerimientos, diseño, codificación, pruebas e integración, aseguramiento de la calidad del software, administración del software y mejoramiento del proceso de software. Todas participaron en la realización del cuestionario de madurez. Pero no sólo basta con que lo realicen, una vez que se tengan los resultados, por medio de la propuesta realizada, el personal debe estar dispuesto a llevar a cabo la propuesta para lograr una mejora en el Proceso de Software, lo que puede derivar en cambios en su sistema de trabajo. Es por esto que es fundamental la participación del personal de la organización para que se lleven a cabo sus objetivos.

Referencias

- [1] Paulk, M., Curtis, B., Chrissis M., Weber, C.: "Capability Maturity Model for Software, Version 1.1", SEI-93-TR-024, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1993
- [2] Pressman, R.: "Ingeniería del Software, Un enfoque práctico", Quinta edición, 2002.
- [3] Henríquez, P.: "Diagnóstico del Desarrollo de Software en Organizaciones Financieras", Tesis de Licenciatura, Ingeniería Civil Informática, Universidad Técnica Federico Santa María, Valparaíso, 1996.
- [4] Paulk, M., Weber, C., Garcia, S., Chrissis, M.B, Bush, M.: "Key Practices of the Capability Maturity Model, Version 1.1", SEI-93-TR-025, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 1993.
- [5] Dunaway, D., Masters, S.: "CMM-Based Appraisal for Internal Process Improvement, Version 1.2", CMU/SEI-2001-TR-033, USA, 2001.
- [6] Zubrow, D., Hayes, W., Siegel, J., Goldenson, D.: "Maturity Questionnaire", CMU/SEI-94-SR-7, 1994.
- [7] Romero, G.: "El Modelo de Capacidad de Madurez y su Aplicación en Empresas Mexicana de Software", Tesis de Licenciatura, Universidad de las Américas, Mexico, 2001.
- [8] Byrnes, P., Phillips, M.: "Software Capability Evaluation Version 3.0", CMU/SEI-96-TR-002, USA, 1996.
- [9] Cabezas Rojas, R.: "Programa para el mejoramiento del proceso de software en el BBVA Banco Bhif", Tesis de Licenciatura, Ingeniería Civil en Computación, Universidad de Chile, Santiago, 2002.
- [10] Garavito, S.: "Desarrollo de un modelo de madurez para valorar la gestión de proyectos en las organizaciones", IX Congreso Internacional de Ingeniería de Proyectos, AEIPRO Asociación Española de Ingeniería de Proyectos, Málaga, España, 2005.

- [11] Peralta, M.: "Asistente para la evaluación de CMMI-SW". Tesis de Maestría. ITBA. Buenos Aires, 2004.
- [12] Chang, L.: "Applying TQM, CMM and ISO 9001 in Knowledge Management for software development process improvement", *Int. J. Services and Standards*, Vol 2, No. 1, 2006.
- [13] Biberoglu, E., Haddad, H.: "A Survey of Industrial Experiences with CMM and the Teaching of CMM Practices", CSIS Department, Kennesaw State University, 2002.
- [14] Xu, R., Nie, P., Xue, Y., Zhang, Y., Li, D.: "Research on CMMI-based Software Process Metrics", Department of Computer Science and Engineering, Fudan University, School of Computer Engineering, Shandong University of Finance, Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences, 2006.
- [15] SEI, "Software Engineering Capability Maturity Model Integration", 2002 (<http://www.sei.cmu.edu/cmmi/models/models.html>).
- [16] SEI, "Capability Maturity Model Integration", 2002 (<http://www.sei.cmu.edu/cmmi/cmmi.html>).
- [17] González, M., Macías, M., Lozano, M., Sánchez, F.: "Métricas de calidad centradas en accesibilidad para KAI". Departamento de Informática Escuela Politécnica, 10071 Cáceres, España, 1999.
- [18] Raymond, D.: "Elements of a Process-Improvement Program," *IEEE Software*, Vol. 9, No. 4, July 1992, pp. 83-85.
- [19] Humphrey, W.S.: *Characterizing the Software Process: A Maturity Framework*, Software Engineering Institute, CMU/SEI-87-TR-11, ADA182895, June 1987. *IEEE Software*, Vol. 5, No. 2, March 1988, pp.73-79.
- [20] Humphrey, W.S.: D.H. Kitson, and J. Gale, "A Comparison of U.S. and Japanese Software Process Maturity," *Proceedings of the 13th International Conference on Software Engineering*, Austin, TX, 13-17 May 1991, pp. 38-49.
- [21] Kitson, D., Masters, S.: *An Analysis of SEI Software Process Assessment Results: 1987-1991*, Software Engineering Institute, CMU/SEI-92-TR-24, July 1992.

- [22] Arnulen, C., Kituyi, M., Mbarika, I., Balunywa, L., Mbarika, V., Kabaale, E.: “Adoption of the Software Capability Maturity Model by Small and Medium Enterprises”, 6th Annual International Conference on ICT for Africa 2014, 2014, pp 71-78.

Anexos

A: Cuestionario de Madurez

Este documento contiene preguntas acerca de la existencia de importantes prácticas en materia de software en su organización. Las preguntas están organizadas en grupos de áreas claves de proceso tales como: planificación del proyecto de software o aseguramiento de calidad del software, de acuerdo al SW-CMM. Un párrafo explicará brevemente cada área clave de proceso, así como también se incluyen algunos términos usados en las preguntas.

