



Universidad Autónoma de Querétaro
 Facultad de Informática
 Maestría en Ingeniería de Software Distribuido

METODOLOGÍA DE ADMINISTRACIÓN DE PROYECTOS DE DESARROLLO DE SOFTWARE
 MULTI-SITIO

TESIS

Que como parte de los requisitos para obtener el diploma/grado de la
 Maestro en Ingeniería de Software Distribuido

Presenta:

Héctor Daniel Martínez Valencia

Dirigido por:

MC Ruth Angélica Rico Hernández

SINODALES

MC Ruth Angélica Rico Hernández
 Presidente

Firma

MISD Juan Salvador Hernández Valerio
 Secretario

Firma

MSI Gerardo Rodríguez Rojano
 Vocal

Firma

MC Luis Corral Velazquez
 Suplente

Firma

Dr. Marco Antonio Aceves Fernández
 Suplente

Firma

MC RUTH ANGÉLICA RICO HERNÁNDEZ
 Director de la Facultad

Dr. LUIS GERARDO HERNÁNDEZ SANDOVAL
 Director de Investigación y Postgrado

Centro Universitario
 Querétaro, Qro.
 Febrero 2011
 México

RESUMEN

El desarrollo de un proyecto de software es cada vez más complejo, extenso y caro. Al mismo tiempo los tiempos de entrega son cada vez más cortos. Ante este panorama existen esfuerzos en diferentes áreas del desarrollo de software, por ejemplo las herramientas CASE, que pueden ayudar a reducir los tiempos de desarrollo y cumplir el 100% de los requisitos asignados al software. Sin embargo, todos estos procesos son ejecutados por equipos de profesionales que se organizan y distribuyen las actividades a fin de lograr los objetivos, a tiempo y con calidad; y es aquí donde la administración de proyectos de ingeniería de software se convierte en un aspecto crítico para el éxito de los proyectos, un reporte del Departamento de Defensa de EU indica que... *“al día de hoy los problemas en el desarrollo de software no son problemas técnicos sino problemas de administración”* [Brooks 1987].

Esta tesis realiza una investigación en las áreas de administración y procesos basándose en el modelo de CMMI y conceptos generales de administración de proyectos, aterrizando dicha investigación en un caso de estudio de una empresa Mexicana transnacional que desarrolla software para sistemas electrónicos cuyos procesos de desarrollo de software adoptan el modelo de CMM y ISO-TS16949, para eventualmente diseñar y proponer un meta-modelo de proceso de administración, basado en CMMI que retome las prácticas positivas del caso de estudio, que ayude a mejorar la eficiencia y eficacia de las actividades de administración de proyectos de software, considerando que la operación de dicha empresa contempla proyectos con equipos localizados en un solo sitio o en múltiples sitios de desarrollo, dadas las exigencias del servicio requerido por sus clientes.

A partir de esta evaluación se emiten conclusiones relacionadas a la propuesta y a la adopción de CMMI como modelo de procesos para la administración de proyectos de software. La expectativa de esta tesis es ser un punto de partida para empresas que deseen adoptar el modelo de CMMI, así como profesionales que sean potenciales administradores de proyectos de desarrollo de software.

(Palabras Clave: CMMI, Administración de Proyectos, KPA, Efectividad, Eficacia, Multi-Sitio.)

ABSTRACT

The software project development has become more complex, large and expensive. At the same time, the required delivery times get shorter. Before this context there are efforts in different aspects of the software development, one example are the CASE tools, that might help reducing delivery times and meeting 100% of the requirements allocated to software. Nevertheless, at the end, all these processes and tools are used and executed by teams of professionals that get activities distributed and organized in order to achieve the goals, on time and with quality, and it is in this point where the software engineering project management becomes critical for the success of the projects, a report from the US Defense Department indicates "*today the problems in the software development are not technical problems but management problems*" [Brooks 1987].

This thesis performs a research on processes and management subjects based on CMMI model and some general concepts of project management, all this research being landed in a study case of a transnational Mexican enterprise that develops software for embedded systems which software development processes are based on CMMI model and ISO-TS16949, so eventually the design and proposal of a management process meta-model, based on CMM and re-taking positive practices from the study case, that help improving the efficiency and effectiveness of the management activities for the software projects, under the consideration that the enterprise operation requires projects with teams allocate in a single or in multiple development sites, based on the service exigencies required by its customers.

From this evaluation the conclusions get raised in relation to the proposal and the adoption of CMMI as a processes model for software project management. The expectation of this thesis is to become a starting point for enterprises looking to adopt CMMI model, and also to professionals in a potential software development project management position.

(Keywords: CMM, Project Management, KPA, Effectiveness, Efficiency, Multi-Site.)

A Angélica, mi compañera

A mi padre, a mi madre

A mi hermana

AGRADECIMIENTOS

Quiero aprovechar esta sección del trabajo de tesis para mencionar primeramente mi gratitud por el apoyo, dirección y docencia a la Maestra en Ciencias Ruth Angélica Rico Hernández, siempre atenta a mis solicitudes diversas y ciertamente una profesora clave en la obtención del grado.

Agradezco la oportunidad recibida por parte de la Universidad Autónoma de Querétaro en particular a la Facultad de Informática de cursar esta maestría así como su cuerpo docente; agradezco el conocimiento compartido, son pilares en mi formación profesional.

Agradezco el apoyo recibido por parte del Consejo Nacional de Ciencia y Tecnología CONACYT.

Agradezco el apoyo de mi esposa Angélica Ambriz Reyes por todo su apoyo, sus sacrificios, el tiempo, fines de semana y largas noches estudiando o haciendo trabajos para cumplir con los objetivos del programa.

ÍNDICE DE CONTENIDO

Introducción	1
1. Marco General de la Investigación	2
1.1 Objetivos de la Investigación	2
2. Método de Investigación	3
2.1 Según Fuente de Información	3
3. Plan de Investigación	4
4. Revisión de Literatura	4
Administración de Proyectos para Ingeniería de Software	6
1. Introducción al Modelo de CMMI	6
2. CMMI para Administración de Proyectos	11
2.1 Project Planning (PP)	11
2.2 Supplier Agreement Management (SAM)	13
2.3 Project Monitoring and Control (PMC)	14
2.4 Risk Management (RSKM)	15
2.5 Integrated Project Management +IPPD (IPM +IPPD)	17
2.5 Quantitative Project Management (QPM)	20
1. Perfil del Objeto de Estudio	24
1.1 Sistemas de Desarrollo	25
1.2 Perfil del Personal	25
1.3 Estructura Organizacional	27
1.4 Infraestructura	30
1.4.1 Tecnología de Información	30
1.4.2 Laboratorio Electrónico	31
1.4.2 Proceso	31
1.4.2 Subject Matter Experts	32
1.5 Interoperabilidad con otras Ingenierías	33
1.6 Clientes	35
1.7 Índices Actuales de Operación	35
1.8 Aseguramiento de Calidad de Software	36
2 Metodología de Administración de Proyectos	38

2.1 Inicio de Planeación de Proyecto	41
2.1.1 Plan de Desarrollo de Software	42
2.2 Ejecución del Plan de Desarrollo de Software	47
2.3 Cierre de Proyectos de Software	48
3 Resultados Actuales	48
3.1 Evaluación de PP	49
3.2 Evaluación de PCM	54
3.3 Evaluación de RSKM	58
3.4 Evaluación de IPM +IPPD	61
3.5 Evaluación de QPM	67
3.6 Resumen	68
1 Mejora PP	70
1.1 Obtener Compromiso con el Plan	70
2 Mejora en PCM	72
2.1 Monitorear el Involucramiento de los Participantes	72
3 Mejoras en IPM+IPPD	74
3.1 Establecer Visión Compartida del Proyecto	74
3.2 Establecer la Estructura Integral del Equipo	75
3.3 Asignar Requerimientos a los Equipos Integrales	78
3.4 Establecer Equipos Integrales	81
3.5 Asegurar Colaboración entre Equipos Interfaceando	82
4 Conclusiones	84
4.1 Trabajo Futuro	85
Apéndice I	87
Glosario	87
Apéndice II	90
Acrónimos	90
Referencias Bibliográficas	91
Libros	91
Reportes	91

ÍNDICE DE FIGURAS

Figura 1 Estructura de Áreas de Proceso de CMMI	8
Figura 2 Ejemplo de Organización Matricial	28
Figura 3 Ejemplo de Estructura Organizacional de Grupo de Software	29
Figura 4 Interoperabilidad	33
Figura 5 Proceso de Administración de Proyectos de Software	39
Figura 6 Antes de Mejora PP SP 3.3	71
Figura 7 Después de Mejora PP SP 3.3	72
Figura 8 Proceso de Administración de Proyectos de Software 3	72
Figura 9 Proceso de Administración de Proyectos de Software 4	75
Figura 10 Ejemplo de Arquitectura de Software	77
Figura 11 Ejemplo de Estructura Integral de Equipo	78
Figura 12 Asignar Requerimientos a los Equipos Integrales	80

ÍNDICE DE TABLAS

Tabla 1 Áreas de Proceso, Categoría y Nivel de CMMI	10
Tabla 2 Ejemplo de Objetivos	44
Tabla 3 Documentación de Riesgos	45
Tabla 4 Lista de Reportes de Progreso de Proyecto	45
Tabla 5 Necesidades de Entrenamiento	46
Tabla 6 Análisis de Área de Proceso PP de CMMI	54
Tabla 7 Análisis de Área de Proceso PCM de CMMI	58
Tabla 8 Análisis de Área de Proceso RSKM de CMMI	61
Tabla 9 Análisis de Áreas de Proceso IPM +IPPD de CMMI	67
Tabla 10 Análisis de Área de Proceso QPM de CMMI	67

CAPÍTULO I. Metodología y Plan de Investigación

Introducción

Las necesidades de desarrollo de software han crecido durante las últimas décadas, demandando la profesionalización, investigación y desarrollo de la ingeniería de software. Ante la crisis de software de los 80's relacionada al escalamiento en costo y problemas de calidad de software, se creó el Instituto de Ingeniería de Software de Estados Unidos "SEI" por sus siglas en inglés, donde una de sus contribuciones más importantes fue "*The Capability Maturity Model: Guidelines for Improving the Software Process*" (SEI, 1995) como una iniciativa para mejorar la calidad de software. Dicho modelo fue adoptado por la comunidad de software como referente en la implantación de procesos de desarrollo de software y su misma estructura (por niveles de madurez) fue adoptada en diversas disciplinas de desarrollo. El modelo más reciente del SEI relacionado a la mejora de procesos de desarrollo de software es llamado "*CMM Integration*" o CMMI. El CMMI resultó de la integración de 3 modelos base:

1. *The Capability Maturity Model for Software (SW-CMM) v2.0 draft C* [SEI 1997b].
2. *The Systems Engineering Capability Model (SECM)* [EIA 1998].
3. *The Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98* [SEI 1997a].

Los modelos del CMMI contienen prácticas que cubren la administración de proyectos, administración de procesos, ingeniería de sistemas, ingeniería de hardware, ingeniería de software y procesos de soporte para desarrollo y mantenimiento. El modelo de CMMI también cubre la administración de equipos integrales de desarrollo y mantenimiento.

Al igual que sus antecesores, el CMMI define niveles de madurez (en su representación "Staged"). Los niveles de madurez son utilizados en CMMI para describir un desarrollo evolutivo recomendado, en términos de proceso, para una organización que decide adoptar el modelo.

La implantación del CMMI y sus Áreas de Proceso (PA) en una organización es internamente dependiente al perfil de los proyectos y mercado que se atiende. La presente tesis utilizará únicamente las áreas de proceso de CMMI relacionadas a la administración de proyectos, particularmente enfocadas a la ingeniería de software, listadas a continuación:

1. Planeación de Proyectos (PP)

2. Monitoreo y Control de Proyectos (PMC)
3. Administración de Acuerdos con Proveedores (SAM)
4. Administración Integral de Proyectos +Desarrollo de Producto y Proceso Integral (IPM+IPPD)
5. Administración de Riesgos (RSKM)
6. Administración Cuantitativa de Proyectos (QPM).

A partir de este referente se buscara entender la evolución necesaria en términos de proceso, para la administración de proyectos de software, al pasar de una operación bajo un contexto de desarrollo local a un contexto global o Multi-sitio.

La importancia de este tema radica en que dicho cambio a un de contexto de operación global, afecta directamente a los ingenieros de software mexicanos involucrándolos en proyectos de desarrollo de software en coordinación con ingenieros alrededor del mundo. El ilustrar y analizar un caso real de aplicación del modelo de CMMI además de potenciales mejoras, habilita a los ingenieros trabajando en un contexto global, a mejorar la eficiencia y eficacia en la administración de sus proyectos de desarrollo de software, así como su desempeño en los mismos.

1. Marco General de la Investigación

1.1 Objetivos de la Investigación

La presente tesis intenta en la medida de su alcance, proponer mejoras a una Metodología de Administración de Proyectos de Desarrollo de Software, que cumpla con los lineamientos de CMMI concernientes a la administración de proyectos y que considere las particularidades de un desarrollo Multi-sitio. Dichas mejoras harán énfasis en las consideraciones particulares de administración pertinentes a un proyecto global, y al mismo tiempo proveerá flexibilidad para la administración de proyectos locales.

Como investigación de campo se analizarán los procesos de una empresa de software que adopto el modelo de SW-CMM [SEI 1997b] para la mejora de su calidad de software. Al momento de diseñar sus procedimientos de desarrollo de software, se encontraba trabajando en un contexto de desarrollo local y ahora ha evolucionado a un contexto de desarrollo global manteniendo los mismos procedimientos. En otras palabras, inicialmente sus actividades de desarrollo de software se realizaban en un solo sitio de ingeniería en Estados Unidos, y

actualmente se encuentra en un entorno con sitios de desarrollo alrededor del mundo, incluyendo países como México, India, China, Polonia, Alemania etc. Bajo dicho cambio en su operación sus procedimientos no evolucionaron con la misma rapidez, o en todos sus aspectos, al contexto de desarrollo global. Analizaremos el impacto y las mejoras potenciales para adaptarse a las nuevas necesidades. Dichas mejoras sugeridas sean comparadas al cumplimiento de las Áreas de Proceso de la Administración de Proyectos de CMMI.

La hipótesis que sustenta el trabajo es la siguiente: La Metodología de Administración de Proyectos de Desarrollo de Software Multi-sitio, provee los mecanismos para una eficiente y eficaz administración de proyectos de desarrollo de software, cuyo staff se encuentra local o distribuido en múltiples sitios separados geográficamente, cumpliendo con las normas documentadas en el modelo de CMMI relacionadas a la administración de proyectos, particularmente para el desarrollo de software.

2. Método de Investigación

A fin de establecer un fundamento teórico actual y lograr los objetivos de la presente tesis los siguientes métodos de investigación fueron utilizados durante el desarrollo de la presente tesis.

2.1 Según Fuente de Información

Investigación Documental

La investigación documental provee mecanismos para analizar el estado del arte actual concerniente a procesos de administración de proyectos particularmente aplicados a la ingeniería de software.

Investigación de Campo

La investigación de campo guía las actividades de análisis de la operación de la empresa de desarrollo de software cuyos procesos y contexto serán utilizados para identificar las potenciales deficiencias o aciertos en su esfuerzo por adaptarse a las nuevas necesidades de operación Multi-sitio.

3. Plan de Investigación

La presente tesis siguió el siguiente plan de investigación:

1. Recolección y revisión de estado del arte, al momento de elaboración de la presente tesis, concerniente a modelos de administración de proyectos relacionado a la ingeniería de software.
2. Identificación de la empresa objeto de estudio.
3. Entrevistas con personal involucrado en la organización en los siguientes aspectos:
 - a. Organización Estructural
 - b. Perfil de los diferentes actores relacionados a la administración de proyectos de software.
 - c. Descripción del contexto de desarrollo, dificultades y aciertos.
 - d. Descripción de planes futuros.
4. Documentación de observaciones.
5. Análisis de potenciales mejoras y documentación de las potenciales mejoras requeridas en una metodología de desarrollo.
6. Elaboración de conclusiones.

4. Revisión de Literatura

La presente tesis trata temas relacionados con la administración de proyectos, la ingeniería de software, el desarrollo de proyectos Multi-sitio, el modelo de proceso de CMMI, el modelo de SW-CMM aplicado.

La literatura de referencia relacionada a la presente tesis es principalmente literatura publicada por el instituto de ingeniería de software “SEI” por sus siglas en inglés. Existen dos razones, primera el objeto de estudio, para la investigación de campo, diseño sus procedimientos de desarrollo de software basado en el modelo del SEI de *The Capability Maturity Model: Guidelines for Improving the Software Process* (SEI, 1995), y segunda, el SEI, siguiendo un modelo de mejora continua, ha continuado la evolución del modelo con base en los resultados recolectados del uso de modelos previos, donde su actual estado del arte se concentra en el modelo de CMMI. A continuación se lista la literatura citada en la presente tesis:

- CMMI Product Team. 2006. CMMI® for Development, Versión 1.2., Technical Report CMU/SEI-2006-TR-008 <http://www.sei.cmu.edu/reports/06tr008.pdf> liga accedida el 6 de Abril 2011.
- Mark C. Paulk, William Curtis, Mary Beth Chrissis, Charlie Weber. 1993. Capability Maturity Model SM for Software, Version 1.1, Technical Report CMU/SEI-93-TR-024. <http://www.sei.cmu.edu/reports/93tr024.pdf> liga accedida el 6 de Abril 2011.
- Mark C. Paulk, Charlie Weber, Suzanne Garcia-Miller, Mary Beth Chrissis, Marilyn Bush. 1993. Key Practices of the Capability Maturity Model Version 1.1, Technical Report CMU/SEI-93-TR-025 <http://www.sei.cmu.edu/reports/93tr025.pdf> liga accedida el 6 de Abril 2011.

CAPÍTULO II. Estado del Arte

Administración de Proyectos para Ingeniería de Software

La presente tesis trata el tema de la administración de proyectos particularmente aplicado al desarrollo de software. A la fecha de elaboración de la presente tesis, la más reciente publicación por parte de Instituto de Ingeniería de Software es el llamado modelo de CMMI (SEI, 2006), el cual abarca aspectos relacionados a la ingeniería de software, ingeniería de sistemas y administración de proyectos en general. Es de notarse que dicho modelo de CMMI es extenso en su alcance, sin embargo para propósitos de la presente tesis se utilizarán los elementos de CMMI relacionados a la administración de proyectos y aplicados a la ingeniería de software, los cuales son descritos en las siguientes secciones del presente capítulo.

El modelo de CMMI fue seleccionado como estado del arte por ser la más reciente publicación del SEI y debido a que la empresa a estudiar utiliza un modelo anterior del mismo SEI para su definición de procesos de desarrollo de software resultado en un mapeo natural de documentación de procesos, sin embargo esto no significa que CMMI sea el único estado del arte referente a procesos de desarrollo de software, existe a su vez, su contraparte Europea llamada SPICE, que no será abordada por la presente tesis.

Actualmente existe una marcada división en la adopción e implantación de modelos de mejora de proceso de desarrollo de software. El criterio de uso y evaluación está generalmente determinado por el lugar en el cual se utilizara el software, si este es usado en Europa, generalmente se utiliza SPICE, si es utilizado en América o Asia generalmente es CMMI, por supuesto existen casos con la flexibilidad de usar uno u otro.

1. Introducción al Modelo de CMMI

El CMMI o “*Capability Maturity Model Integration*” es la más reciente evolución de la serie de modelos de proceso desarrollados y auditados por parte del SEI. Tal como su nombre lo anota, este modelo viene a integrar varios modelos que originalmente se desarrollaban de manera separada dentro del mismo SEI, dichos modelos son los siguientes:

1. *The Capability Maturity Model for Software (SW-CMM) v2.0 draft C* [SEI 1997b].
2. *The Systems Engineering Capability Model (SECM)* [EIA 1998].
3. *The Integrated Product Development Capability Maturity Model (IPD-CMM) v0.98* [SEI 1997a].

CMMI documenta las Áreas de Proceso (PA, por sus siglas en inglés “*Process Area*”) necesarias a considerar para diseñar y/o mejorar los procesos relacionados al desarrollo de software por parte de cualquier empresa o institución que decida adoptarlo. Al igual que las certificaciones por parte de ISO, el SEI provee una certificación que indican el grado de adopción e implantación de un proceso basado en el modelo de CMMI por parte de alguna empresa o institución. CMMI, al igual que sus antecesores, da la posibilidad de indicar el nivel de adopción del modelo reflejado en un nivel de madurez, este es llamado modo “*staged*”, sin embargo también provee la flexibilidad de evaluar individualmente cada una de las PA del modelo de manera individual a un grado de implantación, a este modo se le llama “*continuous*”, y es muy similar al concepto que maneja SPICE. A diferencia de SPICE, CMMI parte de lo general a evaluar lo individual, mientras que SPICE lo hace de manera inversa.

CMMI estructura cada PA en tres componentes principales: Requeridos, Esperados e Informativos.

1. Los componentes requeridos describen lo que la organización debe cumplir a fin de satisfacer el PA. Los componentes requeridos de cada PA son los siguientes:
 - a. Objetivos Específicos.- Describe las características únicas que se deben encontrar en el proceso para satisfacer la PA.
 - b. Objetivos Generales.- Estos objetivos son llamados generales por que dichos objetivos son aplicables a múltiples PA.
2. Los componentes esperados describen lo que la organización podiera implementar para satisfacer la PA. Los componentes esperados de cada PA son los siguientes:
 - a. Prácticas Específicas.- Son las actividades que son consideradas importantes para alcanzar los objetivos específicos del PA.
 - b. Prácticas Genéricas. Son las actividades que son consideradas importantes para alcanzar los objetivos generales del PA.
3. Los componentes informativos dan detalles para ayudar a las organizaciones a pensar como abordar los componentes requeridos y esperados. Los componentes informativos de cada PA son los siguientes:

- a. Enunciados de Propósito.- Describe el propósito del PA.
- b. Notas Introductorias.- Describe los conceptos más importantes cubiertos por el PA.
- c. Áreas de Proceso Relacionadas.- Incluye referencias a otras PA.
- d. Productos de Trabajo Típicos.- Lista ejemplos de salidas obtenidas al ejecutar un PA en particular.
- e. Sub-prácticas.- Son descripciones que dan guía en la interpretación e implementación de una práctica genérica o específica.
- f. Elaboración de Prácticas Genéricas.- Provee guías para indicar como aplicar una Práctica Genérica al PA específico.

La siguiente figura ilustra el modelo de componentes considerado para cada una de las PA de CMMI:

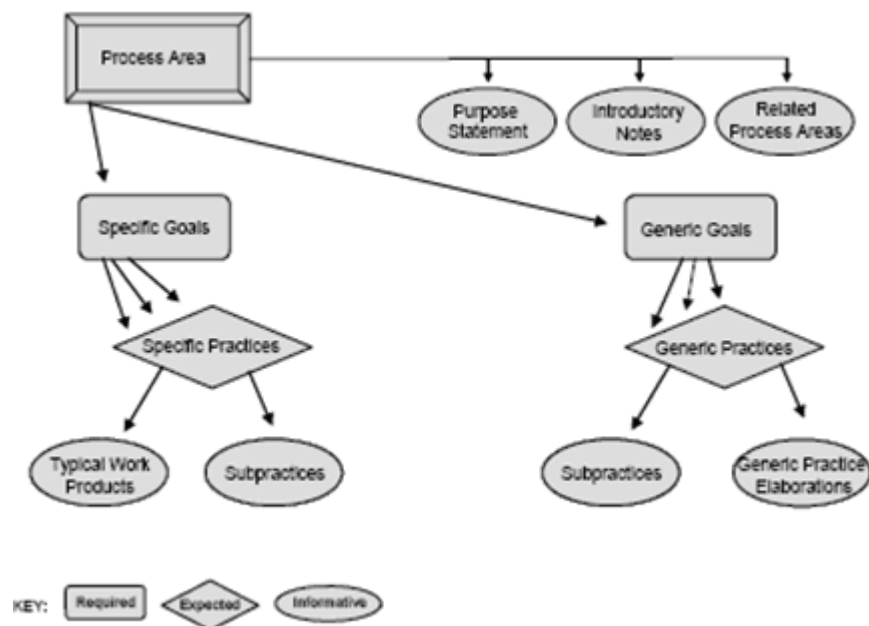


Figura 1 Estructura de Áreas de Proceso de CMMI

El modelo completo de CMMI comprende 22 PA listadas a continuación:

1. *Causal Analysis and Resolution (CAR)*
2. *Configuration Management (CM)*
3. *Decision Analysis and Resolution (DAR)*
4. *Integrated Project Management +IPPD (IPM+IPPD)*
5. *Measurement and Analysis (MA)*
6. *Organizational Innovation and Deployment (OID)*

7. *Organizational Process Definition +IPPD (OPD+IPPD)*6
8. *Organizational Process Focus (OPF)*
9. *Organizational Process Performance (OPP)*
10. *Organizational Training (OT)*
11. *Product Integration (PI)*
12. *Project Monitoring and Control (PMC)*
13. *Project Planning (PP)*
14. *Process and Product Quality Assurance (PPQA)*
15. *Quantitative Project Management (QPM)*
16. *Requirements Development (RD)*
17. *Requirements Management (REQM)*
18. *Risk Management (RSKM)*
19. *Supplier Agreement Management (SAM)*
20. *Technical Solution (TS)*
21. *Validation (VAL)*
22. *Verification (VER)*

Cada una de las PA es asociada a un nivel de madurez, en la representación “staged” de CMMI y a su vez pertenece a una categoría individual como es descrito en la siguiente tabla:

<i>Process Area</i>	<i>Category</i>	<i>Maturity Level</i>
<i>Causal Analysis and Resolution (CAR)</i>	<i>Support</i>	5
<i>Organizational Innovation and Deployment (OID)</i>	<i>Process Management</i>	5
<i>Organizational Process Performance (OPP)</i>	<i>Process Management</i>	4
<i>Quantitative Project Management (QPM)</i>	<i>Project Management</i>	4
<i>Decision Analysis and Resolution (DAR)</i>	<i>Support</i>	3
<i>Integrated Project Management +IPPD (IPM+IPPD)</i>	<i>Project Management</i>	3
<i>Organizational Process Definition +IPPD (OPD+IPPD)</i>	<i>Process Management</i>	3
<i>Organizational Process Focus (OPF)</i>	<i>Process Management</i>	3
<i>Organizational Training (OT)</i>	<i>Process Management</i>	3
<i>Product Integration (PI)</i>	<i>Engineering</i>	3
<i>Requirements Development (RD)</i>	<i>Engineering</i>	3

Risk Management (RSKM)	Project Management	3
Technical Solution (TS)	Engineering	3
Validation (VAL)	Engineering	3
Verification (VER)	Engineering	3
Configuration Management (CM)	Support	2
Measurement and Analysis (MA)	Support	2
Project Monitoring and Control (PMC)	Project Management	2
Project Planning (PP)	Project Management	2
Process and Product Quality Assurance (PPQA)	Support	2
Requirements Management (REQM)	Engineering	2
Supplier Agreement Management (SAM)	Project Management	2

Tabla 1 Áreas de Proceso, Categoría y Nivel de CMMI

Nota: Se resaltan con negrillas las áreas de proceso relacionadas a la administración de proyectos.

