

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

**FUNDAMENTACIÓN E IMPLEMENTACIÓN DE
METODOLOGÍA PARA EL DESARROLLO DE
GRANDES PROYECTOS SOFTWARE PARA LA
EMPRESA INGENIERÍA SOLEM**

ANA ELIZABETH VÁZQUEZ FROW

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMÁTICA

DICIEMBRE 2014

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

**FUNDAMENTACIÓN E IMPLEMENTACIÓN DE
METODOLOGÍA PARA EL DESARROLLO DE
GRANDES PROYECTOS SOFTWARE PARA LA
EMPRESA INGENIERÍA SOLEM**

ANA ELIZABETH VÁZQUEZ FROW

Profesor Guía: **Pamela Hermosilla Monckton**

Profesor Co-referente: **Antonio Moya Villegas**

Carrera: Ingeniería Civil Informática

DICIEMBRE 2014

Dedicatoria

Este trabajo está dedicado primeramente a Dios que por Su gracia y misericordia pude iniciar y finalizar este ciclo.

A mi querida familia, a mis padres Jonatan y Selina, y a mis hermanos Jonathan, Bryan y Daniela, que sin su esfuerzo diario, palabras de aliento y apoyo incondicional, no hubiera logrado sobresalir en esta etapa tan importante de mi vida.

Y con mucho amor y cariño, a mi amado Felipe, por siempre darme los ánimos, haber sido mi compañero en este trayecto y siempre creer en mis capacidades.

Índice

Índice	ii
Lista de Figuras	iv
Lista de Tablas.....	v
Resumen	1
Abstract.....	2
1 Introducción	3
2 Objetivos del Proyecto	5
2.1 Objetivo General.....	5
2.2 Objetivos Específicos	5
3 Estado actual de la Metodología de Desarrollo del Software	6
3.1 Análisis del diagrama de flujo de actividades inicial	7
3.2 Estudio de la percepción del personal sobre la Metodología de Desarrollo utilizada	10
3.3 Método de trabajo actual del personal en el desarrollo de proyectos	15
4 Situaciones a Mejorar.....	18
5 Solución Propuesta.....	20
6 Fundamentación de la Metodología de Desarrollo Software	23
7 Propuesta de Metodología de Desarrollo	26
7.1 Descripción	26
7.2 Metodología de Desarrollo	26
7.2.1 Actividades de Gestión de Proyecto por Procesos	27
7.2.2 Actividades de Ejecución de Proyecto por Procesos	30
7.3 Indicadores Propuestos	34
7.3.1 Indicadores Gestión.....	34
7.3.2 Indicadores Ejecución	36
8 Implementación de Metodología de Desarrollo Propuesta	38
8.1 Herramientas Utilizadas.....	39
9 Conclusión	41
10 Referencias.....	43

Anexos.....	
A. Encuesta Metodología Solem.....	
B. Siglas Documentación.....	
C. Modelos de Proceso Software.....	
1. Modelo Incremental.....	
2. Modelo V.....	
3. Proceso Unificado.....	
D. Procesos de Gestión de Proyectos.....	
E. Fases del Ciclo de Vida Principal de los Proyectos.....	
F. Documentación Interna: Metodología para el Desarrollo y Mantenimiento.....	

Lista de Figuras

Figura 3-1 Metodología de Desarrollo Software Actual	6
Figura 3-2 <i>Project Lifecycle</i>	7
Figura 3-3 <i>Project Management Activities</i>	9
Figura 3-4 <i>Project Implementation Activities</i>	10
Figura 3-5 Cantidad de Personas por Vertical en la Empresa	11
Figura 3-6 Porcentaje de personas con conocimientos sobre la Metodología de Desarrollo	12
Figura 3-7 Porcentaje de personas que cree que la Metodología de Desarrollo se utiliza en la empresa.....	12
Figura 3-8 Porcentaje de trabajadores que consideran importante que la empresa posea una Metodología propia	13
Figura 3-9 Porcentaje de trabajadores que creen que sus tareas aplican en la Metodología	13
Figura 3-10 Porcentaje de trabajadores que aplican actividades según algún ámbito de la Metodología.....	14
Figura 3-11 Porcentaje de trabajadores que encuentran necesario conocer los fundamentos de la Metodología Solem.....	14
Figura 3-12 Porcentaje de trabajadores que aplican distintos modelos o metodologías a su forma de trabajo.....	15
Figura 5-1 Esquema de Solución.....	20
Figura C.1 Modelo Cascada	
Figura C.2 Modelo Incremental.....	
Figura C.3 Modelo V.....	
Figura C.4 Proceso Unificado	
Figura D.1 Procesos del Ciclo de Vida de la Gestión de Proyectos.....	

Lista de Tablas

Tabla 7-1 Entregables por Actividades Gestión Ejecutadas.....	34
Tabla 7-2 Entregables por Actividades Gestión Ejecutadas.....	36
Tabla C.1 Ventajas y Desventajas del Modelo Cascada	
Tabla C.2 Ventajas y Desventajas del Modelo Incremental.....	
Tabla C.3 Flujos de Trabajo Principales RUP.....	

Resumen

La gestión de proyectos es un área importante de una organización, que guía e integra los procesos de planificación, organización de capacidades y administración de recursos, con el objetivo de finalizar todo el trabajo necesario para desarrollar los proyectos y cumplir con los alcances, dentro de límites de tiempo y costo definidos. Es importante que una organización dedicada a la ingeniería del software tenga metodologías definidas para controlar los procesos que conllevan el desarrollo del software, que no sólo considera la creación de aplicaciones, sino que además incluye la generación de la documentación asociada a estas. La gestión del ciclo de vida software incluye procesos con actividades relacionadas al desarrollo del software. Las actividades principales que son parte de todos los modelos de procesos son el análisis, el desarrollo, la validación y la evolución del software. Una buena gestión no puede garantizar el éxito del proyecto, sin embargo, la mala gestión usualmente lleva al fracaso del proyecto, por lo tanto, es esencial una buena gestión de proyectos software para que los proyectos de ingeniería de software se desarrollen a tiempo y según el presupuesto. Se realiza un análisis y se explica el estado inicial de la metodología de la empresa Solem exponiendo las situaciones a mejorar, para luego presentar la solución principal: la fundamentación de la metodología de desarrollo de software, considerando todas las fases del ciclo de vida del software. Luego se propone un nuevo diseño de la metodología de desarrollo, presentando también indicadores que sirvan de complemento a la metodología con la finalidad de que éstas entreguen información para el control sobre la gestión de un proyecto. Finalmente, se presenta un caso de la implementación de la metodología de desarrollo para un proyecto software actualmente en ejecución y las herramientas utilizadas.

Palabras-claves: Gestión de Proyectos, Software, Gestión de Proyectos Software, Metodología de Desarrollo, Indicadores.

Abstract

Project management is an important area of an organization, that guides and integrates the processes of planning, organizing capabilities and resource administration, in order to finalize all the labor necessary to develop projects and meet the scopes, within defined limits of time and cost. It is important that an organization dedicated to software engineering has defined methodologies to control the processes that involve the development of software, which not only consider the creation of applications, but also include the generation of documentation associated with these. The management of life cycle software includes processes with activities related to the development of software. The main activities that are part of every process model are analysis, development, validation and evolution of software. A good management doesn't guaranty the success of the project, however, a mismanagement usually leads the project to failure, and therefore, a good project management software is essential for software engineering projects to be developed on time and within the budget. The initial state of the methodology of the company Solem is first analyzed and explained, exposing the flaws found, followed by the main solution: the basis of the development methodology for software, considering all the phases of the softwares life cycle. Moreover, a new design of the methodology is proposed, presenting also indicators as a complement to the methodology in order that they deliver information for a better control over the management of a project. Finally, a case of implementing the development methodology and its tools used is presented for a software project currently in execution.

Key-words: Project Management, Software, Software Project Management, Development Methodology, Indicators.

1 Introducción

Actualmente en casi todas las áreas, tanto organizacionales o sociales a nivel mundial, ya sea de dirección, administración, operaciones o ventas, por mencionar algunos, existen complejos sistemas informáticos.

Durante la crisis del software, se desarrollaron computadoras hardware basadas en circuitos integrados, lo que produjo que los softwares fueran más grandes y complejos que los sistemas previos. La experiencia obtenida en la construcción de estos sistemas durante esa época mostró que un enfoque informal para el desarrollo del software no era muy bueno. Los grandes proyectos a menudo tenían años de retraso, costaban mucho más de lo presupuestado, eran irrealizables, difíciles de mantener y con un desempeño pobre. Los costos de hardware disminuían considerablemente mientras que los del software se incrementaban con rapidez. Se necesitaban nuevas técnicas y métodos para controlar la complejidad inherente a los sistemas grandes.

Actualmente se han desarrollado métodos efectivos de especificación, diseño e implementación del software, que reducen el esfuerzo para producir sistemas grandes y complejos. En general no existe un enfoque ideal para la ingeniería de software, debido a la amplia diversidad de sistemas y organizaciones que usan estos sistemas, por lo que se necesita diversos enfoques al desarrollo de software.

La forma más efectiva de producir software de alta calidad es adoptando un enfoque sistemático y organizado en el trabajo. Para lograr esto, se busca complementar la aplicación de la ingeniería de software en el desarrollo de sistemas con la disciplina de la gestión de proyectos.

La empresa Ingeniería Solem es una compañía de servicios especializada en el desarrollo de soluciones tecnológicas de vanguardia, altamente comprometida con la modernización e innovación de los procesos de negocios de sus clientes.

Avalados por más de veinte años de experiencia generando importantes aportes al mercado de la Tecnología de la Información, cuenta con un gran equipo de trabajo, compuesto por profesionales ingenieros, analistas, especialistas funcionales y administrativos, capaces de anticipar e identificar tendencias tecnológicas, gracias a un esfuerzo permanente de investigación y desarrollo. Con todo esto, son capaces de generar valor agregado real para sus clientes.

La empresa provee soluciones tecnológicas basadas en la integración de tecnologías de información y comunicaciones para apoyar procesos de negocios y el cumplimiento de los objetivos estratégicos de sus clientes. Además, Ingeniería Solem ha incorporado técnicas y herramientas para la gestión de proyectos según las normas y buenas prácticas propuestas por el PMI® (Project Management Institute) con el propósito de lograr desarrollar y gestionar proyectos de manera exitosa y alcanzar sus objetivos.

La gestión de proyectos busca aplicar conocimientos, habilidades, herramientas y técnicas adecuadas a las actividades de ésta para cumplir con los requisitos exigidos de manera exitosa, aplicando distintas metodologías y procesos en su desarrollo e implementación.

Los proyectos de software se diferencian de otros proyectos de ingeniería tradicional en la naturaleza lógica del producto software. Esto debido a que el software se desarrolla, no se fabrica en un sentido clásico, y además no se deteriora en el paso del tiempo como productos de otra índole. El software es de naturaleza abstracta e intangible, no está restringido por materiales ni está gobernado por leyes físicas o por procesos de manufactura. Debido a que no existen limitaciones físicas del potencial del software, de alguna forma, se tiende a creer que esto simplificaría el proceso de desarrollo del sistema. Sin embargo, el software puede llegar a ser extremadamente complejo y, por lo tanto, muy difícil de entender debido a esta falta de restricciones naturales.

Es por esto que la gestión de los proyectos de software no dejan de ser importante, ya que se debe abarcar procesos de análisis, desarrollo y control, donde es necesario comprender el ámbito del trabajo a realizar, analizar los riesgos en los que puede incurrir, los recursos necesarios, las tareas a llevar a cabo, el esfuerzo o costo y el plan a seguir para tales proyectos.

Debido a que la empresa SOLEM busca desarrollar y gestionar los proyectos software de manera exitosa, manteniendo un control sobre los procesos asociados a este tipo de proyectos, es necesario definir una metodología para el desarrollo de software que pudiera ser aplicado dentro de la organización.

En el presente documento se introducen los objetivos que se busca alcanzar durante el desarrollo del proyecto y el estado inicial de la metodología actual de la empresa. Luego se describen las principales deficiencias encontradas según el análisis a la situación real de la empresa, basado en una encuesta al personal involucrado en el desarrollo de proyectos software.

Posteriormente se detallan las definiciones y los fundamentos principales para la metodología de desarrollo de software para la empresa Solem bajo una perspectiva general de gestión de proyectos. A continuación se especifica la propuesta de diseño y detalle de la metodología de desarrollo de software, con sus actividades y entregables de acuerdo a la gestión y ejecución de proyectos. Además, se proponen indicadores que ayudarán a controlar de mejor manera la gestión y ejecución de los proyectos software y se finaliza con la descripción del seguimiento a la implementación de la Metodología de Desarrollo Propuesto a un proyecto que actualmente se encuentra en ejecución dentro de la empresa Solem, detallando además las herramientas utilizadas para el desarrollo del proyecto software. Finalmente se presentan las conclusiones obtenidas del desarrollo del proyecto.

2 Objetivos del Proyecto

A continuación se darán a conocer los objetivos planteados para este proyecto. Por un lado se mencionará el objetivo general del proyecto, seguido de los objetivos específicos que servirán de apoyo a la obtención del objetivo general.

2.1 Objetivo General

Fundamentar y ajustar las metodologías de desarrollo de grandes proyectos software de la empresa Ingeniería Solem para que cuente con los elementos necesarios para gestionar y ejecutar el desarrollo de sistemas de información de manera exitosa.

2.2 Objetivos Específicos

- Evaluar el conocimiento real del personal del área de desarrollo sobre la metodología con el fin de obtener información útil para ajustar éstas según las necesidades reales de la empresa.
- Especificar las actividades y entregables de la metodología dentro del marco de la gestión de proyectos software.
- Fundamentar la metodología de desarrollo de grandes proyectos software de la empresa Ingeniería Solem.
- Definir métricas adecuadas para medir cuantitativamente el grado de eficacia de la aplicación de la metodología definida para los proyectos software.
- Implementar las metodologías definidas de desarrollo de grandes proyectos software dentro de la empresa.

3 Estado actual de la Metodología de Desarrollo del Software

La empresa Solem, con el fin de desarrollar y gestionar sus proyectos de desarrollo software de forma exitosa y asegurar el logro de los objetivos planteados en éstos, ha buscado incorporar técnicas y herramientas para una óptima gestión de tales proyectos, situación que llevó al diseño inicial de una metodología de desarrollo de proyectos software, de manera que ésta definición pudiera garantizar a sus clientes la calidad de los productos y además obtener un sistema de calidad que asegure un mejoramiento continuo.

Bajo este propósito, la empresa Solem creó un diseño inicial de flujo de actividades para el desarrollo de proyectos software en el año 2009, mostrado en la Figura 3-1, donde se encuentra esbozado las fases del ciclo de vida del proyecto, las actividades de gestión de proyecto y las actividades de implementación de proyecto.

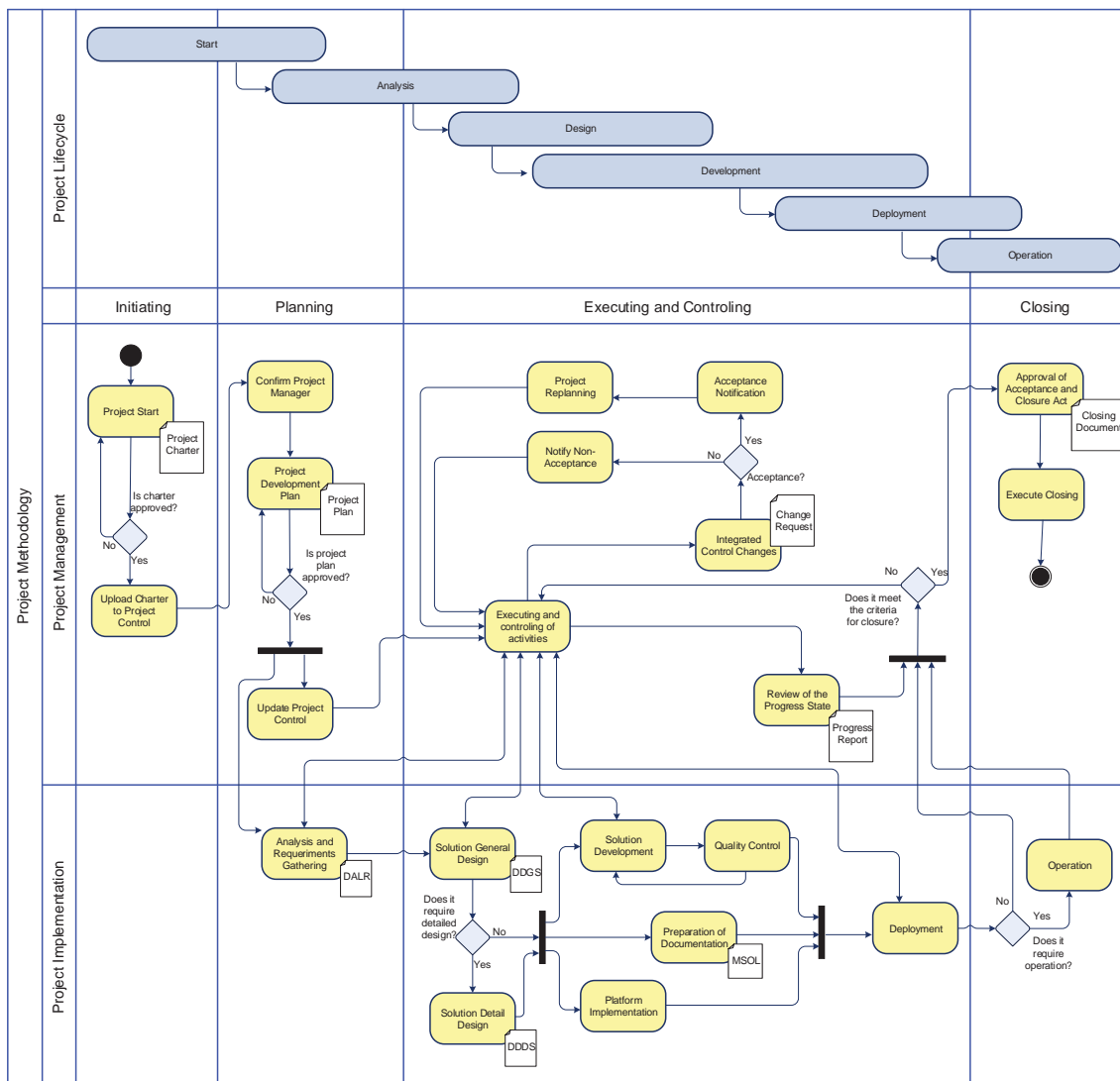


Figura 3-1 Metodología de Desarrollo Software Actual

3.1 Análisis del diagrama de flujo de actividades inicial

Según lo mostrado en el diseño inicial en la Figura 3-1, se puede desglosar el esquema señalando las fases que componen el ciclo de vida de proyectos: *Start*, *Analysis*, *Design*, *Development*, *Deployment* y *Operation*, que se visualiza en la Figura 3-2.

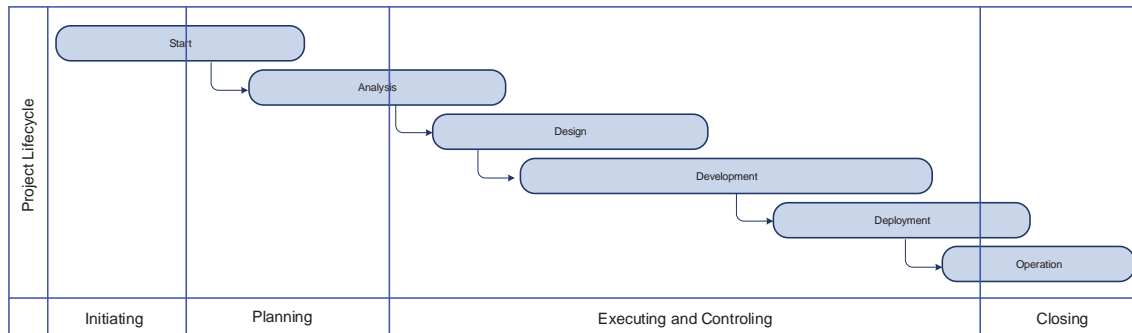


Figura 3-2 *Project Lifecycle*

Por otra parte, se identifica un diseño de flujo de actividades, las cuales se pueden resumir en dos tipos de agrupaciones:

- Por procesos, ya sean de Initiating, Planning, Executing and Controlling y Closing.
- Por estructura de trabajo, ya sea Project Management o Project Implementation.