Favor, conteste todas y cada una de las preguntas basado en su experiencia y conocimiento del proceso de software de su organización y/o en el proyecto en que actualmente trabaja. Las respuestas posibles son las siguientes:

Si: la práctica está bien establecida y se realiza consistentemente;

No: la práctica no está bien establecida o es ejecutada inconsistentemente, es decir, no se realiza o se realiza a veces o frecuentemente, pero se omite ante circunstancias de emergencia;

No aplicable: se tiene el conocimiento necesario sobre el proyecto o la organización y sobre la pregunta formulada, pero se estima que la pregunta no se aplica al proyecto o a la organización;

No sé: no se tiene la suficiente certeza para contestar la pregunta.

Si usted desea ampliar su respuesta, por favor, escriba sus comentarios en los espacios destinados para ello.

Favor, responda todas las preguntas. Las respuestas y los nombres de personas y la organización se mantendrán en absoluta confidencialidad.

Gracias, por sus respuestas.

Sección I - Identificación y Antecedentes

Nombre:

Fecha:

1. Describa su cargo, por favor:

2. En qué actividades trabaja actualmente. Por favor, señale todas aquellas que lo describan:

- | | |
|---|---|
| <input type="checkbox"/> Requerimientos de software | <input type="checkbox"/> Aseguramiento de la calidad del software |
| <input type="checkbox"/> Diseño de software | <input type="checkbox"/> Administración del software |
| <input type="checkbox"/> Codificación | <input type="checkbox"/> Mejoramiento del proceso de software |
| <input type="checkbox"/> Pruebas e integración | <input type="checkbox"/> Otra (especificar) |

3. Ha recibido entrenamiento en CMM? NO SI (especificar)

4. Cuál es su experiencia en software:

en su organización actual? años.

en toda su carrera? años.

5. Ha participado en otras evaluaciones de software?

NO SI, en cuales?

Sección II - Prácticas de Software

1 Administración de requerimientos: su propósito es establecer un entendimiento común entre el cliente y el proyecto de software, acerca de los requerimientos del cliente que serán abordados por el proyecto de software. El acuerdo con los clientes es la base para planificar y administrar el proyecto de software.

Planes de software: El conjunto de planes, formales o informales, usados para señalar como deben ser realizadas las actividades de desarrollo y/o mantención.

Proceso de software: un conjunto de actividades, métodos y prácticas que la gente usa para desarrollar y mantener el software y sus productos asociados (planes del proyecto, documentos de diseño, codificación, casos de prueba, manuales de usuario).

Productos de trabajo de software: cualquier medio creado como parte de la definición, mantención o uso de un proceso de software, esto incluye: descripciones del proceso, planes, procedimientos, programas computacionales y la documentación asociada.

Requerimientos relacionados al software: es el subconjunto de requerimientos del sistema que han de ser implantados en los componentes de software de un sistema. El análisis de requerimientos de software elabora y refina los requerimientos solicitados, lo que genera la documentación de los requerimientos de software.

Aseguramiento de la calidad del software: conjunto de actividades diseñadas para evaluar el proceso con el cual los productos de trabajo de software son desarrollados o mantenidos.

1.1 ¿Se usan los requerimientos relacionados al software para establecer la base para el uso de la ingeniería de software y la administración?

SI NO No aplicable No sé

Comentarios:

1.2 ¿Cuando los requerimientos cambian, se hacen los ajustes necesarios en los planes, productos de trabajo y actividades vinculadas?

SI NO No aplicable No sé

Comentarios:

1.3 ¿El proyecto sigue una política organizacional escrita para la administración y manejo de requerimientos de software?

SI NO No aplicable No sé

Comentarios:

1.4 ¿Las personas encargadas de administrar o manejar los requerimientos están entrenadas en los procedimientos para ello?

SI NO No aplicable No sé

Comentarios:

1.5 ¿Se usan mediciones para determinar el estado de las actividades realizadas para administrar los requerimientos (por ejemplo, número de cambios de requerimientos que han sido propuestos, aprobados, abiertos e incorporados en la base)?

SI NO No aplicable No sé

Comentarios:

1.6 ¿Están las actividades de administración de requerimientos de los proyectos sujetas a la revisión de un área de aseguramiento de calidad del software?

SI NO No aplicable No sé

Comentarios:

2 Planificación del proyecto de software: su propósito es establecer planes razonables para la materialización de la ingeniería de software y la administración del proyecto de software. Sin planes realistas, una administración de proyectos efectiva no puede ser implantada.

Compromiso: acuerdo asumido libremente, que se espera sea cumplido por ambas partes.

Revisión o actividad eventual: una revisión o actividad realizada de acuerdo a la ocurrencia de un evento dentro del proyecto, por ej: una revisión al completarse una etapa del ciclo de vida.

Revisión o actividad periódica: una revisión o actividad realizada en un intervalo regular de tiempo, en lugar de esperar la ocurrencia de un evento determinado.

Plan de software: el conjunto de planes, formales o informales, usados para señalar como deben ser realizadas las actividades de desarrollo y/o mantenimiento de software.

2.1 ¿Las estimaciones de tiempo, tamaño y costo, se documentan para su uso en la planificación y seguimiento del proyecto de software?

SI NO No aplicable No sé

Comentarios:

2.2 ¿El plan de software especifica las actividades a ser realizadas y los compromisos hechos para el proyecto de software?