Finalmente, como se mencionó anteriormente el modelo de CMMI puede ser utilizado en dos modos el “*continuous*” y el “*staged*”. El propósito es facilitar la transición de modelos para las empresas que anteriormente utilizaron algún modelo de SPICE, o CMM. Conceptualmente se manejan algunas diferencias descritas a continuación:

- *Continuous*: esta representación de CMMI permite a la organización seleccionar los PA y niveles de capacidad basado en sus objetivos de mejora. La mejora es medida en términos de niveles de capacidad con rango de 0 a 5. Este modo requiere definir Perfiles de Proceso Objetivo con determinados niveles de capacidad a alcanzar, este concepto es muy parecido a la filosofía de SPICE, donde al momento de auditar una empresa de software, se le comunica con antelación el “Perfil de Proceso” esperado para poder obtener la certificación. CMMI documenta una tabla de equivalencia de los perfiles de proceso a los niveles de madurez de la representación Staged.
- *Staged*: esta es la representación ya conocida de los modelos de CMM, la cual permite a la organización seleccionar sus PA basándose en niveles de madurez. La mejora es medida en términos de nivel de madurez con un rango de 1 a 5. En este caso los objetivos son definidos por nivel de madurez.

2. CMMI para Administración de Proyectos

Esta sección describe las áreas de proceso relacionadas a la administración de proyectos de modelo de CMMI. La descripción de las mismas esta completamente basada directamente en el documento *CMMI® for Development, Version 1.2 (SEI, 2006)*. La secuencia de descripción inicia por las áreas de proceso de niveles de madurez inferiores a superiores del acuerdo a la organización *staged* del modelo de CMMI (ver tabla 1).

2.1 Project Planning (PP)

La planeación de proyectos es un área de proceso perteneciente al nivel de madurez 2 del CMMI *staged*. Su propósito es establecer y mantener planes que definan las actividades de los proyectos. La planeación de los proyectos incluyen las siguientes actividades:

1. Desarrollo del plan de proyecto.
2. Interacción con los participantes.
3. Obtener compromiso con el plan.
4. Mantenimiento del plan.

La planeación inicia con los requerimientos que definen al proyecto o producto (requerimientos esta subrayado por que frecuentemente los requerimientos son demasiado inestables al momento de definir el plan). La planeación incluye la estimación de los atributos de los productos del proyecto y las tareas, determinar los recursos necesarios, negociar acuerdos, definición de un calendario de actividades, identificar y analizar los riesgos del proyecto. La iteración a través de estas actividades podría ser necesaria para afinar el plan del proyecto. El plan del proyecto provee las bases para desempeñar y controlar las actividades que abordan los compromisos con el cliente. Los objetivos específicos y las prácticas de esta Área de Proceso de CMMI se resumen a continuación:

1. SG 1 Establish Estimates (Establecer Estimados)
 - a. SP 1.1 Estimate the Scope of the Project (Estimar el Alcance del Proyecto)
 - b. SP 1.2 Establish Estimates of Work Product and Task Attributes (Establecer Estimados de los Productos del Trabajo y los Atributos de las Tareas)
 - c. SP 1.3 Define Project Lifecycle (Definir el Ciclo de Vida del Proyecto)
 - d. SP 1.4 Determine Estimates of Effort and Cost (Determinar Estimados de Esfuerzo y Costo)

2. SG 2 Develop a Project Plan (Desarrollar un Plan de Proyecto)
 - a. SP 2.1 Establish the Budget and Schedule (Establecer Presupuesto y Calendario)
 - b. SP 2.2 Identify Project Risks (Identificar Riesgos del Proyecto)
 - c. SP 2.3 Plan for Data Management (Planear la Administración de los Datos)
 - d. SP 2.4 Plan for Project Resources (Planear los Recursos del Proyecto)
 - e. SP 2.5 Plan for Needed Knowledge and Skills (Planear las Necesidades de Habilidades y Conocimientos)
 - f. SP 2.6 Plan Stakeholder Involvement (Planear Involucramiento de los Participantes)
 - g. SP 2.7 Establish the Project Plan (Establecer el Plan del Proyecto)
3. SG 3 Obtain Commitment to the Plan (Obtener Compromiso con el Plan)
 - a. SP 3.1 Review Plans That Affect the Project (Revisar los Planes que afecten al Proyecto)
 - b. SP 3.2 Reconcile Work and Resource Levels (Reconciliar Niveles de Trabajo y Recursos)
 - c. SP 3.3 Obtain Plan Commitment (Obtener Compromiso con el Plan)

El plan del proyecto usualmente necesitará ser actualizado mientras que el desarrollo progresa para abordar los cambios en requisitos y acuerdos, estimados inexactos, acciones correctivas, y cambios de proceso. Prácticas específicas, describiendo tanto la planeación como la re-planeación, son contenidas en esta área de proceso. Áreas de Proceso de CMMI relacionadas a la planeación del proyecto son listadas a continuación:

1. Requirements Development (RD).- Desarrollo de Requisitos para obtener información acerca del desarrollo de los requisitos que definen al producto y los componentes del producto.
2. Requirements Management (REQM).- Administración de Requisitos para obtener información acerca la administración de los requisitos necesarios para la planeación y la re-planeación.
3. Risk Management (RSKM).- Administración del Riesgo para obtener información acerca la identificación y administración de los riesgos.
4. Technical Solution (TS).- Solución Técnica para obtener información acerca de la transformación de los requisitos en soluciones del producto y sus componentes.

2.2 Supplier Agreement Management (SAM)

La

administración de acuerdos con proveedores (SAM) es un área de proceso del nivel del madurez 2 del CMMI *staged*. El propósito de esta área de proceso es la gestión de la adquisición de productos de los proveedores. La administración de acuerdos con proveedores involucra lo siguiente:

1. Determinar el tipo de adquisición.
2. Seleccionar proveedores.
3. Establecer y mantener acuerdos con los proveedores.
4. Ejecutar los acuerdos con los proveedores.
5. Monitorear procesos de los proveedores seleccionados.
6. Evaluar los productos de trabajo de los proveedores seleccionados.
7. Aceptar las entregas de los productos adquiridos.
8. Incorporar los productos adquiridos al proyecto.

Esta área de proceso aborda primariamente la adquisición de productos y componentes que serán entregados como parte del proyecto final, aunque puede ser aplicada a productos y componentes, que no serán parte de la entrega sino un medio, como una medida para reducción de riesgos. Típicamente los productos a ser adquiridos son determinados durante las etapas tempranas de planeación del desarrollo de un producto. Esta área de proceso no contempla los casos donde el proveedor esta integrado al mismo equipo de desarrollo y ocupa los mismos procedimientos y reporta actividades al mismo gerente que el resto del equipo. Los objetivos específicos y las prácticas de esta Área de Proceso de CMMI se listan a continuación:

1. SG 1 Establish Supplier Agreements (Establecer Acuerdos con Proveedor)
 - a. SP 1.1 Determine Acquisition Type (Determinar Tipo de Adquisición)
 - b. SP 1.2 Select Suppliers (Seleccionar Proveedores)
 - c. SP 1.3 Establish Supplier Agreements (Establecer Acuerdos con Proveedor)
2. SG 2 Satisfy Supplier Agreements (Satisfacer Acuerdos con Proveedor)
 - a. SP 2.1 Execute the Supplier Agreement (Ejecutar el Acuerdo con Proveedor)
 - b. SP 2.2 Monitor Selected Supplier Processes (Monitorear Procesos de Proveedor Seleccionado)
 - c. SP 2.3 Evaluate Selected Supplier Work Products (Evaluar productos de trabajo del proveedor seleccionado)

- d. SP 2.4 Accept the Acquired Product (Aceptar el producto adquirido)
- e. SP 2.5 Transition Products (Incorporar Productos).

Las áreas de proceso relacionadas son las siguientes:

1. Project Monitoring and Control (PMC). - Control y Monitoreo de Proyecto para obtener información de control de proyectos y la toma de acciones correctivas.
2. Requirements Development (RD).- Desarrollo de Requisitos para obtener información acerca del desarrollo de requisitos.
3. Requirements Management (REQM).- Administración de Requisitos para obtener información acerca de la administración de requisitos, incluyendo la rastreabilidad de los requisitos en los productos adquiridos de los proveedores.
4. Technical Solution (TS). Solución Técnica para obtener información de cómo determinar los productos y componentes de productos que serían adquiridos de los proveedores.

* Nota.- Para propósitos de la presente tesis y el objeto de estudio, esta área de proceso no es utilizada por encontrarse fuera de alcance del tema; solo es documentada como referencia de un área de proceso de CMMI relacionada a la administración de proyectos de software.

2.3 Project Monitoring and Control (PMC)

El

Monitoreo y Control de Proyectos (PMC) es un área de proceso del nivel del madurez 2 del CMMI *staged*. Su propósito es proveer un entendimiento del progreso de proyecto de manera tal que se pueden tomar acciones correctivas cuando el desempeño del proyecto se desvía significativamente del plan.

La planeación documentada de un proyecto es la base para el monitoreo de las actividades, comunicación del status, y la toma de acciones correctivas. El progreso es determinado primariamente comparando los productos de proyecto actuales contra los atributos de las tareas, esfuerzo, costo, y calendario con el plan a través de metas intermedias o niveles de control dentro del calendario del proyecto. En eventos de desviación del plan original será necesario tomar acciones correctivas tales como: revisión del plan, establecer nuevos acuerdos o incluir actividades de mitigación adicionales al plan actual. La siguiente lista resume los objetivos específicos así como las prácticas requeridas por esta área de proceso de CMMI:

1. SG 1 Monitor Project Against Plan (Monitorear Proyecto contra Plan)

- a. SP 1.1 Monitor Project Planning Parameters (Monitorear los Parámetros de Planeación del Proyecto)
 - b. SP 1.2 Monitor Commitments (Monitorear Acuerdos)
 - c. SP 1.3 Monitor Project Risks (Monitorear Riesgos)
 - d. SP 1.4 Monitor Data Management (Monitorear la Administración de Datos)
 - e. SP 1.5 Monitor Stakeholder Involvement (Monitorear Involucramiento de Participantes).
 - f. SP 1.6 Conduct Progress Reviews (Conducir Revisiones de Progreso)
 - g. SP 1.7 Conduct Milestone Reviews (Conducir Revisiones de metas intermedias)
2. SG 2 Manage Corrective Action to Closure (Administrar Acciones Correctivas hasta Cerrarlas)
- a. SP 2.1 Analyze Issues (Analizar Problemas)
 - b. SP 2.2 Take Corrective Action (Tomar Acción Correctiva)
 - c. SP 2.3 Manage Corrective Action (Administrar Acción Correctiva)

Las áreas de proceso relacionadas son las siguientes:

1. Project Planning (PP).- Planeación de Proyecto para obtener información acerca del plan de proyecto, incluyendo como es que especifica un nivel apropiado de monitoreo, las métricas utilizadas para monitorear progreso y los riesgos conocidos.
2. Measurement and Analysis (MA).- Análisis de Mediciones para adquirir información acerca del proceso de medir, analizar y registrar información.

2.4 Risk Management (RSKM)

La

Administración de Riesgos (RSKM) es un área de proceso del nivel del madurez 3 del CMMI *staged*, Su propósito es identificar los problemas potenciales antes de que ocurran de manera que las actividades de manejo de riesgos puedan ser planeadas e invocadas como se necesiten a través de la vida del producto o proyecto para mitigar los impactos adversos mientras que se alcanzan los objetivos.

La administración de riesgos es una actividad continua, el proceso de anticipar es una parte importante de la administración. La administración de riesgos debe abordar problemas que puedan poner en peligro la consecución de objetivos críticos. Una estrategia de administración de

riesgos continua es aplicada para anticipar y mitigar los riesgos que pudieran tener impacto crítico en el proyecto.

Una administración efectiva de riesgos incluye una temprana y agresiva identificación de riesgos a través de la colaboración e involucramiento de los implicados. Un liderazgo fuerte para todos los implicados es necesario para establecer un ambiente de discusión abierta y libre acerca de los riesgos. La administración de riesgos debe considerar fuentes internas y externas para riesgos de costo, calendario y desempeño. La administración de riesgos puede dividirse en tres partes:

1. Definición de la estrategia de administración de riesgos.
2. Identificación y Análisis de los riesgos.
3. Manejo de los riesgos identificados, incluyendo la implementación de los planes de mitigación de riesgos cuando sea necesario.

Los objetivos específicos y las prácticas de esta área de proceso se listan a continuación:

1. SG 1 Prepare for Risk Management (Preparar la Administración de Riesgos)
 - a. SP 1.1 Determine Risk Sources and Categories (Determinar las Fuentes de Riesgo y sus Categorías).
 - b. SP 1.2 Define Risk Parameters (Definir los Parámetros de Riesgo).
 - c. SP 1.3 Establish a Risk Management Strategy (Establecer una estrategia de Administración de Riesgos).
2. SG 2 Identify and Analyze Risks (Identificar y Analizar Riesgos)
 - a. SP 2.1 Identify Risks (Identificar Riesgos)
 - b. SP 2.2 Evaluate, Categorize, and Prioritize Risks (Evaluar, Categorizar, y Priorizar Riesgos)
3. SG 3 Mitigate Risks (Mitigar Riesgos)
 - a. SP 3.1 Develop Risk Mitigation Plans (Desarrollar Planes de Mitigación de Riesgos)
 - b. SP 3.2 Implement Risk Mitigation Plans (Implementar Planes de Mitigación de Riesgos)

Las áreas de proceso relacionadas son las siguientes:

1. Project Planning (PP).- Planeación de Proyecto para obtener información acerca de la identificación de los riesgos del proyecto y planeación para el involucramiento de implicados relevantes.

2. Project Monitoring and Control (PMC).- Monitoreo y Control del Proyecto para obtener información acerca del monitoreo de los riesgos del proyecto.
3. Decision Analysis and Resolution (DAR).- Análisis y Resolución de Decisiones para adquirir información acerca del uso de un proceso formal de evaluación para evaluar las alternativas para la selección y mitigación de los riesgos identificados.

2.5 Integrated Project Management +IPPD (IPM +IPPD)

La administración Integral de Proyectos es un área de proceso perteneciente al nivel de madurez 3 del CMMI *staged*. Su propósito es establecer y administrar el proyecto además del involucramiento de participantes relevantes de acuerdo a un proceso integral y definido para las necesidades del proyecto, con respecto a los estándares de proceso de la organización. Para IPPD también se cubre el establecimiento de una visión compartida para el proyecto y el establecimiento de equipos integrales que completaran los objetivos del proyecto.

La administración integral del proyecto involucra lo siguiente:

- Establecer un proceso definido específicamente para las necesidades del proyecto al inicio del mismo, basado en el grupo de procesos estándar de la organización.
- Administrar el proyecto utilizando el proceso definido para el proyecto.
- Establecer el ambiente de trabajo para el proyecto basado en los estándares de ambiente de trabajo de la organización.
- Utilizar y contribuir con los valuartes de proceso de la organización.
- Considerar las observaciones de participantes relevantes al ser identificadas, y cuando sea apropiado, abordadas durante el desarrollo del producto.
- Asegurar que los participantes relevantes efectúen sus tareas de una manera coordinada y a tiempo para abordar los requerimientos, planes, objetivos, y riesgos del producto y de los componentes del producto; con el fin de cumplir con los compromisos e identificar, rastrear y resolver problemas de coordinación.

IPPD Suplementar también involucra lo siguiente:

- Establecer una visión compartida del proyecto.
- Establecer equipos integrados con tareas para completar los objetivos del proyecto.

La administración del esfuerzo, costo, calendario, staff, riesgo y otros factores están ligados al proceso definido para el proyecto. La implementación y administración del proceso definido para el proyecto es típicamente descrito en el plan de proyecto. Ciertas actividades pudieran ser

cubiertas en otros planes que afectan al proyecto, tales como plan de aseguramiento de calidad, estrategia de administración de riesgos y, plan de administración de configuración. Dado que los procesos definidos para cada proyecto son hechos a la medida con respecto al grupo de procesos estándar de la organización, la variabilidad entre proyectos es típicamente reducida, de manera que los proyectos pueden, más fácilmente, compartir valuartes de proceso, datos y lecciones aprendidas.

Esta área de proceso también aborda la coordinación de todas las actividades asociadas con el proyecto tales como:

- Actividades de desarrollo (ejemplo: desarrollo de requerimientos, diseño y, verificación).
- Actividades de servicio (ejemplo: entrega, ayuda a usuario, operación y, contacto con el cliente).
- Actividades de adquisición (ejemplo: solicitudes, monitoreo de proveedores y, transición de operaciones).
- Actividades de soporte (ejemplo: administración de configuración, documentación, comercialización y, entrenamiento).

Las interfaces de trabajo y las interacciones entre los participantes relevantes, internos y externos al proyecto, son planeadas y administradas para asegurar la calidad e integridad del producto entero. Los participantes relevantes contribuyen, de manera apropiada, en la definición de del proceso definido y el plan del proyecto.

Esta área de proceso aplica a cualquier estructura organizacional, incluyendo proyectos que son estructurados como organizaciones lineales, matriciales o equipos integrales.

Los objetivos específicos y las prácticas de esta área de proceso se listan a continuación:

1. SG 1 Use the Project's Defined Process (Utilizar el Proceso Definido del Proyecto).
 - a. SP 1.1 Establish the Project's Defined Process (Establecer el Proceso Definido para el Proyecto).
 - b. SP 1.2 Use Organizational Process Assets for Planning Project Activities (Utilizar los valuartes de Proceso Organizacionales para las Actividades de Planeación del Proyecto).
 - c. SP 1.3 Establish the Project's Work Environment (Establecer el Ambiente de Trabajo del Proyecto).
 - d. SP 1.4 Integrate Plans (Integrar Planes)

- e. SP 1.5 Manage the Project Using the Integrated Plans (Administrar el Proyecto utilizando los Planes Integrados).
 - f. SP 1.6 Contribute to the Organizational Process Assets (Contribuir a los Valuartes de Proceso Organizacionales)
2. SG 2 Coordinate and Collaborate with Relevant Stakeholders (Coordinar y Colaborar con Participantes Relevantes)
- a. SP 2.1 Manage Stakeholder Involvement (Gestar el Involucramiento de Participantes)
 - b. SP 2.2 Manage Dependencies (Administrar Dependencias)
 - c. SP 2.3 Resolve Coordination Issues (Resolver Problemas de Coordinación)

IPPD Supplemental

3. SG 3 Apply IPPD Principles (Aplicar los Principios de IPPD)
- a. SP 3.1 Establish the Project's Shared Vision (Establecer la Visión Compartida del Proyecto).
 - b. SP 3.2 Establish the Integrated Team Structure (Establecer la Estructura de Equipo Integrada).
 - c. SP 3.3 Allocate Requirements to Integrated Teams (Asignar Requerimientos a los Equipos Integrados).
 - d. SP 3.4 Establish Integrated Teams (Establecer los Equipos Integrados)
 - e. SP 3.5 Ensure Collaboration among Interfacing Teams (Asegurar la Colaboración entre los Equipos de Interfase).

Las áreas de proceso relacionadas son las siguientes:

1. Project Planning (PP).- La planeación de proyecto para obtener más información acerca de la planeación del proyecto, que incluye la identificación de los participantes relevantes y su involucramiento apropiado en el proyecto.
2. Project Monitoring and Control (PMC).- Referirse al área de proceso de Monitoreo y Control de Proyecto para adquirir información acerca del monitoreo y control del proyecto y tomar las acciones correctivas.
3. Organizational Process Definition (OPD).- Definición Organizacional de Proceso para obtener más información acerca de los valuartes de proceso además de los estándares de entorno de trabajo de la organización.

4. Measurement and Analysis (MA).- Medición y Análisis para adquirir información acerca de la definición de un proceso para medir y analizar procesos.

IPPD Addition

5. Organizational Process Definition +IPPD (OPD+IPPD).- Definición Organizacional de Proceso y IPPD+ para obtener información acerca de la creación de las reglas y guías organizacionales para el IPPD.

2.5 Quantitative Project Management (QPM)

La Administración Cuantitativa de Proyectos es un área de proceso perteneciente al nivel de madurez 4 del CMMI staged. El propósito de esta área de proceso es la administrar de manera cuantitativa el proceso definido para el proyecto para alcanzar los objetivos de calidad y desempeño de proceso establecidos.

La Administración Cuantitativa del Proyecto involucra lo siguiente:

- Establecer y mantener los objetivos de calidad y desempeño de proceso.
- Identificar los sub-proceso aplicables que componen el proceso definido del proyecto, basado en datos históricos de estabilidad y capacidad encontrada en modelos y referencias de desempeño de proceso.
- Seleccionar los sub-procesos del proceso definido de proyecto para ser administrados de manera estadística.
- Monitorear el proyecto para determinar si los objetivos de calidad y desempeño de proceso están siendo satisfechos, e identificar las acciones correctivas apropiadas.
- Seleccionar las métricas y técnicas de análisis a ser utilizadas en la administración estadística de los sub-procesos seleccionados.
- Establecer y mantener el entendimiento de la variación de los sub-procesos seleccionados utilizando las técnicas de medición y análisis seleccionadas.
- Monitorear el desempeño de los sub-procesos seleccionados para determinar si ellos son capaces de satisfacer sus objetivos de calidad y desempeño de proceso, e identificar acciones correctivas.
- Colectar datos estadísticos y de administración de calidad en el repositorio de métricas de la organización.

Para abordar de manera efectiva las prácticas específicas de esta área de proceso, la organización debe tener establecidos sus procesos estándares y los valuartes organizacionales

de proceso, tales como: repositorio organizacional de métricas y la librería de valuartes de la organización para la disposición de cada proyecto en proceso de establecer su proceso definido. El proceso definido es un grupo de sub-procesos que forman un ciclo de vida integrado y coherente para el proyecto. El desempeño de proceso es una medición de cómo los resultados del proceso son alcanzados. El desempeño del proceso es caracterizado por métricas de proceso (esfuerzo, tiempo de ciclo y eficiencia de remoción de defectos) y métricas de producto (confiabilidad, densidad de defectos y tiempo de respuesta). Los sub-procesos son componentes definidos de un proceso definido más grande. Los sub-procesos pudieran ser descompuestos en sub-procesos aun más pequeños de acuerdo a las necesidades. Un elemento esencial de la administración cuantitativa es tener confianza en los estimados (ser capaz de predecir el alcance en el cual, el proyecto puede cumplir los objetivos de calidad y desempeño de proceso). Otro elemento esencial es entender la naturaleza y extensión de las variaciones experimentadas durante la ejecución del proceso, y reconocer cuando la ejecución del proyecto no pudiera ser adecuada para alcanzar los objetivos de calidad y desempeño de proceso. La administración estadística involucra un pensamiento estadístico y el uso correcto de una variedad de técnicas estadísticas, tales como diagramas de corridas, de control, intervalos de confianza, intervalos de predicción, y pruebas de hipótesis. La administración cuantitativa usa datos de la administración estadística para ayudar al proyecto a predecir si será capaz de alcanzar sus objetivos de calidad y desempeño de proceso e identificar cuales acciones correctivas pudieran ser tomadas.

Esta área de proceso aplica a la administración de un proyecto, pero los conceptos encontrados aquí, también aplican a la administración de grupos y funciones. Aplicando estos conceptos a la gestión de grupo no necesariamente contribuiría al logro de los objetivos de negocio de la organización, pero pudiera ayudar a estos grupos y funciones a controlar sus propios procesos.