Las actividades diseñadas en el diagrama de flujo son las siguientes, separadas por estructura de trabajo:

- *Project Management* (señalada en la Figura 3-3):
 - Project Start*: actividad que genera un *Project Charter*, la cual si es aprobada, pasa a la siguiente actividad *Upload Charter to Project Control State* (actividad b. de la estructura *Project Management*). En caso de no ser aprobada el *Project Charter*, el flujo se mantiene en esta actividad.
 - Upload Charter to Project Control*: actividad donde se sube el *Project Charter* a un controlador de proyectos (*Project Control*). El flujo continúa con la actividad *Confirm Project Manager State* (actividad c. de la estructura *Project Management*).
 - Confirm Project Manager*: actividad donde se confirma el gestor de proyectos. Luego de esta actividad sigue *Project Development Plan State* (actividad d. de la estructura *Project Management*).
 - Project Development Plan*: actividad donde se planifica el desarrollo del proyecto. La actividad genera un plan de proyecto (*Project Plan*). Si este plan de proyecto es aprobado, se continúa paralelamente en las actividades *Update Project Control State* (actividad e. de la estructura *Project Management*) y *Analysis and Requirements Gathering* (actividad a. de la estructura *Project Implementation*). En caso de no ser aprobado el *Project Plan* el flujo se mantiene en esta actividad.
 - Update Project Control*: actividad donde se actualiza el controlador de proyectos. El flujo de *Project Management* continúa con la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*).

- f. *Executing and Controlling of Activities*: actividad que ejecuta y controla las actividades que se llevan a cabo, tanto para la estructura de trabajo de *Project Management* y *Project Implementation*. Paralelamente el flujo continúa con las actividades *Integrated Control Changes* (actividad g. de la estructura *Project Management*), *Review of the Progress State* (actividad k. de la estructura *Project Management*), *Analysis and Requirements Gathering* (actividad a. de la estructura *Project Implementation*), *Solution General Design* (actividad b. de la estructura *Project Implementation*), *Solution Development* (actividad d. de la estructura *Project Implementation*) y *Deployment* (actividad h. de la estructura *Project Implementation*).
- g. *Integrated Control Changes*: actividad donde se realiza un control integrado de cambios, donde se genera un *Change Request* (una petición de cambio), la cual si es aprobada, el flujo continúa en la actividad *Acceptance Notification State* (actividad h. de la estructura *Project Management*). En el caso de no ser aprobada el *Change Request*, el flujo continúa en la actividad *Notify Non-Acceptance State* (actividad j. de la estructura *Project Management*).
- h. *Acceptance Notification*: actividad donde se notifica la aceptación del *Change Request*. El flujo continúa en la actividad *Project Replanning State* (actividad i. de la estructura *Project Management*).
- i. *Project Replanning*: actividad donde se replanifica el Proyecto. El flujo continúa con la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*).
- j. *Notify Non-Acceptance*: actividad donde se notifica la no aceptación del *Change Request*. El flujo continúa con la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*).
- k. *Review of the Progress State*: actividad donde se revisa el estado de progreso del proyecto. La actividad genera un *Progress Report* la cual, si alcanza el criterio de cierre, el flujo continúa con la actividad *Approval of Acceptance and Closure Act* (actividad l. de la estructura *Project Management*). En el caso de que el *Progress Report* no alcance el criterio de cierre, el flujo retorna a la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*).
- l. *Approval of Acceptance and Closure Act*: actividad donde se aprueba y acepta el cierre del proyecto. La actividad genera un *Closing Document*, y luego continúa el flujo en la actividad *Execute Closing*.
- m. *Execute Closing*: actividad donde se ejecuta el cierre del proyecto.

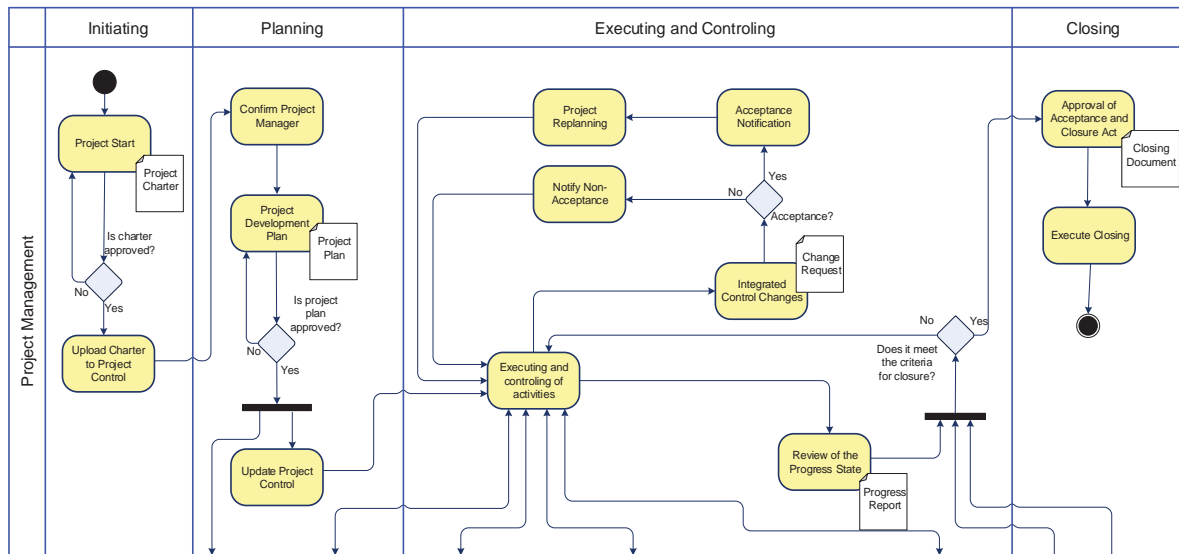


Figura 3-3 Project Management Activities

- *Project Implementation* (señalada en la Figura 3-4):
 - a. *Analysis and Requeriments Gathering*: actividad donde se analiza y recopila los requerimientos. Esta actividad genera un *DALR*. El flujo continúa en la actividad *Solution General Design* (actividad b. de la estructura *Project Implementation*).
 - b. *Solution General Design*: actividad donde se genera un diseño de solución general. La actividad genera un *DDGS*. En el caso de ser necesario un diseño detallado, el flujo continúa en la actividad *Solution Detail Design* (actividad c. de la estructura *Project Implementation*). En caso de no ser necesario un diseño detallado, el flujo continúa paralelamente con las actividades *Solution Development* (actividad d. de la estructura *Project Implementation*), *Preparation of Documentation* (actividad f. de la estructura *Project Implementation*) y *Plataform Implementation* (actividad e. de la estructura *Project Implementation*). Esta actividad además mantiene feedback de avances para la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*).
 - c. *Solution Detail Design*: actividad donde se genera un diseño detallado de la solución. La actividad genera un *DDDS*. El flujo continúa paralelamente con las actividades *Solution Development* (actividad d. de la estructura *Project Implementation*), *Preparation of Documentation* (actividad f. de la estructura *Project Implementation*) y *Plataform Implementation* (actividad e. de la estructura *Project Implementation*).
 - d. *Solution Development*: actividad donde se desarrolla la solución. El flujo continúa con la actividad *Quality Control* (actividad g. de la estructura *Project Implementation*). Esta actividad además mantiene feedback de avances para la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*).
 - e. *Plataform Implementation*: actividad donde se implementa la plataforma de la solución. El flujo de actividades continúa con la actividad *Deployment* (actividad h. de la estructura *Project Implementation*).

- f. *Preparation of Documentation*: actividad donde se prepara la documentación de la solución. Esta actividad genera *MSOL*. El flujo de actividades continúa con la actividad *Deployment* (actividad h. de la estructura *Project Implementation*).
- g. *Quality Control*: actividad donde se realiza el control de calidad de la solución. El flujo de actividades continúa con la actividad *Deployment* (actividad h. de la estructura *Project Implementation*). Esta actividad mantiene un feedback con la actividad *Solution Development* (actividad d. de la estructura *Project Implementation*).
- h. *Deployment*: actividad donde se despliega la solución. En el caso de requerir operación, el flujo continúa en la actividad *Operation* (actividad i. de la estructura *Project Implementation*). En el caso de no requerir operación, y si el despliegue cumple con los criterios de cierre de proyecto, el flujo continúa con la actividad *Approval of Acceptance and Closure Act* (actividad l. de la estructura *Project Management*). En el caso de no requerir operación, y si el despliegue no cumple con los criterios de cierre de proyecto, el flujo continúa con la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*). Esta actividad además mantiene feedback de avances para la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*).
- i. *Operation*: actividad donde se lleva a cabo la operación de la solución. En el caso de que se cumpla con los criterios de cierre de proyecto el flujo continúa con la actividad *Approval of Acceptance and Closure Act* (actividad l. de la estructura *Project Management*). En el caso de que no se cumpla con los criterios de cierre de proyecto, el flujo continúa con la actividad *Executing and Controlling of Activities State* (actividad f. de la estructura *Project Management*).

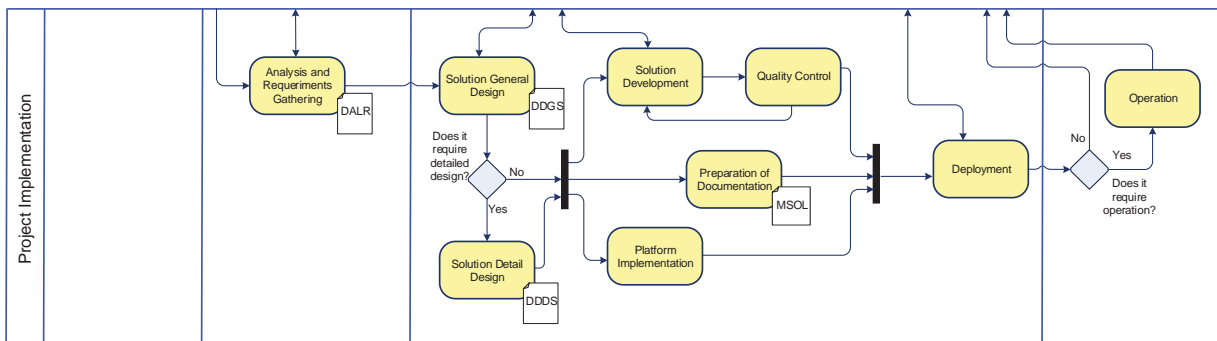


Figura 3-4 *Project Implementation Activities*

3.2 Estudio de la percepción del personal sobre la Metodología de Desarrollo utilizada

El éxito de un proyecto no es fácil y con cuestiones como la falta de liderazgo, comunicación, recursos, elementos y debido a demandas poco realistas, es fácil ver por qué los proyectos pueden fracasar. Reconocer los problemas de la gestión y desarrollo de proyectos de una manera temprana puede ayudar a resolver los problemas y dará lugar a resultados exitosos dentro de una organización. La administración de proyectos software proporciona una ventaja competitiva para la empresa Solem, ya que ésta busca mejorar sus capacidades de servicios de tecnología. El objetivo de una administración de proyecto es controlar las restricciones del

mismo: tiempo, presupuesto y alcance, mientras agrega valor al negocio, por lo que descubrir las falencias de las metodologías actuales y solucionarlas cuanto antes puede resultar en proyectos exitosos.

Para evaluar la realidad del conocimiento del personal de desarrollo de la empresa Solem acerca de la metodología aplicado actualmente para proyectos software, se llevó a cabo la generación y divulgación de una encuesta, para obtener información provechosa que será utilizada para encontrar falencias y ajustar la metodología de acuerdo a necesidades reales de la empresa. La encuesta se presenta en el **Anexo A**.

De una muestra de 56 personas correspondiente a los trabajadores del área de desarrollo de la empresa Ingeniería Solem, se pudo obtener la cantidad de trabajadores pertenecientes a cada vertical dentro de la empresa.

Las verticales existentes en el área de desarrollo son:

- Vertical Workflow y CRM (DSU)
- Vertical Logística y ERP (AHP)
- Vertical Medios de Pago Electrónico y Transaccional (JAS)
- Vertical Biometría y Sistemas Móviles (VPO)
- Vertical de Control de Calidad y Testing (AMG)
- Vertical de Diseño y Documentación (HCV)
- Vertical Grandes Proyectos (VNA)

Cada una de las verticales tiene un jefe asociado que controla a trabajadores que se encuentran a su cargo, como desarrolladores, analistas, etc., además de coordinar sus horas de trabajo y proyectos a los cuales formará parte.

Del universo, tenemos que la mayor cantidad de personas, 14, son parte de la vertical de Workflow y CRM que corresponde a un 25%, seguida por la vertical Lógica y ERP conformada por 13 (23%), 8 pertenecientes a la vertical Control de Calidad y Testing (14%), 6 parte de la vertical Diseño y Documentación (11%), 5 en las verticales de Grandes Proyectos y de Biometría y Sistemas Móviles (9%), 4 de la vertical medios de Pago Electrónico y Transaccional (7%), restando 1 persona perteneciente a otra área de la empresa (2%). Esta información se resume en el gráfico de columnas de la Figura 6.1 y el gráfico de porcentajes de la Figura 3-5.

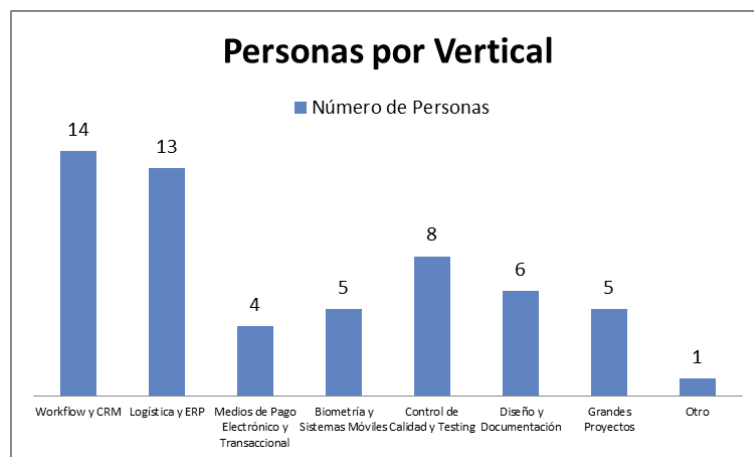


Figura 3-5 Cantidad de Personas por Vertical en la Empresa

Se realizó la medición del grado de conocimiento del personal sobre la metodología aplicado en la empresa Solem para el desarrollo de proyectos software, un 34% de los trabajadores dicen tener noción de la existencia de una metodología de proyecto mientras que sólo el 12% desconoce la metodología. Esto se refleja en la Figura 3-6, donde 1 corresponde que la persona no conoce la metodología y 5 hace referencia a que la persona conoce completamente la metodología.

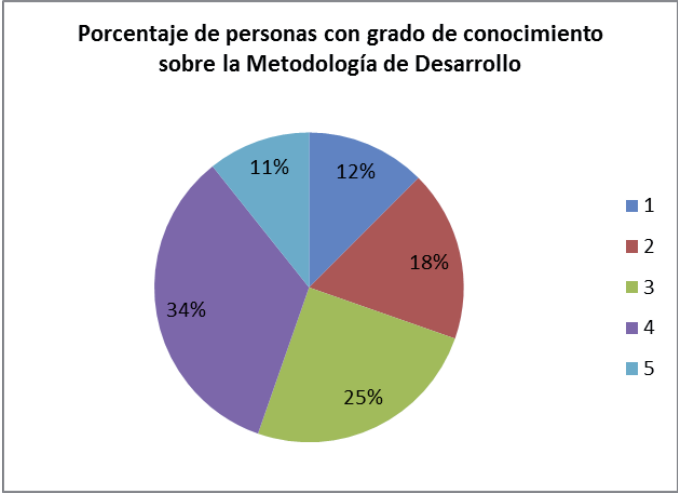


Figura 3-6 Porcentaje de personas con conocimientos sobre la Metodología de Desarrollo

Sin embargo, un 48% de los trabajadores opinan que la metodología de desarrollo se utiliza de manera informal dentro de la empresa, a diferencia de un 7% que cree que no se utiliza nunca. Esto se refleja en el diagrama de la Figura 3-7, donde 1 representa que la persona cree que la metodología no se ocupa nunca para un desarrollo de un proyecto software y 5 representa que sí cree que la metodología se ocupa siempre.

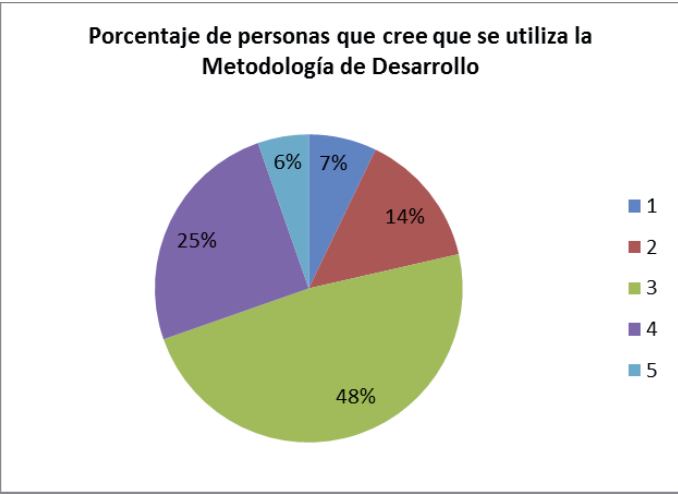


Figura 3-7 Porcentaje de personas que cree que la Metodología de Desarrollo se utiliza en la empresa

Con respecto al grado de importancia de los trabajadores para que la empresa posea una metodología propia de desarrollo de proyectos software un 75% cree que es muy importante. Esta información se muestra en la Figura 3-8, donde 1 corresponde a que la persona no considera importante que la empresa posea una metodología propia y 5 corresponde a que el trabajador considera que es muy importante.

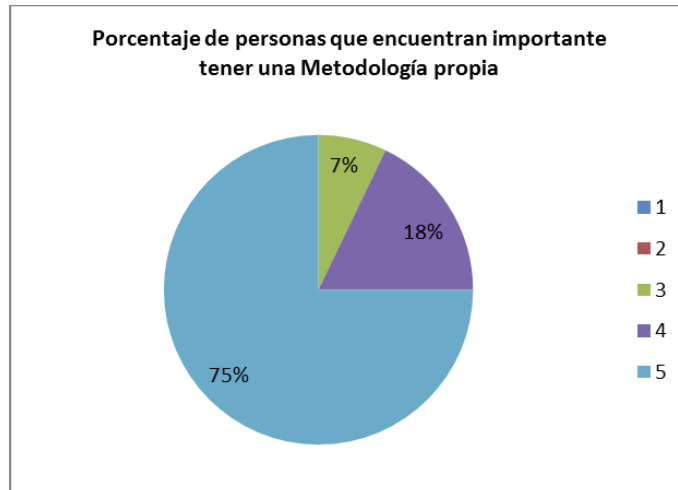


Figura 3-8 Porcentaje de trabajadores que consideran importante que la empresa posea una Metodología propia

Acerca de las tareas y actividades ejercidas por los trabajadores, se les preguntó si éstos consideraban que sus tareas se aplicaban dentro de la metodología, a lo que un 23% respondieron que sus tareas sí aplicaban a diferencia del 2% que creía que no. Esto se ve reflejado en el diagrama de la Figura 3-9, donde 1 corresponde a que la persona no creía que sus tareas aplican en la metodología de desarrollo y 5 corresponde a que las tareas sí aplican en su totalidad.

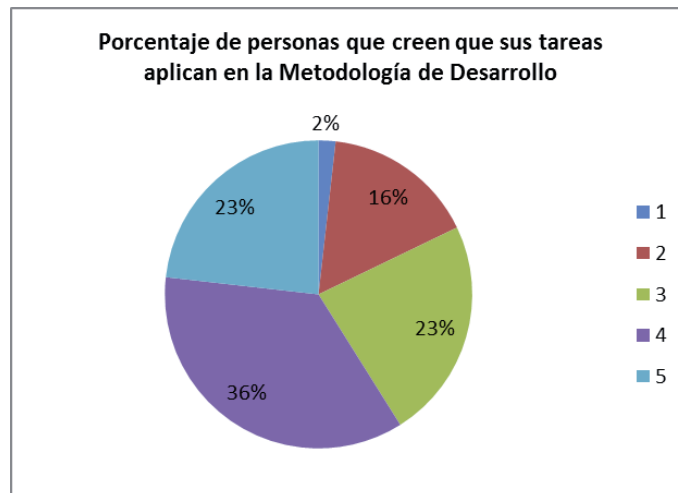


Figura 3-9 Porcentaje de trabajadores que creen que sus tareas aplican en la Metodología

Además se les preguntó a los trabajadores el ámbito de la metodología que creían en el cual empeñaban sus actividades, esto es, si las actividades correspondían a la Gestión de Proyectos, a la Ejecución de Proyectos, ambos, ninguna o desconocía el ámbito. A esto, un 39% de los trabajadores considera que sus actividades son del ámbito de Ejecución de Proyecto, a diferencia de un 9% de los trabajadores considera que sus actividades corresponden al ámbito de Gestión de Proyectos. Un 36% indicó que sus actividades aplicaban a ambas, un 12% desconocía el ámbito en el cual ejecutaba sus actividades y un 4% mencionó que no aplicaba a ninguno de los ámbitos. Esto se visualiza en la Figura 3-10.