SI NO No aplicable No sé

Comentarios:

2.3 ¿Están de acuerdo todos los grupos e individuos afectados con los compromisos relacionados al proyecto de software?

SI NO No aplicable No sé

Comentarios:

2.4 ¿El proyecto sigue una política organizacional escrita para la planificación de un proyecto de software?

SI NO No aplicable No sé

Comentarios:

2.5 ¿Se proveen los recursos adecuados para la planificación del proyecto de software?

SI NO No aplicable No sé

Comentarios:

2.6 ¿Se usan las mediciones para determinar el estado de las actividades para la planificación de un proyecto de software (mediciones de las actividades al completar una etapa o hito, respecto del plan original)?

SI NO No aplicable No sé

Comentarios:

2.7 ¿Revisa el jefe de proyecto las actividades de planificación del proyecto de software, ya sea en forma periódica o eventual?

SI NO No aplicable No sé

Comentarios:

3 Seguimiento del proyecto de software: su propósito es proporcionar una adecuada visión del avance real del proyecto, de forma que la administración pueda tomar acciones efectivas cuando el rendimiento del proyecto de software se desvíe del plan de software.

Productos de trabajo de software: cualquier medio creado como parte de la definición, mantención o uso de un proceso de software, esto incluye: descripciones del proceso, planes, procedimientos, programas computacionales y la documentación asociada.

Revisión o actividad periódica: una revisión o actividad realizada en un intervalo regular de tiempo, en lugar de esperar la ocurrencia de un evento determinado.

3.1 ¿Son comparados los resultados reales del proyecto contra los estimados en el plan de software?

SI NO No aplicable No sé

Comentarios:

3.2 ¿Se toman acciones correctivas cuando los resultados reales se desvían considerablemente del plan de software del proyecto?

SI NO No aplicable No sé

Comentarios:

3.3 ¿Los cambios en los compromisos son acordados por todos los grupos o individuos afectados?

SI NO No aplicable No sé

Comentarios:

3.4 ¿El proyecto sigue una política organizacional escrita para el seguimiento y control de sus actividades de desarrollo de software?

SI NO No aplicable No sé

Comentarios:

3.5 ¿Existe alguien asignado en el proyecto a labores de seguimiento de los productos de trabajo de software y las actividades del proyecto (revisar en términos de esfuerzo, presupuesto, tiempo)?

SI NO No aplicable No sé

Comentarios:

3.6 ¿Se usan mediciones para determinar el estado de las actividades de seguimiento de un proyecto de software, por ejemplo el esfuerzo gastado en el seguimiento del proyecto?

SI NO No aplicable No sé

Comentarios:

3.7 ¿Las actividades de seguimiento del proyecto de software son objeto revisiones periódicas por parte de la alta gerencia?

SI NO No aplicable No sé

Comentarios:

4 Aseguramiento de la calidad del software: su propósito es proveer a la administración una visibilidad adecuada del proceso que está siendo usado y los productos que están siendo construidos. Esto implica revisar y auditar los productos de software usados y desarrollados, verificando que cumplan con los procedimientos y políticas del proyecto y de la organización.

Auditoría: revisión independiente del trabajo o parte de él para asegurar el cumplimiento de las especificaciones, los estándares, los acuerdos contractuales, u otro criterio.

Procedimiento: descripción escrita de un curso de acción para realizar una actividad determinada.

Aseguramiento de la calidad del software: conjunto de actividades diseñadas para evaluar el proceso con el cual los productos de trabajo de software son desarrollados o mantenidos.

4.1 ¿Las actividades de aseguramiento de la calidad del software están planificadas?

SI NO No aplicable No sé

Comentarios:

4.2 ¿La función del aseguramiento de la calidad del software provee de una verificación objetiva de la adhesión a los estándares, procedimientos y requerimientos de los productos de software?

SI NO No aplicable No sé

Comentarios:

4.3 ¿Los resultados de las revisiones y auditorías son entregados a los grupos o individuos afectados?

SI NO No aplicable No sé

Comentarios:

4.4 ¿Existen aspectos de no cumplimiento que no están resueltos en el proyecto y que por ello lo trate la alta gerencia?

SI NO No aplicable No sé

Comentarios:

4.5 ¿El proyecto sigue una política organizacional escrita para la implantación del aseguramiento de la calidad del software?

SI NO No aplicable No sé

Comentarios:

4.6 ¿Se proveen de recursos adecuados para realizar las actividades de aseguramiento de la calidad del software?

SI NO No aplicable No sé

Comentarios:

4.7 ¿Se usan mediciones para determinar el estado y el costo de las actividades realizadas para el aseguramiento de la calidad del software (por ejemplo, trabajo terminado, esfuerzos y fondos gastados respecto al plan original)?

SI NO No aplicable No sé

Comentarios:

4.8 ¿Las actividades de aseguramiento de la calidad del software son revisadas por la alta gerencia en forma periódica?

SI NO No aplicable No sé

Comentarios:

5 Administración de la configuración del software (control de cambios): su propósito es establecer y mantener la integridad de los productos de software del proyecto a través del ciclo de vida del proyecto de software. Esto implica identificar adecuada y oportunamente la configuración del software, es decir como está definido que operen los productos de trabajo de software (herramientas o utilitarios, hardware o software), implica controlar los cambios a la configuración y mantener la integridad y la auditoriabilidad de la configuración a través del ciclo de vida del software. Los productos de trabajo bajo la administración de la configuración incluyen aquellos productos de software que han sido liberados a los clientes y los aspectos requeridos para crear esos productos.