Los objetivos específicos y las prácticas de esta área de proceso se listan a continuación:

1. SG 1 Quantitatively Manage the Project (Administración Cuantitativa del Proyecto)
 - a. SP 1.1 Establish the Project's Objectives (Establecer los Objetivos del Proyecto).
 - b. SP 1.2 Compose the Defined Process (Componer el Proceso Definido)
 - c. SP 1.3 Select the Subprocesses that Will Be Statistically Managed (Seleccionar los Sub-Procesos que serán Administrados Estadísticamente).
 - d. SP 1.4 Manage Project Performance (Administrar el Desempeño del Proyecto)

2. SG 2 Statistically Manage Subprocess Performance (Administrar estadísticamente el desempeño de sub-procesos)
 - a. SP 2.1 Select Measures and Analytic Techniques (Seleccionar métricas y técnicas de análisis).
 - b. SP 2.2 Apply Statistical Methods to Understand Variation (Aplicar métodos estadísticos para entender las variaciones).
 - c. SP 2.3 Monitor Performance of the Selected Subprocesses (Monitorear el desempeño de los sub-procesos seleccionados).
 - d. SP 2.4 Record Statistical Management Data (Colectar Datos Estadísticos de Administración).

Las áreas de proceso relacionadas son las siguientes:

1. Project Monitoring and Control (PMC).- Referirse al área de proceso de Monitoreo y Control de Proyecto para adquirir información acerca del monitoreo y control del proyecto y tomar las acciones correctivas.
2. Measurement and Analysis (MA).- El área de proceso de Medición y Análisis de Proceso da más información acerca del establecimiento de objetivos medibles, especificando las mediciones y análisis a ser realizado, obteniendo y analizando las métricas y proveyendo resultados.
3. Organizational Process Performance (OPP).- El área de Desempeño Organizacional de Proceso provee información acerca de los objetivos de calidad y desempeño de proceso, análisis del desempeño de proceso, referencias de desempeño de proceso, y modelos de desempeño de proceso.
4. Organizational Process Definition (OPD).- El área de proceso de Definición Organizacional del Proceso da información acerca de los valuartes organizacionales de proceso, incluyendo el repositorio de métricas organizacionales.
5. Integrated Project Management (IPM).- Referirse al área de proceso de Administración Integral del Proyecto para obtener información acerca del establecimiento y mantenimiento del proceso definido para el proyecto.
6. Causal Analysis and Resolution (CAR).- El área de proceso de Análisis y Resolución Causal provee información acerca cómo identificar las causas de los defectos y otros problemas, y tomar acción para prevenir que ocurran en el futuro.

7. Organizational Innovation and Deployment (OID).- Referirse al área de proceso de Innovación y Despliegue Organizacional para adquirir información acerca la selección y despliegue de mejoras que den soporte a los objetivos de calidad y desempeño de proceso.

CAPÍTULO III. Estudio de Campo

Para propósitos de la siguiente tesis se eligió una empresa de desarrollo de software, cuya identidad será omitida con propósitos de confidencialidad, que adopto el modelo de SW-CMM como su estrategia de mejora de procesos de desarrollo de software y actualmente se encuentra certificada como CMM nivel 3.

1. Perfil del Objeto de Estudio

La empresa objeto de estudio, se dedica al desarrollo y producción de equipo electrónico de alto volumen a nivel mundial; Dicho equipo conlleva un alto contenido de software para los microcontroladores incluidos en el producto final. Cabe mencionar que esta es una tendencia global en las empresas del ramo, el hecho de que los equipos electrónicos, de manera incremental, incluyan desarrollo de software para microcontroladores. En el entorno industrial, el desarrollo de software para microcontroladores es conocido como software para “Embedded Systems” y es bastante diverso; por ejemplo, este software puede comprender desde sistemas para controlar los teléfonos celulares, hasta software para un microcontrolador que controle una licuadora o un artículo cuyo control sea menos complejo. En el caso de la empresa de estudio, el software esta orientado a equipos electrónicos utilizados en sistemas automotrices. Es importante mencionar que la producción a un alto volumen, en particular para sistemas automotrices, conlleva algunos requisitos extras que afectan directamente al diseño del software como son:

- Generalmente el software debe ser autónomo y funcionar con prácticamente nula intervención humana, además de auto-diagnosticarse para prevenir y/o notificar cualquier falla inherente a su uso. El sistema debe diseñarse pensando en una autonomía absoluta de al menos 10 años, tiempo que es asociado al tiempo de garantía de la mayoría de los componentes electrónicos del producto, incluido el microcontrolador.
- El sistema debe ser desarrollado de manera que se utilice el absoluto mínimo posible de recursos necesario para su funcionamiento, en términos de hardware, para minimizar el costo del microcontrolador, por ende menos costo de producto que a un alto volumen maximiza ganancias. Este requerimiento en particular, pudiera limitar por ejemplo, el tipo de algoritmos que se pudieran usar para procesar datos, dado que las operaciones matemáticas complejas requieren un procesador de más capacidad, y por lo tanto, de

mayor costo por lo que soluciones más simples (y por lo tanto más baratas) son preferibles, por supuesto, siempre y cuando, se cumpla con los requisitos del sistema.

1.1 Sistemas de Desarrollo

Los

“Embedded Systems” comprenden un universo basto de diferentes combinaciones de memorias, cpu, velocidades etc., en el caso particular de la empresa de estudio, la mayoría de sus sistemas cae dentro del siguiente perfil de microcontrolador:

- CPU 16 bits.
- Velocidad de Reloj 12 Mega Hertz.
- ROM 256 kilo bytes.
- RAM 12 kilo bytes.
- Arquitectura Harvard.

Entre los productos principales se encuentran, entre otros:

- Sistemas de Audio.
- Sistemas de Control de Aire Acondicionado.
- Sistemas de Computadora Central del Vehiculo.
- Sistemas de Control de Motor.
- Sistemas de Detección de Impacto.

Los recursos del microcontrolador permiten el uso de tecnología de desarrollo y arquitectura de software que incluye: Compiladores de “C”, sistemas operativos basados en OSEK, Matemática de Punto Fijo y Flotante, Comunicaciones de Clase B, Modelación y Autogeneración de Código C con Matlab etc.

1.2 Perfil del Personal

La

empresa en estudio cuenta con una amplia cantidad de perfiles necesarios para su operación completa, pero para propósitos de esta tesis, se describirán solamente los perfiles involucrados con la ingeniería de desarrollo de software:

1. Ingeniero de Software.
2. Ingeniero de Pruebas de Software.

3. Ingeniero de Sistemas.

El perfil genérico de los Ingenieros de Software es el siguiente:

- Estudios mínimos Nivel Licenciatura relacionado con áreas de Electrónica y/o Sistemas Computacionales.
- Nivel de inglés conversacional (TOEFL \geq 550) y habilidades de comunicación.
- Desempeño sobresaliente bajo presión, convencido del trabajo en equipo.
- Familiarización con Lenguaje Ensamblador y Experiencia con lenguaje C.
- Familiarización con Equipo de Laboratorio Electrónico.
- Deseable experiencia con Metodologías de Diseño de Software y Sistemas en Tiempo Real.
- Deseable experiencia con Protocolos de Comunicaciones utilizados en la Industria Automotriz.
- Deseable experiencia con Arquitectura de Software para Embedded Systems.
- Deseable experiencia con Sistemas de Calidad y Técnicas sistemáticas de resolución de problemas.
- Disponibilidad de Viajar.

El perfil genérico de los Ingenieros de Pruebas de Software es el siguiente:

- Estudios mínimos Nivel Licenciatura relacionado con áreas de Electrónica y/o Sistemas Computacionales.
- Nivel de inglés conversacional (TOEFL \geq 550) y habilidades de comunicación.
- Desempeño sobresaliente bajo presión convencido del trabajo en equipo.
- Experiencia con Equipo de Laboratorio Electrónico.
- Familiarización con Lenguaje Ensamblador y Experiencia con lenguaje C.
- Deseable experiencia técnicas y equipos de Automatización de Pruebas para sistemas electrónicos.
- Deseable experiencia con Protocolos de Comunicaciones utilizados en la Industria Automotriz.
- Deseable experiencia con Sistemas de Calidad y Técnicas sistemáticas de resolución de problemas.
- Disponibilidad de Viajar.

El perfil genérico de los Ingenieros de Sistemas es el siguiente:

- Formación de al menos 3 años, con desempeño sobresaliente, anteriormente como Ingeniero de Software, Pruebas, Diseño Electrónico, Mecánico, Validación o Manufactura dentro de la organización.
- Estudios mínimos Nivel Licenciatura relacionado con áreas de Electrónica y/o Sistemas Computacionales.
- Nivel de inglés conversacional (TOEFL \geq 550) y amplias habilidades de comunicación.
- Desempeño sobresaliente bajo presión convencido del trabajo en equipo.
- Habilidades de Negociación y Definición de Sistemas.
- Alta Disponibilidad de Viajar.

Estos perfiles son útiles como un filtro inicial de selección de ingenieros, sin embargo para propósitos de contratación o promoción existen procedimientos que incluyen evaluaciones técnicas, de inglés y psicológicas además, eventualmente se realizan una o varias entrevistas donde se determina si el candidato se acerca a las necesidades del puesto requerido. Note que el perfil de Ingeniero de Sistemas requiere que se haya laborado al menos en uno de los puestos relacionados de ingeniería del producto.

A los ingenieros de nuevo ingreso, se les provee una serie de entrenamientos relacionados con la organización así como los relacionados con los procedimientos y sistemas que este tiene que llevar para poder realizar su trabajo de manera efectiva y eficiente. Existen procesos de evaluación de desempeño de manera anual donde se evalúa entre otras cosas: el desempeño técnico, la interacción con el grupo, el seguimiento de los procesos, la efectividad y uso de los entrenamientos, etc.

1.3 Estructura Organizacional

La empresa de estudio es de carácter global con presencia en diferentes países y con diferentes líneas de productos que se manejan con presupuestos independientes. La estructura organizacional que involucra a la ingeniería de software es de tipo matricial. Esto es, el Gerente de Software reporta de manera vertical con el Chief Engineer definido para el producto en particular, o línea de producto, para el cual se desarrollara el software; de manera lateral las

actividades se reportan con el Gerente Global de Software, esto se describe en la siguiente figura:

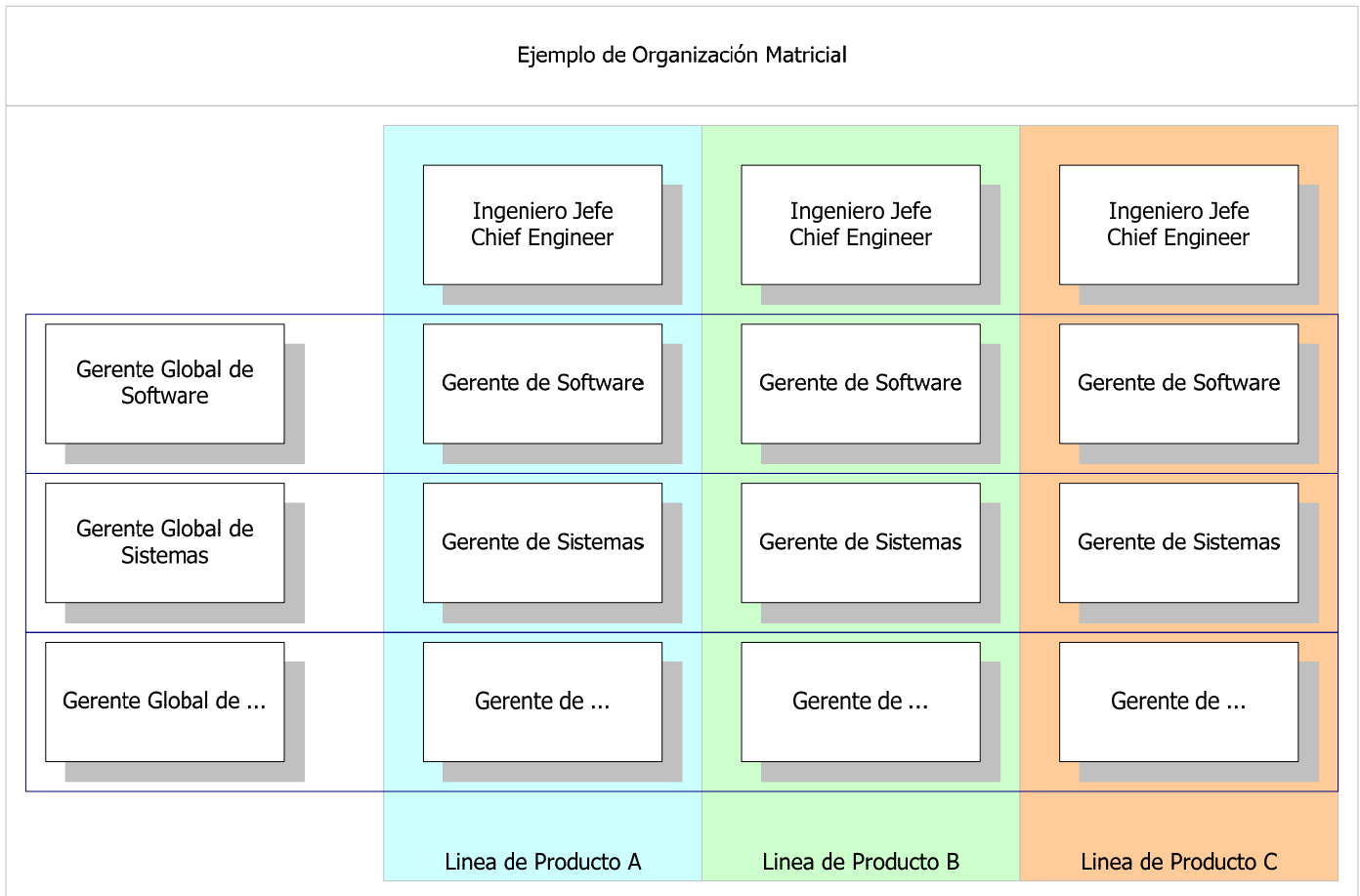


Figura 2 Ejemplo de Organización Matricial

Generalmente las actividades relacionadas a la ingeniería de software relacionadas a la línea de producto, son reportadas con el Chief Engineer, por ejemplo: calendarios, presupuestos, resolución de defectos, algoritmos, diseño, arquitectura etc. de manera similar, las actividades de proceso son reportadas con el Gerente Global de Software tales como Calidad, Métricas, Valuartes de Proceso, Capacitación etc.

Existen ciertas particularidades dependiendo de la línea de producto dentro de la organización, por ejemplo, se puede tener más de un Chief Engineer al cual se reporta, probablemente uno para negocios con norte América y otro para negocios con Asia etc., se puede dar el caso con más de un Gerente de Software que reporta a un solo Chief Engineer etc. estos casos se presentan dependiendo de las necesidades del negocio y la cantidad de proyectos a administrar.

Cada uno de los Gerentes de Ingeniería ya sea software o sistemas cuenta con un staff generalmente de entre 10 a 40 ingenieros dependiendo de la cantidad de proyectos que actualmente este administrando. Cada gerente coordina la administración de cada proyecto asignando a un Ingeniero Líder de su competencia con el que administra y monitorea cada proyecto, la siguiente tabla ejemplifica un la organización de un grupo de ingeniería de software:

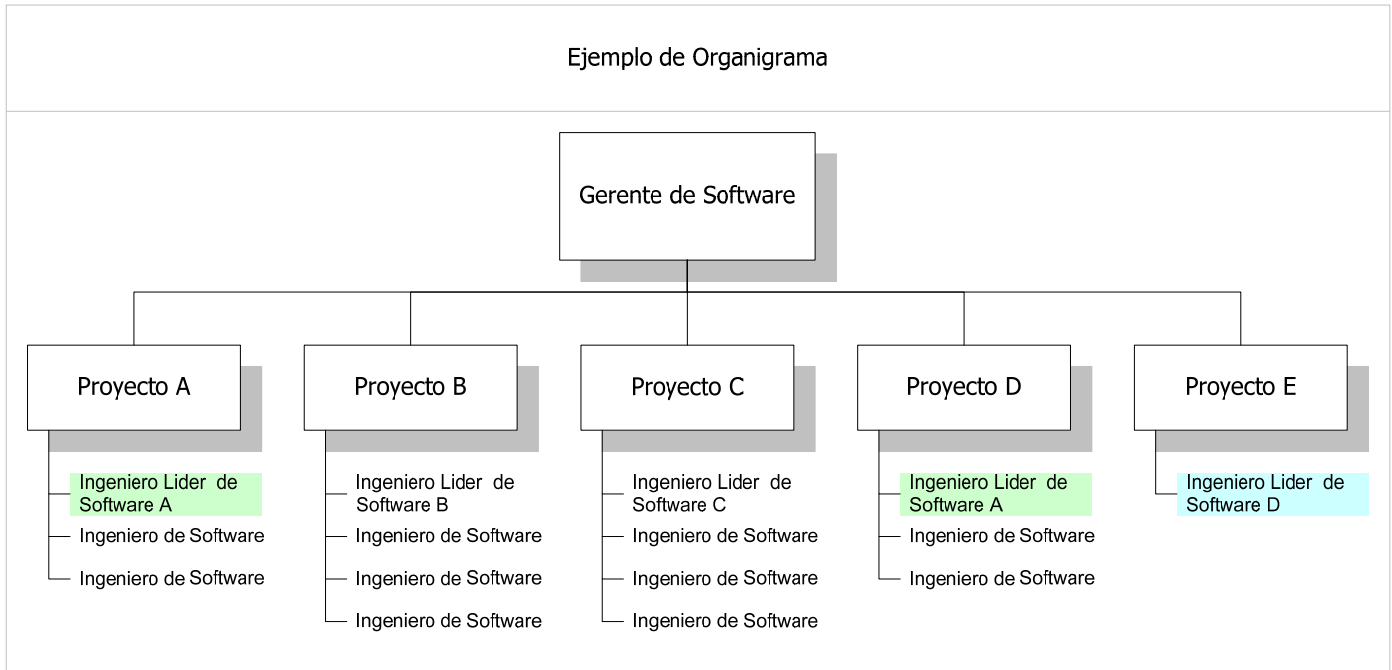


Figura 3 Ejemplo de Estructura Organizacional de Grupo de Software

Note que un ingeniero líder a su vez puede administrar más de un proyecto, y también que un proyecto puede contar solamente con un ingeniero, en este caso este será el ingeniero líder del proyecto.

Los ingenieros asignados a un Gerente de Software no necesariamente estarán localizados en un solo sitio de desarrollo, sino se pueden encontrar en múltiples sitios; dicha asignación de ingenieros localizados en sitios múltiples se determina por los siguientes criterios:

- Centro de mayor experiencia para una tecnología en particular necesaria para el desarrollo del proyecto.
- Disponibilidad de Ingenieros.
- Cercanía con el sitio de desarrollo o producción del cliente.

De manera estratégica el Gerente de Software se encuentra localizado en el sitio principal o donde se concentra la mayoría de las actividades de desarrollo del producto completo, así como

el Chief Engineer. Esta situación es necesaria para poder participar en las actividades estratégicas de la línea de producto, generalmente desde este punto se toma la decisión de donde y cuando se desarrollaran los proyectos de software y quien participará, el entrenamiento, asignaciones necesarias y presupuestos. En el sitio principal se realizan las actividades de cotización de nuevos proyectos de ingeniería de un nuevo producto, involucrando a la ingeniería de software, donde, una vez que se gana un nuevo proyecto, el Gerente de Software identificará al Ingeniero Líder de Software, el staff de software asignado al proyecto así como el sitio o los sitios de desarrollo. Los sitios adicionales son identificados como sitios remotos, generalmente cuentan con un staff más pequeño aunque esta no es la regla. Dentro de los sitios remotos existen Gerentes Locales de Ingeniería de Software los cuales administran a los ingenieros remotos, y trabajan en coordinación con varios Gerentes de Ingeniería de Software de sitios principales.

1.4 Infraestructura

La infraestructura de la empresa de estudio, relacionada al desarrollo de software, al momento de elaboración de la presente tesis comprende varios aspectos que serán descritos a continuación.

1.4.1 Tecnología de Información

Las necesidades de tecnologías de información de la empresa de estudio, para el desarrollo de software incluyen las siguientes instalaciones:

- Red de alta velocidad interconectada con todos los sitios de desarrollo.
- Banco de Servidores distribuidos en los diferentes sitios de desarrollo.
- Bases de Datos para control de Información.
- Sistemas de Seguridad de Información.
- Perfil común, a nivel global, de usuarios, programas y computadoras.
- Correo Electrónico, Teléfono y Sistema de Mensajes Instantáneos.

Además existen instalaciones con acceso mas restringido de acuerdo a las necesidades de cada proyecto, esto incluye: servidores de licencias de programas específicos para el desarrollo tales como compiladores, simuladores, emuladores, comunicaciones con los sistemas

de desarrollo, control de versiones de software, librerías de liberación de software, bases de datos para control de cambios, etc.

1.4.2 Laboratorio Electrónico

Para el desarrollo de software se cuenta con bancos de prueba donde se realiza la verificación del software por parte del equipo de desarrollo. El proyecto cuenta con al menos un banco de prueba, y dependiendo del tamaño del proyecto y número de ingenieros del desarrollo puede incrementarse el número de bancos de prueba. El perfil de un banco de prueba para un proyecto de desarrollo de software es el siguiente:

- Osciloscopio
- Generador de señales
- Fuentes de poder
- Una o varias computadoras, con los programas necesarios
- Simuladores del sistema
- Emulador
- Sistema de comunicaciones con el sistema en desarrollo
- Muestra de Ingeniería del hardware en desarrollo

El equipamiento de un banco de pruebas es costoso, por lo que se busca compartir en la medida de lo posible el mismo. En los sitios de desarrollo generalmente se cuenta con áreas para montar hasta 10 bancos de prueba, lo cual varía dependiendo de la línea de producto y la cantidad de proyectos que se estén desarrollando de manera simultánea.

1.4.2 Proceso

Aunque pudiera resultar infraestructura no tangible, la empresa de estudio cuenta con una infraestructura de proceso que facilita a los ingenieros de desarrollo de software realizar los proyectos de software de acuerdo a un proceso común y cumpliendo con los objetivos del mismo. Dicha infraestructura esta formada principalmente por bases de datos y valuartes de proceso que pueden servir de ejemplo en el cumplimiento del mismo, entre estas se encuentran las siguientes:

- Sistema distribuido para el control de versiones.
- Sistema distribuido para el control de cambios de software.
- Base de Datos donde se alojan las liberaciones de software.

- Librería de Valuartes de Proceso.
- Base de Datos de Lecciones Aprendidas de Ingeniería de Software.
- Base de Datos de Métricas de Software
- Base de Datos de Revisiones entre colegas.
- Base de Datos de Postmortems.
- Programa de Entrenamiento para software.
- Base de datos de Calidad para Software.

Dichos bancos de datos son referenciados desde la documentación de proceso de desarrollo de software de la organización, y están disponibles para toda la comunidad de desarrollo de software a nivel global dentro de la organización, independientemente de la línea de producto.

1.4.2 Subject Matter Experts

La empresa bajo estudio cuenta con “infraestructura” en términos de expertos con al menos 20 años de experiencia en varias áreas relacionadas al desarrollo de software y del desarrollo del producto, aplicado particularmente para sistemas automotrices, por ejemplo:

- Arquitectura de Software (para cada línea de producto).
- Sistemas Operativos.
- Algoritmos (para cada línea de producto).
- Comunicaciones.
- Microcontroladores.
- Compiladores.
- Six Sigma Green Belts y Black Belts.
- Herramientas de Desarrollo.
- Modelación y Auto codificación.

Dichos expertos son consultados en etapas iniciales de los proyectos para poder mejorar el análisis de requisitos y el diseño inicial del sistema, además de buscar incrementar la reusabilidad de software entre líneas de producto. De manera adicional se encargan de proveer

entrenamiento a los ingenieros de manera regular y participar en la identificación de problemas particularmente complejos para los sistemas en desarrollo.