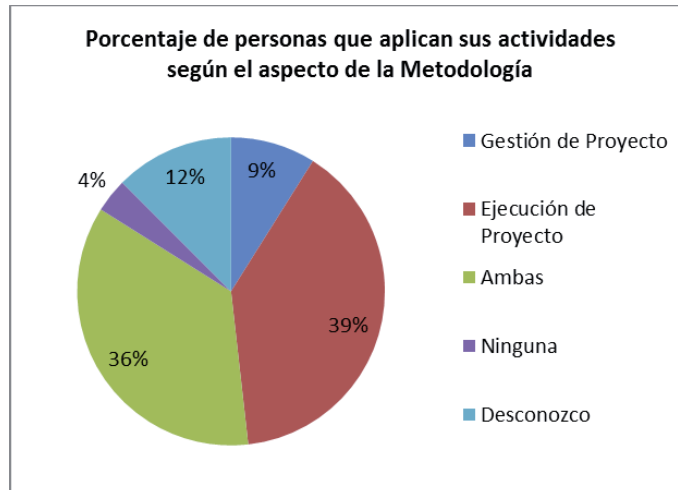


Figura 3-10 Porcentaje de trabajadores que aplican actividades según algún ámbito de la Metodología

Acercas de los trabajadores que encuentran que es necesario que los involucrados en el desarrollo de un proyecto conozcan los fundamentos de la Metodología Solem, un 81% considera que es muy importante que el personal adquiera este conocimiento. El diagrama de la Figura 3-11 señala esta información, donde 1 hace referencia a que no es necesario conocer los fundamentos de la metodología y 5 hace referencia a que es muy importante tener estas bases.

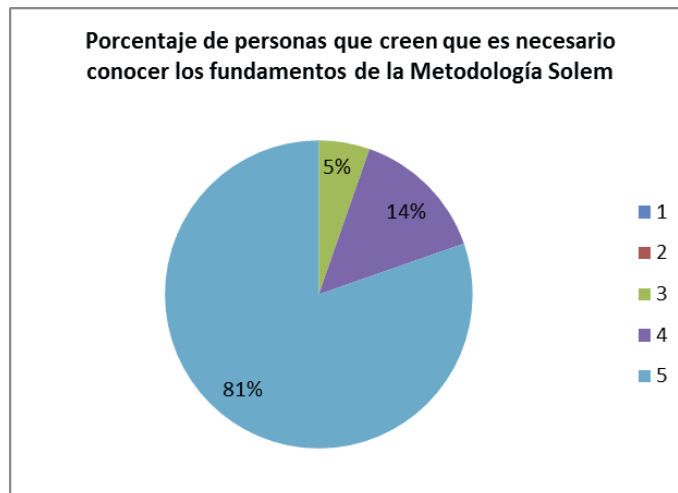


Figura 3-11 Porcentaje de trabajadores que encuentran necesario conocer los fundamentos de la Metodología Solem

Para fundamentar la necesidad del conocimiento de las bases de la Metodología Solem, se preguntó a los trabajadores qué modelos o metodologías creían que se aplicaban a su forma de trabajo en el desarrollo del proyecto. Los resultados arrojaron que la mayoría tienen una noción acerca de los modelos que forman parte de sus actividades diarias. Sin embargo existe un leve porcentaje de trabajadores que seleccionaron modelos que no aplican a la metodología o que desconocían los modelos aplicados. Los resultados se reflejan en la Figura 3-12.

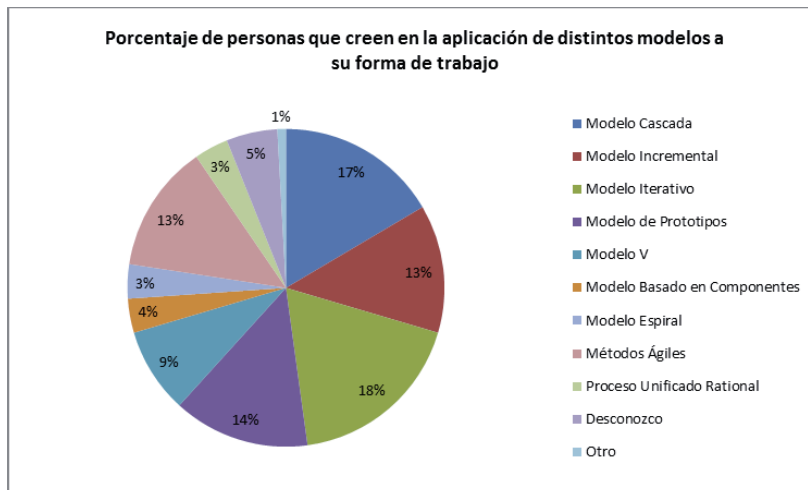


Figura 3-12 Porcentaje de trabajadores que aplican distintos modelos o metodologías a su forma de trabajo

Poder definir correctamente los fundamentos y los elementos de las metodologías presentadas anteriormente, permitirá, por un lado, que los trabajadores de la empresa comprendan de mejor manera la forma de trabajo que se lleva a cabo para realizar un proyecto software de buenos resultados, además de tener claro cuál es el rol y el aporte que cada uno de ellos dan en el desarrollo de un proyecto. Por otra parte, si la empresa posee una metodología con buenos fundamentos que pueda ser expuesta a sus clientes, estos últimos tendrán una mejor confianza al solicitar un servicio a la empresa, además de conocer las fases por la cual atraviesa el servicio solicitado.

3.3 Método de trabajo actual del personal en el desarrollo de proyectos

Los desarrollos de proyectos software en la empresa Solem han sido en su mayoría implementados de manera informal, pero siempre bajo algunos estándares, las cuales dieron con la definición inicial de la metodología. Actualmente el ciclo de vida de los proyectos de desarrollo software contempla seis fases, las cuales indican un funcionamiento de tipo cascada. Este ciclo es utilizado como la estructura base de la metodología, sugiriendo un enfoque sistemático y secuencial hacia el desarrollo del software. También define el flujo de actividades a seguir en el desarrollo de proyectos, guiando la planificación y el desarrollo del software promoviendo una metodología de trabajo efectiva, definiendo el alcance antes de diseñar la solución e implementarlo, para finalmente implantar y dejar la solución operando en ambientes del cliente.

El conjunto de tareas definidas actualmente en la metodología son aplicadas en cualquier proyecto debido a su flexibilidad, pudiendo adaptarse a las necesidades del mismo. Aunque también, según el alcance y complejidad del proyecto, es posible no abarcar alguna de las fases mencionadas anteriormente. Además, el flujo de actividades no es estricto en sus ejecuciones secuenciales, por lo que se puede repetir más de alguna actividad dependiendo del avance del proyecto y de los resultados de cada una de ellas.

En el flujo de actividades actual para la gestión de proyectos primeramente se genera una ficha donde se especifica el alcance general del problema a solucionar, el cual es aprobado por el cliente para dar un inicio formal al desarrollo del proyecto. Esta ficha posteriormente es

subida a un sistema de “Control de Proyectos” para el registro de sus ingresos y egresos, sin embargo, como este documento no siempre es actualizado en el sistema de control, se suele saltar esta actividad para conformar un equipo de trabajo y jefe de proyecto, creando una planificación inicial de las tareas a cumplir durante el desarrollo de la solución. Estas actividades no siempre se realizan en esta secuencia, pues en el momento del planteamiento del problema, ya puede existir un equipo pre-definido que trabajará en el desarrollo de la solución y también un plan básico de las tareas a llevar a cabo.

Con el equipo de trabajo y la planificación de tareas definidas se genera un documento donde se especifica el análisis y levantamiento de requerimientos, la que idealmente debe definir el alcance del proyecto en su totalidad. Sin embargo, a medida que los proyectos avanzan en su desarrollo, suelen surgir nuevos requerimientos, los cuales producen cambios en todo el flujo posterior de actividades, como por ejemplo, el diseño de la solución y la documentación asociada al proyecto (manuales, especificaciones técnicas, etc.). Estos cambios deben ser controlados por el jefe de proyecto, generando los documentos de solicitudes que respalden las modificaciones dadas y actualizar los estados de avance del proyecto, ya sea con o sin modificaciones. Pero no siempre se generan estos respaldos a tiempo, lo que genera retrasos en las actualizaciones de avances provocando desorden en términos de dirección de proyecto. Además, el jefe de proyecto en conjunto con el cliente, deben aprobar los documentos generados para la continuación del flujo de actividades en el desarrollo del proyecto, como por ejemplo, los documentos de requerimientos o diseño. Debido a que estos documentos sufren cambios durante el desarrollo del proyecto, no siempre reciben las aprobaciones formales correspondientes, sin embargo, esto no detiene el flujo de actividades de avance de desarrollo, pero si mantiene un desorden en el seguimiento del proyecto.

Independiente de que el documento de requerimientos reciba las aprobaciones necesarias, el equipo de trabajo procede en diseñar e implementar la solución del proyecto. La implementación de la solución se basa en el diseño lógico de ésta, por lo que debe existir una propuesta sólida de modelo de datos, casos de uso y diagramas de flujo de actividades. Sin embargo, como el diseño sufre constantes cambios debido a las modificaciones solicitadas esto implica que la implementación de la solución también sufrirá cambios. Si no existe un control de versionamiento ya sea de documentación o de procesos de implementación se puede generar una confusión de los últimos cambios a llevar a cabo. Es por esto que es de vital importancia que el jefe de proyecto actúe como controlador de actividades, ya sea documentación o desarrollo, para un óptimo avance de proyecto para todo el equipo de trabajo. Por otra parte, a medida que se implementa la solución esta pasa por un control de calidad, donde se prueban los componentes desarrollados para su aprobación y futura implantación, generando una certificación interna de que el proyecto desarrollado cumple con las necesidades del cliente y genera las salidas esperadas. Se genera un bucle entre la implementación y el control de calidad de la solución, en el caso que los componentes no pasen las pruebas de calidad, esta debe ser mejorada por parte del equipo de desarrollo para nuevamente probarlos.

De forma paralela a la implementación, se lleva a cabo la confección de documentación asociada al proyecto, como manuales de usuario o manuales técnicos. También se realiza la actividad de implementación de la plataforma, la cual se encargará de dar soporte a la solución desarrollada tanto por el lado del proveedor como del cliente. Como se mencionaba, los cambios que se vayan generando en el tiempo de desarrollo del proyecto, implica

modificaciones en estos documentos asociados (manuales). Sin embargo, como los cambios son mayormente funcionales que técnicos, normalmente no se realizan modificaciones en la implementación de la plataforma, las definiciones de especificaciones técnicas quedan congeladas al inicio del proyecto.

Finalizado la implementación y pruebas de la solución, el proyecto se implanta en las dependencias del cliente, esto es, se instala la solución en el ambiente del usuario. Se realiza entrega de documentaciones necesarias, como los manuales de instalación o manuales de usuario. En el caso de ser necesario, la aplicación pasa por la actividad de operación, donde se realizan las pruebas de aceptación de usuario y capacitaciones para cumplir con la satisfacción del cliente. Una vez que se cumpla el alcance definido, se genera un documento donde se especifican los criterios de cierre las cuales son aprobados y aceptados tanto por el proveedor como el cliente. Este documento y su aprobación dan por finalizado el proyecto.

El desarrollo de proyectos software incorpora prácticas que enfatizan en las comunicaciones cara a cara en vez de la documentación, sin embargo durante el desarrollo se presenta una documentación ágil, que se actualiza a medida avanza el proyecto. Los documentos están orientados al respaldo del proyecto de forma de involucrar activamente al cliente en el proceso. La metodología actual de Solem posee documentación para ciertas actividades, las que se van adaptando de acuerdo al desarrollo de incrementos. La ejecución de las tareas se realiza en equipos compuesto por jefes de proyecto, analistas, diseñadores, programadores, que en conjunto, colaboran en la evolución de los requisitos y la solución del proyecto.

4 Situaciones a Mejorar

Mediante el análisis realizado sobre la aplicación de la metodología actual de la empresa, se ha podido identificar algunas falencias dentro de éstas, lo que podría generar problemas al momento de querer mantener un control o una buena gestión sobre el o los proyectos que estén siendo conducidos a través de las actividades. A continuación se detallan algunos problemas encontrados:

1. No existe fundamentación ni documentación con definiciones asociadas a la metodología.
2. El equipo de trabajo no conoce en su totalidad la metodología utilizada actualmente en el desarrollo de proyectos, ya sea de los procesos o las actividades necesarias para llevar a cabo.
3. No existe un documento que lleve un registro de historia, donde se describan los temas tratados y acuerdos conversados en reuniones con el cliente a través de ciertas actividades dadas de la metodología. Actividades Inicio de Proyecto; Ejecución y Control de Actividades; Análisis y Levantamiento de Requerimientos.
4. No existe un documento donde se describan los riesgos asociados al proyecto y se detalle una planificación de cómo éstos serán abordados al momento de ejecutar la actividad de generación de un plan de proyecto.
5. No existe una actividad donde se genere y prepare un equipo de trabajo que será el responsable de llevar a cabo las diversas actividades que conforman la ejecución del proyecto.
6. No se requiere de la actividad de actualización de la ficha de proyecto en el sistema, pues esta se define al inicio del proyecto detallando la problemática a resolver y el alcance que tendrá el desarrollo de la solución.
7. No existe una planificación de los cambios o las modificaciones solicitadas por parte del cliente, sino que se actualiza la planificación general del proyecto, lo que afecta en los tiempos de ejecución de tareas y de entrega del proyecto. Si los tiempos requeridos para los cambios son muy largos, impactará de mayor forma los tiempos del proyecto general.
8. No existe un documento estándar para la definición de cambios o modificaciones a aplicar en el desarrollo del proyecto.
9. No existe una actualización del plan de proyecto luego de las revisiones de estado de avances, que debería considerar cumplimiento de actividades, posibles atrasos, avances de porcentaje, incorporación de cambios, etc. entregando esta actualización para informar al cliente.
10. No existe actividad que controle el versionamiento de documentos, códigos, avances, cambios a aplicar, generación de informes, etc.
11. No existe una actividad donde se defina la arquitectura de la solución del proyecto.
12. No existe una documentación estándar asociada a la actividad de diseño detallado para la definición de diseño del modelo de datos, diseño de infraestructura tecnológica o diseño de infraestructura física.

13. No existe un flujo en la metodología de desarrollo para el caso de desarrollos incrementales, los cuales deben incorporar actualizaciones de diseño e implementación de los incrementos.
14. No existe un flujo de planificación de pruebas para el control de calidad.
15. No existe una actividad relacionada a la instalación y pruebas en el ambiente de pruebas del cliente, para su certificación.
16. No existe un flujo asociado para las capacitaciones requeridas para el uso de la solución en ambientes del cliente.
17. No existe un flujo asociado para la instalación en ambientes de producción del cliente.
18. No existe un flujo asociado para la puesta en marcha de la solución en ambientes del cliente.
19. No existe un flujo asociado para un periodo de marcha blanca de la puesta en operación de la solución en ambientes del cliente.
20. No existe una documentación estándar para la aprobación y cierre de actividades del proyecto.

5 Solución Propuesta

En base a la necesidad que la empresa Ingeniería Solem tenga una metodología estable y útil para el desarrollo de sus proyectos software, se ha generado una propuesta de solución según las actividades llevadas a cabo durante el avance del trabajo, señaladas en el esquema de la Figura 5-1.

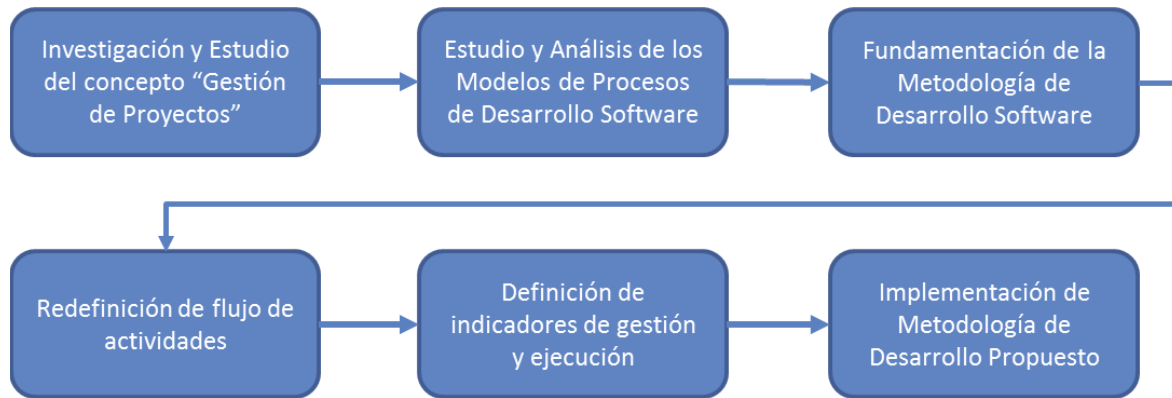


Figura 5-1 Esquema de Solución

Se realizó la investigación y el estudio sobre temas relacionados al concepto de “Gestión de Proyectos” tanto tradicionales como proyectos software, además se analizaron los distintos modelos de procesos de desarrollo software. En base a estos estudios, se surgieron las siguientes soluciones a los problemas presentados en el punto anterior:

a. Fundamentación de la Metodología de Desarrollo Software:

- Con este documento se pretende fundamentar y definir las bases de la metodología de desarrollo.
- Con la metodología documentada correspondiente, se podrá realizar la distribución de sus fundamentos al equipo de trabajo para que conozcan los procesos y actividades asociados al desarrollo del proyecto.

b. Redefinición del flujo de actividades para el desarrollo de proyectos software:

- Se define una actividad de “Conformar y Preparar Equipo de Trabajo” donde se deberá conformar un equipo de trabajo que será el responsable de llevar a cabo las diversas actividades que conforman la ejecución del proyecto.
- Se elimina la actividad de actualización de la ficha de proyecto en el sistema pues el alcance del proyecto ya se encuentra definido en aspectos generales desde la actividad “Inicio del Proyecto”.
- Se define una actividad de “Planificación de Control de Cambio” donde se realizará la planificación de los cambios o las modificaciones solicitadas por parte del cliente.
- Se define una actividad de “Actualización de Plan de Proyecto” que se realizará luego de las revisiones de estado de avances, donde se debe considerar el cumplimiento de actividades, posibles atrasos, avances de porcentaje, incorporación de cambios, etc. entregando esta actualización para informar al cliente.

- Se define una actividad de “Gestión de la Configuración” donde se deberá controlar el versionamiento de documentos, códigos, avances, cambios a aplicar, generación de informes, etc.
 - Se define dentro de la actividad de “Diseño General” la actividad donde se defina la arquitectura de la solución del proyecto, por lo que esta actividad quedará como “Diseño General y Arquitectura de la Solución”.
 - Se define un flujo en la metodología de desarrollo para el caso de desarrollos incrementales, los cuales deben incorporar actualizaciones de diseño e implementación de los incrementos.
 - Se define un flujo de planificación de pruebas para el control de calidad, el cual incluirá las actividades “Plan de Pruebas” y “Fichas de Pruebas”. También se definen documentos estándar asociados a las actividades, siendo estos “DDPP – Documento de Diseño de Plan de Pruebas” y “DDFP – Documento Ficha de Pruebas”.
 - Se define una actividad de “Instalación, Pruebas y Certificación” donde se realizaría a la instalación y pruebas de la solución en el ambiente de pruebas del cliente, para su certificación. También se define el documento estándar asociado a la actividad, siendo este “DPIP - Diseño Plan de Instalación y Puesta en Marcha”.
 - Se define un flujo asociado para las capacitaciones requeridas para el uso de la solución en ambientes del cliente, el cual incluirá la actividad “Capacitaciones”.
 - Se define un flujo asociado para la instalación en ambientes de producción del cliente, el cual incluirá la actividad “Instalación en Producción”. También se define el documento asociado a la actividad, siendo este “DIAP - Documento Instalación Ambiente Producción”.
 - Se define un flujo asociado para la puesta en marcha de la solución en ambientes del cliente, el cual eliminará la actividad “Operación” e incluirá la actividad “Puesta en Marcha”.
 - Se define un flujo asociado para un periodo de marcha blanca de la puesta en operación de la solución en ambientes del cliente, el cual eliminará la actividad “Operación” e incluirá la actividad “Marcha Blanca”.
- c. *Generación de documentos que apoyen el flujo de actividades:* (las siglas de la documentación se encuentran en el **Anexo B**).
- Se define un documento que llevará un registro de historia, donde se describan los temas tratados y acuerdos conversados en reuniones con el cliente a través de ciertas actividades dadas de la metodología, como por ejemplo: Actividades Inicio de Proyecto; Ejecución y Control de Actividades; Análisis y Levantamiento de Requerimientos, que son actividades donde se discuten y definen tareas y compromisos a llevar a cabo. El documento se define como “DMAP – Documento de Minutas y Acuerdos del Proyecto”.
 - Se define un documento donde se describan los riesgos asociados al proyecto y se detalle una planificación de cómo éstos serán abordados al momento de ejecutar la actividad de generación de un plan de proyecto. El documento se define como “DPRI

- Documento Diseño Plan de Riesgos” y se deberá generar en la actividad “Generación de Plan de Proyecto”.
- Se define un documento estándar para la definición de cambios o modificaciones a aplicar en el desarrollo del proyecto. El documento se define como “FC – Ficha Control de Cambio” y se deberá generar en la actividad “Control Integrado de Cambios”.
- Se definen documentos estándar asociados a la actividad de diseño detallado para la definición de diseño del modelo de datos, diseño de infraestructura tecnológica o diseño de infraestructura física. Los documento se definen como “DDIF – Documento de Diseño de Infraestructura Física”, “DDIT – Documento de Diseño de Infraestructura Tecnológica”, “DDMD – Documento de Diseño de Modelo de Datos”
- Se define un documento estándar para la aprobación y cierre de actividades del proyecto. El documento se define como “CAC – Certificado de Aceptación y Cierre”.

d. Definición de Indicadores:

- Se definen Indicadores de Gestión como la cantidad de entregables generados, el porcentaje de objetivos cumplidos o el porcentaje de acierto en estimaciones de tiempo, datos que apoyarán a los jefes de proyecto a llevar un control más detallado del avance del desarrollo del mismo.
- Se definen Indicadores de Ejecución como el porcentaje de actividades de la metodología llevadas a cabo o porcentaje de tareas ejecutadas, datos que apoyarán a los desarrolladores, analistas, etc. a tener un mejor control acerca de los trabajos que deben realizar durante el desarrollo del proyecto.