Auditoría: revisión independiente del trabajo o parte de él para asegurar el cumplimiento de las especificaciones, los estándares, los acuerdos contractuales, u otro criterio.

Base de software: es un conjunto de ítems de configuración (documentación y componentes de software) que han sido formalmente revisados y acordados, como base de desarrollos futuros y que solo pueden ser cambiadas mediante los procedimientos formales de control de cambios.

Ítem de configuración: un conjunto de hardware, software o ambos, que es designado, como tal, por la administración de la configuración y es tratado como una entidad simple en el proceso de administrar la configuración.

Procedimiento: descripción escrita de un curso de acción para realizar una actividad determinada.

Productos de trabajo de software: cualquier medio creado como parte de la definición, mantención o uso de un proceso de software, esto incluye: descripciones del proceso, planes, procedimientos, programas computacionales y la documentación asociada.

5.1 ¿Se planifican actividades de administración de la configuración para el proyecto?

SI NO No aplicable No sé

Comentarios:

5.2 ¿El proyecto ha identificado, controlado y puesto a disposición los productos de trabajo de software mediante el uso de administración de la configuración?

SI NO No aplicable No sé

Comentarios:

5.3 ¿El proyecto sigue algún procedimiento documentado para controlar los cambios en los ítems de configuración?

SI NO No aplicable No sé

Comentarios:

5.4 ¿Existen informes estándar sobre la base de software (por ejemplo, informe de estado o informe sobre requerimientos de cambio)?

SI NO No aplicable No sé

Comentarios:

5.5 ¿El proyecto sigue una política organizacional escrita para la implantación de las actividades de administración de la configuración?

SI NO No aplicable No sé

Comentarios:

5.6 ¿Está debidamente entrenado el personal que es responsable de las actividades de administración de la configuración?

SI NO No aplicable No sé

Comentarios:

5.7 ¿Se usan mediciones para determinar el estado de las actividades de administración de la configuración (por ejemplo: esfuerzo y fondos invertidos en estas actividades)?

SI NO No aplicable No sé

Comentarios:

5.8 ¿Existen revisiones periódicas para verificar que la base de software esté de acuerdo con la documentación que la define?

SI NO No aplicable No sé

Comentarios:

6 Enfoque en el proceso de la organización: su propósito es establecer la responsabilidad organizacional por las actividades del proceso de software que mejoren la capacidad organizacional del proceso. Esto implica mantener y desarrollar el entendimiento de los procesos (como se realiza el proceso de software) a nivel organizacional y a nivel de los proyectos, así como también coordinar las actividades para evaluar, desarrollar, mantener y mejorar estos procesos. La organización establece compromisos de largo plazo y provee de los recursos necesarios para coordinar el desarrollo y mantención de los procesos de software a lo largo de los proyectos actuales y futuros, para tal efecto existe un grupo de ingeniería de software, que se encargad de realizar estas actividades.

Proceso de software: un conjunto de actividades, métodos y prácticas que la gente usa para desarrollar y mantener el software y sus productos asociados (planes de proyecto, documentos de diseño, codificación, casos de prueba, manuales de usuario).

6.1 ¿Están coordinadas a través de la organización las actividades para desarrollar y mejorar los procesos de software de la organización y cada uno de los proyectos?

SI NO No aplicable No sé

Comentarios:

6.2 ¿Se evalúa periódicamente el proceso de software de la organización?

SI NO No aplicable No sé

Comentarios:

6.3 ¿La organización sigue algún plan escrito para desarrollar y mejorar su proceso de software?

SI NO No aplicable No sé

Comentarios:

6.4 ¿La alta gerencia apoya las actividades para el desarrollo y mejoramiento del proceso de software (por ejemplo, estableciendo planes a largo plazo y comprometiendo recursos y fondos)?

SI NO No aplicable No sé

Comentarios:

6.5 ¿Existe en la organización alguien responsable y dedicado full o part time a las actividades del proceso de software de la organización?

SI NO No aplicable No sé

Comentarios:

6.6 ¿Se usan mediciones para determinar el estado de las actividades realizadas para desarrollar y mejorar el proceso de software organizacional?

SI NO No aplicable No sé

Comentarios:

6.7 ¿Las actividades realizadas para el desarrollo y el mejoramiento del proceso de software son revisadas periódicamente con la alta gerencia?

SI NO No aplicable No sé

Comentarios:

- 7 **Definición del proceso de la organización:** Su propósito es desarrollar y mantener un conjunto de activos del proceso de software que mejoran el rendimiento del proceso a través de los proyectos y que provean las bases para beneficios acumulativos y perdurables para la organización. Esto implica desarrollar y mantener el proceso de software estándar de la organización, así como sus elementos, que incluyen por ejemplo, la descripción del modelo de ciclo de vida del software.

Estándar organizacional del proceso de software: la definición operacional del proceso básico que guía el establecimiento de un proceso de software común a través de los proyectos de software de la organización. El estándar describe los elementos fundamentales del proceso de software que cada proyecto de software se espera que incorpore dentro de su proceso de software definido.