1.5 Interoperabilidad con otras Ingenierías

La interoperabilidad de la ingeniería de software dentro de la empresa de estudio es principalmente con los grupos de ingeniería de sistemas (encargada de proveer y aclarar los requisitos del sistema) y el grupo de pruebas independientes de software (encargados de verificar el cumplimiento de los requisitos del sistema). Esto se describe en la siguiente figura:

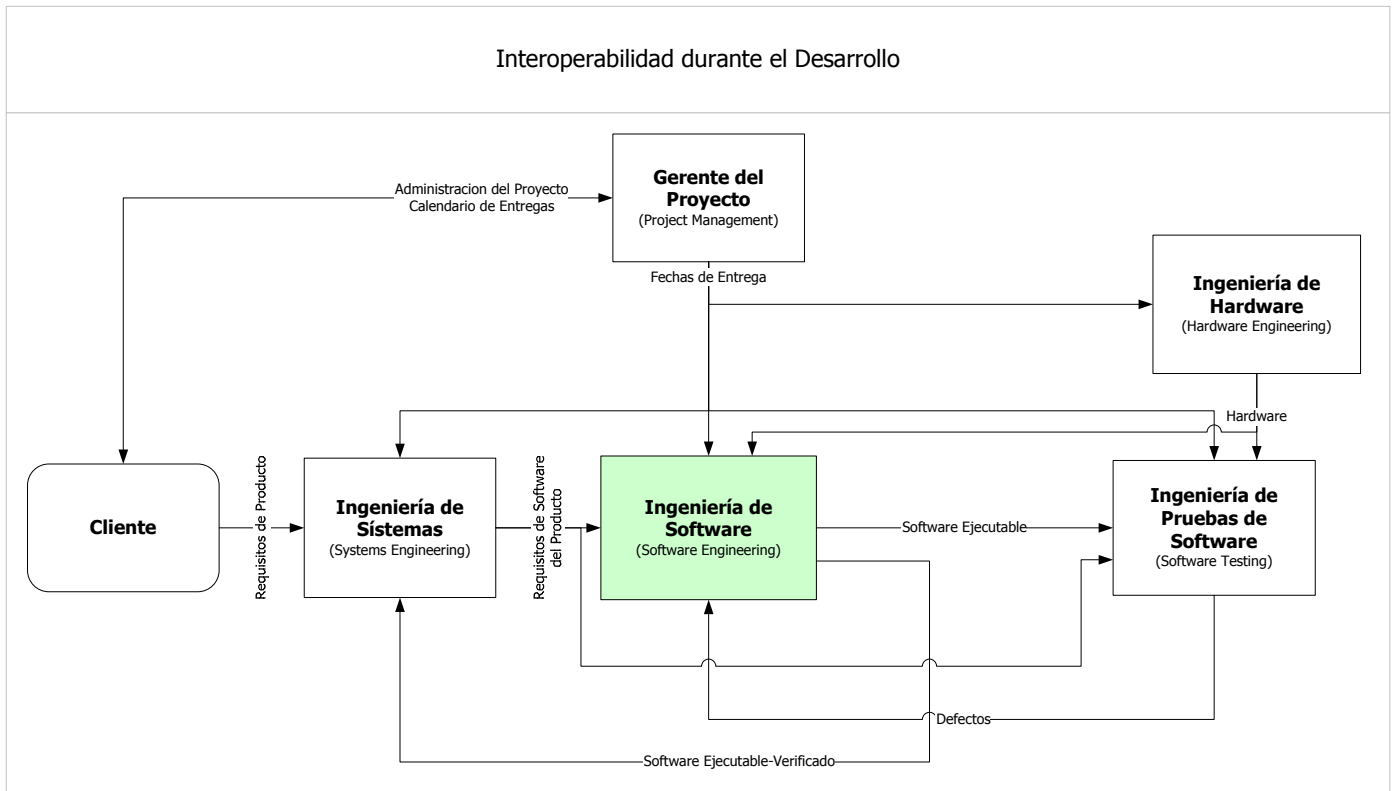


Figura 4 Interoperabilidad

De manera similar existe una interoperabilidad con el Gerente de Proyecto o Project Manager, a fin de sincronizar la planeación del desarrollo de software con las actividades del desarrollo global del producto.

Los clientes de la empresa cuentan con dos interfaces principales: una interfaz técnica, y una interfaz administrativa:

- La interfaz administrativa, o Gerente de Proyecto es responsable de los aspectos de finanzas, negociación de entregas, producción, embarques materiales etc. se encarga de colocar el calendario global del proyecto al cual todos los demás grupos de ingeniería se

deben alinear para lanzar la producción. El Gerente de Proyecto consulta a los grupos de ingeniería, incluida la de software, para proveer estimados de entrega y contenidos parciales.

- La interfaz técnica es realizada por el ingeniero de sistemas. Principalmente se encarga de las actividades de recolección, selección, documentación y control de requisitos, para todas las áreas de ingeniería que participan en el desarrollo del producto. Dentro de los requerimientos documentados del producto, se encuentran los requerimientos correspondientes a la ingeniería de software.

El trabajo del grupo de ingeniería de software inicia a partir de un documento de requisitos definidos y acordados con el cliente, provisto por el grupo de ingeniería de sistemas. La actividad inicial es el análisis de requisitos, dicho análisis permite la planeación del desarrollo de software y eventualmente su diseño, codificación, pruebas y liberación. Una vez liberado el software ejecutable, el equipo de ingeniería de pruebas independientes se encarga de verificar que el software ejecutable cumpla con los requisitos documentados. Para lograr este objetivo, el grupo de ingeniería de pruebas de software prepara procedimientos de pruebas, paralelamente al desarrollo del software, a partir del mismo documento de requisitos de software que el grupo de ingeniería de sistemas provee. Una vez que la verificación termina, el grupo de ingeniería de software es responsable de reparar los potenciales defectos que el grupo de pruebas independientes identifique. En el caso de que se encuentren defectos de implementación en el software ejecutable que no cumplan con requisitos documentados se planea una nueva liberación de software, para corregir los defectos reportados. En este evento, el grupo de pruebas independientes realiza pruebas de regresión para asegurar que se removieron los defectos y no se afectó la funcionalidad. Una vez terminado este ciclo es cuando el cliente final puede recibir la versión de software que se liberó. En este punto el ingeniero de sistemas verificará que el software ejecutable cumple con los requisitos documentados ahora ejecutándose en el entorno de trabajo del cliente, que generalmente es un vehículo prototipo.

Este proceso regularmente cuenta con algunas complicaciones que los grupos de ingeniería deben que sobreponer, estos son algunos ejemplos:

- Los requisitos generalmente no están completos desde el inicio del proyecto, por lo cual se realizan liberaciones parciales.
- El cliente no necesariamente está seguro de lo que quiere o necesita por lo cual las liberaciones parciales resultan en cambios de requisitos.

- El ciclo de Análisis de Requisitos efectuado por el grupo de software y de pruebas independientes identifica clarificaciones, actualizaciones y/o correcciones que el grupo de sistemas requiere resolver, lo cual afecta la planeación de software y de pruebas independientes.
- Las pruebas de software reportan defectos que no se pueden reproducir por parte del grupo de software. Estos defectos potenciales resultan defectos en el procedimiento de ejecución de pruebas o ambigüedad en los requisitos.
- El entorno de trabajo del cliente tiene diferencias con el entorno de pruebas de ingeniería de software o de pruebas independientes, resultando en defectos no identificados antes que el cliente los vea.

La interoperabilidad con otros grupos de ingeniería es independiente de sitio donde se este desarrollando el software, es decir, puede darse el caso que el ingeniero de sistemas, software y pruebas de software se encuentren distribuidos en sitios de desarrollo diferentes.

1.6 Clientes

Los clientes de la empresa de estudio son primordialmente productores de vehículos tales como General Motors, Ford, Chrysler, Volkswagen, Toyota, Fiat, Honda etc. La diversidad de clientes y productos de la empresa, hace que los proyectos de desarrollo así como sus procesos permitan flexibilidad a fin de cumplir con las expectativas de los clientes.

Algunos clientes comparten ciertos sistemas, estándares y tecnologías, y generalmente están distribuidos por regiones por ejemplo: CMMI es generalmente más promovido por empresas norteamericanas, mientras que SPICE es más popular en Europa.

En el caso de la ingeniería de software, además de los clientes “externos” también es necesario cumplir requisitos para clientes internos dentro de la organización. Algunos de los clientes internos para la ingeniería de software son los siguientes: ingeniería de pruebas de software de software, ingenieros de calibración, ingeniería de manufactura, ingeniería de hardware etc.

1.7 Índices Actuales de Operación

Los índices generales de operación del área de desarrollo de software, son representativos de una

empresa de nivel CMM nivel 3, donde se puede observar un proceso institucionalizado donde los ingenieros usuarios del proceso de desarrollo de software, están concientes de los requerimientos del proceso para el desarrollo de proyectos de software, los planes requeridos, documentación, recolección de métricas, estándares, y herramientas de desarrollo. Se puede identificar que la organización esta acostumbrada a recolectar métricas de diferentes tipos, de producto y proceso como parte de sus actividades de desarrollo, y esta iniciando el proceso de aprender a utilizar dichos datos de manera estadística para la toma de decisiones en proyectos subsecuentes, a fin de mejorar la planeación y estimados de desarrollo de los proyectos de software.

Como objetivo a mediano plazo, por parte del área de ingeniería de software, se tiene contemplado lograr que el proceso evolucione para cumplir con los estándares de SPICE, a fin de cumplir con uno de los requisitos de los clientes europeos. Asimismo se tiene el objetivo de obtener la certificación del proceso de desarrollo de software ahora bajo el estándar de CMMI involucrando a las áreas de ingeniería de sistemas (Systems Engineering) y administración de proyectos (Project Management).

1.8 Aseguramiento de Calidad de Software

Debido a que los procesos de desarrollo de software están diseñados bajo el modelo de CMM, el sistema de calidad también se encuentra diseñado para auditar bajo las áreas claves de proceso de Aseguramiento de Calidad de Software y Administración de la Calidad de Software de CMM.

El sistema de calidad se fundamenta en dos actividades específicas de calidad a efectuarse de manera periódica en los proyectos de software:

- **Revisión de Actividades de Proyecto.-** El objetivo de estas revisiones de calidad es verificar que los procesos descritos para el desarrollo de software se ejecutan de manera correcta y con los elementos adecuados, justo en el momento en que ocurren las actividades. Solamente en la planeación inicial del proyecto no hay revisiones de actividades relacionadas al proceso de planeación del proyecto debido a que la definición del plan de calidad depende del plan de desarrollo de software. En la practica se observa un trabajo preliminar cercano entre en ingeniero líder del proyecto y el ingeniero de aseguramiento de calidad de software a fin de definir de manera precisa entre otras

principalmente la sección 5 de *Proceso Definido de Desarrollo de Software para el Proyecto*.

- Auditorías.- Las auditorías de calidad se enfocan en la revisión de los productos terminados de los diferentes pasos del desarrollo de software para asegurar que cumplen con las características requeridas por el proceso desarrollo de software.

Los formatos y cuestionarios utilizados por los ingenieros de aseguramiento de calidad de software son comunes para todos los proyectos de software, independientemente de la línea de producto o del cliente al cual se le este desarrollando el proyecto de software. Dichos cuestionarios se encuentran alojados en una base de datos de aseguramiento de calidad de software la cual cumple con la función de centralizar la información, como generar reportes estadísticos para tomas de decisión. Esto reduce la variabilidad de criterio en las evaluaciones a través de la organización y en los diferentes sitios de desarrollo.

Las actividades de calidad son planeadas y calendarizadas para cada ciclo de desarrollo de software por parte del Ingeniero de aseguramiento de calidad de software. El plan de aseguramiento de calidad es compartido, revisado y autorizado por el Ingeniero Líder de Software así como el gerente de software del proyecto al que se aplica las actividades de aseguramiento de calidad de software. Los resultados de estas auditorías se almacenan en una base de datos a fin de lograr información estadística así como reportes de seguimiento a las fallas de ejecución más comunes de los procesos.

En caso de identificarse no-conformidades al proceso durante las auditorías o revisiones de actividades de proyecto se levantará un reporte de no-conformidad. Estos reportes de no-conformidades deben ser corregidos en no más de 30 días calendario, entendiendo que para cerrar el reporte se debe corregir todas las no-conformidades documentadas en el mismo. Si las no-conformidades no son corregidas dentro de este plazo se procede a un “proceso escalación” que involucra a gerentes de más altos niveles con el fin de acelerar la corrección de las no-conformidades, o explicar a gerentes de más altos niveles, la razón de por que no se han cerrado así como el plan para hacerlo.

La información recolectada de métricas de calidad de software se analiza con herramientas como análisis de causa efecto para mejorar las áreas del proceso de desarrollo de software donde se presenta no-conformidades de manera más recurrente.

Algunas veces se realiza soporte de actividades de calidad para auditores externos solicitadas por los clientes. Para dichas actividades se prepara la documentación pertinente y se requiere un aviso anticipado por parte del cliente, indicando el alcance de la auditoria externa. Las auditorías externas regularmente son requeridas por los mismos clientes antes de otorgar el desarrollo de un producto a la empresa, razón por lo cual son de alta importancia y se toman muy en serio.

2 Metodología de Administración de Proyectos

La siguiente sección describe la metodología y procedimientos que se utilizan actualmente la empresa bajo estudio para la administración de proyectos de software en todos sus proyectos globalmente.

El procedimiento esta dividido en tres partes:

- El inicio del proyecto y las actividades iniciales de planeación
- La ejecución del plan de desarrollo de manera iterativa.
- El cierre del proyecto.

El proceso de planeación de proyecto de software, es el segundo proceso que se ejecuta al iniciar un nuevo proyecto, y es ejecutado cíclicamente (en algunos de sus pasos) durante la vida del proyecto principalmente durante cambios significativos en el proyecto. El principal insumo para la planeación inicial del desarrollo de software es la definición de los requerimientos de software del producto, que como se mencionó anteriormente, es provisto por parte de los ingenieros de sistemas, de ahí que el primer procedimiento en ejecutarse, es el de Administración de Requisitos, donde uno de sus pasos es análisis de requerimientos. El análisis de requisitos se ejecuta inicialmente con el fin de obtener un entendimiento común entre lo que el cliente solicita y lo que el equipo diseñara a fin de poder definir el plan de desarrollo.

Dentro del proceso de planeación de software el actor principal es identificado como el Ingeniero Líder de Software, sobre el cual recae la responsabilidad de este procedimiento y es el encargado de definir el plan de desarrollo del proyecto. El perfil de un ingeniero líder de proyecto es básicamente el mismo al de un ingeniero de software, pero con cualidades adicionales de comunicación, liderazgo, negociación, experiencia técnica y el reconocimiento de su equipo entre otros. El Ingeniero Líder de Software es la interfaz principal con el gerente de software para la administración y monitoreo del proyecto de software.

El procedimiento de planeación de proyectos de software se muestra en la siguiente figura:

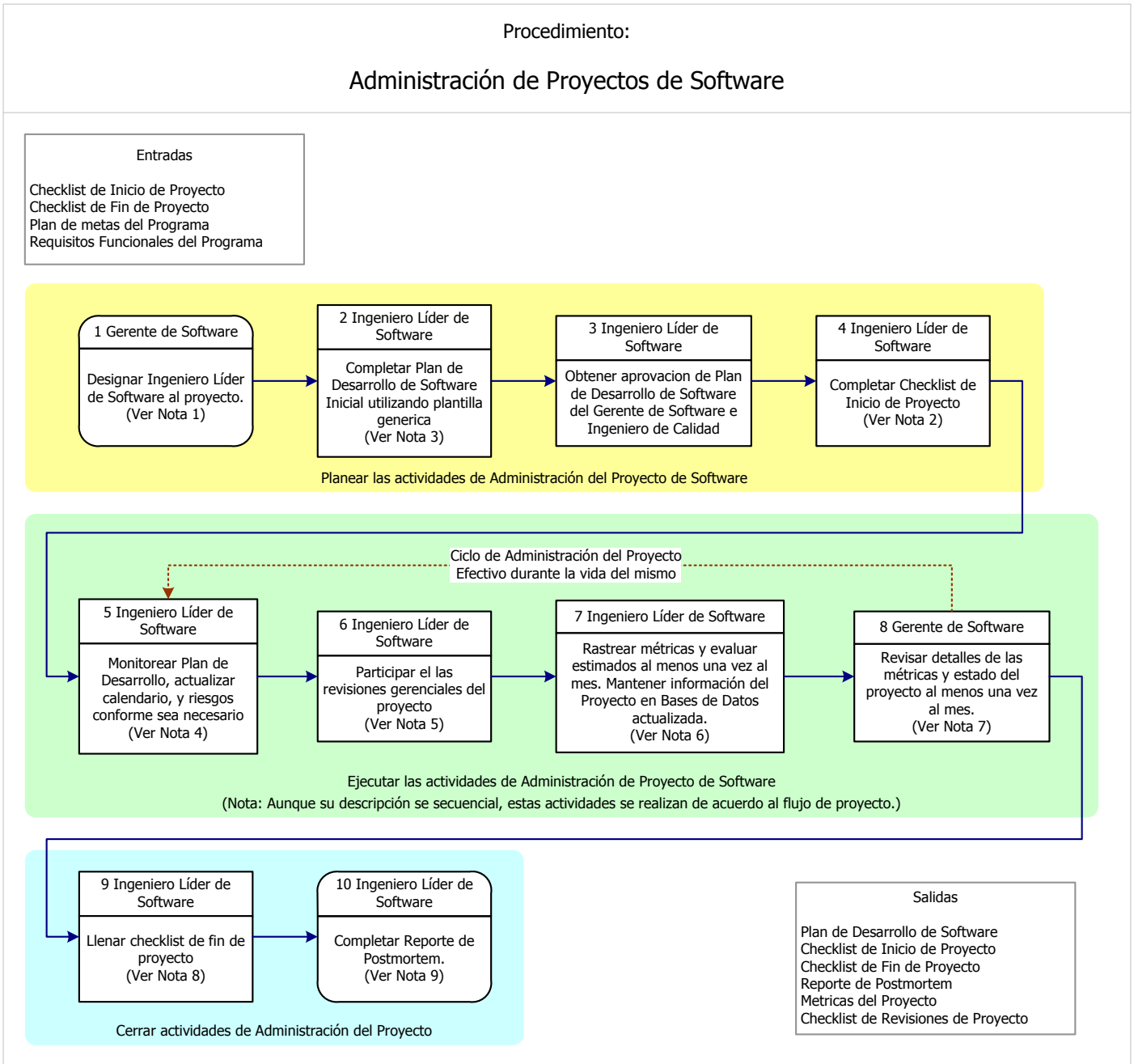


Figura 5 Proceso de Administración de Proyectos de Software

Notas:

1. El Ingeniero Líder de Software será responsable por la administración del proyecto de software. De manera adicional a este rol, en este paso se identificarán al Administrador de la Configuración de Software, y al Ingeniero de Aseguramiento de Calidad de Software.

2. Refiérase a la librería de valuartes de proceso para obtener la plantilla de checklist de Inicio de Proyecto de Software. Dicha plantilla contiene la mayoría de las tareas de inicio de proyecto, sin embargo, tareas adicionales relacionadas a la línea de producto pueden ser agregadas.
3. Refiérase a la librería de valuartes de proceso para obtener la plantilla del Plan de Desarrollo de Software. Todas las secciones deben ser llenadas, el ingeniero líder del software no debe omitir sección alguna.
4. El Ingeniero Líder de Software deberá abordar los problemas ínter-competencia durante las juntas de equipo del proyecto con el Gerente del Proyecto. Todos los riesgos categorizados con un alto impacto deberán ser monitoreados durante esta junta.
5. De acuerdo con el procedimiento de desarrollo de productos, se programaran varias revisiones de diseño durante la vida del desarrollo del producto. Refiérase a la librería de valuartes de proceso para obtener el checklist de revisiones de diseño de software en preparación para dichas revisiones de diseño. Todos los puntos de checklist no acreditados deberán ser justificados durante la exposición del estado del diseño del software en estas revisiones.
6. Considere el uso de indicadores, métodos estadísticos, y reportes disponibles en la base de datos para evaluar el desempeño del proyecto y actualizar sus estimados.
7. Cuando las revisiones de estado del proyecto son ejecutadas, se sugiere los siguientes puntos en la agenda:
 - a. Revisión de actividades completadas desde la última revisión.
 - b. Revisión de las actividades planeadas en el siguiente periodo (regularmente semanal).
 - c. Rastreo del calendario, progreso técnico, y estado de los recursos críticos de microcontrolador contra los estimados.
 - d. Rastreo de dependencias críticas en el calendario con otros grupos de ingeniería.
 - e. Actividades de prevención de defectos.
 - i. Comunicación de lecciones aprendidas.
 - ii. Monitoreo de Riesgos.
 - iii. Comunicación de mejores prácticas.

iv.

as de verificación del software.

- f. Problemas donde se requiera ayuda para resolver.
 - g. Otros problemas técnicos.
8. Refiérase a la librería de valuartes de proceso para obtener la plantilla de checklist de Fin de Proyecto de Software.
9. El postmortem del proyecto debe ser conducido como mínimo al momento de completar el proyecto. Postmortems intermedios pueden ser ejecutados en cualquier fase del desarrollo bajo discreción del Ingeniero Líder de Software. La convocatoria del postmortem debe ser enviada al menos 24 hrs. antes de la junta. Los objetivos del postmortem son los siguientes:
- a. Discusión y documentación de lo que se hizo bien.
 - b. Discusión y documentación de lo que se hizo mal.
 - c. Revisión de los objetivos del programa y si alguno no se cumplió, discutir y documentar por que no se cumplió.
 - d. Revisión de los estimados iniciales contra los resultados finales, discutir las razones de las discrepancias.
 - e. Lecciones aprendidas.
 - f. Sugerencias de mejoras para proyectos futuros.

2.1 Inicio de Planeación de Proyecto

El desarrollo del un proyecto de software es un proceso íntimamente relacionado al desarrollo de un producto como un todo; el proceso inicia cotizando proyectos para diferentes clientes, en donde el desarrollo de software es parte de la cotización. Una vez que un proyecto es negociado con algún cliente, la actividad inicial reside en el área de ingeniería de sistemas, quienes trabajarán con el cliente para obtener los requerimientos específicos del producto.

Una vez que se tiene un nivel de requerimientos de software suficiente para iniciar el proyecto se procede a ejecutar el Análisis de Requisitos con el objetivo inicial de establecer el Plan de Desarrollo de Software. El primer paso es entender el alcance y objetivo del proyecto a fin de ser capaz de definir el plan de desarrollo, estimar recursos necesarios y calcular tiempos. El tamaño del equipo de software esta asociado al tamaño, fechas de entrega objetivo, complejidad y presupuesto asignado para el proyecto; así que generalmente esta actividad se determina en

conjunto con el Gerente de Software. Durante la definición del equipo de desarrollo de software asignado al proyecto se pueden identificar recursos que se encuentren localizados en diferentes sitios de desarrollo, que debido a su experiencia en áreas necesarias y específicas para el proyecto, será necesario considerarlos.

2.1.1 Plan de Desarrollo de Software

La documentación del Plan de Desarrollo de Software se lleva al cabo por medio de una plantilla genérica de planeación que ya incluye los campos necesarios a considerar para el desarrollo. La plantilla genérica contiene los siguientes campos a llenar por parte del Ingeniero Líder de Software:

1. Introducción.- En esta sección se hace referencia a la empresa, proceso de desarrollo de software utilizado, y una descripción del alcance del proyecto. De manera adicional se incluye información general de identificación del proyecto, incluyendo el cliente, año modelo del vehículo, lenguaje de programación utilizado, microcontrolador, identificación en la base datos del proyecto.
2. Visión General.- En esta sección se describe de manera genérica las funcionalidades principales del proyecto, y en caso de existir funcionalidades de terceras partes, es necesario describir las mismas y quien la proveerá.
 - 2.1. Software adquirido por Cliente, Vendedor, Tercero, Contrato.- En esta sección se identifica los módulos obtenidos por medio de la subcontratación de servicios de software para el proyecto por parte de empresas externas a la organización.
 - 2.2. Entregables.- Esta sección detalla los productos del desarrollo de software que se acordaron para ser incorporados en el paquete de entrega al cliente.
3. Personal, Responsabilidades y la Coordinación Intergrupar.- Esta parte del plan identifica los diferentes sitios participantes en el desarrollo así como el sitio líder de desarrollo y quienes son los sitios de soporte. De manera adicional describe las responsabilidades primarias de cada disciplina de ingeniería que influye en el desarrollo del software (por ejemplo, ingeniería de sistemas, ingeniería electrónica, validación etc.).
 - 3.1. Roles y Responsabilidades.- Incluye una lista de los diferentes participantes y sus responsabilidades dentro del desarrollo del proyecto de software. Dentro de esta lista se identifican roles específicos y grupos de toma de decisión que afectan al proyecto de desarrollo de software:

- Ingeniero Líder de Software
- Ingeniero de Software
- Arquitecto de Software
- Gerente de Software
- Administrador de la Configuración de Software
- Buró de control de cambios de software
- Ingeniero residente
- Ingeniero de Pruebas
- Ingeniero de Sistemas
- Ingeniero de Hardware
- Ingeniero de Aseguramiento de Calidad
- Cliente (voz del cliente)
- Administrador de Proveedor de Software.
- Buró de Revisión Experta

3.2. Personal.- Esta sección documenta la información de contacto de cada uno de los participantes descritos en la sección anterior. Un rol puede tener mas de un ingeniero asignado, por ejemplo el rol de Ingeniero de Software, sin embargo existen roles que regularmente existe solamente un responsable por proyecto como el Ingeniero Líder de Software.

4. Estimados de Software.- Esta sección describe los métodos y las suposiciones (experiencia, complejidad, herramientas etc.) que se han tomado para hacer el estimado inicial del desarrollo del software, algunos ejemplos de métodos de estimación pueden ser: Análisis histórico de proyectos similares, Cocomo, y algunas metodologías internas de la organización. Los estimados son almacenados en la base de datos de métricas donde eventualmente se les da mantenimiento y seguimiento mes con mes.

5. Proceso Definido de Desarrollo de Software para el Proyecto.- Esta es una de las secciones principales del plan de desarrollo ya que describe un proceso de desarrollo específico para el proyecto que se ajusta a las necesidades y prácticas particulares del cliente, por ejemplo, si el cliente solicita realizar una prueba específica, esta se documenta en esta sección para su eventual ejecución. Esta sección describe el flujo de las tareas y las actividades que deben llevar al proyecto al desarrollo exitoso del software. Cada uno de estos pasos del

proceso de desarrollo (planeación, análisis de requisitos, diseño, codificación, pruebas) se asocian a un ciclo de desarrollo de software tales como: Cascada, Espiral, etc. La descripción de cada paso incluye la siguiente información:

- Documentos generados en la etapa específica de desarrollo.
- Metodología de verificación de los productos de trabajo (pruebas unitarias, revisiones entre colegas, pruebas de integración etc.).
- Identificación y localización física de la documentación, ya sea electrónica o impresa.
- Control de cambios y los procedimientos de liberación de software. Esta sección describe, para cada etapa del proceso, la forma en que los artefactos del proyecto serán administrados y controlados, es decir el Plan de Administración de Configuración.