Estas soluciones propuestas se describen a continuación, señalando inicialmente los fundamentos teóricos de la Metodología de Desarrollo y luego se detalla la metodología propuesta junto con la definición de los Indicadores de Gestión y Ejecución.

6 Fundamentación de la Metodología de Desarrollo Software

Toda gestión de proyecto tiene asociado una metodología que debe estar reducida a responder a preguntas como: qué se está abordando, qué se debe hacer, qué se va a hacer, cómo se hará, cómo se sabrá que se hizo y qué tan bien se hizo. Si alguna metodología no llegase a responder alguna de estas preguntas, es considerada incompleta.

Diferentes tipos de sistemas necesitan diferentes procesos de desarrollo. Estas actividades genéricas pueden organizarse de diferentes formas y describirse en diferentes niveles de detalle para diferentes tipos de software. Existen distintos modelos de procesos del software que describen de manera simplificada los procesos y las actividades necesarias para el desarrollo de software. Las actividades principales que son utilizados mayormente en el desarrollo de software son: el análisis, el diseño, el desarrollo y la validación del software. En el **Anexo C** se muestran los modelos en los que se fundamentan las metodologías.

La empresa considera proyectos grandes a aquellos proyectos que abarcan un tiempo de desarrollo de un año aproximadamente, las que se pueden dividir en etapas con duraciones mensuales considerando además sobretiempos, y que contempla un equipo de trabajo de 10 a 12 personas entre desarrolladores, diseñadores y jefes de proyecto.

Un *proyecto* es una secuencia planificada de actividades únicas, complejas e interrelacionadas, que buscan lograr un objetivo o propósito, y deben ser completadas dentro de un tiempo específico, un presupuesto y de acuerdo a una especificación [3]. Es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único [1], esto es, tiene un principio y un final definidos. Un proyecto opera bajo cinco limitaciones o restricciones: alcance, calidad, costo, tiempo y recursos, mostrado en el triángulo del alcance de la Figura 6-1.



Figura 6-1 Triángulo del Alcance de un Proyecto

Las restricciones forman un conjunto interdependiente, esto es, el cambio de alguna limitación puede requerir el cambio de otro, a fin de restaurar el equilibrio del proyecto. El área dentro del triángulo representa el alcance y la calidad del proyecto. Las líneas que representan el tiempo, el costo y los recursos disponibles limitan el alcance y la calidad.

La *gestión de proyectos* es considerada una disciplina que guía e integra los procesos de planificar, captar, organizar capacidades y administrar recursos, con el objetivo de finalizar todo el trabajo necesario para desarrollar un proyecto y cumplir con el alcance, dentro de límites de tiempo y costo definidos. Es la aplicación de conocimientos, habilidades,

herramientas y técnicas a las actividades del proyecto para cumplir con los requisitos del mismo [1].

El Project Management Institute (PMI) y el International Project Management Association (IPMA) han confirmado una ampliación del campo de la gestión de proyectos que va más allá que el enfoque en la administración de estos, ya que también abarca la administración de programas, portafolios y organizaciones que logran sus objetivos estratégicos a través de proyectos, programas y portafolios, esto también se conoce como la “gestión de proyectos organizacionales”. El objetivo de la gestión de proyectos organizacional no es sólo entregar los proyectos a tiempo, dentro del presupuesto y conforme a las especificaciones técnicas y de calidad. El objetivo es crear valor para el negocio [2].

Un *programa* se define como un grupo de proyectos relacionados, administrado de forma coordinada para obtener beneficios y control, que no se obtendrían si se gestionaran en forma individual. Un proyecto puede o no formar parte de un programa, pero un programa siempre incluye proyectos.

Un *portafolio* se refiere a un conjunto de proyectos o programas y otros trabajos que se agrupan para facilitar la dirección eficaz de ese trabajo para cumplir con los objetivos estratégicos del negocio. La Figura 6-2 ilustra la relación existente entre los proyectos, programas y portafolios desde el punto de vista ofrecido por el PMI.

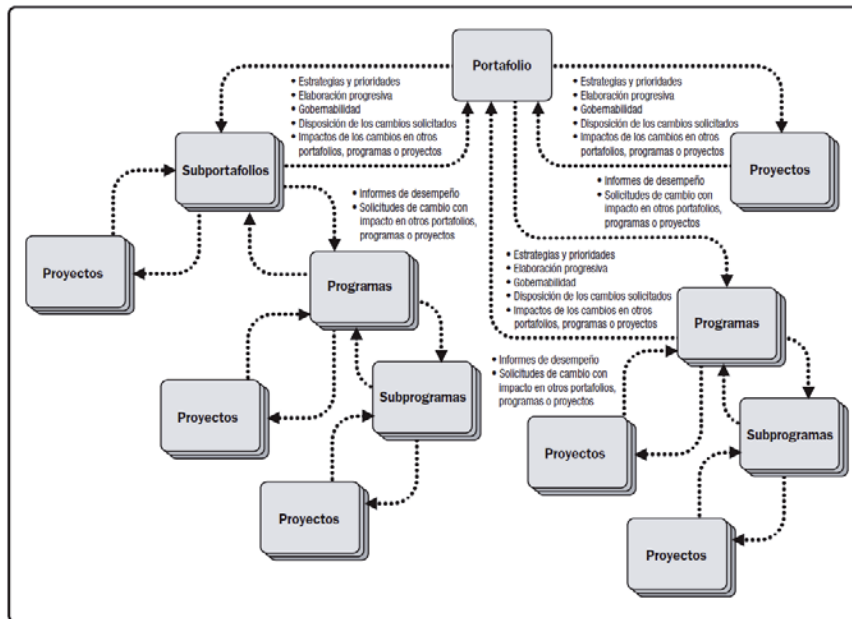


Figura 6-2 Relación entre Portafolio, Programas y Proyectos

La gestión de proyectos software es una parte esencial de la ingeniería del software. La buena gestión no puede garantizar el éxito del proyecto, sin embargo, la mala gestión usualmente lleva al fracaso del proyecto [4].

Cualquier modelo de ciclo de vida de gestión de proyecto debe contener los procesos mostrados en la Figura 6-3. Los procesos y sus actividades se describen en el **Anexo D**.



Figura 6-3 Procesos del Ciclo de Vida de la Gestión de Proyectos

A grandes rasgos la gestión de proyectos abarca identificar requisitos; establecer objetivos claros y posibles de realizar; equilibrar los costos, plazos, alcances y calidad; y adaptar las especificaciones, estrategias, planes y enfoque a las inquietudes de los diferentes interesados.

Algunas definiciones que se deberá tener en cuenta para comprender de mejor manera la metodología de desarrollo son las siguientes:

- Los *activos de los procesos* son los planes, los procesos, las políticas, los procedimientos y las bases de conocimiento específicos de la organización ejecutora y utilizados por la misma. También incluyen bases de conocimiento de la organización, como lecciones aprendidas e información histórica.
- Los *factores ambientales* de la empresa hacen referencia a condiciones que no están bajo el control del equipo del proyecto y que influyen, restringen o dirigen el proyecto. Estos pueden influir de manera positiva o negativa sobre el resultado. Se consideran algunos factores ambientales, la cultura de la organización, la distribución geográfica o la infraestructura.
- Un *interesado* es un individuo, grupo u organización que puede afectar, verse afectado o percibir a sí mismo como afectado por una decisión, actividad o resultado de un proyecto. Pueden participar activamente o tener intereses a los que puede afectar la ejecución o la terminación del proyecto y pueden ejercer influencia sobre el proyecto, los entregables y el equipo del proyecto.
- Un *proceso* es un conjunto de acciones y actividades interrelacionadas realizadas para obtener un producto, resultado o servicio definido [1].
- Un *software* es todos los documentos asociados y la configuración de datos que se necesitan para hacer que estos programas operen de manera correcta [4].
- Un *proceso del software* es un conjunto de actividades y resultados asociados que producen un producto software.
- La *ingeniería del software* es una disciplina de la ingeniería cuya meta es el desarrollo costeable de sistemas software. Comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de éste después de que se utiliza.

La metodología Ingeniería Solem incorpora estas definiciones para el desarrollo de sus proyectos software aplicando las técnicas y herramientas de gestión de proyectos según las normas y buenas prácticas propuestas por el Project Management Institute.

7 Propuesta de Metodología de Desarrollo

7.1 Descripción

El ciclo de vida principal de los proyectos software de la empresa Ingeniería Solem, mostrada en la Figura 7-1, contempla seis fases principales y una fase transversal que forman la estructura de la metodología de gestión de proyectos, siendo estas fases las siguientes:

- Inicio
- Análisis
- Diseño
- Implementación
- Implantación y Operación
- Complementado con un proceso transversal de Monitoreo y Control.



Figura 7-1 Fases del Ciclo de Vida de Proyectos Software

Las fases, tanto del proceso de desarrollo como el de mantenimiento, se encuentran insertas dentro de los procesos claves propuestos por el PMI para la gestión de proyectos. En el **Anexo E** se describen las fases del ciclo de vida de los proyectos. Esto se debe a que los procesos de desarrollo software tradicionales carecen de normas y prácticas necesarias para una buena gestión y control del avance de un proyecto software. Esta integración se refleja en la Figura 7-2.



Figura 7-2 Integración de Procesos Claves con el Ciclo de Vida de Proyectos Software

A continuación se describen las metodologías de Desarrollo y de Mantenimiento de la empresa Solem en forma general. Se realizaron modificaciones al diagrama de flujo de actividades, indicando en cada una de ellas los entregables que debieran generar.

7.2 Metodología de Desarrollo

Esta metodología integra las buenas prácticas propuestas por el PMI y los activos propios de la empresa.

La metodología tiene dos vistas:

- Las fases del desarrollo del Proyecto.
- Las actividades de dirección del Proyecto.

En la Figura 7-3, se presenta el diagrama con las actividades que contempla la metodología de desarrollo utilizada por Ingeniería Solem, las cuales están agrupadas en:

- Ciclo de Vida del Proyecto: Fases que sigue el Proyecto, lo que permite mantener una planificación, control y seguimiento, mejorar la relación costo-beneficio y facilita la comunicación y la evaluación de cumplimientos de objetivos con nuestros Clientes.
- Gestión de Proyecto: Actividades que se desarrollan para definir, guiar y controlar el proyecto, esta actividades son ejecutadas en su mayoría por el Jefe de Proyectos.
- Ejecución de Proyecto: Actividades que se desarrollan (llevan a cabo) para desarrollar los productos, los cuales van generando una serie de entregables a lo largo del ciclo de vida.

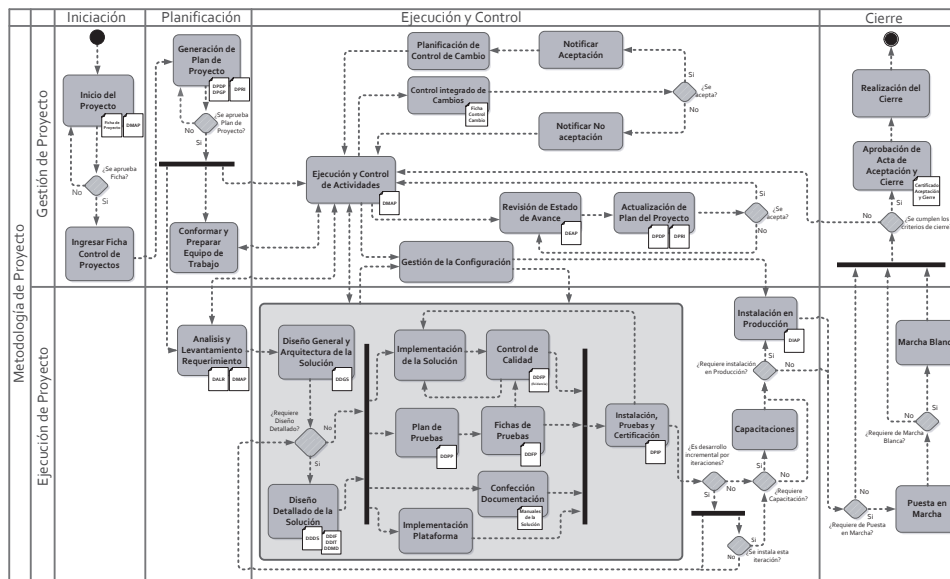


Figura 7-3 Metodología de Desarrollo Propuesta

A continuación se describen las actividades de la metodología de desarrollo software seccionado por los procesos de Iniciación, Planificación, Ejecución y Control y Cierre de acuerdo a las tareas llevadas a cabo por la gestión o la ejecución del proyecto.

7.2.1 Actividades de Gestión de Proyecto por Procesos

En la Figura 7-4 se muestra las actividades asociadas al grupo de gestión de proyectos software.

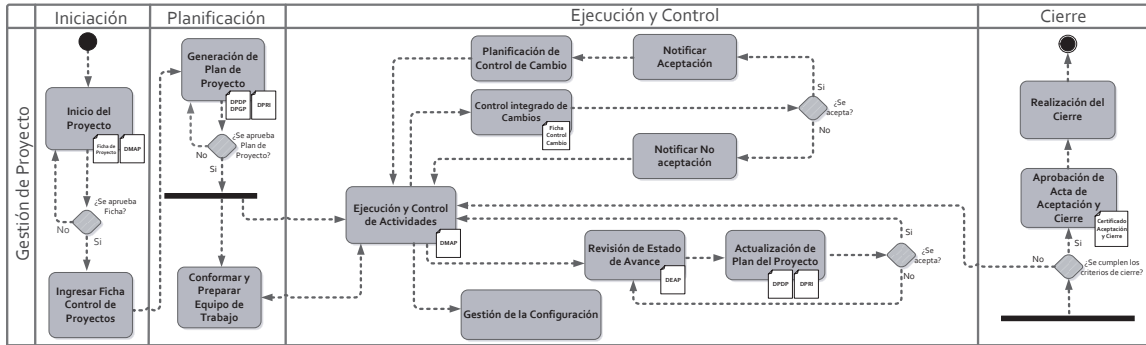


Figura 7-4 Actividades de la Gestión de Proyecto para la Metodología de Desarrollo Propuesta

A continuación se describen las actividades asociadas a la gestión de proyecto por cada proceso para la metodología de desarrollo junto con los entregables correspondientes.

7.2.1.1 Iniciación

Las actividades de la fase de Iniciación son:

- **Inicio del Proyecto:** Es la primera actividad que se realiza para iniciar un proyecto dentro de la metodología de desarrollo de un proyecto software. Dentro de esta actividad se genera y presenta al cliente una propuesta de solución y presupuesto inicial, el cual debe ser aprobado. Se debe recopilar todos los documentos asociados al proyecto: cotizaciones, evaluaciones, contrato, etc.

Luego que el cliente haya aprobado la propuesta inicial del proyecto. Se elabora y acuerda el *Acta de Constitución del Proyecto* o *Ficha de Proyecto*. Se debe documentar los requisitos iniciales que satisfacen las expectativas de los interesados.

Se concluye la actividad con una reunión kick off del proyecto, para conocer a los interesados, ajustar los plazos, definir planes de proyecto y dar por iniciado el proyecto formalmente. Aquí es donde el cliente autoriza formalmente el inicio del proyecto. Si no se aprueba el documento, ésta se corrige de acuerdo a los requerimientos del cliente hasta ser aceptado. Se genera un documento de minutas y acuerdos del proyecto de lo revisado en las reuniones para mantener un seguimiento a lo conversado con los clientes en las reuniones. (*Documento de Minutas y Acuerdo – DMAP*). Además esta actividad tiene como entregable la *Ficha de Proyecto*.

- **Ingresar Ficha Control de Proyectos:** Si no existe, se crea en el sistema gestión el proyecto asociándolo a un programa donde se pueden manejar sus ingresos, egresos, archivos (legales y documentación) y notas de pedido. En caso de que el proyecto exista, se cargan los documentos legales asociados. En el sistema gestión se realiza el seguimiento y control del proyecto software a desarrollar.

7.2.1.2 Planificación

Las actividades de la fase de Planificación son:

- **Generación Planes de Proyecto:** En esta actividad se genera una planificación general de las actividades e hitos del proyecto, donde se considera el nombre de las tareas a realizar, su duración, inicio, fin, porcentaje completado y sus predecesoras.

El documento de planificación del proyecto corresponde a la programación del proyecto una vez que se ha efectuado el cierre de venta con el cliente. Este proceso consiste en la

especificación, el gestor y el cliente de los compromisos que deben ser cumplidos durante la ejecución. Aquí se especifican todas las etapas acordadas y los hitos que deben ser cumplidos en cada una de ellas.

Se genera el documento plan de riesgos y comunicaciones del proyecto, que son relevantes para el buen desarrollo del proyecto. Además se considera otros planes, como costos y calidad, que en este caso, deberán ser provistos por la parte encargada de la gestión del proyecto. La planificación general y de riesgos debe ser aprobada por parte del cliente, o reformulada hasta que el cliente esté satisfecho.

Esta actividad tiene como entregable el *Documento Plan General (DPGP)*, el *Documento Planificación del Proyecto (DPDP)* y el *Documento Plan de Riesgo (DPR)*.

- **Conformar y Preparar el Equipo de Trabajo:** Una vez aprobado el plan de proyecto, se realiza la formación del equipo de trabajo que se encargará de realizar las tareas para el desarrollo del proyecto. Se asignan los analistas, diseñadores y desarrolladores que serán parte del avance del proyecto.

Para esta actividad además, se debe realizar una reunión introductoria al equipo para presentar los conceptos y las reglas del negocio del proyecto que se deben aplicar.

Durante el control de actividades, los participantes del equipo de trabajo pueden ir cambiando a medida que avanza el proyecto. Esto conlleva a tener nuevas reuniones o juntas con los nuevos integrantes del equipo para presentar las reglas del negocio del proyecto y la resolución de dudas de integrantes antiguos.

7.2.1.3 Ejecución y Control

Las actividades de la fase de Ejecución y Seguimiento son:

- **Ejecución y Control de Actividades:** Actividad transversal donde se supervisa y analiza el progreso y el desempeño del proyecto, controlando en todo momento que vaya de acuerdo a lo planificado y dentro del alcance definido, de forma que se puedan detectar anomalías para hacer acciones correctivas en el momento que corresponda.

Corresponde a todas las reuniones que se tendrán durante todo el proyecto entre los jefes de proyecto, con la finalidad que se conversen los distintos acuerdos entre las partes, se llegue a consensos y se aprueben los informes entregados durante el transcurso del proyecto, aclarando así de mejor forma las observaciones que puedan aparecer desde las distintas partes interesadas. Estas reuniones se proponen hacerlas en forma semanal, quedando abierta la posibilidad de modificarlas si el proyecto lo requiere o si alguna de las partes lo solicita. Se generan documentos de minutas (*DMAP*) y acuerdos del proyecto de lo revisado en las reuniones para mantener un seguimiento a lo conversado con los clientes en las reuniones.

- **Revisión de Estado de Avance:** Actividad encargada de revisar el estado de avance del proyecto. Se debe verificar el cumplimiento de las actividades presentadas en el Documento Plan General Proyecto. El registro de las actividades se debe realizar en el *Documento Estado Avance Proyecto (DEAP)*. En el documento se presenta la carta Gantt de las etapas a declarar asociadas al proyecto junto con tablas resumen del desarrollo y estado de las respectivas etapas y sus actividades. Incluye la planificación establecida del proyecto para contextualizar temporalmente los estados de avance de las tareas involucradas.