Proceso de software: un conjunto de actividades, métodos y prácticas que la gente usa para desarrollar y mantener el software y sus productos asociados (planes de proyecto, documentos de diseño, codificación, casos de prueba, manuales de usuario).

Aseguramiento de la calidad del software: conjunto de actividades diseñadas para evaluar el proceso con el cual los productos de trabajo de software son desarrollados o mantenidos.

- 7.1 ¿Su organización ha desarrollado y mantenido un estándar del proceso de software?

SI NO No aplicable No sé

Comentarios:

- 7.2 ¿La organización recopila, revisa y divulga información relacionada al uso del estándar organizacional para el proceso de software?

SI NO No aplicable No sé

Comentarios:

7.3 ¿La organización sigue alguna política escrita para desarrollar y mantener su estándar del proceso de software?

SI NO No aplicable No sé

Comentarios:

7.4 ¿Las personas responsables de desarrollar y mantener el estándar organizacional del proceso de software han recibido el adecuado entrenamiento para desempeñar estas actividades?

SI NO No aplicable No sé

Comentarios:

7.5 ¿Se usan mediciones para determinar el estado de las actividades realizadas para definir y mantener el estándar organizacional del proceso de software?

SI NO No aplicable No sé

Comentarios:

7.6 ¿Las actividades y productos de trabajo (herramientas) para el desarrollo y mantención del estándar organizacional del proceso de software están sujetas a revisión y/o auditoría por parte de un área de aseguramiento de la calidad del software?

SI NO No aplicable No sé

Comentarios:

8 Programa de capacitación: Su propósito es desarrollar las habilidades y conocimientos necesarios en las personas, para que ellos puedan cumplir sus roles en forma eficaz y efectiva. La capacitación es una responsabilidad organizacional, pero los proyectos de software deben identificar las habilidades necesitadas y proporcionar la capacitación necesaria cuando las necesidades del proyecto son únicas.

8.1 ¿Están planificadas las actividades de capacitación?

SI NO No aplicable No sé

Comentarios:

8.2 ¿La capacitación es entregada para desarrollar las destrezas y adquirir el conocimiento necesario para desempeñar roles técnicos y administrativos de software?

SI NO No aplicable No sé

Comentarios:

8.3 ¿Los miembros del grupo de ingeniería de software y otros grupos relacionados con el software, reciben la capacitación necesaria para desempeñar sus roles?

SI NO No aplicable No sé

Comentarios:

8.4 ¿La organización sigue una política organizacional escrita para encontrar y satisfacer las necesidades de entrenamiento?

SI NO No aplicable No sé

Comentarios:

8.5 ¿Se proveen de recursos adecuados para establecer un adecuado programa de entrenamiento (fondos, herramientas de software, instalaciones de capacitación apropiadas)?

SI NO No aplicable No sé

Comentarios:

8.6 ¿Se usan mediciones para determinar la calidad del programa de capacitación?

SI NO No aplicable No sé

Comentarios:

8.7 ¿Las actividades del programa de capacitación son revisadas con la alta gerencia periódicamente?

SI NO No aplicable No sé

Comentarios:

9 Administración integrada de software: su propósito es integrar las actividades de ingeniería de software y de administración en un proceso definido y coherente, que es adaptado del proceso de software estándar de la organización. Esto involucra desarrollar el proceso de software definido para cada proyecto y administrarlo usando el proceso de software definido. El proceso de software a definir para cada proyecto se debe derivar, basar y guiar del estándar organizacional del proceso de software adaptándolo a cada uno de los proyectos. El plan de desarrollo de software está basado en el proceso de software definido para el proyecto y define cómo se deben implantar y administrar las actividades definidas.

Estándar organizacional del proceso de software: la definición operacional del proceso básico que guía el establecimiento de un proceso de software común a través de los proyectos de software de la organización. El estándar describe los elementos fundamentales del proceso de software que cada proyecto de software se espera que incorpore dentro de su proceso de software definido.

Proceso de software definido de un proyecto: la definición operacional del proceso de software usada por el proyecto, descrito en términos de estándares de software, procedimientos, herramientas y métodos.

9.1 ¿El proceso definido de software para los proyectos ha sido desarrollado siguiendo el estándar organizacional de proceso de software?

SI NO No aplicable No sé

Comentarios:

9.2 ¿El proyecto está planificado y administrado de acuerdo al proceso definido de software para el proyecto?

SI NO No aplicable No sé

Comentarios:

9.3 ¿El proyecto sigue una política organizacional escrita que requiera que el proyecto de software sea planificado y administrado usando el estándar organizacional del proceso de software?

SI NO No aplicable No sé

Comentarios:

9.4 ¿Las personas a cargo de desarrollar el proceso definido de software del proyecto reciben la capacitación requerida sobre cómo adaptar el proceso de software estándar de la organización?

SI NO No aplicable No sé

Comentarios:

9.5 ¿Se usan mediciones para determinar la efectividad de las actividades para la administración integrada de software (frecuencia, causas y magnitud de los esfuerzos para replanificación)?

SI NO No aplicable No sé

Comentarios:

9.6 ¿Las actividades y productos de trabajo usados para administrar el proyecto de software están sujetos a revisión y/o auditoría por parte de un área de aseguramiento de la calidad del software?