6. Proceso de Desarrollo y Metas de Producto y Proceso.- Esta sección define metas específicas para la mejora continua tanto en proceso como en producto. Se describe la meta seleccionada, el tipo, unidad medible, estimado original y algún comentario relacionado. Entre las metas más comunes se encuentra, tiempo total de desarrollo, total de cambios de software por errores de implementación, tiempo de respuesta etc. La siguiente tabla ejemplifica los objetivos de un proyecto.

Tipo de Objetivo	Parámetro	Objetivo	Estimado	Comentario
Proceso	Personal-Mes	< 12 Meses	15 Meses	El desarrollo se puede expeditar utilizando herramientas CASE
Proceso	Defectos	< 20	30	
Producto	RAM	< 10k bytes	12k bytes	
Producto	ROM	< 100k bytes	110k bytes	

Tabla 2 Ejemplo de Objetivos

7. Administración de Riesgos.- Esta sección lista los riesgos principales que pueden afectar el desempeño del proyecto de software. El siguiente es el formato genérico de documentación de cada riesgo identificado:

Riesgo # <>	
Severidad de Riesgo	<Alta, Media, Baja>
Descripción de Riesgo	<¿Qué es lo que puede salir mal?>
Medición del Riesgo	<¿Cómo puedo medir el riesgo?>
Señal de Riesgo	<¿Cómo saber cuando tomar alguna acción?>

Mitigación del Riesgo	<¿Qué puedo hacer para prevenir el riesgo?>
Contingencia del Riesgo	<¿Qué debo hacer para contener el riesgo?>.
Persona Responsable	<¿Quién es responsable de dar seguimiento al riesgo? >
Fecha de Identificación	
¿Qué puede hacer la gerencia para mitigar el riesgo?	

Tabla 3 Documentación de Riesgos

8. Reporte de Progreso.- Esta sección describe los diferentes reportes y/o juntas a ser utilizados, a lo largo del desarrollo, para rastrear el progreso del desarrollo del proyecto, la siguiente tabla ejemplifica algunas de los eventos utilizados:

Reporte o Junta	Contenido de Reporte o Agenda	Lista de Distribución o localización de minutas	Persona Responsable de Reportar o asistir a Junta	Frecuencia de Reporte o Junta
Notificación de Liberación de Software	Descripción de contenido, versión	<Gerente de Software, Ingeniero de Calidad, Ingeniero de Pruebas, Ingeniero de Sistemas>	Ingeniero Líder de Software	Cada Liberación
Actualización de Métricas	Predefinido por la base de datos	Base de Datos de Métricas	Ingeniero Líder de Software	Mensual
Revisión de estatus de Proyecto con el Gerente de Software	De acuerdo a agenda sugerida	<Gerente de Software, Ingeniero Líder de Software >	< Gerente de Software >	<Mensual>
<Reporte de Actividades>	<Actividades Completas, Actividades Planeadas, Problemas. >	< Gerente de Software, Ingeniero Líder de Software >	< Ingeniero Líder de Software >	<Bi-Semanal>
<Reporte de Estatus de Proyecto>	<Calendario, Riesgos, Staff>	< Gerente de Software, Ingeniero Líder de Software >	< Ingeniero Líder de Software >	<Mensual>
<Junta de Desarrollo del Producto>	<Calendario, Riesgos, Planes>	< Gerente de Software, Ingeniero Líder de Software >	<Gerente de Proyecto>	<Semanal>

Tabla 4 Lista de Reportes de Progreso de Proyecto

9. Herramientas e Instalaciones.- Esta sección describe las herramientas de software necesarias para completar el proyecto de software, las bases de datos requeridas (métricas, calidad, administración de versiones, liberaciones etc.), herramientas de escritorio y formato,

herramientas CASE para Análisis, Diseño, Codificación u Auto-codificación etc., Computadoras, Laboratorios, Simuladores. En el caso de desarrollos multi-sitio esta descripción se hace necesaria para cada uno de los sitios participantes en el desarrollo.

10. Entrenamiento Específico para el Proyecto.- Esta sección identifica el entrenamiento que los ingenieros requieren para desarrollar correctamente el proyecto. Este plan incluye, como, cuando y de quien se espera el entrenamiento.

Clase	¿Quién es requerido a tomar el entrenamiento?	El entrenamiento es requerido para ¿Cuándo?	¿Cómo debe ser provisto este entrenamiento?	¿Cómo debe ser verificado el entrenamiento?
Métodos de Diseño	Ingeniero 1,....			
Lenguaje X	Ingeniero 1,...			

Tabla 5 Necesidades de Entrenamiento

11. Calendario del Proyecto.- Este calendario debe estar alineado con el calendario general del producto que controla, negocia y define el Gerente de Proyecto. El calendario es un llamado documento vivo por lo que generalmente se mantiene en un archivo independiente y se actualiza conforme progresa el proyecto.

12. Administración de Subcontratación de Software.- Esta sección lista las actividades que el administrador de subcontratación de software realizará para el proyecto. Esta sección generalmente no es utilizada. Esta sección cuenta con los siguientes apartados:

- 12.1. Información de Contactos.
- 12.2. Reporte de Progreso de Subcontratistas.
- 12.3. Criterio de Aceptación.
- 12.4. Plan de Aseguramiento de Calidad, esta sección hace referencia al plan de aseguramiento de calidad que define el ingeniero de calidad que es asignado al proyecto.

13. Estándares Organizacionales y Documentos de Referencia

14. Apéndices.

- 14.1. Alojamiento de Registros de Calidad.
- 14.2. Información de Acrónimos.

Se puede observar en el contenido del Plan de Desarrollo de Software que no se tocan aspectos financieros del proyecto, esto se completa por parte del Gerente de Proyecto por lo que quedan fuera del alcance de la administración del proyecto de software.

2.2 Ejecución del Plan de Desarrollo de Software

Una vez que el plan es autorizado y comunicado al equipo de desarrollo, el equipo procede a ejecutarlo de acuerdo a lo que se encuentra documentado principalmente en la Proceso Definido de Desarrollo de Software para el Proyecto de la plantilla del Plan de Desarrollo de Software. Dicha ejecución es administrada por el Ingeniero Líder de Software y con seguimiento por parte del Gerente de Calidad. Las actividades de calidad se alinean con el plan de desarrollo. Generalmente se planean liberaciones alpha, beta etc. con intervalos de 2 a 3 meses a fin de permitir la ejecución de pruebas por otras competencias de ingeniería y el ejercicio parcial del producto por parte del cliente. Esta condición es generalmente favorable para el desarrollo, dado que permite evaluar el progreso del desarrollo en términos de funcionalidad, sin embargo, da entrada al cambio requisitos de manera más frecuente por parte del cliente que derivan en la re-planeación y re-estimación de calendarios. Estas condiciones generalmente derivan en re-trabajo:

- Ingeniería de sistemas en administrar los nuevos requisitos además de los cambios de requisitos actuales,
- Ingeniería de pruebas de software a generar nuevos casos de prueba o re-diseñar los actuales.
- Ingeniería de software a cumplir con los contenidos planeados para las entregas parciales además de incorporar los nuevos requerimientos por parte del cliente.

La gran mayoría de los proyectos de software operan en este contexto de implementación, de cambios de requisitos y análisis y resolución de reportes de falla como entradas para los ciclos de desarrollo de software, la mayor parte de la vida del mismo. Conforme el proyecto evoluciona los cambios de requisitos se decrementa paulatinamente, de la misma manera las necesidades de recursos del proyecto, es decir, el personal de inicio de los proyectos regularmente resulta grande a fin de soportar toda la carga de tiempo de ingeniería para atender las necesidades del proyecto, conforme el proyecto madura, estas necesidades se van decrementando y entonces el personal del proyecto es sujeto a reducción o re-asignando a otros proyectos.

Es importante hacer notar que en el procedimiento se define que los pasos 5-8 son cíclicos con un periodo determinado, lo que implica que el Ingeniero Líder de Software esta periódicamente revisando estados, actualizando métricas y bases de datos además de preparando presentaciones y checklist para atender las diferentes revisiones del proyecto.

2.3 Cierre de Proyectos de Software

La actividad de cierre de proyectos se conoce como Cierre y Transferencia de proyectos. Esta actividad denota una fecha en la cual ningún cambio adicional se autoriza (a menos que sea algún defecto de garantía o represente una ganancia importante a la organización) por lo cual no existe personal asignado en presupuesto para su mantenimiento. Es requisito mantener los registros de calidad almacenados por al menos 15 años de manera tal que esta información permanece disponible para consulta. Se ejercita un checklist de cierre para asegurar que los registros del proyecto están completos y cerrados, y que se han enviado las notificaciones apropiadas.

Existen casos de cierre de proyectos debido a que el cliente decide terminar con el proyecto antes de su término. Esto puede obedecer a varias causas, cambio de tecnología, falta de presupuesto, fallas en la operación, ruptura de relaciones con clientes, mejores alternativas, estas ultimas son celosamente cuidadas por los niveles directivos de la organización para evitarlos.

3 Resultados Actuales

La implantación del procedimiento de planeación de proyectos de software ha permitido a la competencia de ingeniería de software establecer un lenguaje común en términos de administración de proyectos de software independientemente del (los) sitio(s) donde se lleve el desarrollo, llevar registros de operación y métricas de desempeño, utilizar técnicas de estimación, definir calendarios de actividades, identificar a los actores principales del proyecto, identificar procedimientos y documentación necesaria etc. La ejecución del procedimiento por si solo, ha contribuido a identificar las necesidades mínimas del proyecto para planearse y ejecutarse. La sola ejecución del plan contribuye a seguir un orden más eficiente de acciones en el desarrollo del proyecto, sin embargo, no garantiza la efectividad de las mismas. El Plan de Desarrollo de Software define la referencia de ejecución del proyecto de una manera sistemática para cumplir

con los requisitos del proyecto, sin embargo, no mejora la exactitud y calidad de los requisitos mismos.

En resumen el procedimiento de planeación de proyectos de software cubre las necesidades de la organización para iniciar, ejecutar y cerrar un desarrollo de software. Establece un plan de desarrollo adaptado a las necesidades específicas del proyecto, buscando alinear su calendario con el plan general de desarrollo del producto.

Con el fin de entender el grado de implantación del procedimiento de administración de proyectos de software en la organización, contra las áreas de proceso de CMMI relacionadas con la administración de proyectos, las siguientes secciones sintetizan los resultados de la evaluación de los objetivos específicos (SG por sus siglas en inglés) y sus prácticas específicas (PA) para las áreas de proceso de CMMI relacionados a la administración de proyectos. Esta evaluación considera tanto los objetivos genéricos como los específicos de las áreas de proceso de CMMI relacionadas, solamente omitiendo SAM (Supplier Agreement Management – Administración de Acuerdos con Proveedores) dado que las actividades de sub-contratación de software son prácticamente nulas dentro de esta organización y Quantitative Project Management (QPM) debido a que esta área de proceso no se encuentra implantada aún en la organización debido a ser CMM nivel 3.

Para referencia a las secciones del Plan de Desarrollo de Software, refiérase a la sección 2.1.1 Plan de Desarrollo de Software.

3.1 Evaluación de PP

- Objetivo específico SG1, Establecer Estimados
 - Práctica Específica SP 1.1 Estimar el Alcance del Proyecto: La sección 2 Visión General del plan de desarrollo describe el alcance del proyecto, el cliente del proyecto así como una descripción general de la funcionalidad requerida del proyecto. Esta sección cuenta con varias sub-secciones que contribuyen con la descripción del alcance del proyecto: sección 2.1 Software adquirido por Cliente, Vendedor, Tercero, Contrato, sección 2.2 Entregables.
 - Prácticas Específicas SP1.2 Establecer Estimados de los Productos de Trabajo y Atributos de las Tareas y SP1.4 Determinar Estimados de Esfuerzo y Costo: El documento de Plan de Desarrollo de Software incluye la sección 4 Estimados de Software únicamente para documentar la información relacionada con los

estimados del proyecto, dicha sección describe el método utilizado para la estimación, un estimado inicial así como una referencia a la base de datos donde se documentaran los estimados subsecuentes. Es responsabilidad del gerente de software revisar periódicamente estas métricas a fin de ajustar planes y recursos según sea necesario. Dentro de la información de estimados se incluye, los recursos necesarios para ejecutar el proyecto basado en las fechas compromiso, los recursos computacionales que el proyecto usara, considera la reusabilidad de proyectos anteriores o de arquitecturas similares y costos asociados con la compra de componentes de software de terceros, por ejemplo algún driver de video etc. El Plan de Desarrollo de Software no tiene información concerniente a costos, debido a que esta información es administrada por el Gerente del Proyecto.

- Práctica Específica SP 1.3 Definir el ciclo de vida del proyecto: La definición del ciclo de vida del proyecto es documentada como parte de la sección 5 Proceso Definido de Desarrollo de Software para el Proyecto. El ciclo de vida se determina a partir de las características del desarrollo de software, el grado de reusabilidad y las metas intermedias negociadas con el cliente para la entrega de funcionalidad parcial. El ciclo de vida mas utilizado en esta organización es el de espiral.
- Objetivo específico SG 2, Desarrollar un Plan de Proyecto
 - Práctica Específica 2.1 Establecer presupuesto y calendario: El Plan de Desarrollo de Software incluye la sección 11 Calendario del Proyecto para documentar el las tareas y responsables de cumplir con ellas; sin embargo debido a que el calendario es información sumamente dinámico, generalmente en esta sección se hace referencia a un archivo externo de calendario que detalla tanto las tareas, relaciones, progreso y responsables incluyendo metas intermedias y actividades de soporte del proyecto como actividades de aseguramiento de calidad etc. El Plan de Desarrollo de Software no tiene información concerniente al presupuesto, debido a que esta información es administrada por el Gerente del Proyecto.
 - Práctica Específica 2.2 Identificar riesgos del proyecto: La sección 7 Administración de Riesgos documenta la información relacionada con los riesgos identificados para el proyecto de software. Además de la identificación de los riesgos el plan documenta entre otras cosas, actividades de mitigación y contingencia además de responsable para cada riesgo identificado. Esta información es revisada

periódicamente con el Gerente del Proyecto y con el Gerente de Software para tomar acción en caso de que un riesgo se transforme en un problema.

- Práctica Específica 2.3 Planear la Administración de Datos: Dentro de la sección 5 Proceso Definido de Desarrollo de Software para el Proyecto se documenta la información referente a los productos de trabajo e identificación de elementos de configuración donde se listan los documentos que se generaran y controlaran como parte del proceso de desarrollo del proyecto. Esta lista incluye información del nombre y método de control de versiones así como el lugar donde serán alojados los documentos. Esta misma sección documenta la metodología de administración de solicitudes de cambio para cada documento que se encuentra directamente relacionada con el control de configuración del proyecto. La organización tiene carácter global por lo que los lugares donde los documentos son localizados se encuentran en red para su acceso desde los diferentes sitios de desarrollo. La metodología de control de cambios permite que la modificación de la información sea identificada y registrada.
- Práctica Específica 2.4 Planear Recursos del Proyecto: Esta práctica se logra en tres secciones distintas del Plan de Desarrollo de Software. La sección 3 Personal, Responsabilidades y la Coordinación Intergrupala describe las necesidades de recursos humanos para el desarrollo del proyecto, además de los roles y responsabilidades. La sección 5 Proceso Definido de Desarrollo de Software para el Proyecto describe los procesos específicos requeridos para el desarrollo del software del proyecto, adaptándose a las necesidades particulares del proyecto. La sección 9 Herramientas e Instalaciones describe las herramientas e instalaciones necesarias para el desarrollo del proyecto.
- Práctica Específica 2.5 Planear las Necesidades de Habilidades y Conocimientos: La sección 10 Entrenamiento Específico para el Proyecto del Plan de Desarrollo de Software documenta la información relacionada con la identificación y planeación del entrenamiento requerido por los participantes del desarrollo. De manera adicional y como complemento a esta práctica del modelo, la sección 3 Personal, Responsabilidades y la Coordinación Intergrupala describe los roles a desempeñar así como quien ejecuta cada uno de los roles.

- Práctica Específica 2.6 Planear el Involucramiento de los Participantes: La sección 3 Personal, Responsabilidades y la Coordinación Intergrupar del Plan de Desarrollo de Software documenta quien participara, que rol desempeñara así como cuales son las responsabilidades de dicho rol. Esta información incluye a participantes directos e indirectos del proyecto.
- Práctica Específica 2.7 Establecer el Plan del Proyecto: El desarrollo de software implantado en la organización, requiere y audita la creación uso y mantenimiento del Plan de Desarrollo de Software de manera periódica para cada proyecto de desarrollo de software. El documento del Plan de Desarrollo de Software es la evidencia misma del cumplimiento de esta práctica.
- Objetivo específico SG 3, Obtener compromiso con el Plan
 - Práctica Específica 3.1 Revisar los Planes que afecten al Proyecto: El documento de Plan de Desarrollo de Software es autorizado por el Gerente del Proyecto, el Gerente de Software y el Ingeniero de Aseguramiento de Calidad de Software. En el caso del Gerente del Proyecto y el Ingeniero de Aseguramiento de Calidad de Software ellos, a su vez, tienen sus propios planes en relación al desarrollo de producto y el software, la búsqueda de su aprobación es con el fin de asegurar coordinación, y alcanzar los objetivos globales del proyecto logrando el alineamiento de todos los planes alrededor del producto. El Plan de Desarrollo de Software incluye el registro las aprobaciones para cada modificación del plan.
 - Práctica Específica 3.2 Reconciliar Niveles de Trabajo y Recursos: Haciendo referencia a las aprobaciones del Plan de Desarrollo de Software, la aprobación del gerente de software busca que la información de los estimados, recursos, herramientas, calendario etc. incluidos para el desarrollo del proyecto, estén de conciliados con los recursos reales disponibles de la organización y esto quede sentado en el plan antes de firmar la autorización del plan. El Plan de Desarrollo de Software incluye un registro de aprobación del Gerente de Software.
 - Práctica Específica 3.3 Obtener Compromiso con el Plan: Actualmente el compromiso con el plan se obtiene por medio de las aprobaciones de parte del Gerente del Proyecto, el Ingeniero de Aseguramiento de calidad así como el Gerente de Software, sin embargo, no se encontró evidencia de la obtención del compromiso por parte de los participantes directos en el desarrollo. Esta práctica es

un área de mejora para los procedimientos de desarrollo de software de la organización. Al parecer dentro de esta organización una vez que se obtienen las aprobaciones se da por hecho la aceptación del plan por parte de los participantes, sin embargo esto no abona con el establecimiento de una comunicación efectiva en equipo así como la oportunidad de mejorar el mismo con beneficio en la ejecución del proyecto.

La siguiente tabla sintetiza los resultados del análisis de esta área de proceso.

Project Planning (PP)		Cumplimiento	Notas
SG 1 Establecer Estimados			
SP 1.1	Estimar el Alcance del Proyecto	Sección 2 Visión General	
SP 1.2	Establecer Estimados del Trabajo del Producto y los Atributos de las Tareas	Sección 4 Estimados de Software	Los estimados de software utilizan los requisitos del sistema para estimar recursos, tiempo y el microcontrolador adecuado para la aplicación. Como práctica general se utiliza la arquitectura de software del producto para particionar la funcionalidad y estimar cada sección.
SP 1.3	Definir el Ciclo de Vida del Proyecto	Sección 5 Proceso definido de software del proyecto	SPBB = Software Process Building Blocks. Estos bloques describen pasos del proceso de desarrollo de software para una aplicación, desde análisis de requisitos hasta pruebas y liberación.
SP 1.4	Determinar Estimados de Esfuerzo y Costo	Sección 4 Estimados de Software	Se observa que el costo está relacionado al tiempo y cantidad de recursos asignados al proyecto, esto generalmente lo administra el Gerente del Proyecto.
SG 2 Desarrollar un Plan de Proyecto			
SP 2.1	Establecer Presupuesto y Calendario	Sección 11 Calendario del Proyecto	No se incluyen estimados de costos por que esta información queda en el plan de desarrollo del producto
SP 2.2	Identificar Riesgos del Proyecto	Sección 7 Administración de Riesgos del Proyecto	Se documentan y evalúan planes de riesgos únicamente relacionados al desarrollo del software.
SP 2.3	Planear la Administración de los Datos	Sub-sección 5.1 Productos de Trabajo y Elementos de Configuración. Sub-sección 5.4 Administración de solicitudes de cambio de software.	La subsección 5.4 hace referencia a la información para el control de los documentos considerados como elementos de configuración y sus versiones.

SP 2.4	Planear los Recursos del Proyecto	sección 3 Personal, Responsabilidades y Coordinación Intergrupala, sección 5 Proceso Definido de Software para el proyecto, sección 9 Herramientas e Instalaciones	
SP 2.5	Planear el Habilidades y Conocimientos Necesarios	Sección 10 Plan específico de entrenamiento para el proyecto. sección 3 Personal, Responsabilidades y Coordinación Intergrupala,	De manera adicional la organización cuenta con una currícula de entrenamientos mandatorios para el puesto de ingeniero de software.
SP 2.6	Planear la Intervención de Participantes	sección 3 Personal, Responsabilidades y Coordinación Intergrupala,	
SP 2.7	Establecer el Plan del Proyecto	Documento de Plan de Desarrollo de Software	
SG 3 Obtener compromiso con el Plan			
SP 3.1	Revisar los Planes que afecten al Proyecto	Registro de aprobaciones del plan	El plan es revisado y aprobado por parte del Gerente de Proyecto y el Ingeniero de Aseguramiento de Calidad de Software para verificar que los planes están alienados con los objetivos de proyecto.
SP 3.2	Reconciliar Niveles de Trabajo y Recursos	Registro de aprobaciones del plan	El plan es revisado y aprobado por parte del Gerente de Software, evaluando y garantizando los recursos y niveles de trabajo.
SP 3.3	Obtener Compromiso con el Plan	Evidencia parcial del cumplimiento de la práctica.	Se obtiene compromiso por parte del Gerente de Software, el Ingeniero de Aseguramiento de Calidad de Software y el Gerente de Proyecto. Se da por hecho el compromiso y acuerdo con el plan por parte de los miembros directos e indirectos del equipo.

Tabla 6 Análisis de Área de Proceso PP de CMMI

SG = Specific Goal (Meta específica)

SP = Specific Practice (Práctica específica)

3.2 Evaluación de PCM

- Objetivo específico SG 1, Monitorear proyecto contra plan
 - Práctica específica 1.1 Monitorear los Parámetros de Planeación del Proyecto: La sección 6 Proceso de Desarrollo y Metas de Producto y Proceso del Plan de Desarrollo de Software específica los objetivos generales del desarrollo del proyecto, además en la sección 11 Calendario del Proyecto se establecen las

tareas, responsables, tiempos estimados y dependencias para el cumplimiento de ellas. Ante estos parámetros en la sección 8 Reporte de Progreso se establecen los reportes periódicos a ser revisados por el Gerente de Software y el Gerente de Proyecto.

- Práctica específica 1.2 Monitorear Compromisos: El monitoreo de los compromisos se logra a través de la actualización del documento de calendario del desarrollo de software establecido en la sección 11 Calendario del Proyecto del Plan de Desarrollo de Software. Este documento incluye entregas específicas a solicitud de los clientes, y es revisado periódicamente con el Gerente de Software, el Gerente de Proyecto y el/los clientes. No existe un formato o procedimiento en particular para el monitoreo de compromisos, más que el progreso reportado ante las tareas planeadas en cada entrega.
- Práctica específica 1.3 Monitorear Riesgos del Proyecto: Los riesgos del proyecto son monitoreados periódicamente por el Gerente de Software así como la ejecución de su plan de mitigación y en caso de convertirse en problema, su contingencia y escalación. La sección 7 Administración de Riesgos documenta el plan de administración de los riesgos y la sección 8 Reporte de Progreso documenta su reporte y seguimiento periódico.
- Práctica Específica 1.4 Monitorear la Administración de Datos: La sección 6 Proceso de Desarrollo y Metas de Producto y Proceso del Plan de Desarrollo de Software documenta las metas de producto y proceso específicas para el desarrollo de software. Dichas metas hacen referencia específica a métricas del desarrollo que se monitorean periódicamente por parte del Gerente de Software de acuerdo a lo documentado en la sección 8 Reporte de Progreso.
- Práctica Específica 1.5 Monitorear el Involucramiento de los Participantes: El Plan de Desarrollo de Software describe tanto los participantes directos como indirectos del proyecto así como las responsabilidades de cada uno de ellos en la sección 3 Personal, Responsabilidades y la Coordinación Intergrupar. Sin embargo no se encontró evidencia de planeación específica para el monitoreo de que los participantes, sus responsabilidades y la coordinación intergrupar esta ocurriendo de acuerdo a lo planeado esta es un área de oportunidad para la mejora del proceso.