Este documento sirve de entregable al cliente para una mejor comprensión del avance real en el que se encuentra el proyecto y además se utiliza como documento de control interno de la empresa SOLEM.

- **Actualización Plan del Proyecto:** Esta actividad tiene como objetivo actualizar tanto el plan del proyecto (*DPDP*) como el plan de riesgos (*DPRI*), las que deben ser aceptadas por el cliente. Durante las reuniones sostenidas durante el desarrollo del proyecto, los hitos y actividades de la planificación pueden ir variando para ajustarse a los requisitos generales del proyecto. Lo mismo sucede con los requerimientos iniciales de los clientes, que pueden ir alterándose de forma no significativa, a medida que el grupo vaya aclarando las reglas del negocio, tanto por lado del cliente como por lado del proveedor. Estas modificaciones pueden realizarse a lo largo del proyecto, aunque puede sufrir un congelamiento en el transcurso, ya sea por decisión del proveedor o del cliente.

- **Control Integrado de Cambios:** La actividad tiene como objetivo controlar los cambios producidos en el desarrollo del proyecto mediante la documentación y seguimiento del cambio. Se registran cambios de requerimientos o de planificación que son propuestos por el cliente, y se presenta para su aceptación. El registro de cambios se realiza en la *Ficha Control de Cambio*, documento entregable de la actividad.

- **Notificar No Aceptación:** Cuando un cambio que ha sido planteado y puesto entre la lista de necesidades a implementar no es posible de ser realizado, es muy importante informar lo antes posible de esta situación al cliente y así no tener que esperar hasta la próxima reunión de seguimiento para proporcionar esta información crítica.

- **Notificar Aceptación:** Cuando un cambio que ha sido planteado por el cliente y analizado por parte del proveedor es aceptado, éste forma parte a la lista de necesidades a implementar del proyecto, situación que se debe informar al cliente por cualquiera de los medios disponibles y no esperar la siguiente reunión de control de actividades.

- **Planificación de Control de Cambio:** Se realiza una planificación de hitos, actividades y tiempos de entrega para el cambio a implementar.

- **Gestión de la Configuración:** Esta actividad está presente durante todo el proceso de desarrollo del software, y tiene una serie de actividades y responsabilidades frente al proceso, entre las cuales se encuentran el control de versiones, control de cambios y generación de informes.

7.2.1.4 Cierre

Las actividades de la fase de Cierre son:

- **Aprobación Acta de Aceptación y Cierre:** Una vez que se hayan cumplido con todos los requisitos del proyecto, el encargado debe generar el *Certificado de Aceptación y Cierre* donde se describen los criterios de finalización, documento que se considera como entregable de la actividad. El documento debe ser aprobado por el cliente para finalizar el proceso de entrega del proyecto con éxito. Se podría modificar el documento de acuerdo a observaciones dadas por el cliente al hacer la entrega del proyecto.

- **Realización del Cierre:** Esta actividad tiene como finalidad la reunión final donde el cliente formaliza la aceptación del proyecto, firmando el Certificado de Aceptación y Cierre, y se finaliza con éxito la entrega del proyecto.

7.2.2 Actividades de Ejecución de Proyecto por Procesos

En la Figura 7-5 se muestra las actividades asociadas al grupo de gestión de proyectos software.

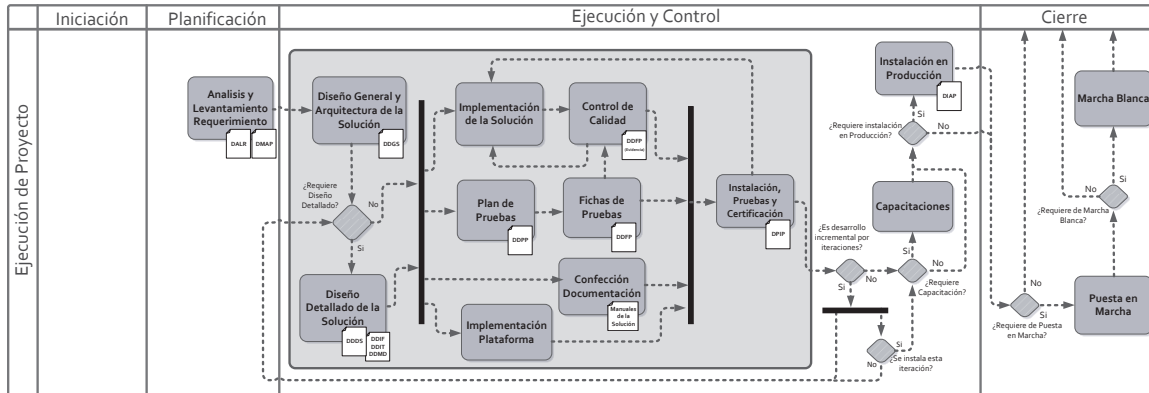


Figura 7-5 Actividades de Ejecución de Proyecto para la Metodología de Desarrollo

A continuación se describen las actividades asociadas a la gestión de proyecto por cada proceso para la metodología de desarrollo junto con los entregables correspondientes.

7.2.2.1 Planificación

Las actividades de la fase de Planificación son:

- **Análisis y Levantamiento de Requerimientos:** Una vez aprobado el plan del proyecto por parte del cliente, se debe levantar y documentar todos los requerimientos del proyecto, de manera de poder determinar en forma exacta el alcance del proyecto y todas las partes estén de acuerdo con lo que será construido y entregado como producto final. En esta actividad se genera un documento que indica cuales son los requerimientos del proyecto y de qué tipo, siendo el entregable de la actividad el *Documento Análisis y Levantamiento de Requerimientos (DALR)*.

El documento debe ser revisado por el cliente y se debe generar los ajustes necesarios producto de la revisión, de manera de llegar a un documento final acordado entre las partes. El cliente debe realizar una aceptación formal del documento para la continuación correcta del diseño del proyecto.

7.2.2.2 Ejecución y Control

Las actividades de la fase de Ejecución y Control son:

- **Diseño General y Arquitectura de la Solución:** En esta actividad el equipo de proyecto define la solución, tanto funcional como física (cuando corresponda) que va a satisfacer los requerimientos de negocio. Se planifica cómo se va a implementar y aceptar la solución, se planifica el soporte de la solución y se reciben consideraciones a la solución propuesta y a los servicios de soporte.

Además, se genera el entregable *Documento Diseño General Solución (DDGS)*, donde se establecen las definiciones necesarias para comprender el funcionamiento del sistema propuesto y todas sus interrelaciones con su entorno. En el documento se utilizan diagramas en lenguaje de modelamiento UML los que permiten conocer lo que debe hacer la aplicación, adjuntando la documentación asociada a cada uno de ellos. Además, se brinda propuestas del diseño de las interfaces, las cuales pueden ayudar en la comprensión de las funcionalidades de los casos de uso.

Se debe realizar las modificaciones al diseño general según las revisiones entregadas por el cliente, de manera que se llegue a común acuerdo entre las partes. Una vez que se acepta el diseño general de la solución, se da por cerrada la tarea.

- **Diseño Detallado de la Solución:** Esta actividad se encarga de generar un diseño detallado de la solución, que se lleva a cabo en caso de que el cliente encuentre que el documento de diseño general de la solución esté incompleto. Para esto, se agrega al documento general el diseño de los diagramas de estados o de secuencia, las interfaces del sistema definitivas, el modelo de datos con su respectivo diccionario y se especifican las reglas de negocio del sistema. Este documento completo es el entregable *Documento de Diseño Detallado de la solución (DDDS)* de la actividad.

El cliente debe revisar el documento de diseño detallado y aprobarlo. En caso de obtener observaciones por parte del cliente, se debe realizar las modificaciones al diseño general según las revisiones, de manera que se llegue a común acuerdo entre las partes. Una vez que se acepta el diseño detallado de la solución, se da por cerrada la tarea.

En caso de ser requerido por el cliente y dependiendo del proyecto, se deben adjuntar el *Documento de Diseño de Infraestructura Física (DDIF)*, de *Diseño de Infraestructura Tecnológica (DDIT)* y de *Diseño de Modelo de Datos (DDMD)*.

En el caso de un desarrollo incremental, se verifica si la etapa de incremento requiere de diseño un diseño detallado para su implementación.

- **Implementación de la Solución:** Una vez que se haya aprobado el diseño general (o diseño detallado) de la solución, corresponde la actividad de desarrollar el software solicitado según las especificaciones de diseño aprobadas por el cliente. Se construye o se obtiene cada componente de la solución, luego se ensamblan, se prueba y se integra con los otros componentes relacionados. Estos componentes se prueban como un todo y pueden estar en categorías de hardware, software, aplicación o documento.

Es esencial que el equipo aplique los principios de control de cambios y administración de la configuración. Esta fase involucra a muchas personas trabajando simultáneamente, pero con independencia en tareas complejas.

- **Plan de Pruebas:** Esta actividad tiene como objetivo desarrollar un plan de pruebas, que es un factor crítico de éxito para la puesta en práctica de un proceso de pruebas que permita entregar un software de mejor nivel. Se genera un documento de *Diseño de Plan de Pruebas (DDPP)* FAT e Integradas provistas por Solem. Una vez que el sistema se ha construido, es necesario hacerlo pasar por una serie de pruebas antes de entrar a la fase de producción. Mediante dichas pruebas, se medirá su reacción frente a diversas acciones que realizarán los usuarios desde sus páginas. El objetivo del plan de pruebas es encontrar la mayor cantidad de errores para garantizar la calidad del sistema que se entregará. Se identifican los casos y escenarios que serán probados posteriormente en la etapa de implantación durante el periodo de pruebas integradas.

- **Fichas de Pruebas:** En esta actividad se detalla el *Documento Diseño de las Fichas de Pruebas (DDFP)* FAT e Integradas. El objetivo de las pruebas es encontrar la mayor cantidad de errores para garantizar la calidad del sistema que se entregará. Para esto se genera el documento de diseño ficha de pruebas donde se implementan, tanto las pruebas unitarias, integradas y las de no regresión. Aquí se diseñan casos de prueba poniéndose en todos los casos para lograr cumplir el objetivo. Las fichas de prueba pasan a ser parte de la ejecución del testing en la actividad de control de calidad, en las pruebas de certificación y en la aprobación del sistema ya en producción.

- **Control de Calidad:** En esta actividad se realizan las pruebas funcionales y de no regresión de la solución, de forma de asegurar que se cumplan las especificaciones de la mantención acordada y no se haya incorporado un error en la solución general. Las evidencias de las pruebas se detallan en el documento diseño fichas de prueba.

- **Confeción Documentación:** Esta actividad tiene como objetivo generar documentos de capacitación y manuales de usuario para el cliente. Dependiendo de los requerimientos solicitados por el cliente, se genera el documento de plan de capacitación, manual de usuario, materiales de capacitación técnica y usuario, entre otros. Algunos entregables de esta actividad son el *Documento Plan de Capacitación (DDPC)* y *Documento Manual de Usuario (DMUS)*.

- **Implementación Plataforma:** Durante esta actividad se preparan los ambientes necesarios tanto para el ambiente de desarrollo, testing, certificación y producción cuando corresponda. Además, se preparan los ambientes para el control de las versiones de los distintos ambientes.

- **Instalación, Pruebas y Certificación:** En esta actividad los componentes de la solución se distribuyen al cliente y se implantan, se instalan todas las componentes de la solución en un ambiente de certificación. La solución se prueba como un todo en un ambiente operacional para la prueba de aceptación formal por parte del cliente. El plan de pruebas cumple con demostrar al cliente que la solución cumple los requerimientos de la especificación funcional. El cliente aquí confirma por escrito que todas las pruebas han sido exitosas y que acepta la solución mediante la aprobación del documento *Diseño de Plan de Instalación y Puesta en Marcha (DPIP)*.

- **Capacitaciones:** Se llevan a cabo los diferentes cursos de capacitación a los clientes usuarios que hayan sido comprometidos en el alcance del proyecto.

- **Instalación en Producción:** Durante esta actividad, todas las componentes de la solución se encuentran listas para su explotación normal. La solución se instala en el ambiente de producción para su revisión.

Se solicita al cliente final o encargado de la solución por parte del cliente, la revisión y aprobación formal del *Documento Instalación Ambiente Producción (DIAP)*. En caso de encontrarse cualquier problema o error, se generarán los protocolos de rollback para dejar el ambiente de producción operativo, tal como se encontraba antes del cambio.

7.2.2.3 Cierre

Las actividades de la fase de Cierre son:

- **Puesta en Marcha:** Esta actividad tiene como objetivo asegurar que las distintas partes de un sistema involucradas en el proyecto se desenvuelvan según las especificaciones del cliente en el ambiente de producción. En caso de requerir ajustes o modificaciones, se realizan las actividades correspondientes a control de cambios.

- **Marcha Blanca:** El objetivo de tener la marcha blanca es realizar pruebas en producción. En caso de que el software desarrollado no fue sometido a pruebas exhaustivas antes de ser puestos en producción, o en caso de que el cliente lo solicite, es necesario realizar la actividad de marcha blanca.

7.3 Indicadores Propuestos

Los indicadores aportan la información necesaria para verificar el progreso hacia el logro de los objetivos establecidos en el proyecto. Un indicador es un algoritmo o fórmula que expresa la relación cualitativa o cuantitativa entre dos o más variables y que sirve para medir cuánto se ha logrado del objetivo. A continuación se describen indicadores para la fase de gestión y la fase de ejecución de la metodología:

7.3.1 Indicadores Gestión

1. **Entregables por Actividades Ejecutadas:** cantidad de documentos entregables que se hayan generado mediante las actividades de Gestión a lo largo de proyecto. Durante el proyecto se deberá generar ocho entregables en el transcurso de las actividades. Si se cumple la entrega de todos los documentos, se mantuvo un buen seguimiento y control sobre el proyecto.

- a. Ficha Proyecto
- b. Documento Plan General Proyecto
- c. Documento Planificación del Proyecto
- d. Documento Planificación de Riesgos
- e. Minutas de Acuerdos
- f. Ficha Control de Cambios
- g. Documento Estado Avance Proyecto
- h. Certificado de Aceptación y Cierre

Tabla 7-1 Entregables por Actividades Gestión Ejecutadas

<=3	Mal Seguimiento y Control sobre el Proyecto (a, b, h)
>3,<=5	Seguimiento y Control Regular sobre el Proyecto (a, b, e, f, h)
>=6,>=8	Muy buen Seguimiento y Control sobre el Proyecto (a, b, d, e, f, h o todas + versiones)

2. **Grado de Satisfacción del Cliente:** busca medir el nivel de conformidad y aceptación del cliente del producto o servicio entregado por la empresa una vez finalizado el desarrollo del proyecto. Una mayor satisfacción por parte del cliente, indica mayor posibilidad de que el cliente vuelva a comprar o contratar los servicios de la empresa. La medición de satisfacción del cliente se realizará mediante una encuesta de cumplimiento de requerimientos, el nivel de trato y atención recibida por parte del personal de la empresa y una evaluación del producto final, además de una sección de opinión, donde el cliente pueda expresar con mayor libertad su grado de conformidad según el producto desarrollado o el servicio entregado. Las encuestas por proyecto varían de acuerdo al alcance de cada una de ellas. Luego de obtener las respuestas por parte del cliente, estas se analizan para definir el nivel de satisfacción. La definición de la escala del nivel de satisfacción será de acuerdo al proyecto que se desarrollará.

- Insatisfacción: Se produce cuando el desempeño percibido del producto no alcanza las expectativas del cliente.
- Satisfacción: Se produce cuando el desempeño percibido del producto coincide con las expectativas del cliente.

- Complacencia: Se produce cuando el desempeño percibido excede a las expectativas del cliente.

3. **Porcentaje de Actividades Ejecutadas:** este indicador mantiene un control sobre la cantidad de actividades de gestión llevadas a cabo durante el proyecto. Las actividades principales que se deben llevar a cabo para que se cumpla correctamente la fase de gestión de proyectos bajo un flujo normal son doce.

- a. Inicio del Proyecto
- b. Ingresar Ficha Control de Proyectos
- c. Generación de Plan de Proyecto
- d. Conformar y Preparar Equipo de Trabajo
- e. Ejecución y Control de Actividades
- f. Gestión de la Configuración
- g. Revisión de Estado de Avance
- h. Actualización de Plan del Proyecto
- i. Control Integrado de Cambios
- j. Notificar Aceptación/No Aceptación del Control
- k. Aprobación de Acta de Aceptación y Cierre
- l. Realización del Cierre

Se calcula el porcentaje de actividades cumplidas calculando con la siguiente fórmula:

$$(\text{Número de Actividades Ejecutadas} / \text{Número de Actividades Principales}) * 100$$

- Si el porcentaje de actividades realizadas es del 100%, se considera que se aplicó la metodología en su totalidad al proyecto.

4. **Porcentaje de Objetivos Logrados:** este indicador mantiene un control sobre la cantidad de objetivos cumplidos durante el desarrollo del proyecto. El número de objetivos varía de acuerdo al proyecto. Se obtiene este indicador mediante la siguiente fórmula:

$$(\text{Número de Objetivos Cumplidos} / \text{Número de Objetivos Definidos}) * 100$$

- Si el porcentaje de objetivos logrados es del 100%, se considera un desarrollo de proyecto exitoso.

5. **Porcentaje de Acierto en las Estimaciones de Tiempo:** este indicador busca medir el rendimiento de las horas hombre presupuestados al inicio del proyecto versus las horas hombre realmente invertidas a lo largo del desarrollo y finalización del proyecto. Para obtener el indicador se debe realizar el siguiente cálculo:

$$(\text{HH Reales Aplicados} / \text{HH Presupuestados}) * 100$$

- Si el porcentaje obtenido es $\leq 80\%$ se ha estimado una inversión de horas por trabajar muy altas, y el cobro al cliente podría haber sido excesivo, lo que generaría cierta desconfianza por parte de estos últimos en el momento de contratar un nuevo servicio de la empresa.

- Si el porcentaje obtenido es $> 80\%$ y $\leq 100\%$ se ha logrado una buena estimación de tiempo. Además se podría concluir que el jefe de proyecto armó un equipo de desarrollo con altas capacidades técnicas y el equipo de trabajo ha ejecutado sus tareas de manera eficiente.

- Si el porcentaje obtenido es $> 100\%$, el jefe de proyecto ha estimado el tiempo de trabajo a invertir de mala forma o puede que el equipo de trabajo no fue completamente

eficiente al ejecutar sus tareas. Se deberá realizar una evaluación de las posibilidades según cada proyecto.

7.3.2 Indicadores Ejecución

1. **Entregables por Actividades Ejecutadas:** lista de documentos entregables que se hayan generado por actividades de Ejecución a lo largo del proyecto. Durante el proyecto se deberá generar seis entregables en el transcurso de las actividades. Si se cumple la entrega de todos los documentos, se mantuvo una buena especificación y ejecución sobre el proyecto.

- a. Documento de Análisis y Levantamiento de Requerimientos
- b. Documento de Diseño General de la Solución
- c. Documento de Diseño Detallado de la Solución
- d. Documento de Diseño Plan de Pruebas
- e. Documento de Diseño Ficha de Pruebas
- f. Manual de Usuario

Tabla 7-2 Entregables por Actividades Gestión Ejecutadas

<=2	Mala especificación y ejecución del Proyecto (a, b)
>2,<=4	Especificación y ejecución Regular del Proyecto (a, b, e, f)
>=5,>=6	Muy buena especificación y ejecución del Proyecto (a, b, d, e, f o todas + versiones)

2. **Porcentaje de Actividades Ejecutadas:** este indicador mantiene un control sobre la cantidad de actividades de ejecución llevadas a cabo durante el proyecto. Las actividades principales que se deben llevar a cabo para que se cumpla correctamente la fase de ejecución de proyectos bajo un flujo normal son nueve.

- a. Análisis y Levantamiento de Requerimientos
- b. Ingresar Ficha Control de Proyectos
- c. Generación de Plan de Proyecto
- d. Conformar y Preparar Equipo de Trabajo
- e. Ejecución y Control de Actividades
- f. Gestión de la Configuración
- g. Revisión de Estado de Avance
- h. Actualización de Plan del Proyecto
- i. Control Integrado de Cambios
- j. Notificar Aceptación/No Aceptación del Control
- k. Aprobación de Acta de Aceptación y Cierre
- l. Realización del Cierre

Se calcula el porcentaje de actividades cumplidas calculando con la siguiente fórmula:

$$(\text{Número de Actividades Ejecutadas} / \text{Número de Actividades Principales}) * 100$$

- Si el porcentaje de actividades realizadas es del 100%, se considera que se aplicó la metodología en su totalidad al proyecto.