SI NO No aplicable No sé

Comentarios:

10 Ingeniería de productos de software: su propósito es ejecutar consistentemente un proceso de ingeniería bien definido que integre todas las actividades de ingeniería de software, para producir efectiva y eficientemente productos de software correctos y consistentes. La *Ingeniería de Productos de Software* describe las actividades técnicas del proyecto, como el análisis de requerimientos, el diseño, la codificación y el testing del software.

Proceso de software definido de un proyecto: la definición operacional del proceso de software usado por el proyecto, descrito en términos de estándares de software, procedimientos, herramientas y métodos.

Productos de trabajo de software: cualquier medio creado como parte de la definición, mantención o uso de un proceso de software, esto incluye descripciones de proceso, planes, procedimientos, programas computacionales y la documentación asociada.

Aseguramiento de la calidad del software: conjunto de actividades diseñadas para evaluar el proceso con el cual los productos de trabajo de software son desarrollados o mantenidos.

10.1 ¿Los productos de trabajo de software son producidos de acuerdo al proceso de software definido para el proyecto?

SI NO No aplicable No sé

Comentarios:

10.2 ¿Se mantiene consistencia en los productos de trabajo de software (por ejemplo, se mantiene la documentación que hace el seguimiento de los requerimientos a través del diseño, codificación y casos de prueba)?

SI NO No aplicable No sé

Comentarios:

10.3 ¿El proyecto sigue una política organizacional escrita para realizar las actividades ingeniería de software (por ejemplo, políticas que establezcan el uso de métodos y herramientas apropiadas para la construcción y mantenimiento de productos de software)?

SI NO No aplicable No sé

Comentarios:

10.4 ¿Se destinan los recursos adecuados para desempeñar las tareas de ingeniería de software (fondos, personas hábiles y capacitadas, herramientas apropiadas)?

SI NO No aplicable No sé

Comentarios:

10.5 ¿Se usan mediciones para determinar la funcionalidad y calidad de los productos de software (por ejemplo, número, tipos y gravedad de los defectos encontrados)?

SI NO No aplicable No sé

Comentarios:

10.6 ¿Las actividades y productos de trabajo usados para ingeniería de software están sujetos a revisión y/o auditoría por parte de un área del aseguramiento de la calidad del software?

SI NO No aplicable No sé

Comentarios:

11 Coordinación intergrupala: su propósito es establecer los medios para que el grupo de ingeniería de software participe activamente con los otros grupos de ingeniería, de modo que el proyecto mejore su habilidad de satisfacer las necesidades del cliente. Este aspecto interdisciplinario involucra la participación del grupo de ingeniería de software con grupos de ingeniería de otros proyectos para tratar: los requerimientos de sistema, los objetivos y los asuntos que puedan ser de interés. Representantes del grupo de ingeniería del proyecto participan en el establecimiento del nivel o tipo de requerimientos, de los objetivos y planes trabajando con el cliente y usuario final; esto será la base de las actividades de ingeniería.

11.1 ¿En el proyecto, el grupo de ingeniería de software u otro grupo de ingeniería colaboró con el cliente para establecer los requerimientos del sistema?

SI NO No aplicable No sé

Comentarios:

11.2 ¿El grupo de ingeniería está de acuerdo con los compromisos señalados en el plan del proyecto?

SI NO No aplicable No sé

Comentarios:

11.3 ¿Los grupos de ingeniería identifican, siguen y resuelven asuntos intergrupales (riesgos técnicos, planificación de actividades incompatibles o problemas de sistema)?

SI NO No aplicable No sé

Comentarios:

11.4 ¿Existe una política organizacional escrita que dirija el establecimiento de equipos interdisciplinarios de ingeniería?

SI NO No aplicable No sé

Comentarios:

11.5 ¿Las herramientas de soporte usadas por los distintos grupos de ingeniería hacen posible la comunicación efectiva y la coordinación (por ejemplo, sistemas de procesadores de texto compatibles, sistemas de base de datos, sistemas de seguimiento de problemas)?

SI NO No aplicable No sé

Comentarios:

11.6 ¿Se usan mediciones para determinar el estado de las actividades de coordinación intergrupales (por ejemplo, esfuerzo gastado por el grupo de ingeniería de software para dar soporte a otros grupos)?

SI NO No aplicable No sé

Comentarios:

11.7 ¿Las actividades de coordinación intergrupales son revisadas por el jefe de proyecto en forma periódica y/o eventual?

SI NO No aplicable No sé

Comentarios:

12 Revisiones de Pares: su propósito es eliminar defectos de los productos de trabajo de software temprana y eficientemente. Un efecto importante de ésta práctica, es que se obtiene un mejor entendimiento de los productos de trabajo de software y los defectos pueden ser prevenidos. Revisión de pares involucra una revisión metódica de los productos de trabajo de software por profesionales pares, con el propósito de identificar defectos y hacer cambios donde sea necesario.

Productos de trabajo de software: cualquier medio creado como parte de la definición, mantención o uso de un proceso de software, esto incluye descripciones de proceso, planes, procedimientos, programas computacionales y la documentación asociada.

Aseguramiento de la calidad del software: conjunto de actividades diseñadas para evaluar el proceso con el cual los productos de trabajo de software son desarrollados o mantenidos.

12.1 ¿Se planifican revisiones de pares?

SI NO No aplicable No sé

Comentarios:

12.2 ¿Las acciones correctivas asociadas con los defectos detectados por revisiones de pares son objeto de seguimiento hasta que son resueltas?