- Práctica Específica 1.6 Conducir Revisiones de Progreso: Las revisiones del progreso del desarrollo de software se llevan al cabo de manera periódica de acuerdo a lo planeado y documentado en la sección 8 Reporte de Progreso con el Gerente de Proyecto en la llamada “junta de desarrollo del producto”, donde representantes de los demás grupos de ingeniería participan, lo cual motiva la interacción y acuerdos. Por otro lado el líder de desarrollo de software lleva juntas periódicas con los ingenieros de software tanto para saber del progreso u obstáculos como cambios de dirección del desarrollo.
- Práctica Específica 1.7 Conducir Revisiones de Metas Intermedias: Las revisiones de metas intermedias son administradas por el Gerente de Proyecto y cada proyecto tiene metas muy específicas de revisión con la dirección de ingeniería, en las cuales la competencia de ingeniería de software es responsable de presentar información pre-definida y responder las preguntas de la dirección, dichas revisiones son las siguientes:
 - Revisión de Requerimientos.
 - Revisión preliminar de Diseño.
 - Revisión de Diseño.
 - Revisión Crítica de Diseño.
 - Aprobación de Pre-producción “PPAP”.
 - Producción.
- Objetivo Específico 2, Administrar Acciones Correctivas hasta cerrarse
 - Práctica Específica 2.1 Analizar Problemas: Los problemas relacionados al desarrollo de software generalmente son relacionados al incumplimiento de algún requisito. La administración de las acciones correctivas es gestada como una “solicitud de cambio” con etiqueta de “Defecto”. El proceso de administración de cambios esta documentado en la sección 5 Proceso Definido de Desarrollo de Software para el Proyecto en particular en la porción que describe el plan de administración de cambios de software. Existen problemas relacionados con recursos, calendario y riesgos, los cuales su análisis y plan de mitigación son documentados en la sección 7 Administración de Riesgos. Varios de los problemas sistémicos del desarrollo serán abordados con técnicas organizacionales como 5 pasos, Análisis de Efecto-Causa raíz, análisis con herramientas de Six Sigma.

- Práctica Específica 2.2 Tomar Acción Correctiva: De manera muy similar a la práctica anterior se observa que el Plan de Desarrollo de Software utiliza diferentes técnicas u estrategias dependiendo del tipo de problema que afronta. Nuevamente las secciones que documentan la planeación de la ejecución de las acciones correctivas son las siguientes: sección 5 Proceso Definido de Desarrollo de Software para el Proyecto en particular en la porción que describe el plan de administración de cambios de software y la sección 7 Administración de Riesgos.
- Práctica Específica 2.3 Administrar la Acción Correctiva: El monitoreo y control de las acciones correctivas son llevados al cabo por parte de Ingeniero Líder de Software y reportadas periódicamente al Gerente de Software como parte del reporte de estatus del proyecto. La sección que documenta este reporte y seguimiento es la sección 8 Reporte de Progreso.

La siguiente tabla sintetiza los resultados del análisis de esta área de proceso:

Project Monitoring and Control (PMC)		Cumplimiento	Notas
SG 1 Monitorear proyecto contra el plan			
SP 1.1	Monitorear los Parámetros de Planeación del Proyecto	Sección 6 Objetivos de Producto y Proceso del Desarrollo. Sección 11 Calendario. Sección 8 Reporte de Progreso.	El monitoreo del gasto es llevado por el Gerente de Proyecto del proyecto no por el líder de software. El monitoreo de riesgos y entrenamiento es llevado por el Gerente de Software.
SP 1.2	Monitorear Compromisos	Sección 11 Calendario.	Acuerdos establecidos con el Project Análisis en términos de contenido y fechas.
SP 1.3	Monitorear Riesgos	Sección 7 Administración de Riesgos del Proyecto. Sección 8 Reporte de Progreso.	Revisión Periódica con Gerente de Software
SP 1.4	Monitorear la Administración de Datos	Sección 6 Objetivos de Desarrollo de Producto y Proceso. Sección 8 Reporte de Progreso.	Revisión Periódica con Gerente de Software
SP 1.5	Monitorear el Involucramiento de Participantes	Sección 3 Personal, Responsabilidades y Coordinación Intergrupala. No se encontró evidencia del Monitoreo, solo de la descripción del involucramiento.	Revisión Periódica con Gerente de Software
SP 1.6	Conducir Revisiones de Progreso	Juntas periódicas de proyecto. Minutas de Junta de Proyecto.	Participación en juntas de proyecto periódicas con el Gerente de Proyecto. Revisión de estatus y progreso con los ingenieros de software.
SP 1.7	Conducir Revisiones de metas intermedias	Revisiones administradas por el	Estas son revisiones formales del

		Gerente de Proyecto.	proyecto donde se tienen metas específicas para cada proyecto con requisitos de la información a presentar así como la posibilidad de que los diseños sean retados.
SG 2 Administrar Acciones Correctivas hasta Cerrarse			
SP 2.1	Analizar Problemas	sección 5 Proceso definido de software del proyecto Sub-sección 5.4 Administración de solicitudes de cambio de software Sección 7 Administración de Riesgos del Proyecto	Dependiendo del tipo de problema que enfrente el desarrollo de software su análisis y resolución utiliza diferentes áreas de la planeación del desarrollo para.
SP 2.2	Tomar Acción Correctiva	sección 5 Proceso definido de software del proyecto Sub-sección 5.4 Administración de solicitudes de cambio de software Sección 7 Administración de Riesgos del Proyecto	En la definición de los proceso se incluye las acciones necesarias para ejecutar la acción correctiva. Por ejemplo, la resolución de un defecto de software requiere varias revisiones, aprobaciones y reporte de resolución del defecto.
SP 2.3	Administrar Acción Correctiva	Sección 8 Reporte de Progreso.	Las acciones correctivas son monitoreadas y reportadas periódicamente con el Gerente de Software.

Tabla 7 Análisis de Área de Proceso PCM de CMMI

SG = Specific Goal (Meta específica)

SP = Specific Practice (Práctica específica)

3.3 Evaluación de RSKM

- Objetivo específico SG 1, Prepararse para la Administración de Riesgos
 - Práctica Específica 1.1 Determinar Fuentes de Riesgos y sus Categorías: La plantilla del Plan de Desarrollo de Software sugiere el uso del procedimiento de Administración de Riesgos organizacional, sin embargo no limita el uso de técnicas externas o sugeridas por los clientes. Dicho procedimiento de administración de riesgos establece una definición común de riesgo y define inicialmente las siguientes fuentes de riesgo que afecten el cumplimiento del proyecto:
 - Fechas de Entrega, Costo, Funcionalidad, Desempeño en tiempo real, Confiabilidad, Disponibilidad de Recursos.

El procedimiento de administración de riesgos establece el método de identificación de riesgos así como su eventual categorización por prioridades, impacto e mitigación-beneficio.

- Práctica Específica 1.2 Definir los Parámetros de Riesgo: Los parámetros de definición de los riesgos están definidos como parte de la plantilla de Plan de Desarrollo de Software y están documentados en la sección 7 Administración de Riesgos del Plan de Desarrollo de Software. El procedimiento de planeación no limita los parámetros a los que se encuentran en la plantilla pero al menos debe contar con la siguiente información:

- Numero de Riesgo
- Severidad del Riesgo (alta, media, baja)
- Descripción del Riesgo
- Medición del Riesgo
- Umbral del Riesgo para tomar Acción
- Plan de mitigación del Riesgo
- Plan de Contingencia del Riesgo
- Responsable
- Fecha de Identificación
- Como puede ayudar la Gerencia para reducir el Riesgo
- Práctica Específica 1.3 Establecer Estrategia de Administración de Riesgos: La sección 7 Administración de Riesgos del Plan de Desarrollo de Software lista los riesgos y documenta la estrategias de administración de cada riesgo identificado para el desarrollo del software del proyecto, donde se incluyen acciones específicas para medir, mitigar, monitorear y en caso necesario aplicar acciones de contingencia de cada riesgo.

- Objetivo Específico SG2, Identificar y Analizar Riesgos

- Práctica Específica 2.1 Identificar Riesgos: La identificación de riesgos del proyecto es una actividad realizada principalmente por el Ingeniero Líder de Software y el Gerente de Software donde durante el proceso de planeación del proyecto revisan lecciones aprendidas, fechas de entrega, recursos disponibles y los requerimientos técnicos del proyecto para determinar la lista inicial de riesgos que se administrarán. Estos riesgos serán monitoreados periódicamente y los planes de mitigación se

implantarán. Típicamente la revisión y actualización de los riesgos se realiza antes de cada revisión formal de diseño del proyecto para ser reportada con la dirección. La lista de riesgos identificada queda como parte del Plan de Desarrollo de Software en la sección 7 Administración de Riesgos del Plan de Desarrollo de Software.

- Práctica Específica 2.2 Evaluar, Categorizar, y Priorizar Riesgos: Cada uno de los riesgos identificados es evaluado, categorizado y priorizado de acuerdo a los parámetros anteriormente definidos en la plantilla del Plan de Desarrollo de Software. La lista de riesgos con sus parámetros de planeación completos constituyen el plan de administración de riesgos del proyecto documentado en la sección 7 Administración de Riesgos del Plan de Desarrollo de Software.
- Objetivo Específico SG3, Mitigar Riesgos
 - Práctica Específica 3.1 Desarrollar Planes de mitigación de Riesgos: Parte de los parámetros de administración de riesgos pre-definidos en la plantilla de Plan de Desarrollo de Software incluye la definición del plan de mitigación de cada riesgo de manera específica. El objetivo es planear las acciones necesarias para que las métricas que nos indican el estado del riesgo no excedan el umbral definido. Si exceden el umbral, se procede a las acciones planeadas de contingencia para el riesgo. La planeación de la mitigación para cada riesgo identificado esta documentada en la sección 7 Administración de Riesgos del Plan de Desarrollo de Software.
 - Práctica Específica 3.2 Implementar Planes de mitigación de Riesgos: Con base en la planeación de las actividades de mitigación, el Ingeniero Líder de Software es responsable de la implementación de los planes de mitigación, sin embargo en ocasiones necesitara el apoyo del Gerente de Software; por ejemplo, un riesgo seria que las herramientas de desarrollo no estén disponibles a tiempo para etapa de codificación, lo cual retrasaría la entrega, en este caso el Gerente de Software, como mitigación, puede expeditar el pago al proveedor, o conseguir herramientas prestadas en otras líneas de producto etc. Estas acciones están planeadas, implantadas y periódicamente monitoreadas hasta que el riesgo no exista. El estado de los riesgos que puedan afectar el desarrollo del proyecto se reportan formalmente durante las revisiones de diseño con la dirección.

La siguiente tabla sintetiza los resultados del análisis de esta área de proceso:

Risk Management (RSKM)		Cumplimiento	Notas
SG1 Prepararse para la administración de riesgos			
SP 1.1	Determinar las Fuentes de Riesgo y sus Categorías	Instrucción de trabajo de Administración de Riesgos	
SP 1.2	Definir los Parámetros de Riesgo	Plantilla genérica de Plan de Desarrollo de Software	Los parámetros de riesgo pre-definidos en la plantilla son los mínimos requeridos pero el líder de proyecto puede definir cualquier adicional que considere necesario.
SP 1.3	Establecer Estrategia de Administración de Riesgos	Sección 7 Administración de Riesgos del Proyecto. Instrucción de trabajo de Administración de Riesgos	
SG 2 Identificar y analizar riesgos			
SP 2.1	Identificar Riesgos	Sección 7 Administración de Riesgos del Proyecto.	El plan de riesgos lista los riesgos identificados para el desarrollo del proyecto a partir de un análisis de lecciones aprendidas así como requerimientos del proyecto.
SP 2.2	Evaluar, Categorizar, y Priorizar Riesgos	Plantilla genérica de Plan de Desarrollo de Software Sección 7 Administración de Riesgos del Proyecto.	
SG 3 Mitigar riesgos			
SP 3.1	Desarrollar Planes de Mitigación de Riesgos	Sección 7 Administración de Riesgos del Proyecto.	
SP 3.2	Implementar Planes de Mitigación de Riesgos	Sección 7 Administración de Riesgos del Proyecto.	

Tabla 8 Análisis de Área de Proceso RSKM de CMMI

SG = Specific Goal (Meta específica)

SP = Specific Practice (Práctica específica)

3.4 Evaluación de IPM +IPPD

- Objetivo específico SG 1, Usar el Proceso Definido del Proyecto
 - Práctica Específica 1.1 Establecer el Proceso Definido del Proyecto: La sección 5 Proceso Definido de Desarrollo de Software para el Proyecto del documento del Plan de Desarrollo de Software documenta la definición específica de los procedimientos de desarrollo a utilizar por el equipo con base en las necesidades

específicas del proyecto de software. Este proceso es diseñado a partir de los procedimientos organizacionales estándar de desarrollo de software, los requerimientos del proyecto, así como necesidades o solicitudes específicas de los clientes, por ejemplo, algunos clientes solicitan específicamente un análisis estático del código de acuerdo a algunas reglas específicas de MISRA 1998, por lo tanto parte de la verificación del código implementado requerirá dicha verificación. El proceso es revisado y aprobado por el Gerente de Software, Ingeniero de Aseguramiento de Calidad de Software y el Gerente de Proyecto.

- Práctica Específica 1.2 Utilizar Valuartes de Proceso Organizacionales para la Planeación del Proyecto: Al inicio del proyecto, una vez recibidos los requisitos e identificado el Ingeniero Líder de Software, este es responsable de definir el Plan de Desarrollo de Software, así como algunas actividades administrativas extra relacionadas al proyecto. Con tal motivo se utiliza un Checklist de Inicio de Proyecto donde uno de los puntos requeridos para iniciar la planeación del desarrollo es la revisión de las lecciones aprendidas y los valuartes de proceso alojados en la librería de valuartes de proceso “PAL (Process Assets Library)” por sus siglas en Inglés.
- Práctica Específica 1.3 Establecer el entorno de Trabajo del Proyecto: El entorno de trabajo para el desarrollo de software es establecido por la organización de manera anticipada, es decir, instalaciones como, oficinas, computadoras, red, servidores, bases de datos, equipo y área de laboratorio, personal, entrenamiento esta preparado y es re-utilizado para los múltiples proyectos que se desarrollan. El Plan de Desarrollo de Software, identifica los recursos específicos a ser utilizados por el desarrollo de software en la sección 9 Herramientas e Instalaciones del Plan de Desarrollo de Software.
- Práctica Específica 1.4 Integrar Planes: El Plan de Desarrollo de Software es un documento que incluye secciones que documentan la planeación de varios aspectos del desarrollo de software tales como:
 - Proceso Definido de Software para el Proyecto
 - Plan de Verificación del Software
 - Administración de Control de Cambios
 - Administración de Configuración

- Administración de Riesgos
- Plan de Entrenamiento

El Plan de Desarrollo de Software a su vez está relacionado con el Plan de Aseguramiento de Calidad de Software y con el Plan de Desarrollo del Producto del Gerente de Proyecto. Con el fin de estar coordinados, el Plan de Desarrollo de Software que incluye el proceso definido de software para el proyecto debe obtener la aprobación del Ingeniero de Aseguramiento de Calidad de Software y de Gerente de Proyecto a fin de lograr la integración y coordinación de los planes. Cada versión del plan obtiene nuevas aprobaciones, las cuales están documentadas en la sección de registro de aprobaciones del plan.

- Práctica Específica 1.5 Administrar el Proyecto utilizando los Planes Integrados: Una vez establecido y aprobado el Plan de Desarrollo de Software este es utilizado para la ejecución del proyecto. Cada ingeniero participante del desarrollo utiliza como referencia, y basándose en la tarea a realizar, la información documentada en el Plan de Desarrollo de Software, por ejemplo, si la actividad programada en el calendario para el ingeniero es el análisis de requerimientos, entonces el ingeniero deberá referirse a la sección de Proceso Definido de Software donde se documenta la información que describe lo que se debe hacer para la ejecución de este proceso, entradas y salidas esperadas, y donde se encuentra alojada la información así como donde alojar las salidas del procedimiento. En caso de requerir desviaciones, estas serán discutidas con el Ingeniero Líder de Software para su resolución. Conforme las tareas son terminadas y los productos de trabajo de los procesos se generan el Ingeniero Líder de Software colecta la información para reportar el estado del desarrollo conforme a su calendario y lo que indica el Plan de Desarrollo de Software con respecto al reporte de progreso.
- Práctica Específica 1.6 Contribuir a los Valuartes de Proceso Organizacionales: Esta práctica específica no está documentada como parte del Plan de Desarrollo de Software para un proyecto en específico, sino que durante el entrenamiento del proceso de desarrollo de software que reciben los ingenieros, se hace referencia a la librería de valuartes de proceso PAL, tanto para la referencia de documentos, como para su contribución. En este segundo caso, existe un proceso organizacional que establece un comité de expertos que evalúa las solicitudes de contribución a la

PAL con el fin de asegurar que las contribuciones son realmente valiosas para el proceso antes de ser agregados nuevos valuartes.

- Objetivo Específico SG2, Coordinar y Colaborar con Participantes Relevantes
 - Práctica Específica 1.1 Administrar el Involucramiento de los Participantes: Esta práctica inicia con la identificación de los participantes del proyecto así como sus roles y responsabilidades, lo cual está documentado en la sección 3 Personal, Responsabilidades y la Coordinación Intergrupala, del Plan de Desarrollo de Software. Seguidamente cada participante del desarrollo de software, se le asignan tareas específicas de acuerdo al calendario del proyecto documentado en la sección 11 Calendario del Proyecto, del Plan de Desarrollo de Software. Por este mismo medio se le da seguimiento a la tarea hasta su cierre. El cierre de una tarea es entendido como que dicha tarea fue completa de acuerdo a lo establecido en el proceso definido de software del proyecto incluyendo todas sus salidas. Es responsabilidad de Ingeniero Líder de Software asegurarse que esto se cumple para cerrar la actividad en el calendario y asignar una nueva; así como resolver malos entendidos en la coordinación de las tareas de desarrollo.
 - Práctica Específica 2.2 Administrar Dependencias: Las mayoría de las tareas de desarrollo de software tienen dependencias con otras tareas, debido a esto y a la participación de múltiples ingenieros, es responsabilidad del Ingeniero Líder de Software administrar dichas dependencias para, de preferencia, mantenerlas al mínimo entre los participantes relevantes. En la práctica es importante que la funcionalidad del software sea agrupada de acuerdo a la arquitectura y dicho desarrollo sea asignado a un solo ingeniero o grupo de ingenieros en la medida de lo posible, esto reduce dependencias y el riesgo de tener participantes esperando debido a la dependencia de otros módulos. La principal herramienta para administrar las dependencias está en la sección 11 Calendario del Proyecto, del Plan de Desarrollo de Software.
 - Práctica Específica 2.3 Resolver Problemas de Coordinación: Durante el desarrollo del proyecto, los estimados para las tareas asignadas a cada participante varían, y debido a esto las dependencias de actividades se verán afectadas resultando potencialmente en tiempos muertos. Estos problemas de coordinación se presentan tanto con los participantes internos como externos, de acuerdo al impacto de los

problemas de coordinación es responsabilidad del Gerente de Software facilitar la resolución de estos problemas de coordinación al Ingeniero Líder de Software a fin de que el desarrollo continúe de acuerdo a lo planeado; o en su defecto, proceder a procesos de escalación, con la dirección o con los clientes para remover obstáculos que afecten al desarrollo del proyecto. Nuevamente la herramienta más importante para, anticipar y reportar los problemas de coordinación esta en la sección 11 Calendario del Proyecto del Plan de Desarrollo de Software.

- Objetivo Específico IPPD+ SG 3 Aplicar los principios de IPPD (Integrated Product and Process Development)
 - Práctica Específica 3.1 Establecer la Visión Compartida del Proyecto: El Plan de Desarrollo de Software no documenta la visión compartida de proyecto. No se identifica que los procedimientos organizacionales de desarrollo de software requieran esta información como parte de la planeación.
 - Práctica Específica 3.2 Establecer la Estructura de Equipo Integral: El Plan de Desarrollo de Software no documenta la estructura de equipo integrado. No se identifica que los procedimientos organizacionales de desarrollo de software requieran esta información como parte de la planeación. Sin embargo, en la ejecución de los proyectos se identifica la partición de equipos en sub-equipos funcionales basados en los diferentes componentes de la arquitectura de software del producto que desarrollaran. Esto hace suponer que la implantación de esta práctica es de facto.
 - Práctica Específica 3.3 Asignar Requerimientos a los Equipos Integrales: El Plan de Desarrollo de Software no documenta estrategia de asignación de requisitos a equipos integrales. No se identifica que los procedimientos organizacionales de desarrollo de software requieran esta información como parte de la planeación.
 - Práctica Específica 3.4 Establecer Equipos Integrales: El Plan de Desarrollo de Software no documenta el establecimiento de equipos integrales. No se identifica que los procedimientos organizacionales de desarrollo de software requieran esta información como parte de la planeación.
 - Práctica Específica 3.5 Asegurar Colaboración entre Equipos Interfaceando: El Plan de Desarrollo de Software no documenta el establecimiento acuerdos o planes que aseguren la colaboración entre equipos integrales. No se identifica que los

procedimientos organizacionales de desarrollo de software requieran esta información como parte de la planeación.

La siguiente tabla sintetiza los resultados del análisis de esta área de proceso:

Integrated Project Management +IPPD (IPM +IPPD)		Cumplimiento	Notas
SG 1 Usar el Proceso definido del proyecto			
SP 1.1	Establecer el proceso definido del proyecto	sección 5 Proceso Definido de Desarrollo de Software para el Proyecto	
SP 1.2	Utilizar Valuartes de Proceso Organizacionales para la Planeación del Proyecto	Checklist de Inicio de Proyecto PAL (Librería de valuartes de Proceso)	
SP 1.3	Establecer Entorno de Trabajo del Proyecto	sección 9 Herramientas e Instalaciones	
SP 1.4	Integrar Planes	Registro de aprobaciones del plan	
SP 1.5	Administrar el Proyecto utilizando los Planes Integrados	Documento de Planeación de Desarrollo de Software	
SP 1.6	Contribuir a los Valuartes de Proceso Organizacionales	Proceso Organizacional del Área de ingeniería de Software	Existe el Equipo de Proceso de Software con expertos responsables de la evaluación de los potenciales de valuartes de proceso.
SG 2 Coordinar y Colaborar con Participantes Relevantes			
SP 2.1	Administrar el Involucramiento de Participantes	sección 3 Personal, Responsabilidades y la Coordinación Intergrupala sección 11 Calendario del Proyecto	En la práctica, esta actividad es muy dinámica por lo cual el calendario y sus actualizaciones son de suma importancia para la administración efectiva de los participantes.
SP 2.2	Administrar Dependencias	sección 11 Calendario del Proyecto	
SP 2.3	Resolver Problemas de Coordinación	sección 11 Calendario del Proyecto	El hecho que el calendario del proyecto describa el estado real del proyecto es fundamental para poder gestionar efectivamente la coordinación y las dependencias.
IPPD + Addition SG 3 Aplicar los principios de IPPD (Integrated Product and Process Development)			
SP 3.1	Establecer la Visión Compartida del	No definido	

	Proyecto		
SP 3.2	Establecer la Estructura de Equipo Integral	No definido	
SP 3.3	Asignar Requerimientos a los Equipos Integrales	No definido	
SP 3.4	Establecer los Equipos Integrados	No definido	
SP 3.5	Asegurar colaboración entre Equipos interfaceando	No definido	

Tabla 9 Análisis de Áreas de Proceso IPM +IPPD de CMMI

SG = Specific Goal (Meta específica)

SP = Specific Practice (Práctica específica)

3.5 Evaluación de QPM

Esta área de proceso no se encuentra implantada en la organización por ser del nivel 3 de CMMI, y QPM de nivel 4. Por esta razón queda fuera del alcance del presente análisis. Cabe mencionar que el Plan de Desarrollo de Software incluye la sección 6 Objetivos de Producto y Proceso del Desarrollo, donde se inicia la educación del grupo de ingeniería de software a pensar en términos de objetivos y métricas.

Quantitative Project Management (QPM)		Cumplimiento	Notas
SG 1 Administrar el proyecto de manera cuantitativa			
SP 1.1	Establecer los Objetivos del Proyecto	Sección 6 Objetivos de Producto y Proceso del Desarrollo	No todos los procesos definidos en la sección 5 Proceso Definido de Software para el Proyecto tienen un objetivo de proceso específico.
SP 1.2	Componer el Proceso Definido	No definido	Procedimiento de Nivel 4 de CMMI fuera del alcance de la madurez de procesos de la empresa bajo estudio
SP 1.3	Seleccionar los Sub-Procesos que serán Administrados Estadísticamente	No definido	
SP 1.4	Administrar el Desempeño del Proyecto	No definido	
SG 2 Administrar estadísticamente el desempeño de los subprocesos			
SP 2.1	Seleccionar métricas y técnicas de análisis	No definido	Procedimiento de Nivel 4 de CMMI fuera del alcance de la madurez de procesos de la empresa bajo estudio
SP 2.2	Aplicar métodos estadísticos para entender las variaciones	No definido	
SP 2.3	Monitorear el desempeño de los sub-procesos seleccionados	No definido	
SP 2.4	Colectar Datos Estadísticos de Administración	No definido	

Tabla 10 Análisis de Área de Proceso QPM de CMMI

SG = Specific Goal (Meta específica)

SP = Specific Practice (Práctica específica)

3.6 Resumen

Con base en los resultados de la Tablas:

- Tabla 6 Análisis de Área de Proceso PP de CMMI
- Tabla 7 Análisis de Área de Proceso PCM de CMMI
- Tabla 8 Análisis de Área de Proceso RSKM de CMMI
- Tabla 9 Análisis de Áreas de Proceso IPM +IPPD de CMMI
- Tabla 10 Análisis de Área de Proceso QPM de CMMI

Se puede observar que existen varias prácticas específicas (SP) del modelo de proceso de CMMI, las cuales no son cumplidas, o no se mostró evidencia de su implantación en el proceso de planeación de proyectos de software utilizado en la organización, y es aquí donde se fundamentan las oportunidades de mejora. Particularmente se observa que los procesos de:

- Project Monitoring and Control (PMC)
 - SP 1.5 Monitorear el Involucramiento de Participantes
- Integrated Project Management +IPPD (IPM +IPPD)
 - SP 3.1 Establecer la Visión Compartida del Proyecto
 - SP 3.2 Establecer la Estructura de Equipo Integral
 - SP 3.3 Asignar Requerimientos a los Equipos Integrales
 - SP 3.4 Establecer los Equipos Integrados
 - SP 3.5 Asegurar colaboración entre Equipos interfaceando
- Quantitative Project Management (QPM)*
 - SP 1.2 Componer el Proceso Definido
 - SP 1.3 Seleccionar los Sub-Procesos que serán Administrados Estadísticamente
 - SP 1.4 Administrar el Desempeño del Proyecto
 - SP 2.1 Seleccionar métricas y técnicas de análisis
 - SP 2.2 Aplicar métodos estadísticos para entender las variaciones
 - SP 2.3 Monitorear el desempeño de los sub-procesos seleccionados
 - SP 2.4 Colectar Datos Estadísticos de Administración

Presentan las prácticas específicas que no se cumplen (identificados como “No definido” en las tablas de resumen).