3. **Porcentaje de Tareas Ejecutadas:** este indicador mantiene un control sobre la cantidad de tareas llevadas a cabo por el equipo de trabajo durante el proyecto. Cada proyecto

tendrá su lista de tareas propias que se deberá ejecutar por el equipo para desarrollar el proyecto. Se calcula este indicador mediante la siguiente fórmula:

$$(\text{Tareas Ejecutadas} / \text{Tareas Totales Planificadas}) * 100$$

4. **Porcentaje de Módulos Funcionales:** este indicador mantiene un control sobre los módulos correctamente desarrollados durante la ejecución del proyecto. Si se alcanza un 100% de módulos funcionales sin errores, el desarrollo del proyecto se considera exitoso. Para calcular este porcentaje se aplica la fórmula:

$$(\text{Número de Módulos Correctamente Desarrollados} / \text{Número Total de Módulos Desarrollados}) * 100$$

5. **Porcentaje de Módulos Integrados:** este indicador mantiene un control sobre aquellos módulos que se deberán integrar con otros módulos desarrollados durante la ejecución del proyecto. Si se alcanza un 100% de integración, el desarrollo del proyecto se considera exitoso. Para calcular este porcentaje se aplica la fórmula:

$$(\text{Número de Módulos Integrados} / \text{Número Total de Módulos por Integrar}) * 100$$

6. **Grado de Eficiencia de Desarrollo y Ejecución:** este indicador busca medir el rendimiento de las horas de desarrollo propuestos al inicio del desarrollo del proyecto versus las horas de desarrollo realmente invertidas por los trabajadores a lo largo de la ejecución del proyecto. Para obtener el indicador se debe realizar el siguiente cálculo:

$$(\text{Horas Reales de Desarrollo Empleadas} / \text{Horas Estimadas de Desarrollo}) * 100$$

- Si el porcentaje obtenido es $\leq 60\%$ se ha estimado una inversión de horas por trabajar muy altas, y el cobro al cliente podría haber sido excesivo, lo que generaría cierta desconfianza por parte de los clientes en el momento de contratar un nuevo servicio de la empresa.

- Si el porcentaje obtenido es $> 60\%$ y $\leq 100\%$ se ha estimado una inversión de horas por trabajar regular, y el cobro al cliente podría haber sido aceptable, lo que generaría confianza por parte del cliente para volver a contratar servicios de la empresa. Además se podrá deducir que el equipo de trabajo realizó el desarrollo de la solución de manera eficiente, cumpliendo las tareas planificadas bajo los plazos estimados.

- Si el porcentaje obtenido es $> 100\%$ el jefe de proyecto ha estimado una inversión de horas por trabajar muy bajas, y el equipo de trabajo no ha podido cumplir con los plazos establecidos. Ambas situaciones deberán ser evaluadas por un jefe mayor para verificar las deficiencias según cada proyecto.

8 Implementación de Metodología de Desarrollo Propuesta

Luego de haber definido un documento interno (véase **Anexo F**) con la metodología de desarrollo, ésta se distribuyó para su estudio y su futura aplicación a los proyectos que se han de desarrollar.

Se realizó el seguimiento del proyecto llamado “Agenda Web”, la que propone generar una herramienta multi-navegador de Agendamiento de Horas Médicas, que permitirá a los pacientes y agentes del hospital realizar reservas de atención en línea. El proyecto fue dividido en cuatro fases de desarrollo de funcionalidades, por lo que no aplica estrictamente el método cascada, sino más bien, una forma de desarrollo iterativa incremental. Por cada fase desarrollado y probado se genera un módulo del sistema, la que pasará a producción en el ambiente del cliente y podrá ser utilizado por el usuario. Las iteraciones de las fases que siguen, serán para mejoras del módulo entregado en la fase anterior. Actualmente la primera fase está por finalizar la fase de “Ejecución y Control”.

El inicio del proyecto tuvo una falta de formalidad en la entrega de la Ficha de Proyecto que se debería haber generado en la primera actividad de la etapa de gestión de proyecto “Inicio del Proyecto”, donde el ejecutivo comercial debería haber generado la ficha físicamente y oficializar la entrega de ésta al área de desarrollo. Independiente de esto, la ficha fue ingresada correctamente al sistema gestión que posee la empresa, la que corresponde a la actividad “Ingresar Ficha Control de Proyectos” de la fase de Iniciación de la etapa de gestión de proyectos.

Una vez que se aprobó la iniciación del proyecto, el jefe de proyecto generó la planificación de proyecto mediante una carga Gantt (DPGP). Esto forma parte de la actividad “Generación de Plan de Proyecto”, la que una vez que fue aprobada, se conformó y preparó un equipo de trabajo. Luego, mediante reuniones que se sostuvieron con el cliente se fueron generando minutas de acuerdos, según lo detallado para la actividad “Ejecución y Control de Actividades”. Actualmente, se está generando una Ficha de Control de Cambios, bajo la petición del cliente, situación que pasa por la actividad “Control Integrado de Cambios” de la fase “Ejecución y Control” de la etapa de gestión de proyectos. Por otra parte, las revisiones de estado de avance se realizan a través de la carta Gantt, y no generando un documento de estado de avance proyecto como se definió según la actividad “Revisión de Estado de Avance” de la fase “Ejecución y Control”. Como las revisiones de avance se realizan mediante la Gantt, la actividad de “Actualización de Plan del Proyecto” se lleva a cabo.

En la etapa de ejecución de proyecto, según la actividad “Análisis y Levantamiento Requerimiento” de la fase de “Planificación”, se obtuvo como entregable el documento de Diseño de Análisis y Levantamiento de Requerimientos (DALR). Basado en este documento, no se pasó por la actividad de generación de diseño general, sino que se realizó un diseño detallado de la solución, actividad que tiene como entregable el Documento de Diseño Detallado de la Solución (DDDS). Una vez finalizado y aprobado el documento de diseño, se ejecutaron las actividades de “Implementación de la Solución”, donde el proyecto se encuentra actualmente, además de la “Implementación de Plataforma” donde se definió que la plataforma de testing se generaría por parte de la empresa Solem y la plataforma final será definida y generada por parte del cliente.

No se realizó la actividad de planificación de pruebas, sin embargo, el desarrollo de la primera fase pasó por un estricto control de calidad el cual generó una repetitiva iteración de pruebas generando varias versiones del documento de fichas de pruebas junto con sus evidencias. Una vez que la primera fase pasara el control de calidad, ésta fue instalada en producción, sin necesidad de pasar por la fase de capacitación. El flujo de actividades para la primera fase no requirió pasar por la puesta en marcha, debido al profundo nivel de pruebas que obtuvo, por lo que luego del paso a producción, se realizó el cierre de la primera fase del proyecto.

El inicio de la segunda fase partió por la actividad de definición de requerimientos, donde se describió las mejoras que se incorporarían a la solución dada en la primera fase. Esta segunda fase también pasó por un control de calidad estricto para su paso a producción. Se generaron las actualizaciones correspondientes de los diseños de solución, fichas de pruebas y documentos de despliegues. Para las modificaciones incorporadas, se generaron las fichas de control de cambios necesarias que mantuvieran documentados todos los cambios solicitados por el cliente. La segunda fase también tuvo un paso a producción, sin necesidad de pasar por la puesta en marcha por la iteración de pruebas realizadas. Con esto, se finalizó el desarrollo de la segunda fase. Actualmente el proyecto se encuentra en el desarrollo de una tercera fase, que también incorpora mejoras para el sistema entregado en las últimas dos fases. Se siguen las mismas actividades dadas para la primera y segunda fase, siempre considerando que el control de calidad es la actividad más iterada e importante para el paso a producción de los desarrollos.

8.1 Herramientas Utilizadas

La empresa cuenta con las herramientas necesarias para llevar a cabo la ejecución y la gestión del proyecto, las cuales simplifican el trabajo a ser realizado por el equipo durante las actividades. Se considera el caso del proyecto implementado que fue descrito en el punto anterior para describir algunas de las herramientas que fueron por el equipo de trabajo.

El equipo de trabajo utiliza la herramienta de control de versiones *Tortoise SVN*, una herramienta de control de versiones open source basada en un repositorio, que funciona de forma parecida a un sistema de ficheros, denominado Subversion (SVN). El SVN permite al equipo de trabajo crear, copiar y borrar carpetas con la misma flexibilidad con la que lo haría si estuviese en su disco duro local, y se puede acceder al repositorio a través de redes, lo que le permite ser usado por todo el equipo de trabajo que se encuentran en distintas computadoras en las oficinas de la organización. Esta herramienta se utiliza tanto para el ámbito de gestión de proyectos como para la ejecución de ésta, ya que mantiene las versiones de los documentos o código generado.

Tanto el equipo de gestión de proyectos y el de ejecución utiliza las herramientas ofrecidas por *Microsoft Office*, como el *Word*, *Excel* o *Project*, para realizar la documentación asociadas al proyecto. Para los documentos estándares definidos, como la ficha de proyecto, documento de requerimientos, diseños, manuales, etc., son generados con *Word*. La planificación del proyecto se realiza utilizando la herramienta de *Project*, definiendo los hitos a llevar a cabo y manteniendo actualizado los porcentajes de los avances.

Una herramienta utilizada para el modelado de workflow por parte del equipo de gestión de proyectos es el *Bizagi Modeler*, donde se debe diseñar los procesos involucrados en el modelo de negocio que se debe abarcar en el desarrollo de la solución. Además para el

modelado de procesos se utiliza la herramienta *Enterprise Architect*, para el diseño de casos de usos, diagramas de actividad, diagrama de procesos, entre otros. Para el diseño de propuestas de interfaces que se presentarán al cliente, se utilizan herramientas como *Pencil* o *Snagit*.

Para la implementación de la solución, según los requerimientos del sistema a desarrollar, existen varias herramientas utilizadas por el equipo de ejecución de proyecto. Algunas de las más utilizadas son: *Microsoft SQL Server*, *Windows Server System*, *Visual Studio*, *Eclipse*, *Oracle WebLogic*, *JBoss*, *Oracle SQL Developer*, *CentOS*, *Notepad++*, *AquaData Studio*, *VMWare Workstation*, entre otros más. Algunos lenguajes de desarrollo utilizados son *ASP.net*, *Visual Basic*, *C*, *Java*, *JavaScript*, *HTML5*, *XML*, etc. Según las herramientas mencionadas, algunas plataformas implementadas son: *Java*, *Linux*, *Microsoft.net*, *iOS*, *Android*.

La variedad de herramientas es amplia y el uso de alguno se basa en la solución a desarrollar, variando según el conocimiento del equipo de trabajo y de acuerdo a los proyectos a implementar.

9 Conclusión

En cualquier proyecto de desarrollo de sistemas, es necesario disponer de una gestión de proyecto eficaz para garantizar que ésta cumpla los objetivos y que se desarrolle dentro de un presupuesto rentable, es por esto que se define la gestión de proyectos como el proceso por el cual se planifica y controla el desarrollo de un sistema costeable dentro de un periodo de tiempo específico.

Si bien para el desarrollo de un sistema las herramientas y técnicas del análisis y del diseño de sistemas desempeñan un papel fundamental, estos métodos no son suficientes por sí mismos, ya que requieren de la administración por parte de un director de proyectos. Una buena gestión no puede garantizar el éxito del proyecto, sin embargo, la mala gestión de proyectos puede llevar al fracaso de éste aún si se ejecutara bajo los mejores métodos de análisis y diseño de proyectos. En una deficiente gestión de proyectos las necesidades del usuario no quedan satisfechas o no se identifican correctamente, no existe un control de cambios en el proyecto, los costes aumentan tanto en el desarrollo como en el mantenimiento del software y existen retrasos en las entregas. En un deficiente desarrollo de proyecto puede darse que los requerimientos no se cumplan en su totalidad, que exista una alta tasa de errores en código o un control de calidad de software ineficiente.

Es importante que una empresa dedicada a la ingeniería del software, como Ingeniería Solem, tenga metodologías definidas para dirigir y controlar los procesos que conllevan el desarrollo y mantenimiento de sus proyectos software. Esto debido principalmente a que el desarrollo de los que poseen un gran alcance, contiene muchas actividades y procedimientos que se deben ejecutar para cumplir con los objetivos que plantea, las cuales, si se llevan a cabo sin una dirección o método definido, podría conllevar al fracaso del proyecto. Otra buena manera de controlar el proceso de desarrollo de los proyectos software, es tener definidos indicadores, tanto de gestión como de ejecución de la misma, donde se pueda medir la eficiencia de la metodología a aplicar, controlando las actividades cumplidas, las horas trabajadas, el grado de satisfacción del cliente, los objetivos logrados y los módulos funcionales. Este tipo de indicadores aportará información necesaria para verificar el progreso del proyecto hacia el logro de su alcance.

Según lo señalado anteriormente, una metodología propia permite la planificación y estimación más acertada de los tiempos de desarrollo, ordenando el proceso que conlleva el desarrollo del proyecto software, además de tener claro lo que se está desarrollando tanto para los que se dedican a desarrollar como para los jefes de proyecto. Con la metodología se podría mantener un orden lógico de las cosas, como a su vez, se puede obtener el desarrollo de proyectos exitosos, logrando cumplir los objetivos del proyecto, mejorando la integración y adaptación de los distintos procesos que conllevan el desarrollo.

Es sumamente importante que todo el equipo participante del desarrollo del proyecto conozca los fundamentos de las metodologías, ya que con esto, se podría lograr un desarrollo más eficaz de los proyectos, pues los trabajadores podrán entender la forma de trabajo que se lleva a cabo para realizar un proyecto software de buenos resultados, además de tener claro cuál es el rol y el aporte que cada uno de ellos da en el desarrollo de un proyecto. La empresa debe preocuparse que aquellos que dirigen tanto la gestión como la ejecución del proyecto, conozcan la metodología completa y sus fundamentos, ya que una mala gestión de proyectos puede llevar al fracaso de éste.

La principal medida de distribución de la información dentro de la empresa considerada es la capacitación personalizada constante de la metodología mediante presentaciones a distintos grupos de trabajadores donde se muestren las definiciones, se describan las actividades a seguir, sus entregables y los roles asociados. Otra medida de distribución considera subir un documento manual para el personal en el sitio de la empresa, para que éstos tengan acceso a su lectura en el momento que sea.

El ciclo de vida software ofrece un marco de trabajo básico para llevar a cabo un proyecto que incluye procesos con actividades relacionadas al desarrollo y mantenimiento. Las actividades principales que forman parte de todos los modelos de procesos son la especificación, el desarrollo, la validación y la evolución del software. En todas las actividades en necesario el trabajo en equipo, un seguimiento y una gestión del ciclo de vida.

Actualmente se han desarrollado procesos efectivos de especificación, diseño e implementación del software, que reducen el esfuerzo para producir sistemas grandes y complejos bajo distintos modelos de desarrollo. Aunque en general, no existe un enfoque ideal para una metodología de creación de software, debido a la amplia diversidad de sistemas y organizaciones que usan estos sistemas, por lo que se necesita diversos enfoques y combinaciones de procesos para el desarrollo de software.

Además de requerir una buena gestión, los proyectos software deben estar basados en alguna metodología para producir el software. Estas metodologías organizadas de desarrollo incluyen sugerencias para el proceso que se debe seguir, la notación que se va a utilizar, los modelos del sistema que hay que desarrollar y las reglas que gobiernan estos modelos y las pautas de diseño.

En la implementación de la metodología propuesta se pudo observar que independiente de que no se apliquen todas las actividades propuestas para la ejecución y gestión del proyecto, se puede aún mantener un control sobre el avance de éste hacia el cumplimiento de los objetivos definidos inicialmente, ofreciendo así un flujo de actividades secuenciales con documentos de apoyo que brindan una base estable para un buen seguimiento por parte de los encargados del proyecto, para que éste último finalice exitosamente.

Una de las herramientas más importantes utilizadas por el equipo de trabajo es el control de versiones SVN, ya que al permitir que varios usuarios puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones, fomenta la colaboración entre ellos y dada esta flexibilidad, se aplican buenas prácticas que conducen a una correcta gestión de las versiones del software y documentación generada.

Como se pudo observar en el ejemplo de implementación de la metodología en un proyecto, no existe un estándar de herramientas utilizadas a nivel de empresa ya que existe una amplia gama de éstas ofrecidas por el mercado Sin embargo existe una definición de las herramientas a utilizar según el proyecto a desarrollar por el equipo, teniendo en cuenta que la selección de éstas puede darse según la preferencia de algún trabajador, el cual puede sugerir el uso de alguna herramienta en particular al equipo de desarrollo. Cabe señalar que esta selección de herramientas depende de la necesidad requerida por la solución.

10 Referencias

[1] Project Management Institute, Inc., Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK®), Cuarta Edición, Junio 2008.

[2] Aubry Monique, Hobbs Brian, Thuillier Denis, A new framework for understanding organisational Project management through the PMO. International Journal of Project Management 25 (2007) 328-336.

[3] Wysocki Robert K., Effective Project Management: Traditional, Agile, Extreme, Fifth edition. Wiley Publishing, Inc. 2009.

[4] Sommerville Ian, Ingeniería del Software, Séptima Edición. Pearson Education. 2005

[5] Cataldi Zulma, Metodología de Diseño, Desarrollo y Evaluación de Software Educativo. Tesis de Magíster en Informática. Facultad de Informática. Universidad Nacional de La Plata. 2000

[6] Comisión de Reglamentos Técnicos y comerciales – INDECOPI. Tecnología de la Información: Procesos del Ciclo de Vida del Software, Segunda Edición. Norma Técnica Peruana NTP-ISO/IEC 12207, 2006.

[7] Pressman Roger S., Ingeniería del Software: Un enfoque Práctico, Sexta Edición. McGraw-Hill, NY, 2005.

[8] Szypersky, C. Component Software. Beyond Object-Oriented Programming. Addison-Wesley, 1998.

[9] ISO/IEC 14764:2006 Software Engineering — Software Life Cycle Processes — Maintenance.

[10] Vara C. Marcela, Gestión de Proyectos de Desarrollo de Software, Departamento de ingeniería Informática y ciencias de la Computación, Facultad de Ingeniería, Universidad de Concepción. 2000.

[11] Procesos de Ingeniería de Software, Métricas de Proceso y Proyecto, Lorena León, Universidad Técnica Particular de Loja <http://www.slideshare.net/loreknelamorena/mtricas-de-proceso-y-proyecto-de-software>.

Anexos

A. Encuesta Metodología Solem

Objetivo: Evaluar el conocimiento acerca de la metodología aplicado actualmente para proyectos software. Mediante la misma se busca obtener información útil para ajustar la metodología según las necesidades reales de la empresa.

Muestra o Población: Personal del área de desarrollo de la empresa Ingeniería Solem.

Preguntas:

1. Indique la vertical a la cual usted pertenece:
 - Vertical Workflow y CRM (DSU)
 - Vertical Logística y ERP (AHP)
 - Vertical Medios de Pago Electrónico y Transaccional (JAS)
 - Vertical Biometría y Sistemas Móviles (VPO)
 - Vertical de Control de Calidad y Testing (AMG)
 - Vertical de Diseño y Documentación (HCV)
 - Vertical Grandes Proyectos (VNA)
 - Otro
2. ¿Qué grado de conocimiento cree usted que tiene acerca de la metodología de desarrollo Solem que se aplica para los proyectos software? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que lo desconoce y 5 que lo conoce completamente.
3. ¿Qué tan importante considera usted la necesidad de que la empresa posea una metodología propia de desarrollo de proyectos software? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
4. Según el grado de importancia seleccionado en la pregunta anterior, indique el porqué es o no es importante.
5. Si conoce la metodología y lo considera importante para la empresa, ¿cuál cree usted que es el grado de uso que se le da a la metodología en el desarrollo de proyectos? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no se ocupa nunca la metodología y 5 que se ocupa siempre.
6. ¿Cuál cree usted que sería un buen canal de distribución para difundir la metodología dentro de la empresa?
7. ¿Cree usted que las tareas que realiza diariamente aplican en la metodología de desarrollo? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no aplican nunca y 5 que siempre aplican.
8. Liste las tareas que usted considere que no aplican en la metodología.
9. ¿En qué aspecto de la metodología cree usted que aplican las actividades que realiza?
 - Gestión de Proyectos
 - Ejecución de Proyectos
 - Ambas
 - Ninguna
 - Desconozco
10. ¿Cree usted que es necesario que todos los involucrados en el proceso de desarrollo de proyectos software conozcan los fundamentos de la metodología Solem? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no necesario y 5 muy necesario.

11. ¿Cuál(es) de los siguientes modelos o metodologías se adecua más a su forma de trabajo actual en el desarrollo de un proyecto software?
- Modelo Cascada
 - Modelo Incremental
 - Modelo Iterativo
 - Modelo de Prototipos
 - Modelo V
 - Modelo Basado en Componentes
 - Modelo Espiral
 - Métodos Ágiles
 - Proceso Unificado Rational
 - Desconozco
 - Otros
12. ¿Cuál es el grado de importancia que usted le da a la ficha de proyecto?
13. ¿Cuál es el grado de utilidad que usted le da a la ficha de proyecto?
14. ¿Cuál es el grado de importancia que usted le da al documento plan general del proyecto (DPGP)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
15. ¿Cuál es el grado de utilidad que usted le da al documento plan general del proyecto (DPGP)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo ocupa nunca y 5 que lo ocupa siempre.
16. ¿Cuál es el grado de importancia que usted le da al documento de planificación del proyecto (DPDP)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
17. ¿Cuál es el grado de utilidad que usted le da al documento de planificación del proyecto (DPDP)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo ocupa nunca y 5 que lo ocupa siempre.
18. ¿Cuál es el grado de importancia que usted le da al documento de plan de riesgos (DPRI)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
19. ¿Cuál es el grado de utilidad que usted le da al documento de plan de riesgos (DPRI)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo ocupa nunca y 5 que lo ocupa siempre.
20. ¿Cuál es el grado de importancia que usted le da al documento de análisis y levantamiento de requerimientos (DALR)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
21. ¿Cuál es el grado de utilidad que usted le da al documento de análisis y levantamiento de requerimientos (DALR)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo ocupa nunca y 5 que lo ocupa siempre.
22. ¿Cuál es el grado de importancia que usted le da al documento de diseño general de la solución (DDGS)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
23. ¿Cuál es el grado de utilidad que usted le da al documento de diseño general de la solución (DDGS)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo ocupa nunca y 5 que lo ocupa siempre.

24. ¿Cuál es el grado de importancia que usted le da al documento de diseño plan de pruebas (DDPP)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
25. ¿Cuál es el grado de utilidad que usted le da al documento de diseño plan de pruebas (DDPP)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo ocupa nunca y 5 que lo ocupa siempre.
26. ¿Cuál es el grado de importancia que usted le da al documento de diseño ficha de pruebas (DDFP)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
27. ¿Cuál es el grado de utilidad que usted le da al documento de diseño ficha de pruebas (DDFP)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo ocupa nunca y 5 que lo ocupa siempre.
28. ¿Cuál es el grado de importancia que usted le da al certificado de cierre y aprobación (Acta de Entrega)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
29. ¿Cuál es el grado de utilidad que usted le da al certificado de cierre y aprobación (Acta de Entrega)? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo ocupa nunca y 5 que lo ocupa siempre.
30. ¿Cree usted que es necesario realizar un control de calidad sobre los sistemas desarrollados por la empresa? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que es completamente innecesario y 5 que es muy necesario.
31. ¿Cree usted que es importante que la empresa tenga un área de control de calidad independiente? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
32. ¿Usted utiliza el área de control de calidad de la empresa?
 - Si
 - No
33. Si su respuesta anterior fue No, indique de manera breve el(los) motivo(s) del porqué.
34. ¿Cree usted que es importante almacenar las versiones del desarrollo del proyecto en el SVN? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.
35. ¿Usted tiene claro cómo obtener una versión específica de un código o documento sobre el cual se está trabajando?
 - Si
 - No
36. Si su respuesta anterior fue Si, explique de forma breve cómo lo obtiene.
37. ¿Usted ha tenido problemas para identificar u obtener la última versión de algún documento o desarrollo de un proyecto?
 - Si
 - No
38. Según usted, ¿cuál es el grado de importancia de obtener un acta de entrega firmada por el cliente? Debiendo seleccionar de una escala del 1 al 5, siendo 1 no importante y 5 muy importante.

39. ¿Usted cree que su equipo realiza las actas formales de entrega al proyecto? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que no lo generan y 5 que lo generan siempre.
40. ¿Usted conoce el estado de avance real versus el esperado de su proyecto? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que nunca conoce el estado de avance y 5 que siempre conoce el estado de avance.
41. ¿Usted conoce cuál es el siguiente hito o entregable de su proyecto? Debiendo seleccionar de una escala del 1 al 5, siendo 1 que nunca sabe cuál es el siguiente hito o entregable y 5 que siempre sabe cuál es el siguiente hito o entregable.
42. Indique cuáles son los roles que ejecuta dentro de la empresa.
43. Para los roles previamente definidos, ¿cuáles son las tareas o actividades que realiza, o cree que debería realizar?
44. ¿Cómo mediría y controlaría el cumplimiento de sus tareas?
45. ¿Cómo mediría y controlaría el cumplimiento de las tareas de su equipo? Si corresponde.
46. ¿Cómo le gustaría que fuera se medido o evaluado su desempeño cada año?

B. Siglas Documentación

Sigla	Documento
CAC	Certificado de Aceptación y Cierre
DALR	Documento Análisis y Levantamiento de Requerimientos
DDDS	Documento Diseño Detallado de la Solución
DDFP	Documento Diseño Ficha de Pruebas
DDGS	Documento Diseño General de la Solución
DDIF	Documento Diseño Infraestructura Física
DDIT	Documento Diseño Infraestructura Tecnológica
DDMD	Documento Diseño Modelo de Datos
DDPC	Documento Diseño Plan Capacitación
DDPP	Documento Diseño Plan de Pruebas
DEAP	Documento Estado Avance Proyecto
DESM	Documento Estado de Situación de Mantenimiento
DESS	Documento Estado Situación de Soporte de Proyecto
DIAP	Documento Instalación Ambiente Producción
DIAT	Documento Implementación Ambiente Tecnológico
DIIF	Documento Implementación Infraestructura Física
DIIT	Documento Implementación Infraestructura Tecnológica
DISO	Documento Implementación de la Solución
DMAP	Documento de Minutas y Acuerdos del Proyecto
DMCT	Documento Material de Capacitación Técnica
DMCU	Documento Material de Capacitación Usuario
DMIC	Documento Manual de Instalación y Configuración
DMUS	Documento Manual de Usuario
DPDP	Documento Planificación del Proyecto
DPGP	Documento Plan General Proyecto
DPIP	Diseño Plan de Instalación y Puesta en Marcha
DPMD	Diseño Plan Migración de Datos

Sigla	Documento
DPRI	Diseño Plan de Riesgos
DTCO	Documento Especificación Técnica de Componentes
DPPM	Diseño Plan Puesta en Marcha
FCC	Ficha Control de Cambio
FM	Ficha de Mantenición
FP	Ficha Proyecto
MIAT	Manual Instalación Ambiente Tecnológico

C. Modelos de Proceso Software

Para resolver problemas reales de una industria, un ingeniero del software o un equipo de ingenieros deben incorporar una estrategia de desarrollo que acompañe al proceso, métodos y capas de herramientas. Esta estrategia se conoce como un modelo de proceso o paradigma de ingeniería del software. Un modelo de procesos del software es una descripción simplificada de un proceso del software que presenta una visión de ese proceso [4]. Los modelos pueden incluir actividades que son parte de los procesos y productos de software y el papel de las personas involucradas en la ingeniería del software. Selecciona un modelo de proceso para la ingeniería del software según la naturaleza del proyecto y de la aplicación, los métodos y las herramientas a utilizarse.

Los modelos se propusieron originalmente para ordenar el caos del desarrollo de software, proporcionando estructuras útiles para el trabajo en la ingeniería del software y caminos efectivos para los equipos de software. Son importantes ya que proporciona estabilidad, control y organización a una actividad que si no se controla puede volverse caótica. Todos los modelos de proceso se pueden adaptar para usarlos de forma efectiva y en un proyecto de software específico.

El proceso conduce a un equipo de software a través de un conjunto de actividades del marco de trabajo que se organizan en un flujo de proceso, el cual puede ser lineal, incremental o evolutivo. A continuación se describen los modelos tradicionales mayormente conocidos.

Modelo Cascada

La versión original del modelo en cascada, o también conocido como el ciclo de vida clásico, fue presentada por Winston Royce en 1970, aunque son más conocidos los refinamientos realizados por Barry Boehm (1981), Ian Sommerville (1985) y Kevin Sigwart (1990)[5].

El modelo sugiere un enfoque sistemático, secuencial hacia el desarrollo del software [7]. Este modelo considera las actividades anteriores y las representa como fases de procesos separados. Aquí, el producto evoluciona a través de una secuencia de fases ordenadas en forma lineal, permitiendo iteraciones al estado anterior. Las etapas del modelo cascada se señalan en la Figura C.1.

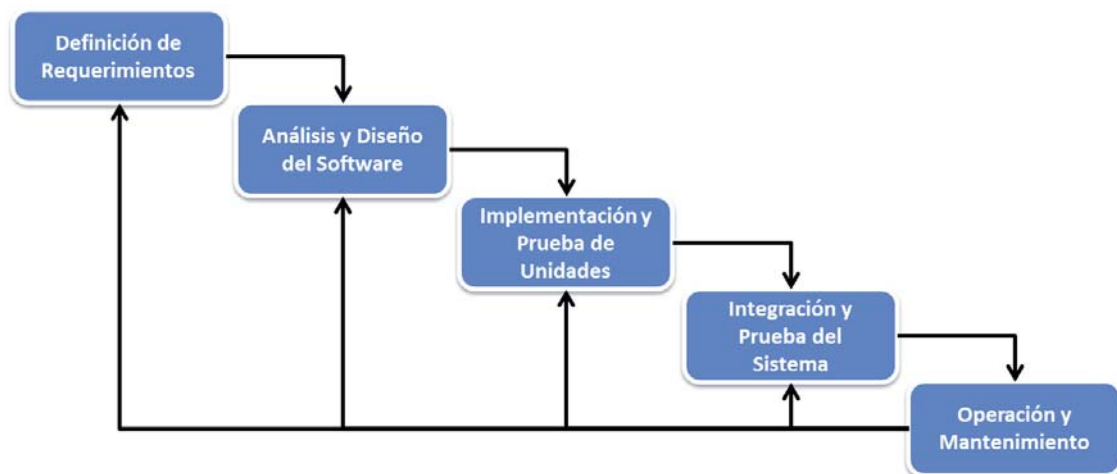


Figura C.1 Modelo Cascada

El número de etapas puede variar, pero en general suelen ser:

- *Definición de Requerimientos*: se analizan las necesidades de los usuarios finales del software para determinar los objetivos que se deben cubrir. Se especifica lo que debe hacer el sistema sin entrar en detalles. Se debe acordar con el cliente todo lo que se requiere del sistema.
- *Análisis y Diseño del Software*: se descompone y se organiza el sistema en elementos que puedan elaborarse por separado. Se describe la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes además de la manera en que éstas se combinan una con otras. Se debe distinguir el diseño de alto nivel o arquitectónico y el diseño detallado, esto es, el primero tiene como objetivo definir la estructura de la solución identificando los grandes módulos funcionales que van a estar asociadas y sus relaciones definiendo la arquitectura de la solución y el segundo define los algoritmos empleados y la organización del código para comenzar la implementación.
- *Implementación y Prueba de Unidades*: Se realizan los algoritmos necesarios para el cumplimiento de los requerimientos del usuario. Se implementa el código fuente haciendo prototipos de pruebas y ensayos para corregir errores.
- *Integración y Prueba del Sistema*: Se integran los componentes unitarios para componer el sistema y se comprueba que funcione correctamente y que cumplan con los requisitos. Se realizan pruebas al sistema para comprobar que el sistema no falle, detectar errores y asegurar que la entrada definida produzca las salidas reales de acuerdo con los resultados requeridos.
- *Operación y Mantenimiento*: El usuario final ejecuta el sistema. El software sufrirá cambios después de la entrega al cliente. El software debe adaptarse a cambios del entorno externo o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

Generalmente en proyectos reales rara vez se sigue una linealidad tal, y casi siempre hay iteraciones que van más allá de la etapa anterior. Una comparación entre las ventajas y desventajas se presentan en la Tabla C.1:

Tabla C.1 Ventajas y Desventajas del Modelo Cascada

Ventajas	Desventajas
<ul style="list-style-type: none"> ▪ La planificación es sencilla. ▪ Modelo conocido y utilizado con frecuencia. ▪ Modelo fácil de implementar y entender. ▪ Orientado a documentos. ▪ La calidad del producto resultante es alta. ▪ Realiza un buen funcionamiento en equipos débiles y productos maduros. ▪ Requiere menos capital y herramientas para hacerlo funcionar de manera óptima. ▪ Promueve una metodología de trabajo efectiva: Definir antes de diseñar, diseñar antes de codificar. 	<ul style="list-style-type: none"> ▪ No refleja realmente el proceso de desarrollo del software. ▪ Un proyecto rara vez sigue la secuencia lineal que propone el modelo. El modelo incluye iteraciones, pero de manera indirecta. ▪ El modelo cascada requiere que todos los requisitos se encuentren establecidos de manera explícita, lo que es difícil para el cliente. ▪ El proceso de creación de software tarda mucho tiempo ya que debe pasar por el proceso de prueba y hasta que el software no esté completo no se opera. ▪ El mantenimiento se realiza en el código

	<p>fuelle.</p> <ul style="list-style-type: none"> ▪ Cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando costos de desarrollo. ▪ Las revisiones de proyectos de gran complejidad son muy difíciles. ▪ Una etapa determinada del proyecto no se puede llevar a cabo a menos que se haya finalizado la etapa anterior.
--	--

La naturaleza lineal del modelo conduce a “estados de bloqueo” en los cuales algunos miembros del equipo deben esperar a otros para terminar tareas dependientes [7].

1. Modelo Incremental

El modelo incremental fue sugerido por Harlan Mills en 1980 como una forma de reducir la repetición del trabajo en el proceso de desarrollo y dar oportunidad de retrasar la toma de decisiones en los requisitos hasta adquirir experiencia con el sistema.

Este modelo combina elementos del modelo en cascada aplicado en forma iterativa, en la Figura C.2 se puede apreciar la estructura del modelo incremental [7]. El modelo aplica un proceso de desarrollo incremental, donde los clientes identifican, a grandes rasgos, los servicios que proporcionará el sistema. Identifican qué servicios son más importantes y cuáles menos. Entonces, se definen varios incrementos en donde cada uno proporciona un subconjunto de la funcionalidad del sistema [4].

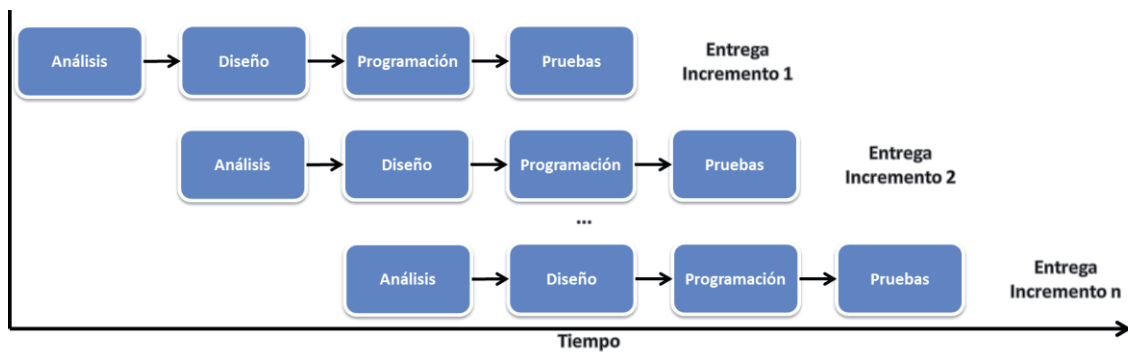


Figura C.2 Modelo Incremental

El primer incremento es un producto esencial. Es decir, se incorporan los requisitos básicos, pero muchas características suplementarias (algunas conocidas, otras no) no se incorporan. El producto esencial se entrega al cliente o se somete a una evaluación detallada. Como resultado de la evaluación se desarrolla un plan para el siguiente incremento, la que abarca la modificación del producto esencial con el fin de satisfacer de mejor manera las necesidades del cliente, entregando características y funcionalidades adicionales. Este proceso se repite después de la entrega de cada incremento mientras no se haya elaborado el producto completo. La Tabla C.2 muestra una comparación entre las ventajas y desventajas del modelo incremental.

El modelo incremental se enfoca en la entrega de un producto operacional con cada incremento. Los primeros incrementos son versiones “incompletas” del producto final, pero proporciona al usuario la funcionalidad que necesita y una plataforma para evaluarlo. Los clientes pueden experimentar con el sistema, lo cual les ayuda a clarificar sus requerimientos para los incrementos posteriores y para las últimas versiones del incremento actual. Tan pronto como se completan los nuevos incrementos, se integran en los existentes de tal forma que la funcionalidad del sistema mejora con cada incremento entregado.

Tabla C.2 Ventajas y Desventajas del Modelo Incremental

Ventajas	Desventajas
<ul style="list-style-type: none"> ▪ Los clientes no tienen que esperar hasta que el sistema completo se entregue. ▪ Se reduce el tiempo de desarrollo inicial, ya que se implementa la funcionalidad parcial ▪ El primer incremento satisface los requerimientos más críticos. ▪ Los clientes pueden utilizar los incrementos iniciales como prototipos y obtener experiencia sobre los requerimientos de los incrementos posteriores. ▪ Resulta más sencillo acomodar cambios al acotar el tamaño de los incrementos. ▪ Existe un bajo riesgo de un fallo total del proyecto. ▪ Los incrementos posteriores se integran a los servicios de más alta prioridad. ▪ Es menos probable que los clientes encuentren fallos de funcionamiento del software en las partes más importantes del sistema, ya que se realizan más pruebas a los servicios más importantes. ▪ Requiere de una planeación cuidadosa tanto a nivel administrativo como técnico. 	<ul style="list-style-type: none"> ▪ Los incrementos deben ser relativamente pequeños (no más de 20.000 líneas de código), cada uno debe entregar una funcionalidad del sistema. ▪ Es difícil adaptar los requerimientos del cliente a incrementos de tamaño apropiado. ▪ Muchos sistemas requieren un conjunto de recursos que se utilizan en diferentes partes del sistema. ▪ Debido a que los requerimientos no se definen en detalle hasta que un incremento se implementa, puede ser difícil identificar los recursos comunes que requieren todos los incrementos. ▪ No es recomendable para sistemas de tiempo real, de alto nivel de seguridad, de procesamiento distribuido y/o de alto índice de riesgos. ▪ Requiere mucha planeación, tanto administrativa como técnica. ▪ Requiere de metas claras para conocer el estado del proyecto.

El desarrollo incremental es útil sobre todo cuando el personal necesario para una implementación completa no está disponible. Los primeros incrementos se pueden implementar con menos gente. Si el producto esencial es bien recibido se agrega más personal para implementar el incremento siguiente. Los incrementos se pueden planear para manejar los riesgos técnicos.

2. Modelo V

El Modelo V, representado en la Figura C.3, es una representación gráfica del ciclo de vida del desarrollo de sistemas. En él se resumen las principales medidas que deben adoptarse

en relación con las prestaciones correspondientes en el marco del sistema informático de validación.

Es un proceso que representa la secuencia de pasos en el desarrollo del ciclo de vida de un proyecto. Se describen las actividades y resultados que deben producirse durante el desarrollo del producto. El lado izquierdo de la V representa la descomposición de las necesidades, y la creación de las especificaciones del sistema. El lado derecho de la V representa la integración de las piezas y su verificación. V significa «Verificación y validación». Es muy similar al modelo de la cascada clásico ya que es muy rígido y contiene una gran cantidad de iteraciones.

La versión Método-V XT se terminó en febrero del 2005. No es comparable con CMMI. Mientras que CMMI solo describe «qué» se ha hecho, el Método-V XT describe el «cómo» y el «cuándo» y «quién» es el responsable de haberlo hecho.

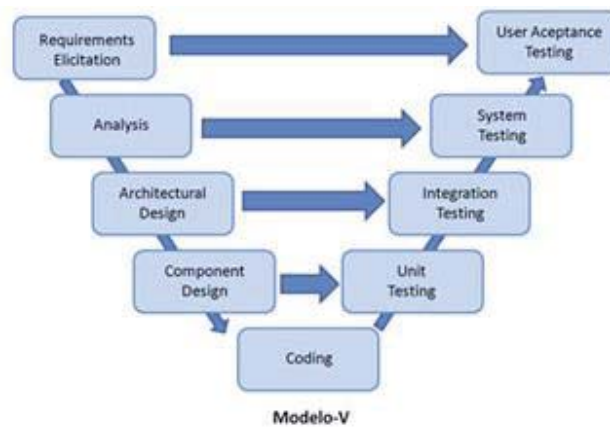


Figura C.3 Modelo V

La corriente de especificación (parte izquierda, Project definition) consiste principalmente de:

- Conceptos de operaciones: que debe hacer el sistema a grandes rasgos.
- Requisitos del sistema y arquitectura del mismo.
- Diseño detallado.

La corriente de pruebas (parte derecha, Project test and integration), por su parte, suele consistir de:

- Integración de las distintas partes, prueba y verificación de las mismas.
- Verificación y validación del sistema en conjunto.
- Mantenimiento del sistema.

La corriente de desarrollo puede consistir (depende del tipo de sistema y del alcance del desarrollo) en personalización, configuración o codificación.

El Método-V fue desarrollado para regular el proceso de desarrollo de software por la Administración Federal Alemana. Describe las actividades y los resultados que se producen durante el desarrollo del software. Proporciona una guía para la planificación y realización de proyectos. Los siguientes objetivos están destinados a ser alcanzados durante la ejecución del proyecto:

- Minimización de los riesgos del proyecto: Mejora la transparencia del proyecto y control del proyecto, especificando los enfoques estandarizados, describe los resultados correspondientes y funciones de responsabilidad. Permite una detección temprana de las desviaciones y los riesgos y mejora la gestión de procesos, reduciendo así los riesgos del proyecto.

- Mejora y Garantía de Calidad: Como un modelo de proceso estándar, asegura que los resultados que se proporcionan sean completos y contengan la calidad deseada. Los resultados provisionales definidos se puede comprobar en una fase temprana. La uniformidad en el contenido del producto mejora la legibilidad, comprensibilidad y verificabilidad.

- Reducción de los gastos totales durante todo el proyecto y sistema de Ciclo de Vida: El esfuerzo para el desarrollo, producción, operación y mantenimiento de un sistema puede ser calculado, estimado y controlado de manera transparente mediante la aplicación de un modelo de procesos estandarizados. Reduciendo la dependencia en los proveedores y el esfuerzo para las siguientes actividades y proyectos.

- Mejora de la comunicación entre todos los inversionistas: La descripción estandarizada y uniforme de todos los elementos pertinentes y términos es la base para la comprensión mutua entre todos los inversionistas. De este modo, se reduce la pérdida por fricción entre el usuario, comprador, proveedor y desarrollador.

Desventajas:

- El riesgo es mayor que el de otros modelos, pues en lugar de hacer pruebas de aceptación al final de cada etapa, las pruebas comienzan a efectuarse luego de haber terminado la implementación, lo que puede traer como consecuencia un “roll-back” de todo un proceso que costó tiempo y dinero.

- El modelo no contempla la posibilidad de retornar a etapas inmediatamente anteriores, cosa que en la realidad puede ocurrir.

- Se toma toda la complejidad del problema una vez y no en iteraciones o ciclos de desarrollo, lo que disminuye el riesgo.

3. Proceso Unificado

El Proceso Unificado de Rational (RUP), señalado en la Figura C.4 es un modelo de proceso híbrido, que reúne elementos de todos los modelos de procesos genéricos (cascada, evolutivo y basado en componentes), iteraciones de apoyo e ilustra las buenas prácticas en la especificación y el diseño [4]. El RUP reconoce que los modelos de procesos genéricos presentan un sólo enfoque del proceso. En contraste, el RUP se describe normalmente desde una perspectiva dinámica, que muestra las fases del modelo sobre el tiempo; una perspectiva estática, que muestra las actividades del proceso que se representan; y una perspectiva práctica, que sugiere buenas prácticas a utilizar durante el proceso.

El RUP es un modelo en fases que identifica cuatro fases diferentes en el proceso del software. A diferencia del modelo en cascada donde las fases se equiparan con las actividades del proceso, las fases en el RUP están mucho más relacionadas con asuntos de negocio más que técnicos. Las fases son:

- i. Inicio: donde se establece un caso de negocio para el sistema. Se deben identificar todas las entidades externas (personas y sistemas) que interactuarán con el sistema y definir estas interacciones. Esta información se utiliza para evaluar el aporte del sistema al negocio. Si

el aporte es de poca importancia, se puede cancelar el proyecto después de esta fase. Las salidas obtenidas de esta fase son: Documento de la visión; Modelo inicial del caso de uso; Glosario inicial del proyecto; Caso inicial del negocio; Evaluación inicial del riesgo; Plan de proyecto, fases e iteraciones; Modelo del negocio; Uno o más prototipos.

ii. Elaboración: los objetivos de esta fase son desarrollar una comprensión del dominio del problema, establecer un marco de trabajo arquitectónico para el sistema, desarrollar el plan del proyecto e identificar los riesgos clave del proyecto. Al terminar esta fase, se debe tener un modelo de los requerimientos del sistema (casos de uso UML), una descripción arquitectónica y un plan de desarrollo del software. Las salidas asociadas a esta fase son: Modelo de casos de uso; Requisitos suplementarios, se incluyen los no funcionales; Modelo de análisis; Descripción de la arquitectura del software; Prototipo arquitectónico ejecutable; Modelo de diseño preliminar; Lista revisada de riesgos; Plan de proyectos que incluye plan de iteración, flujos de trabajo adaptados, fundamentos, productos técnicos del trabajo, manual preliminar del usuario.

iii. Construcción: comprende el diseño del sistema, la programación y las pruebas. Se desarrollan e integran las partes del sistema. Al terminar esta fase, se debe tener un sistema software operativo y la documentación correspondiente lista para entregarla a los usuarios. Esta fase tiene como entrega: Modelo del diseño; Componentes del software; Incremento integrado del software; Plan y procedimiento de pruebas; Casos de prueba; Documentación del soporte, manuales del usuario, manuales de instalación, descripción del incremento actual.

iv. Transición: se ocupa de mover el sistema desde la comunidad de desarrollo a la comunidad del usuario y hacerlo trabajar en un entorno real. Esto se deja de lado en la mayor parte de los modelos de procesos del software, pero es, en realidad, una actividad de alto costo y a veces problemática. Al terminar esta fase, se debe tener un sistema software documentado que funciona correctamente en su entorno operativo. Esta fase tiene como entrega: Incremento de software integrado; Reportes de las pruebas beta; Retroalimentación general del usuario.

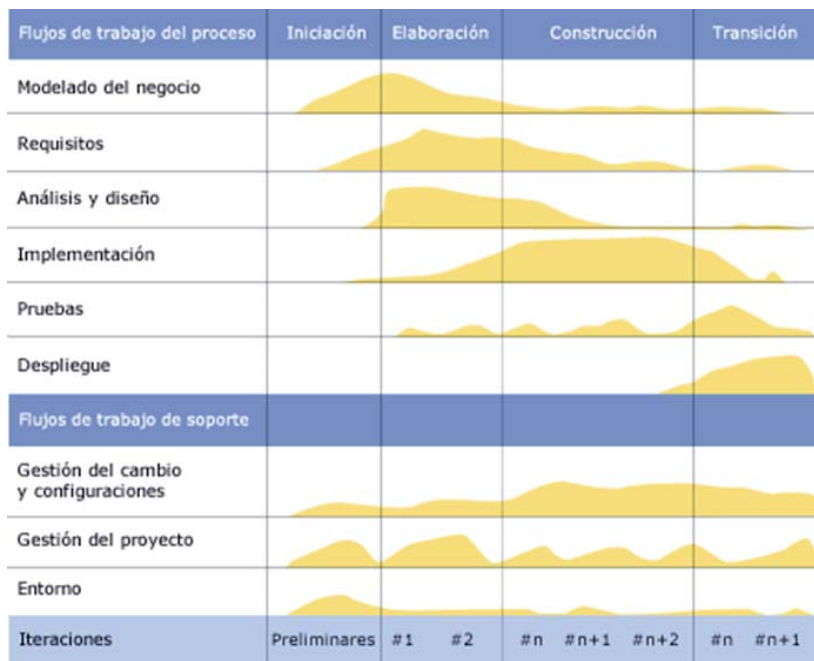


Figura C.4 Proceso Unificado

Cada fase se puede representar de un modo iterativo con los resultados desarrollados incrementalmente. Además, dentro del conjunto de fases puede también representarse de forma incremental. La vista estática del RUP se centra en las actividades que tienen lugar durante el proceso de desarrollo. Éstas se denominan flujos de trabajo. Existen seis principales flujos de trabajo del proceso identificados en el proceso y tres principales flujos de trabajo de soporte [4]. La Tabla C.3 señala los flujos de trabajo principales.

Tabla C.3 Flujos de Trabajo Principales RUP

Flujos de Trabajo del Proceso	Actividades
<ul style="list-style-type: none"> ▪ Modelo del Negocio 	<ul style="list-style-type: none"> ▪ Los procesos del negocio se modelan utilizando casos de uso de negocio.
<ul style="list-style-type: none"> ▪ Requerimientos 	<ul style="list-style-type: none"> ▪ Se definen los actores que interactúan con el sistema y se desarrollan casos de uso para modelar los requerimientos del sistema.
<ul style="list-style-type: none"> ▪ Análisis y diseño 	<ul style="list-style-type: none"> ▪ Se crea y documenta un modelo del diseño utilizando modelos arquitectónicos, modelos de componentes, modelos de objetos y modelos de secuencias.
<ul style="list-style-type: none"> ▪ Implementación 	<ul style="list-style-type: none"> ▪ Se implementan y estructuran en subsistemas los componentes del sistema. La generación automática de código de los modelos del diseño ayuda a acelerar este proceso.
<ul style="list-style-type: none"> ▪ Pruebas 	<ul style="list-style-type: none"> ▪ Las pruebas son un proceso iterativo que se llevan a cabo conjuntamente con la implementación. A la finalización de la implementación tienen lugar las pruebas del sistema.
<ul style="list-style-type: none"> ▪ Despliegue 	<ul style="list-style-type: none"> ▪ Se crea un lanzamiento del producto, se distribuye a los usuarios y se instala en su lugar de trabajo.
Flujos de Trabajo de Soporte	Actividades
<ul style="list-style-type: none"> ▪ Configuración y cambios de gestión 	<ul style="list-style-type: none"> ▪ Este flujo de trabajo de soporte gestiona los cambios del sistema.
<ul style="list-style-type: none"> ▪ Gestión del proyecto 	<ul style="list-style-type: none"> ▪ Este flujo de trabajo de soporte gestiona el desarrollo del sistema.
<ul style="list-style-type: none"> ▪ Entorno 	<ul style="list-style-type: none"> ▪ Este flujo de trabajo se refiere a hacer herramientas software apropiados disponibles para los equipos de desarrollo de software

El modelo RUP recomienda seis buenas prácticas fundamentales:

i. Desarrolle el software de forma iterativa: planifique incremento del sistema basado en las prioridades del usuario y desarrollo y entregue las características del sistema de más alta prioridad al inicio del proceso de desarrollo.

ii. Gestione los requerimientos: documente explícitamente los requerimientos del cliente y manténgase al tanto de los cambios de estos requerimientos. Analice el impacto de los cambios en el sistema antes de aceptarlos.

iii. Utilice arquitecturas basadas en componentes: estructure la arquitectura del sistema en componentes.

iv. Modele el software visualmente: utilice modelos gráficos UML para representar vistas estáticas y dinámicas del software.

v. Verifique la calidad del software: asegure que el software cumple los estándares de calidad organizacionales.

vi. Controle los cambios del software: gestione los cambios del software usando un sistema de gestión de cambios y procedimientos y herramientas de configuraciones.

Los beneficios en aplicar el proceso RUP es que está basada en las mejores prácticas de las demás metodologías, pudiendo reducir los riesgos del proyecto debido a su enfoque de análisis de riesgo tanto en el desarrollo como en los cambios, incorpora fielmente el objetivo de calidad e integra el desarrollo con mantenimiento. Sin embargo, el modelo presenta algunos inconvenientes como por ejemplo que busca prever y tener todo el control desde el inicio, lo que en situaciones de proyectos reales es casi imposible, el modelo genera un trabajo adicional y costos elevados y no es recomendable su aplicación a proyectos pequeños.

D. Procesos de Gestión de Proyectos

Un proceso es un conjunto de acciones y actividades interrelacionadas realizadas para obtener un producto, resultado o servicio definido [1]. Cada proceso se caracteriza por sus entradas, herramientas y técnicas que puedan aplicarse y por las salidas que se obtiene. Cualquier modelo de ciclo de vida de gestión de proyecto debe contener los procesos mostrados en la Figura D.1:



Figura D.1 Procesos del Ciclo de Vida de la Gestión de Proyectos

i. *Iniciación*: procesos realizados para definir un nuevo proyecto o una nueva fase de un proyecto ya existente, mediante la obtención de autorización para comenzar dicho proyecto o fase. No se incluyen procesos relacionados a realizar el trabajo del proyecto. Incluye la definición de los criterios de éxito del negocio que serán las métricas utilizadas para responder la pregunta ¿Cómo vas a saber que lo hiciste? Incluye todos los procesos asociados a responder la pregunta ¿Qué necesitas hacer? Incluye actividades como:

- Contratando un director de proyecto
- Obtener las necesidades reales del cliente
- Documentación de las necesidades del cliente
- Negociación con el cliente sobre cómo se van a satisfacer las necesidades.
- Redacción de una descripción del proyecto
- Obtener autorización de la dirección para planificar el proyecto

ii. *Planificación*: procesos requeridos para establecer el alcance del proyecto, refinar los objetivos y definir el curso de acción necesario para alcanzar los objetivos cuyo logro se emprendió el proyecto. Incluye los procesos asociados a responder la pregunta ¿Cómo lo vas a hacer? Algunas actividades son:

- Definición del trabajo del proyecto
- Estimar cuánto tiempo se invertirá en completar el trabajo
- Estimar los recursos necesarios para completar el trabajo
- Estimar el costo total del trabajo
- Ordenar/secuenciar el trabajo
- Construir la planificación inicial del proyecto
- Analizar y ajustar la planificación del proyecto
- Redactar un plan de administración de riesgos
- Documentar el plan del proyecto
- Obtener autorización de la dirección para poner en marcha el proyecto

iii. *Ejecución*: procesos realizados para completar el trabajo definido en el plan para la gestión de proyectos a fin de cumplir con las especificaciones del mismo. Incluye procesos relacionados al contrato y organización del equipo y estableciendo las reglas de operación del equipo. También incluye procesos relacionados a la inicio del trabajo del proyecto. Algunas actividades pueden ser:

- Contratación del equipo de proyecto
- Redacción de una descripción del proyecto
- Establecer las reglas de operación del equipo
- Establecer el alcance del proceso de gestión de cambios
- Gestión de las comunicaciones del equipo
- Finalizar la planificación del proyecto
- Redactar paquetes de trabajo

i. *Seguimiento y Control*: procesos requeridos para dar seguimiento, analizar y regular el proceso y el desempeño del proyecto, para identificar áreas en las que el plan requiera cambios y para iniciar los cambios correspondientes. Aquí es donde se realiza el trabajo real del proyecto. Ocupa al director de proyecto en actividades internas y externas al equipo de proyecto y trata con el cliente, el patrocinador y el director superior. Incluye procesos relacionados al trabajo en curso del proyecto. Las actividades son:

- Establecer informes rendimiento y presentación del proyecto
- Monitoreo del rendimiento del proyecto
- Monitoreo del riesgo
- Informes de estado del proyecto
- Procesar solicitudes de cambios de alcance
- Encontrar y solucionar problemas

ii. *Cierre*: procesos realizados para finalizar todas las actividades a través de todos los grupos de procesos, a fin de cerrar formalmente el proyecto o una fase del mismo. Procesos relacionados a la terminación del proyecto. Incluye procesos relacionados a la terminación del proyecto, incluyendo a las respuestas de las preguntas asociadas a ¿Qué tan bien lo hiciste? Las actividades son las siguientes:

- Obtener la aprobación del cliente de haber cumplido los requerimientos del proyecto
- Planificación e instalación de los entregables
- Redacción de un reporte final del proyecto
- Realizar la auditoría posterior a la implementación

Estos son los procesos claves para el ciclo de vida de un proyecto. En el caso más simple cada uno finalizará una vez en la secuencia descrita. En situaciones más complejas, algunos o todos los procesos serán repetidos varias veces.

Por lo tanto a grandes rasgos la gestión de proyectos abarca identificar requisitos; establecer objetivos claros y posibles de realizar; equilibrar los costos, plazos, alcances y calidad; y adaptar las especificaciones, estrategias, planes y enfoque a las inquietudes de los diferentes interesados.

E. Fases del Ciclo de Vida Principal de los Proyectos

A continuación se detallan las diferentes fases:

- La fase de *Inicio*, se encuentra dentro de la etapa gestión de proyectos y además está asociado al proceso de *Iniciación* de la gestión de proyectos propuesto por el PMI Institute. Contempla las primeras actividades al iniciar el proyecto donde se elabora y acuerda el acta de constitución del proyecto (ficha de proyecto), el cual autoriza formalmente la primera fase y se documentan los requisitos iniciales que satisfacen las expectativas de los interesados.
- La fase de *Mantenimiento y Control*, que dentro de la gestión de proyectos ofrecida por el PMI se señala como *Seguimiento y Control*, pero que en este caso se considera desde el Inicio al Cierre, contiene actividades transversales, donde se supervisa y analiza el progreso y el desempeño del proyecto, controlando en todo momento que vaya de acuerdo a lo planificado y dentro del alcance definido, de forma que se puedan detectar anomalías para hacer acciones correctivas en el momento que corresponda. Aquí se coordina y se supervisa el proyecto mediante reuniones que se tendrán durante todo el trabajo del proyecto entre los jefes del mismo. Estas reuniones tienen como finalidad que se conversen los distintos acuerdos entre las partes, se llegue a consenso y se aprueben los informes entregados durante el transcurso del mismo, aclarando así de mejor forma las observaciones que puedan aparecer desde las distintas partes interesadas.
- La fase de *Análisis* (inserta dentro de las fases *Planificación, Ejecución y Seguimiento y Control* del PMI), comprende el análisis de la situación actual del(os) procesos a soportar por la solución a diseñar. Durante esta fase se analiza la especificación de requerimientos, se analizan las alternativas y se establece una propuesta de arquitectura. Además se toman las definiciones claves del negocio, lo que afectará en la decisión de la solución a diseñar e implementar. Basado en todo esto, se genera un plan inicial. El objetivo de esta fase, es identificar la mejor arquitectura que satisfaga los requerimientos y seleccionar la más adecuada para el proyecto, como también documentar los requerimientos levantados y preparar los planes iniciales del mismo. La solución debe contemplar la estrategia, organización, riesgos, contingencias, recursos y tareas, por lo menos para la fase de diseño. Las actividades de esta fase son el análisis y levantamiento de requerimientos; revisión y ajustes de informes; aceptación de informe de análisis y levantamiento de requerimientos.
- En la fase de *Diseño* (inserta dentro del proceso de *Ejecución*), el equipo de proyecto define la solución, tanto funcional como física (cuando corresponda) que va a satisfacer los requerimientos de negocio. Además el equipo planifica cómo se va a implementar y aceptar la solución, el soporte y recibe las consideraciones a la solución planteada. Aquí se documentan formalmente los acuerdos, en términos de una comprensión detallada de la solución requerida y se aplican disciplinas como control de cambios y administración del riesgo. Los objetivos de esta fase son identificar las diversas soluciones que satisfagan los requerimientos y seleccionar la mejor, seleccionar y documentar la arquitectura lógica de la solución (así como también la arquitectura física si es que aplica), documentar la solución funcional propuesta, actualizar los planes de proyecto y

calidad para finalmente especificar los planes de pruebas y especificar un diseño de solución.

- En la fase de *Implementación* (también inserta dentro del proceso de *Ejecución*), cada componente de la solución identificado en la especificación de diseño de sistema, se desarrolla, se prueba y se integra con los otros componentes relacionados. Estos componentes se prueban como un todo. Es esencial que el equipo aplique los principios de control de cambios, administración de la configuración y reportes. Teniendo en cuenta que esta fase involucra a muchas personas trabajando simultáneamente, pero con independencia en tareas complejas, es determinante que los planes para apoyar esta fase sean conocidos, y los roles y responsabilidades estén claramente definidas. La fase de implementación se preocupa de obtener los productos externos, combinarlos con los componentes construidos y generar el paquete de plan de pruebas integrado.
- En la fase de *Implantación* (inserta dentro de los procesos de *Ejecución* y *Cierre*), todas las componentes de la solución se distribuyen al cliente y se implantan. La solución se prueba como un todo en un ambiente operacional hasta que esté lista para la prueba de aceptación formal por parte del cliente solicitante. Es donde el plan de pruebas cumple con demostrar al cliente que la solución cumple los requerimientos de la especificación funcional y el cliente confirma por escrito que todas las pruebas han sido exitosas y que acepta la solución. Los objetivos de esta fase es que el cliente esté de acuerdo en que la solución cumple la especificación funcional, que ha sido distribuida y que la organización acepta la propiedad de la solución. Además, se preocupa de distribuir e instalar todas las componentes de la solución, ejecutar el plan de pruebas integradas, registrar, revisar y corregir la solución para los defectos identificados, capacitar al personal correspondiente y finalmente obtener la aceptación del cliente.
- Durante la fase de *Operación* (que abarca los procesos *Ejecución* y *Cierre*), la aplicación ha sido probada y los usuarios se encuentran capacitados para el uso del sistema. Aquí se hace funcionar el sistema en forma real. El objetivo de esta fase, es que el cliente compruebe en su entorno de trabajo que el sistema funciona correctamente y entregue el visto bueno para su función definitiva, para que posteriormente la organización se acostumbre al sistema entregado, brindando soporte a los usuarios durante el periodo de marcha blanca del mismo.

F. Documentación Interna: Metodología para el Desarrollo y Mantenimiento