SI NO No aplicable No sé

Comentarios:

12.3 ¿El proyecto sigue una política organizacional escrita para realizar las revisiones de pares?

SI NO No aplicable No sé

Comentarios:

12.4 ¿Los participantes en revisiones de pares han recibido el entrenamiento adecuado para desempeñar sus roles?

SI NO No aplicable No sé

Comentarios:

12.5 ¿Se usan mediciones para determinar el estado de las actividades de revisión de pares (por ejemplo, número de revisiones efectuadas, esfuerzo gastado en las revisiones pares, número de productos de trabajo de software comparados contra el plan)?

SI NO No aplicable No sé

Comentarios:

12.6 ¿Las actividades de revisión de pares y productos de trabajo están sujetas a revisión y/o auditoría por parte de un área de aseguramiento de la calidad del software?

SI NO No aplicable No sé

Comentarios:

13 Administración cuantitativa de proceso: Su propósito es controlar cuantitativamente el desempeño del proceso en el proyecto de software. El desempeño del proceso de software representa los resultados reales logrados al seguir un proceso de software. El énfasis está en identificar las causas especiales de variación dentro del proceso y corregir las circunstancias que dieron lugar a la variación, para lograr las metas definidas.

Capacidad de proceso: el rango de resultados esperados que pueden ser logrados siguiendo un proceso.

Estándar organizacional del proceso de software: la definición operacional del proceso básico que guía el establecimiento de un proceso de software común a través de los proyectos de software de la organización. El estándar describe los elementos fundamentales del proceso de software que cada proyecto de software se espera que incorpore dentro de su proceso de software definido.

Proceso de software definido de un proyecto: la definición operacional del proceso de software usado por el proyecto, descrito en términos de estándares de software, procedimientos, herramientas y métodos.

13.1 ¿El proyecto sigue un plan documentado que conduzca a la administración cuantitativa del proceso?

SI NO No aplicable No sé

Comentarios:

13.2 ¿El desempeño del proceso de software definido para el proyecto es controlado cuantitativamente (por ejemplo, mediante el uso de métodos de análisis cuantitativos)?

SI NO No aplicable No sé

Comentarios:

13.3 ¿La capacidad de proceso del estándar organizacional del proceso de software es conocida en términos cuantitativos?

SI NO No aplicable No sé

Comentarios:

13.4 ¿El proyecto sigue una política organizacional escrita para medir y controlar el desempeño del proceso de software definido para el proyecto (por ejemplo, planes de proyecto que indican como identificar, analizar y controlar las causas en las variaciones respecto a lo preestablecido)?

SI NO No aplicable No sé

Comentarios:

13.5 ¿Se destinan los recursos adecuados para las actividades de administración cuantitativa del proceso (fondos, herramientas de software, programa organizacional de mediciones)?

SI NO No aplicable No sé

Comentarios:

13.6 ¿Se usan mediciones para determinar el estado de las actividades de administración cuantitativa del proceso (por ejemplo, costos de las actividades y logro de objetivos para las actividades de administración cuantitativa del proceso)?

SI NO No aplicable No sé

Comentarios:

13.7 ¿Las actividades de administración cuantitativa del proceso son revisadas con el jefe de proyecto en forma periódica o eventual?

SI NO No aplicable No sé

Comentarios:

14 Administración de la calidad del software: su propósito es desarrollar una comprensión cuantitativa de la calidad de los productos de software del proyecto y lograr metas de calidad específicas. Esto implica establecer metas de calidad cuantitativas para los productos de software basadas en las necesidades de la organización, del cliente y de los usuarios finales. Para que estas metas puedan ser logradas, la organización establece estrategias y planes, y el proyecto ajusta específicamente su proceso de software definido para lograr las metas de calidad definidas.

14.1 ¿Se planifican para el proyecto las actividades de administración de la calidad de software?

SI NO No aplicable No sé

Comentarios:

14.2 ¿Usa el proyecto metas medibles y priorizadas para administrar la calidad de sus productos de software (funcionalidad, confiabilidad, utilidad, capacidad de ser mantenido, auditoriabilidad, seguridad)?

SI NO No aplicable No sé

Comentarios:

14.3 ¿Se hacen mediciones de calidad para compararlas con las metas de calidad del producto de software y determinar si son satisfechas?

SI NO No aplicable No sé

Comentarios:

14.4 ¿El proyecto sigue una política organizacional escrita para administrar la calidad del software?

SI NO No aplicable No sé

Comentarios:

14.5 ¿Los miembros del grupo de ingeniería de software u otros grupos relacionados al software reciben entrenamiento en administración de la calidad del software?

SI NO No aplicable No sé

Comentarios:

14.6 ¿Se usan mediciones para determinar el estado de las actividades para administrar la calidad del software (por ejemplo, el costo de la baja calidad)?

SI NO No aplicable No sé

Comentarios:

14.7 ¿Las actividades para administrar la calidad de software son revisadas periódicamente por la alta gerencia?

SI NO No aplicable No sé

Comentarios:

15 Prevención de defectos: Su propósito es identificar las causas de los defectos y prevenir que se repitan.

Causa común de un defecto: una causa de un defecto que es parte inherente de un proceso o sistema. Las causas comunes afectan cada salida del proceso y cada trabajo en el proceso.

Reunión de análisis causal: una reunión, realizada después de completar una tarea específica, con el fin de analizar defectos no cubiertos durante el desarrollo de la tarea.