Evaluando el impacto de la carencia de la implantación de dichas prácticas específicas, y dejando fuera de la evaluación QPM, se puede anticipar que estas tienen un impacto directo en el desempeño de desarrollo de proyectos multi-sitio, debido a que estas contribuyen directamente a mejorar la comunicación y administración de equipos integrales, aspecto fundamental en el desarrollo de un proyecto cuyos participantes se encuentran geográficamente en diferentes locaciones.

En el siguiente capítulo se revisará a detalle las prácticas específicas no cumplidas con el fin de entender su particular efecto en la operación de desarrollo de software Multi-sitio, así como servirán como fundamento para definir la propuesta de mejora.

Capítulo IV Análisis y Propuestas de Mejora

El

presente capítulo desarrolla la propuesta de mejora metodológica de los procesos de administración de software basado en el análisis de resultados actuales del desempeño de los procesos contra las áreas de proceso relacionadas a la administración de proyectos de desarrollo de software de CMMI. Los principales puntos de referencia para lograr las propuestas de mejora será desarrollada sobre la base de las prácticas específicas de las áreas de proceso evaluadas las cuales no se encontró evidencia de su implementación, haciendo énfasis en su impacto para el desarrollo de los proyecto multi-sitio.

Para una mejor referencia del procedimiento completo de planeación de proyectos de desarrollo de software refiérase a la sección 2 Metodología de Administración de Proyectos.

1 Mejora PP

1.1 Obtener Compromiso con el Plan

En la revisión de la documentación relacionada a la Planeación de Proyecto (PP) por parte de la organización no demostró evidencia de la completa implantación de la práctica específica 3.3 modelo de CMMI Obtener Compromiso con el Plan. Esta práctica específica solicita la obtención del compromiso con el plan de desarrollo por parte tanto de los participantes relevantes (en este caso los ingenieros de software desarrollando el proyecto) así como los participantes que respaldan el desarrollo, o el caso de la gerencia. La práctica específica 3.3 muestra que se cumple de manera parcial dado que se obtiene la firma de aprobación del plan por parte del gerente de software, el ingeniero de aseguramiento de calidad y el Gerente de Proyecto, pero no se muestra alguna evidencia del compromiso adquirido de parte del equipo de desarrollo, implica que se asume el compromiso de los ingenieros a cumplir con la planeación para el desarrollo del proyecto. Esto es particularmente sensible cuando los participantes del proyecto se encuentran distribuidos en diferentes sitios de desarrollo, donde los problemas relacionados con la comunicación y coordinación se amplifican.

La

propuesta de mejora con respecto a esta práctica específica simplemente es la formalización de la comunicación del Plan de Desarrollo de Software a los participantes directos del desarrollo,

esto incluso antes de su autorización. Es muy importante para el Ingeniero Líder de Software asegurarse que todos los miembros del equipo de desarrollo entienden y están de acuerdo con el plan para lograr que su ejecución minimice las desviaciones; es también importante que todos los participantes del desarrollo tengan la oportunidad de aportar a la planeación del desarrollo con el fin de facilitar la ejecución. La porción del procedimiento actual Administración de Proyectos de Software se muestra en la siguiente figura:

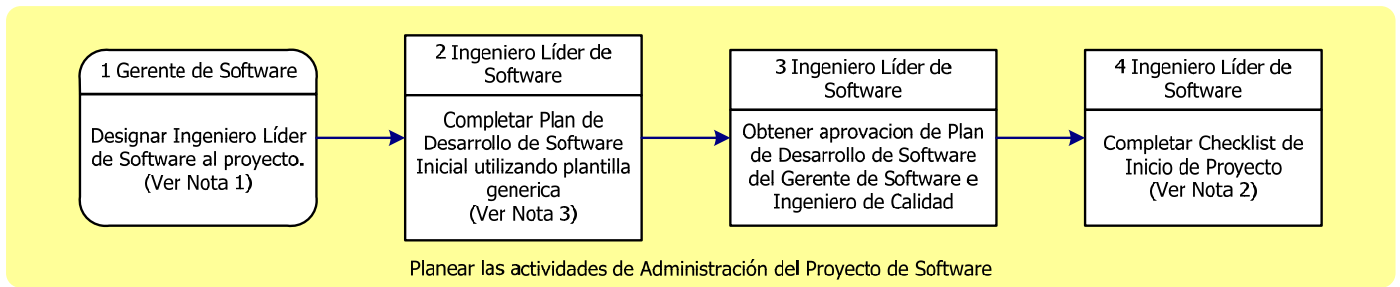


Figura 6 Antes de Mejora PP SP 3.3

La siguiente figura se describe la propuesta de mejora al procedimiento actual. En color azul se puede identificar un paso extra sugerido dentro del proceso de Administración de Proyectos de Software de la organización donde expresamente se responsabiliza al Ingeniero Líder de Software a presentar de manera formal el Plan de Desarrollo de Software a todos los miembros del equipo que participaran en el desarrollo del proyecto. Este paso puede implicar que el plan será presentado múltiples veces a múltiples audiencias en el caso que el equipo esta distribuido en múltiples sitios, lo cual es de esperarse dado las características de los desarrollos globales de la organización. Es de considerarse que en los desarrollos globales la comunicación efectiva entre los miembros del equipo es un aspecto fundamental para la eficiente y efectiva ejecución del desarrollo. El hecho de obtener compromiso con la estrategia de desarrollo definida por el líder del proyecto es un buen paso inicial a mejorar de inicio dicha comunicación. Como nota se puede anticipar que a partir de las presentaciones formales se derivaran potenciales mejoras al plan de desarrollo, siendo esto un aspecto deseable para contribuir a la mejora en la ejecución del desarrollo.

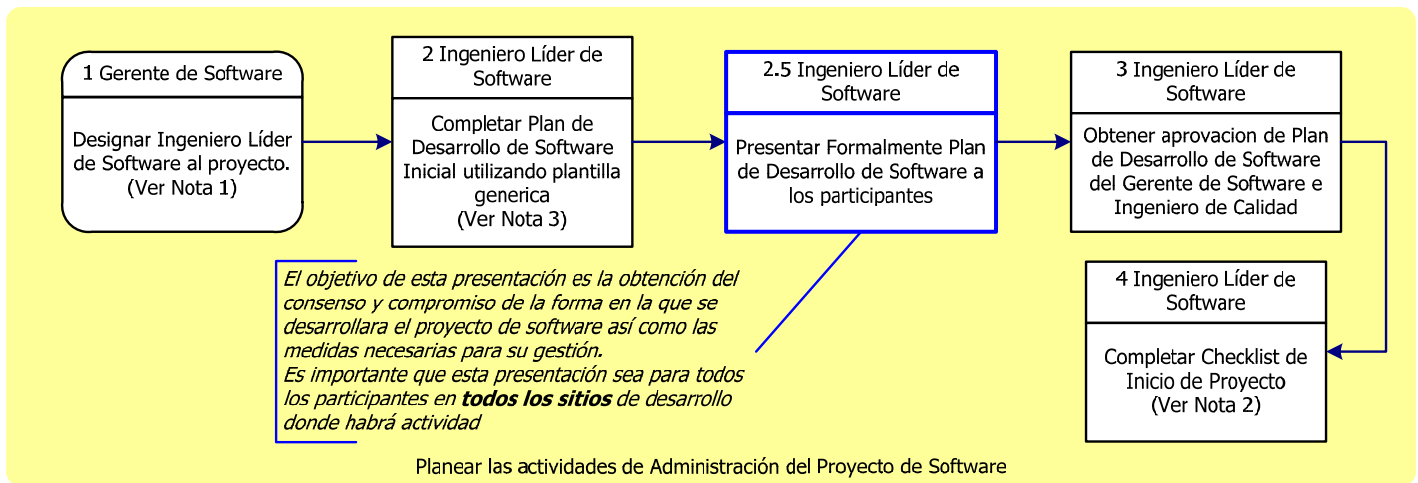


Figura 7 Después de Mejora PP SP 3.3

2 Mejora en PCM

2.1 Monitorear el Involucramiento de los Participantes

En el área de proceso de monitoreo y control del proyecto se identifico como potencial mejora la implantación de la práctica específica 1.5 Monitorear el Involucramiento de los Participantes. El proceso actual solamente identifica y define los roles y responsabilidades de los participantes sin describir una acción específica relacionada con el monitoreo de su involucramiento ya en la ejecución del desarrollo. La siguiente figura muestra la porción del proceso (Paso 5 y 7) donde se solicita el rastreo las actividades dentro del procedimiento de Administración de Proyectos de Software de la organización.

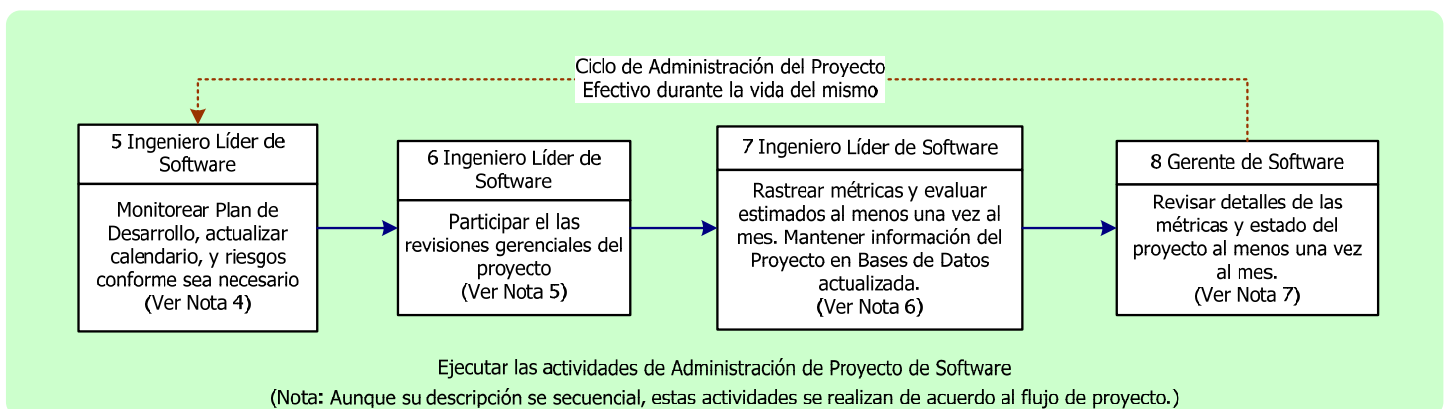


Figura 8 Proceso de Administración de Proyectos de Software 3

Como se observa el procedimiento, actualmente existe un rastreo periódico de varios aspectos del proyecto en el procedimiento de Administración del Proyecto de Software. Hablando en particular de la revisión estado del proyecto ejecutado por el Gerente de Software con el Ingeniero Líder de Software, los procedimientos de desarrollo de software organizacionales sugieren la siguiente agenda:

1. revisión de actividades completadas desde la última revisión.
2. revisión de las actividades planeadas en el siguiente periodo (regularmente semanal).
3. Rastreo del Calendario, progreso.
4. Estado de los recursos críticos de microcontrolador contra los estimados.
5. Rastreo de dependencias críticas de las actividades y con otros grupos de ingeniería.
6. Actividades de prevención de defectos.
7. Problemas y riesgos.

Como se observa en la agenda sugerida, el monitoreo del involucramiento de los participantes no es parte de la misma. Por lo tanto la sugerencia de mejora radica en incluir como parte de la agenda de revisión del proyecto cualquier cambio en el involucramiento de los participantes a lo planeado, es decir, al inicio de la planeación del proyecto de software definen los participantes y roles, aunque esto cambia conforme el proyecto se desarrolla; dichos cambios deben ser revisados, evaluados y autorizados por el Gerente de Software. Asimismo la evaluación del involucramiento de los participantes directos, en términos de las actividades que desarrollan, se vera reflejado en la actualización del calendario del proyecto. La documentación final de los cambios de los roles y responsabilidades se documentaran en una actualización al documento del Plan de Desarrollo de Software del proyecto en específico.

Llevando esta sugerencia de mejora a un entorno de desarrollo global, este monitoreo se hace particularmente mas complejo debido a que generalmente cuando los recursos están distribuidos en múltiples sitios de desarrollo, es debido a que la experiencia de cierta funcionalidad del proyecto se encuentra localizada en dicho sitio remoto; involucrar recursos de dicho sitio es de particular beneficio para el proyecto debido a que se maximiza la reusabilidad y se minimizan los errores. Sin embargo también dichos recursos son generalmente compartidos con varios proyectos, donde el monitorear su involucramiento ayudara a anticipar y mitigar potenciales retrasos por dependencias con actividades críticas del proyecto.

3 Mejoras en IPM+IPPD

La

evaluación de la implantación de las prácticas específicas de la Administración Integral del Proyecto y Desarrollo Integral de Producto y Proceso IPM+IPPD arrojaron en esencia la falta de definición en todas las prácticas específicas de +IPPD.

3.1 Establecer Visión Compartida del Proyecto

La práctica específica 3.1 Establecer la Visión Compartida del Proyecto establece la necesidad de entender la misión, objetivos, expectativas y restricciones organizacionales para alinear la dirección, acciones y visión compartida del proyecto con la organización. Revisando el proceso actual de Administración de Proyectos de Software, se puede identificar que establecer esta visión compartida del proyecto debe ser una de las actividades iniciales del proyecto, prácticamente como la indicación del inicio formal del proyecto a todos los participantes del mismo. Este es un punto ideal ya que generalmente existe el ánimo y excitación del inicio de un nuevo proyecto lo cual puede ser aprovechado por el gerente de software para generar dicha visión.

Una vez que se ha identificado a los participantes internos y externos del proyecto, es el momento donde se puede reunir a todos los participantes a fin de generar dicha visión en la ejecución del proyecto. En el caso de proyectos con desarrollo Multi-sitio, llega a ser difícil coincidir con todos los sitios en un horario en particular para llevar esta actividad simultáneamente, de ahí que será muy importante la coordinación de esta actividad por parte del gerente de software a fin de lograr que todos los miembros del equipo puedan expresar y acordar la visión compartida del proyecto.

Establecer la visión compartida del proyecto ayuda a tener un lineamiento común para todos los participantes de la dirección del proyecto, lo cual abona a facilitar la comunicación en el equipo, cuestión crítica en el desarrollo de proyectos Multi-sitio. Establecer la visión compartida del proyecto debe considerar estar en alineamiento con la misión y visión de la empresa, y requiere que el mensaje sea común para todos los proyectos, de ahí que el mejor facultado para esta actividad sea el gerente de software.

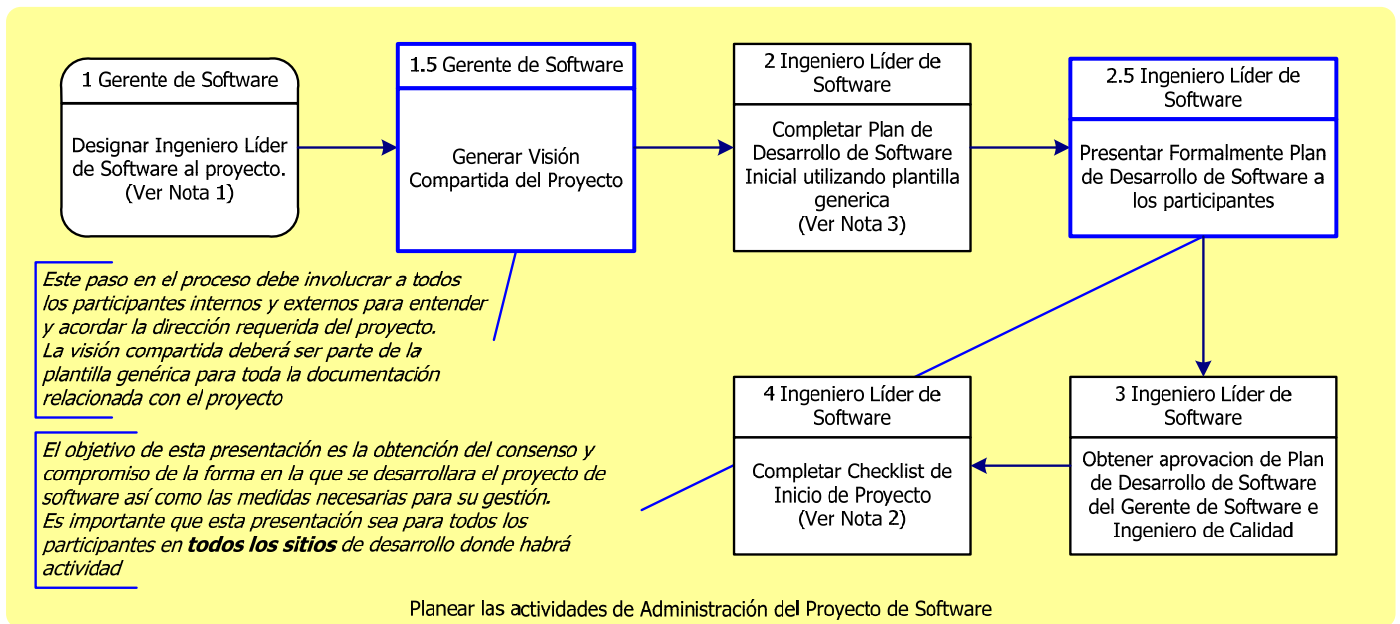


Figura 9 Proceso de Administración de Proyectos de Software 4

3.2 Establecer la Estructura Integral del Equipo

Durante la evaluación de las prácticas específicas de IPPD se encontró la oportunidad de mejorar la implantación del establecimiento de la estructura integral del equipo de software. Tal como indica CMMI, dicha estructura es dinámica por lo tanto cambiante de acuerdo a las necesidades del proyecto. Sin embargo es importante que, su establecimiento de manera inicial, así como sus eventuales cambios sean documentados y actualizados en el Plan de Desarrollo de Software y más importante, que sean comunicados y entendidos claramente por todos los participantes del proyecto.

La propuesta de mejora se encuentra localizada dentro del procedimiento de Administración de Proyectos de Software, en el punto donde el Ingeniero Líder de Software diseña y documenta el Plan de Desarrollo de Software. La propuesta de mejora sugiere los siguientes pasos a implantar durante la elaboración del Plan de Desarrollo de Software:

1. Siguiendo con la práctica que ya ocurre de facto en la organización (sub-equipos funcionales basados en los componentes de la arquitectura de software), la primera sugerencia es incluir como parte de las entradas al proceso de planeación del proyecto de software, información referente a la arquitectura de software del producto (en alto nivel, solo componentes estáticos)

que se desarrollara; el proveedor de dicha información será el arquitecto de software, y muy probablemente podrá ser anticipada (al menos inicialmente) desde el momento en el que se esta cotizando el proyecto.

2. Una vez con esta información, el ingeniero líder deberá de definir la estructura de su equipo basado en los componentes funcionales que se estarán implementando. Es posible que este análisis ayude a afinar los estimados de los recursos de ingeniería necesaria para el proyecto, sin que este sea su objetivo principal. La siguiente figura muestra un ejemplo genérico de los componentes funcionales de una arquitectura de software para un producto. Con esta información el Ingeniero líder de proyecto puede establecer una estructura con 4 sub-equipos de trabajo, cada uno de ellos enfocados en un aspecto diferente del software y con la ayuda del arquitecto de software, establecer interfaces y dependencias entre los componentes. Tal como se muestra en la figura, observe que los componentes de la arquitectura de software no necesariamente están distribuidos en igual numero para cada sub-equipo de trabajo, sino que la distribución debe tomar en cuenta varios factores entre los que se pueden ejemplificar:
 - a. Complejidad de los componentes.
 - b. Experiencia y disponibilidad de los miembros del equipo.
 - c. Prioridad de Entregas.
 - d. Acoplamiento y Cohesión de los componentes.
 - e. Ubicación de los miembros del equipo (Multi-sitio).

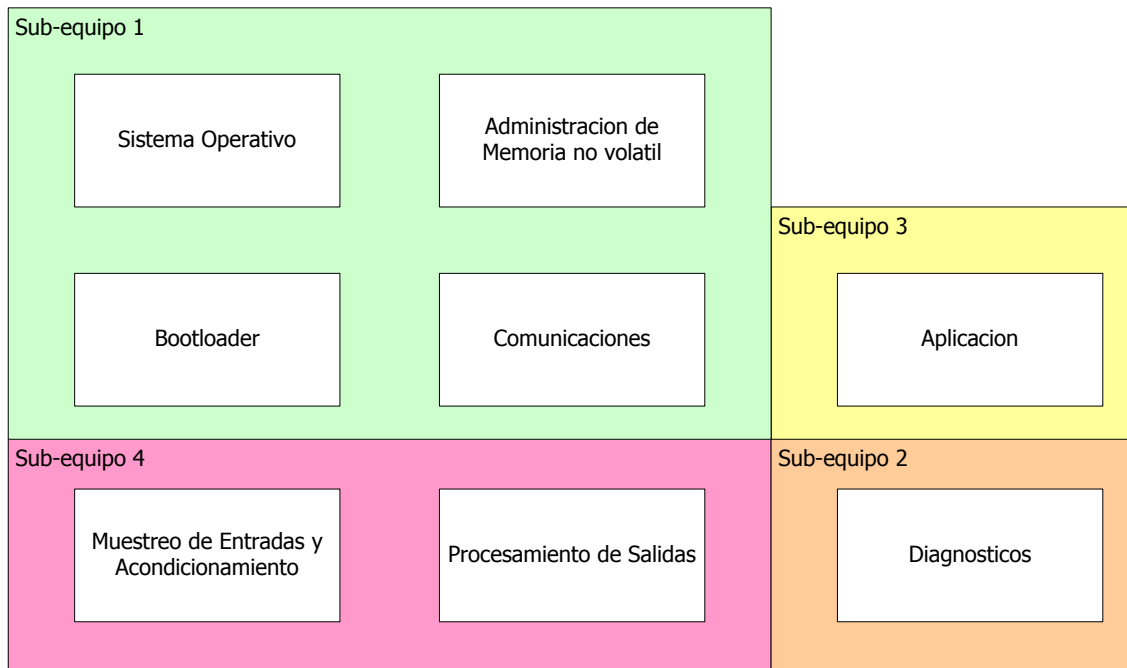


Figura 10 Ejemplo de Arquitectura de Software

A partir de este análisis el Ingeniero Líder de Software puede establecer y documentar y actualizar la estructura integral de su equipo, en el Plan de Desarrollo de Software. Esta información en conjunto con el calendario de actividades del proyecto ayudará a administrar los recursos y entender dependencias críticas dentro de las actividades del desarrollo, así como tomar acciones preventivas para evitar riesgos en la ejecución del proyecto.

Sub-equipo 1			
Participante	Ubicacion	Experiencia	Disponibilidad
Ingeniero 1	Sitio 1	x	x
Ingeniero 2	Sitio 1	x	x
Ingeniero 3	Sitio 1	x	x

Sub-equipo 2			
Participante	Ubicacion	Experiencia	Disponibilidad
Ingeniero 1	Sitio 2	x	x
Ingeniero 2	Sitio 1	x	x

Sub-equipo 1			
Participante	Ubicacion	Experiencia	Disponibilidad
Ingeniero 1	Sitio 3	x	x
Ingeniero 2	Sitio 3	x	x
Ingeniero 3	Sitio 3	x	x

Sub-equipo 1			
Participante	Ubicacion	Experiencia	Disponibilidad
Ingeniero 1	Sitio 4	x	x

Figura 11 Ejemplo de Estructura Integral de Equipo

3. El último paso en el establecimiento de la estructura integral del equipo, será incluir un reporte de su estado y cambios durante las revisiones periódicas con el gerente de software.

La implantación de esta propuesta de mejora requiere que el proceso de planeación incluya como entrada al mismo una arquitectura inicial del proyecto, lo cual implica un conocimiento a priori de los requisitos preliminares. Esta información deberá ser obtenida a partir del proceso de cotización del proyecto.

3.3 Asignar Requerimientos a los Equipos Integrales

La siguiente práctica específica de IPPD requiere la asignación de requisitos, responsabilidades, tareas e interfaces a cada equipo integral. De manera adicional, esta práctica requiere que esta actividad sea definida antes de que los equipos sean formados. Para lograr cumplir con esto, dentro del procedimiento de Administración de Proyectos de Software en el paso inicial ejecutado por el Gerente de Software, “Asignar al Ingeniero líder del Proyecto y roles adicionales al proyecto”, es necesario agregar esta práctica de IPPD.

La propuesta de mejora al proceso de desarrollo de software organizacional sugiere los siguientes pasos:

- El Gerente de Software deberá identificar el Ingeniero Líder de Software para ejecutar este análisis.
- El Gerente de software en conjunto con el Ingeniero Líder de Software realizarán el siguiente proceso:
 - Utilizando la información de arquitectura de software del producto a desarrollar y los requerimientos del proyecto establecer una estructura integral de equipo inicial.
 - Una vez identificada la distribución inicial, contactar la gerencia relacionada con cada equipo integral identificado, solicitando la revisión de los requisitos que les correspondiesen, con el fin de comunicar la disponibilidad de la ejecución, y sus estimados de desarrollo.
 - Revisar las respuestas de los equipos integrales y actualizar la estructura integral de equipo conforme sea necesario. Una vez que se logre confirmar una estructura integral de equipo, confirmar dicha estructura integral de equipo con las diferentes gerencias asociadas a los miembros del equipo.
 - Una vez confirmadas las respuestas, se procederá a la formación del equipo integral y se establecerán autoridades y responsabilidades.
 - Ejemplos, la determinación de un líder del equipo integral, la subdivisión en más equipos, los reportes de progreso.

Cabe mencionar que en esta organización, los recursos son establecidos desde el momento de la cotización del proyecto, por lo que las gerencias asociadas a los miembros de los equipos integrales saben de antemano que, al participar sus ingenieros en la estructura integral de equipo, todo el tiempo de ingeniería que trabajen en el proyecto será pagado el mismo.

Esta práctica específica es de suma importancia para la ejecución en proyectos Multi-sitio debido a que actualmente que la gerencia y disponibilidad de los ingenieros se administra de manera independiente en cada sitio, y por lo tanto, el establecer un acuerdo desde el inicio del proyecto ayuda a:

- El sitio líder para poder definir y obtener compromisos con el plan de desarrollo del proyecto.

- Los sitios de soporte, poder anticipar los recursos necesarios para poder cumplir con las solicitudes.

Entradas

Checklist de Inicio de Proyecto
 Checklist de Fin de Proyecto
 Plan de metas del Programa
 Requisitos Funcionales del Programa
Arquitectura Preliminar del Proyecto

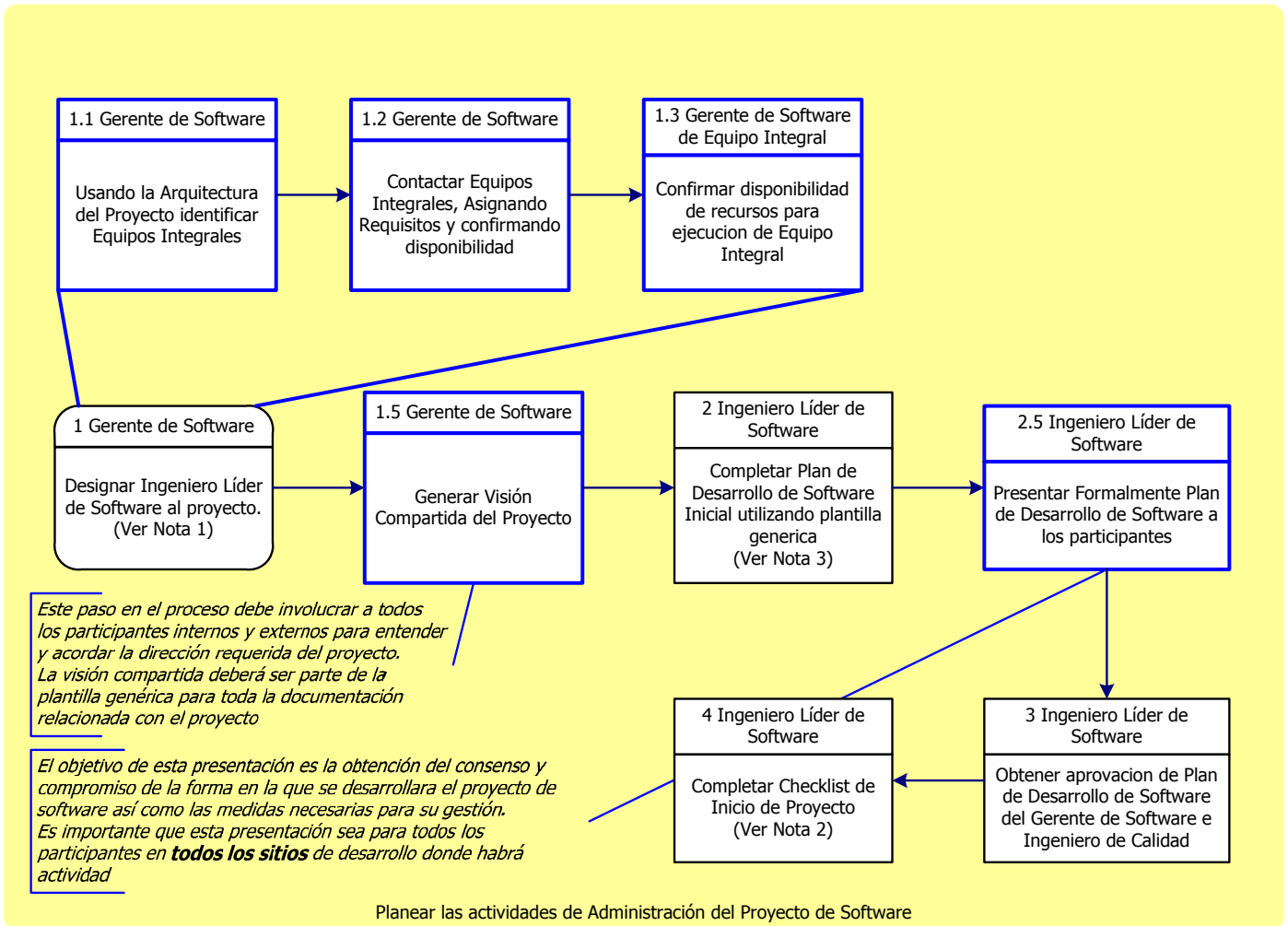


Figura 12 Asignar Requerimientos a los Equipos Integrales

3.4 Establecer Equipos Integrales

La práctica específica de IPPD para establecer los equipos integrales asigna la responsabilidad de su ejecución a los patrocinadores del equipo, que en el caso de la organización bajo revisión se trata de los gerentes locales a cada equipo integral de desarrollo. Esta práctica solicita la identificación del líder del equipo integral así como de sus participantes.

La propuesta de mejora es la siguiente:

1. Una vez que el sitio líder del desarrollo ha enviado la confirmación de la estructura integral del equipo, cada gerente local al equipo integral procederá a identificar el líder del equipo integral y de los participantes del desarrollo.
2. El Ingeniero Líder de Software del proyecto documentara y mantendrá la información de los ingenieros participantes en cada uno de los equipos integrales así como sus líderes en el Plan de Desarrollo de Software.
3. Cualquier cambio en un equipo integral deberá ser autorizado por el Gerente de Software del sitio líder así como del sitio local, comunicado y actualizado en el Plan de Desarrollo de Software.
4. El Ingeniero Líder de Software establecerá y documentara en el plan de desarrollo de software los parámetros para evaluar el desempeño del equipo integral y periódicamente revisara el estado y comunicara resultados a cada equipo integral y sus gerentes locales. Se sugieren los siguientes parámetros:
 - a. Entregas de acuerdo a calendario.
 - b. Asistencia a juntas y revisiones.
 - c. Estado de trazabilidad de requerimientos.
 - d. Reportes de Aseguramiento de Calidad.
5. Basado en la visión y el Plan de Desarrollo de Software cada líder del equipo integral distribuirá las tareas asignadas para los componentes de software que es responsable y mantendrá un calendario local de actividades.
6. El líder del equipo integral será responsable de reportar al Ingeniero Líder de Software un estado periódico de las actividades del desarrollo para los requisitos alojados a su equipo integral.
 - a. El estado periódico de las actividades deberá seguir el mismo contenido del reporte periódico de estado del proyecto documentado en el Plan de Desarrollo de Software.

7. El Gerente de Software revisara el estado de las actividades de cada equipo integral con el Ingeniero Líder de Software y compartirá dicho reporte con los gerentes locales de desarrollo para cada uno de los equipos integrales. Se tomaran acciones correctivas al identificar desviaciones al plan de desarrollo.

Esta práctica específica IPPD formaliza la definición de los equipos integrales así como su ejecución y seguimiento durante el desarrollo del proyecto. Cuando los equipos se encuentran bajo un solo Gerente de Software se puede simplificar la administración del desarrollo, pero cuando el desarrollo se encuentra distribuido con múltiples Gerentes de Software y múltiples sitios de desarrollo su ejecución para evitar confusión y facilitar la comunicación de información para la toma de decisiones se hace critica.

3.5 Asegurar Colaboración entre Equipos Interfaceando

La siguiente práctica específica es utilizada una vez que se ha establecido la estructura integral del equipo, monitorear que la colaboración entre los equipos integrales ocurre de manera exitosa y los conflictos son resueltos de manera satisfactoria.

La propuesta de mejora en términos de procesos para la organización se da ya durante la ejecución del proyecto. Como se recordara se sugiere la definición de los equipos integrales de trabajo con base en la arquitectura de software del producto. Dicha arquitectura es elaborada por el Arquitecto de Software del producto y entre las características de cada modulo funcional se encuentra la de tener bajo acoplamiento y alta cohesión, es decir son módulos funcionales altamente independientes. Si esta característica se mantiene al máximo, se puede definir que las interfaces entre los módulos deben ser claras y determinadas de antemano por la arquitectura de software, contribuyendo a identificar claramente los límites del desarrollo para cada equipo integral. Para lograr que la colaboración entre los equipos integrales sea efectiva y eficiente se proponen las siguientes guías:

- El Arquitecto de Software será responsable de que las interfaces entre los componentes de software estén documentadas y sean comunicadas a los equipos integrales. En caso de que un equipo requiera interfaces adicionales, el Arquitecto de Software deberá revisar y autorizar dichas interfaces.
- El Arquitecto de Software deberá resolver las confusiones o desacuerdos de las interfaces definidas para el proyecto.

- El Ingeniero Líder de Software es responsable de negociar y definir las metas intermedias en términos de funcionalidad para el proyecto de software, de ahí que, la prioridad de la implementación de los componentes de software será definida por el Ingeniero líder de Proyecto.
- El Ingeniero Líder de Proyecto resolverá las diferencias en los calendarios de los equipos integrales para minimizar dependencias y minimizar tiempos muertos por espera de funcionalidad. De esto se responsabiliza a los líderes de cada equipo integral, en mantener su calendario y traer a la atención del Ingeniero Líder de Software dichas dependencias que pueden poner en riesgo la cadencia del desarrollo.
- Los líderes de cada equipo integral mantendrán la lista de compromisos adquiridos y reportara su estado periódicamente al Ingeniero Líder de Software. En caso de retrasarse se tomaran acciones correctivas para mitigar el efecto.

4 Conclusiones

La

presente tesis es desarrollada con la finalidad de proponer una metodología de administración de proyectos de software, los cuales son desarrollados por múltiples ingenieros en múltiples sitios de desarrollo distribuidos geográficamente en el mundo, que permita mejorar la efectividad de este proceso basado en el modelo de CMMI y tomando como base un proceso ya establecido en una organización ya operando bajo una versión anterior del modelo, CMM-1997, y ya certificada con nivel 3 dentro del modelo.

- La primera conclusión es que el modelo de CMMI definitivamente ofrece mejores practicas para lograr establecer un proceso que permita la planeación y ejecución de proyecto de software con participantes que se encuentren distribuidos en diferentes sitios de desarrollo, diferentes horarios y tal vez con desarrollos asíncronos y en cierto grado independientes. En particular las prácticas específicas relacionadas a IPDD están mas cercanamente diseñadas con este propósito.
- El modelo de CMMI establece el concepto de equipos integrales, para la distribución de las actividades, sin embargo me parece que no hace un énfasis más estricto en el apoyo de una arquitectura de software para facilitar el establecimiento y administración de dichos equipos integrales. Se puede observar que en las propuestas de mejora, utilicé el concepto de arquitectura de software como base para la toma de decisión de la planeación del proyecto, y eventual evaluación de la ejecución del mismo, creo que esto a nivel de ejecución será de utilidad durante la ejecución de este tipo de proyectos.
- Se puede observar que la mayoría de las propuestas de mejora se localizan al inicio del proceso de planeación, esto es interesante dado que podemos leer del modelo, que es de suma importancia establecer un plan de desarrollo de software lo mas robusto posible antes de comenzar a gastar en recursos. Pensando en la forma en la que se llevan los proyectos actualmente en la organización creo que se puede observar un inicio de proyectos con cierto caos y redundancia de tareas donde eventualmente se estabiliza. Parece que el establecimiento firme de este proceso, permitirá evitar dicho caos mejorando la efectividad y eficiencia del los recursos asignados al desarrollo.

- Aunque las propuestas de mejoras al proceso de planeación son pensadas en el alcance de un contexto de desarrollo multi-sitio, creo que su aplicación es útil aun cuando el proyecto no esta distribuido en múltiples sitios, como una forma de lograr una organización y distribución del trabajo de desarrollo mas efectiva, y aun, en preparación para la necesidad del potencial traslado del desarrollo de un componente a ser desarrollado en un sitio remoto o por otra empresa.
- Revisando las propuestas y colocándose en un diferente punto de vista, se puede resumir que dichas propuestas establecen reglas y métodos para finalmente mejorar la comunicación en beneficio del “trabajo en equipo”, donde finalmente parece que el desarrollo distribuido multi-sitio comparativamente con un desarrollo localizado en un solo sitio de desarrollo, simplemente amplifica las deficiencias de comunicación entre el equipo en detrimento de la eficiencia del desarrollo de software.
- La comunicación efectiva es un aspecto que no necesariamente se desarrolla como parte de las habilidades técnicas de un ingeniero. Si además agregamos ingredientes como, diferencias de idiomas, uso horario y culturales, nos encontramos con que los ingenieros se enfrentan a retos donde su eficiencia y efectividad no solo depende de que sepan lenguajes de programación, estándares de calidad, metodologías de desarrollo etc. Sino también en su capacidad para coordinarse y trabajar en equipo. De ahí que formalizar los pasos que ayuden a que la coordinación ocurra será de fundamental importancia para mitigar dichos aspectos.

4.1 Trabajo Futuro

Un aspecto que no cubre esta tesis es el proceso de planeación de un proyecto que actualmente se ejecuta en un contexto local y se decide distribuir su trabajo en múltiples sitios de desarrollo. En primera instancia se parece que esto requiere nuevamente la ejecución del proceso de planeación desde el inicio, tal vez omitiendo ciertos pasos como la definición de Ingeniero Líder de Software, debido a que esto ya estaría hecho, sin embargo, el caso de uso deberá ser evaluado.

A partir de las conclusiones obtenidas en la presente tesis también se abre la puerta para evaluar la formación de ingenieros de software en las universidades. Como trabajo futuro queda la evaluación del perfil de egreso de ingenieros de software para poder desatacar en proyectos de desarrollo de software en coordinación y “competencia” con ingenieros en diferentes países.

Apéndice I

Glosario

Abstracción – (1) Nivel de detalle técnico de una representación del software; (2) Modelo cohesivo de datos o procedimientos

Administración de Calidad – Grupo de acciones de ingeniería que ayudan a asegurar que el software se construye de una forma en la que se alcanza una alta calidad.

Administración de Cambios – Grupo de acciones de ingeniería que ayudan a asegurar que los cambios de software son identificados, controlados y reportados.

Administración de Riesgos – Plan para mitigar, monitorear y administrar los riesgos.

Alcance – Enunciado delimitado de lo que se debe cumplir.

Alcance del Proyecto – Enunciado de los requisitos básicos del software a ser construido.

Análisis – Grupo de actividades que buscan entender y modelar las necesidades del cliente y sus restricciones.

Análisis de Requisitos o Requerimientos – Actividad de modelado cuyo objetivo es entender lo que el cliente realmente quiere.

Análisis de Riesgos – Técnicas para identificar y evaluar riesgos.

Arquitectura – Estructura total de los componentes de software, los datos y las relaciones entre ellos.

Aseguramiento de Calidad de Software (SQA) – Serie de actividades que ayudan a la organización en la producción de software de alta calidad.

Auditoría de Configuración – Actividad realizada por el grupo de aseguramiento de calidad con la intención de asegurar que el proceso de control de cambios está funcionando adecuadamente.

Baseline – Un punto en el cual un(os) producto de trabajo es/son puesto bajo un proceso de control de cambios formal durante el proceso de ingeniería de software.

Calidad – El grado en el que un producto cumple tanto con los requisitos explícitos como implícitos.

Calidad de Software – ver Calidad

CASE – Ingeniería de Software Asistida por Computadora

Caso de Uso – Descripción escrita que describe una interacción muy específica entre el usuario y el sistema.

Cliente – Persona o grupo que solicita el software y que pagará por su desarrollo.

Complejidad – Medida cuantitativa de la complejidad de un programa.

Componente (también llamado Componente de Software) – Bloque funcional de Software.

Confiabilidad – Medida del grado en el que el software opera confiablemente en un período de tiempo.

Configuración – Colección de programas, documentos y datos que deben ser controlados cuando es necesario hacer cambios.

Control de Cambios – Procesos que habilitan al equipo para aceptar, evaluar y ejecutar cambios de una manera sistemática.

Control de Configuración – Control de cambios a programas, documentos y datos.

Control de Proyecto – Control de calidad y de cambios.

Debugging – Actividad asociada a encontrar y corregir defectos.

Descomposición Funcional – Técnica utilizada durante la planeación, análisis y diseño, donde se crea una jerarquía funcional del software.

Diagrama de Flujo de Datos (DFD) – Notación que representa una descomposición funcional de sistema.

Diagrama de Transición de Estados (STD) – Notación que representa un comportamiento funcional de sistema.

Diseño – Actividad que traduce los requisitos a modelos detallados que guían su implementación.

Diseño Arquitectónico – Actividad que intenta describir disposición general del software.

Diseño de Caso de Prueba – Actividades que buscan genera casos de prueba efectivos y eficientes.

Diseño Detallado – Actividad de diseño que se enfocan en el refinamiento de la arquitectura de software.

Documentación – Información descriptiva.

Documentos – Resultados producidos como parte del proceso de ingeniería de software.

Elemento de Configuración – Partes individuales de programas, documentos y datos que son controlados utilizando procesos de control de cambios.

Errores – Falta de cumplimiento de requisitos.

Esfuerzo – Relación tiempo-trabajo-producto asociado al proyecto.

Especificación de Diseño – Documento que describe el diseño.

Especificación de Requisitos de Software – Documento que describe todos los requisitos de funcionalidad y comportamiento del software. Todas las restricciones y todos los requisitos de validación del software.

Estimación – Actividad de planeación de software que intenta proyectar el esfuerzo y costo del desarrollo de software.

Estructura de desglose de trabajo (WBS) – Grupo de tareas requeridas para desarrollar el software, definidas como parte del modelo de proceso.

Flujo de Trabajo – Secuencia de tareas requerida para completar alguna actividad o acción.

Generación Automática de Código – Herramientas que generan código fuente a partir de una fuente que no es código.

Herramientas – Software de aplicación que es utilizado para desarrollar la ingeniería de software.

Ingeniería de Requisitos o Requerimientos – Actividades requeridas para obtener, elaborar, negociar, especificar y validar requisitos de software.

Ingeniería de Reversa – Intento de desarrollar modelos de diseño a partir del entendimiento de código de un programa como punto de inicio.

Ingeniería de Sistemas – Disciplina enfocada en el análisis y el diseño de todos los elementos de un producto, servicio o tecnología para la transformación o control de información.

Ingeniería de Software – Disciplina que engloba los procesos asociados al desarrollo de software, los métodos utilizados para analizar, diseñar y probar software, las técnicas de administración asociadas con el control y monitoreo del proyecto de software y las herramientas utilizadas para dar soporte al proceso, métodos y técnicas.

Integración – Integración de elementos funcionales de software desarrollados de manera separada.

ISO 9001:2000 – Standard de aseguramiento de calidad.

LOC – Líneas de Código

Mantenibilidad – El grado de facilidad de cambiar un programa.

Mantenimiento – Actividades asociadas con los cambios al software después de que se ha liberado a los usuarios.

Mantenimiento de Software – Ver mantenimiento.

Mejora de Proceso de Software (SPI) – Grupo de actividades de ingeniería de software que intentan mejorar el estado de las prácticas de ingeniería de software en una organización en particular.

Métricas – Mediciones específicas utilizadas para monitorear el progreso y desempeño del desarrollo de software.

Métricas de Calidad – Mediciones del desempeño de la ejecución de los procedimientos y prácticas de la ingeniería de software para un proyecto.

Métricas de Software – Mediciones cuantitativas del proceso y del producto.

Milestones – Un punto en el tiempo que se utiliza para indicar el progreso durante un proyecto.

Modelo Espiral – Modelo evolutivo de proceso desarrollo de software. Modelo para describir el ciclo de vida del

proyecto de software.

Modularidad – Atributo cualitativo de un diseño que contribuye a la creación de componentes.

Outsourcing – Contratación de desarrollo de software a una tercera parte.

Plan de Proyecto – Descripción del enfoque de administración de un proyecto.

Plan y Procedimiento de Pruebas – Descripción de la estrategia de prueba.

Planeación del Proyecto – Actividad que crea un Plan de Desarrollo del Proyecto.

Portabilidad – Habilidad de transportar la funcionalidad de un software de un ambiente de ejecución a otro.

Productividad – Cantidad de trabajo por unidad de tiempo.

Producto de Trabajo – Información cohesiva y persistente que se produce como consecuencia de una o mas acciones de ingeniería de software.

Pruebas – Actividad que intenta encontrar errores.

Pruebas de Caja Blanca – Técnica de diseño de pruebas que utiliza el conocimiento de la lógica interna del programa.

Pruebas de Caja Negra – Técnica de diseño de pruebas que no utiliza el conocimiento de la lógica interna del programa, sino tan solo en los requisitos de interfaces externas.

Pruebas de Integración – Etapa de pruebas que verifica que el software funciona de acuerdo sus requisitos una vez que las unidades funcionales dispersas han sido integradas.

Pruebas de Regresión – Pruebas que son conducidas de manera repetitiva para evaluar que no se han generado efectos secundarios a un cambio de software.

Pruebas de Software – Grupo de actividades conducidas para encontrar errores en el software.

Pruebas Unitarias – Estrategia de pruebas que se enfoca en probar componentes individuales de software identificados como unidades.

Rastreo del Proyecto – Actividad que habilita al gerente a entender el estado del un proyecto.

Recursos – Todo lo requerido para ejecutar el proyecto.

Reporte de Cambio – Detalle del trabajo realizado para completar un cambio.

Reusabilidad – Habilidad del software a ser re-utilizado en otra aplicación.

Riesgo – Problema potencial.

Riesgos de Proyecto – Grupo de problemas potenciales que pueden causar la falla del proyecto.

Six sigma – Estrategia estadística para el aseguramiento de calidad.

Solicitud de Cambio – Detalle de tipo y expectativas del un cambio solicitado.

Stakeholders o Implicados – Cualquier persona o grupo que tiene acción en la consecución exitosa de un proyecto de software.

Tamaño del Proyecto – Indicación del esfuerzo total a ser ejecutado o el número total de personas trabajando en el proyecto.

Usabilidad – Medida informal de la facilidad de uso de un software.

Usuario – Persona que usa el software o el producto donde se ejecuta el software.

Validación – Pruebas para asegurar el cumplimiento de los requisitos de software.

Apéndice II

Acrónimos

CMM	Capability Maturity Model. Modelo de Madurez de Capacidad.
CMMI	Capability Maturity Model Integrated. Model de Madurez de Capacidad Integrado.
EU	Estados Unidos de América
ISO	International Organization for Standardization (Organización Internacional de Estandarización)
KPA	Key Process Area (Área Clave de Proceso).
SEI	Software Engineering Institute.
SW	Software
TS	Technical Specification (Especificación Técnica)

Referencias Bibliográficas

Libros

Carnegie Mellon Univ. Software Engineering Inst., Mark C. Paulk, Charles V. Weber, Bill Curtis, Mary Beth Chrissis. 1994. The Capability Maturity Model: Guidelines for Improving the Software Process. Addison-Wesley Professional ISBN 0-201-54664-7.

Richard H. Thayer (Editor), Edward Yourdon (Foreword). 1997. Software Engineering Project Management. Wiley-IEEE Computer Society Pr. 2000 ISBN 0-8186-8000-8.

Ishikawa, Kaoru. 1976. Guide to Quality Control. Asian Productivity Organisation, Tokyo ISBN 9283310357

Reportes

CMMI Product Team. 2006. CMMI® for Development, Versión 1.2., Technical Report CMU/SEI-2006-TR-008 <http://www.sei.cmu.edu/reports/06tr008.pdf> liga accedida el 6 de Abril 2011.

CMMI Product Development Team. 2000. .CMMI for Systems Engineering/Software Engineering, Version 1.02, Staged Representation (CMMI-SE/SW, V1.02, Staged), Technical Report CMU/SEI-2000-TR-018 <http://www.sei.cmu.edu/reports/00tr018.pdf> liga accedida el 4 de Noviembre de 2009.

National Aeronautics and Space Administration, Goddard Space Flight Center National Technical Information Service. 1990. Manager's Handbook for Software Development Revision 1 <http://homepages.inf.ed.ac.uk/dts/pm/Papers/nasa-manage.pdf>, liga accedida el 4 de Noviembre de 2009.
ASIN: B00010N998