15.1 ¿Se planifican actividades de prevención de defectos?

SI NO No aplicable No sé

Comentarios:

15.2 ¿El proyecto realiza reuniones de análisis de causas para identificar causas comunes de defectos?

SI NO No aplicable No sé

Comentarios:

15.3 ¿Una vez identificadas las causas comunes de defectos, son priorizadas y sistemáticamente eliminadas?

SI NO No aplicable No sé

Comentarios:

15.4 ¿El proyecto sigue una política organizacional escrita para las actividades de prevención de defectos?

SI NO No aplicable No sé

Comentarios:

15.5 ¿Los miembros del grupo de ingeniería de software u otros grupos relacionados al software reciben entrenamiento requerido para realizar actividades de prevención de defectos (por ejemplo, entrenamiento en métodos de prevención de defectos y la realización de reuniones de análisis causal)?

SI NO No aplicable No sé

Comentarios:

15.6 ¿Se usan mediciones para determinar el estado de las actividades de prevención de defectos (por ejemplo, el tiempo y el costo para identificar y corregir los defectos)?

SI NO No aplicable No sé

Comentarios:

15.7 ¿Las actividades y productos de trabajo de la prevención de defectos están sujetos a revisiones por parte de un área de aseguramiento de calidad del software?

SI NO No aplicable No sé

Comentarios:

16 Administración de cambios tecnológicos: su propósito es identificar las nuevas tecnologías y transferirlas a la organización en forma ordenada. La organización establece un grupo que trabaja en los proyectos de software para introducir y evaluar nuevas tecnologías y administrar los cambios de las tecnologías en uso.

16.1 ¿La organización sigue un plan para administrar los cambios en tecnología?

SI NO No aplicable No sé

Comentarios:

16.2 ¿Se evalúan nuevas tecnologías para determinar sus efectos en productividad y calidad?

SI NO No aplicable No sé

Comentarios:

16.3 ¿Sigue la organización un procedimiento documentado para incorporar nuevas tecnologías dentro del estándar organizacional del proceso de software?

SI NO No aplicable No sé

Comentarios:

16.4 ¿La alta gerencia auspicia las actividades de la organización para administrar el cambio en tecnología (por ejemplo, estableciendo planes de largo plazo y comprometiendo fondos, personal y otros recursos)?

SI NO No aplicable No sé

Comentarios:

16.5 ¿Existe procesamiento de datos para asistir en la selección de nuevas tecnologías?

SI NO No aplicable No sé

Comentarios:

16.6 ¿Se usan mediciones para determinar el estado de las actividades organizacionales para administrar los cambios tecnológicos (por ejemplo, el efecto de implementar cambios tecnológicos)?

SI NO No aplicable No sé

Comentarios:

16.7 ¿Las actividades para administrar los cambios tecnológicos son revisadas periódicamente por la alta gerencia?

SI NO No aplicable No sé

Comentarios:

17 Administración del cambio de procesos: su propósito es mejorar los procesos de software usados en la organización, con el intento de mejorar la calidad del software, aumentar la productividad, y acortar el tiempo de desarrollo del producto. Esto implica definir metas para el mejoramiento del proceso, con el apoyo de la alta gerencia; implica identificar, evaluar e implementar mejoras al proceso de software estándar de la organización en forma continua, proactiva y sistemática. Se deben establecer programas de incentivos y entrenamiento para permitir y promover la participación de todos en las actividades de mejoramiento del proceso. Se realizan planes pilotos para validar los cambios propuestos a los procesos, los que revisados y adecuados apropiadamente al proceso de software estándar y a los proyectos.

Estándar organizacional del proceso de software: la definición operacional del proceso básico que guía el establecimiento de un proceso de software común a través de los proyectos de software de la organización. El estándar describe los elementos fundamentales del proceso de software que cada proyecto de software se espera que incorpore dentro de su proceso de software definido.

Proceso de software definido de un proyecto: la definición operacional del proceso de software usado por el proyecto, descrito en términos de estándares de software, procedimientos, herramientas y métodos.

17.1 ¿La organización sigue un procedimiento documentado para desarrollar y mantener planes para el mejoramiento continuo del proceso?

SI NO No aplicable No sé

Comentarios:

17.2 ¿El personal de la organización participa en actividades de mejoramiento del proceso de software?

SI NO No aplicable No sé

Comentarios:

17.3 ¿Se efectúan mejoramientos continuos al estándar organizacional del proceso de software y a los procesos de software definidos para los proyectos?

SI NO No aplicable No sé

Comentarios:

17.4 ¿La organización sigue una política escrita para implantar mejoras al proceso de software?

SI NO No aplicable No sé

Comentarios:

17.5 ¿En su organización se entrega entrenamiento para mejorar el proceso de software al personal técnico y la gerencia?

SI NO No aplicable No sé

Comentarios:

17.6 ¿Se usan mediciones para determinar el estado de las actividades de mejoramiento del proceso de software (por ejemplo, el efecto de implementar cada mejora del proceso comparada con sus metas definidas)?

SI NO No aplicable No sé

Comentarios:

17.7 ¿Los esfuerzos para mejorar el proceso de software son revisados periódicamente por la alta gerencia?

SI NO No aplicable No sé

Comentarios:

17.8 ¿Se registran cambios en el proceso debido a las causas de errores encontradas en las revisiones?

SI NO No aplicable No sé

Comentarios: