

Universidad de las Ciencias Informáticas

Estrategia para la estandarización de la documentación y las
actividades para el desarrollo de software en la Universidad de las
Ciencias Informáticas

Tesis en opción al título de
Máster en Ciencias

Por

Autor: Ing. Ramsés Delgado Martínez

Tutor: Dra. Ailyn Febles Estrada

Ciudad de La Habana, Cuba

Noviembre de 2010

Agradecimientos

Al escribir este epígrafe de la tesis me vienen a la mente muchas personas que durante todo este tiempo me han apoyado y dado ánimo y otros que por un motivo u otro ya no están, pero que donde quiera que estén quisiera agradecerles también por su amor y cariño.

A riesgo de olvidar a alguien quiero agradecer

a mi Nana que por más que haga nunca bastará para agradecerle todo lo que hizo por mi,

a mis padres y a Albert que me han apoyado y han estado a mi lado todos los días,

a mi Tita por las alegrías y orgullo que me embriaga todos los días,

a mi hermano y abuelos por su cariño,

a mi tutora que sin su apoyo no se cómo hubiera llegado hasta este momento,

a Mari por regalarme su cariño,

a Iri por aguantarme durante tanto tiempo,

al Gordo, Gio y Fito que en la distancia siempre están ahí para mí,

a Sonny por competir conmigo y hacerme pelear tanto,

a Tay por estresarme y ayudarme tanto,

a mis niñas y Dennis porque aquí hay un pedacito de ellos también,

a mis compañeros de oficina que me han apoyado en todo momento,

y por si se me fuera a quedar alguien.... a todos aquellos que en algún momento me preguntaron “eh, y la tesis qué?”.

Resumen

El mercado de soluciones informáticas constituye uno de los de más auge en los últimos años. Las estadísticas muestran el crecimiento de las ganancias de los países por este concepto, pero penetrarlo constituye un reto dado el creciente número de empresas y la especialización de las mismas tanto en los tipos de aplicaciones como en el área de mercado.

Sin dudas uno de los factores de éxito para entrar en el mercado y mantenerse es ofrecer productos con mayor calidad, pero para lograr esto es necesario lograr estandarizar los procesos y artefactos para el procesos de desarrollo.

El presente trabajo propone una estrategia para la implantación del Expediente de Proyecto y los Lineamientos de Calidad en la Universidad de las Ciencias Informáticas.

Se realiza un estudio bibliográfico de los principales modelos, normas, estándares y metodologías para confeccionar la propuesta.

Como resultados del trabajo se obtuvo el Expediente de Proyecto, los Lineamientos de Calidad, cómo realizar la integración con los resultados obtenidos en el Programa de Mejora de Procesos y cómo implantarlo en la producción.

Índice de contenidos

Introducción.....	1
1 Fundamentación teórica.....	8
1.1 Introducción.....	8
1.2 Calidad de software.....	8
1.2.1 Gestión de la calidad.....	9
1.3 Estándares, normas y modelos de calidad para el desarrollo de software.....	9
1.3.1 ISO	10
1.3.1.1 ISO 90003:2004.....	10
1.3.1.2 ISO/IEC 12207:2005.....	11
1.3.2 IEEE.....	13
1.3.3 CMMI.....	14
1.3.4 MoProSoft.....	18
1.4 Procesos para el desarrollo de software.....	20
1.4.1 RUP.....	21
1.4.2 Metodologías ágiles.....	23
1.5 Buenas prácticas para el desarrollo de software.....	24
1.6 Documentación de proyectos software.....	26
1.6.1 Artefactos del Proceso Unificado de Rational (RUP).....	27
1.6.2 Artefactos NASA.....	28
1.6.3 Ready Set.....	28
1.7 Conclusiones.....	31
2 Propuesta de solución.....	32
2.1 Introducción.....	32
2.2 Estrategia.....	32
2.3 Estandarización de la documentación. Expediente de proyecto.....	33
2.3.1 El expediente de proyectos, su estructura general.....	34
2.3.1.1 Ingeniería.....	35
2.3.1.2 Gestión de proyecto.....	37
2.3.1.3 Soporte.....	38
2.3.2 Necesidades organizacionales.....	39
2.3.3 Integración con estándares.....	39
2.3.4 Adiciones.....	41
2.4 Estandarización de la producción. Lineamientos de calidad.....	42
2.4.1 Antecedentes.....	42
2.4.2 Lineamientos de calidad.....	43
2.5 Estrategia de continuidad. Programa de mejora.....	45
2.5.1 Integración del Expediente de proyecto al programa de mejora	45
2.5.2 Integración lineamientos de calidad.....	48
2.5.3 Evolución.....	49
2.6 Conclusiones.....	50
3 Validación de la hipótesis y presentación de resultados.....	51

3.1	Introducción.....	51
3.2	Estrategia de implantación.....	51
3.2.1	Reglas para la implantación.....	51
3.2.2	Acciones.....	53
3.2.3	Cronograma.....	53
3.2.4	Evaluación.....	54
3.3	Diagnóstico, revisiones y pruebas.....	55
3.3.1	Revisión 1.....	57
3.3.2	Revisión 2.....	57
3.3.3	Revisión 3.....	58
3.3.4	Revisión 4.....	59
3.3.5	Revisión 5.....	60
3.4	Pruebas de software.....	61
3.5	Mejoras.....	62
3.6	Conclusiones.....	66
4	Conclusiones.....	68
5	Recomendaciones.....	69
6	Referencias bibliográficas.....	70
7	Anexos.....	75
7.1	Anexo 1. Ciclo de vida del software según la ISO 12207.....	75
7.2	Anexo 2. Guía exploratoria para revisiones.....	75

Índice de figuras

Figura 1. Áreas de procesos por niveles de CMMI.....	15
Figura 2. Flujos de trabajo y fases de RUP.....	22
Figura 3. Ciclo de vida básico para metodologías ágiles.....	23
Figura 4. Estructura del Expediente de Proyecto.....	35
Figura 5. Proceso de integración del Expediente de Proyecto.....	47
Figura 6. Principales problemas revisión de febrero.....	57
Figura 7. Principales problemas revisión de marzo.....	58
Figura 8. Principales problemas revisión mayo.....	59
Figura 9. Principales problemas revisión junio.....	60
Figura 10. Principales problemas revisión octubre – noviembre – diciembre.....	61
Figura 11. Costo de defectos por etapa de detección.....	62
Figura 12. Estructura Expediente de Proyecto v2.0.....	65

Índice de tablas

Tabla 1. Distribución de niveles CMMI en América Latina.....	1
Tabla 2. Relación de normas ISO para software.....	10
Tabla 3. Estructura de la norma ISO/IEC 12207:2005.....	12
Tabla 4. Comparación de propuesta para documentación del software.....	30
Tabla 5. Plantillas del Expediente de Proyecto.....	41
Tabla 6. Planificación de revisiones según tipo de proyecto.....	57
Tabla 7. Artefactos del Expediente de Proyecto.....	66

Introducción

La Industria del Software, a pesar de ser de las últimas en llegar al mercado, es la que, en estos momentos, cuenta con los índices de desarrollo más elevados. Entrar y posicionarse al mercado de software presupone un reto para toda empresa del ramo. Son muchos los países que han apostado a esta rama de la industria como una fuente importante de ingresos para la economía. Como ejemplo ilustrativo podrían mencionarse los alrededor de 5 mil millones de dólares que se calculan como ingresos de la India anualmente [Síntesis, 2005], incluso países en vías de desarrollo como Venezuela en el año 2003 reportaron ingresos cercanos a los \$237 millones de dólares por el concepto de producción de software. Estos números son indicadores de la existencia de un mercado que, aunque nada fácil de conquistar, puede constituir en pilar del crecimiento económico de no pocos países.

Para poder mantenerse en un mercado tan competitivo la calidad de las producciones se ha convertido en un factor determinante. Dado lo abstracto que pudiera resultar el término calidad las empresas desarrolladoras de software han optado por obtener certificaciones de sus procesos y productos basados en estándares y normas internacionales como pueden ser ISO, IEEE y Six Sigma. Tan solo en América Latina, en el año 2007 existían un total de 66 empresas con nivel 2 de CMMI o superior. La tabla 1 muestra la distribución por países y niveles [Pleue 2007].

País	Cantidad nivel 2	Cantidad nivel 3	Cantidad nivel 4	Cantidad nivel 5
Venezuela	1	0	0	0
Colombia	1	0	0	1
Perú	1	0	0	0
Chile	16	8	0	3
Bolivia	0	1	0	0
Brasil	24	6	1	1
Argentina	1	1	0	0
Uruguay	0	0	0	1

Tabla 1. Distribución de niveles CMMI en América Latina

Sin dudas no es despreciable el número de empresas con algún nivel de CMMI entré las que resaltan las de Brasil y Chile que a su vez constituyen los países con mayor ingreso por concepto de software en la región.

Atendiendo a los elevados dividendos de la producción de software, Cuba ha visto en la esfera potencialidades para que la misma llegue a ocupar las primeras posiciones en las exportaciones del país. Pensando en este objetivo, se crea en el año 2002 la Universidad de las Ciencia Informáticas (UCI).

La UCI se ha propuesto como objetivo estratégico para el 2012 el ingreso de mil millones de dólares por concepto de software, sin dudas una cifra ambiciosa pero alcanzable[UCI 2008]. Para esto es necesario penetrar en otros mercados además del venezolano con el que, desde hace algún tiempo se viene trabajando. Los mayores mercados para el software a nivel mundial exigen como requisito indispensable la certificación mediante organismos autorizados del sistema de calidad de la empresa productora. De ahí la importancia de ir paso a paso estandarizando los procesos productivos en la Universidad pensando en una futura certificación.

A nivel mundial existen entidades que se han encargado de dictar pautas para la producción de software como: ISO, SEI, MoProSoft, IBM, IEEE y la NASA que proponen en algunos casos soluciones muy específicas y en otros muy generales, pero que pueden servir como guía para que una empresa establezca sus propias normas y estándares.

También algunos autores en sus publicaciones hacen referencia a elementos que deben tener en cuenta para el éxito de los proyectos. Angélica de Antonio, brinda su apreciación en [de Antonio 2001], sobre cómo el éxito de un proyecto y por consiguiente de manera sostenida el de una empresa de software depende de la correcta ejecución de cuatro tipos de funciones:

- La gestión del proyecto.
- El desarrollo técnico.
- El Sistema de Calidad, que incluye las actividades de Validación.

- El Sistema de Gestión de Configuración (GCS).

Basado en algunos de los elementos que establecen los estándares antes mencionados y referencias de la literatura se realizó en la UCI en diciembre 2007 un diagnóstico por parte de la Dirección de Calidad a los proyectos de la UCI [DCS 2008a]. Fueron evaluados 137 proyectos de diferentes tipos y características, detectándose de forma resumida las siguientes dificultades [DCS 2008a]:

Organización

- Gran diversidad de roles.
- Solapamiento de roles.
- No se tiene definición de las actividades de los roles.
- Poca capacitación en el rol a desempeñar.
- No todos los proyectos se encuentran regidos por una metodología.

Planificación y seguimiento

- No se gestionan los requisitos del sistema.
- No todos los desarrolladores participan en la comprensión de los requerimientos del sistema.
- No se gestionan los costos de los proyectos.
- Gran diversidad de lenguajes de programación.
- No se realizan actividades básicas dentro de todo proceso de desarrollo de software.

Calidad

- No se utilizan estándares para el desarrollo
- Los artefactos que se entregan a los clientes poseen diferentes formatos
- Se entregan al cliente artefactos con alto número de no conformidades
- Muy difícil ejecutar acciones de control de la calidad
- No se documenta el trabajo realizado

Mediciones

- No se cuenta con datos históricos para realizar futuras mediciones.

Con una simple mirada salta espontáneamente la interrogante de cómo revertir esta situación, que, sin lugar a dudas, pone en riesgo los esfuerzos de la UCI de incorporarse de manera seria en el mercado mundial de software ocasionando que no se pueda realizar un seguimiento efectivo de los proyectos, no se puedan transmitir experiencias de un proyecto a otro, los tiempos de desarrollo sean muy extensos y por consecuencia los ciclos de vida y el conocimiento se quede en las personas que lo generan.

También tiene un impacto significativo en los clientes finales de los productos. En ocasiones un proyecto está formado por varios subproyectos (multimedias, portales, plataformas de gestión) y estos a su vez pueden estar siendo desarrollados por distintos equipos de desarrollo. Al no existir una estandarización en la documentación para los proyectos no resulta extraño que un subproyecto entregue un artefacto en un formato y otro subproyecto entregue el mismo artefacto en otro formato, aspecto este muy negativo de cara al cliente. Por otro lado, al no existir una norma que indique cuáles son las actividades básicas a desarrollar por cada proyecto el cliente puede sentirse perdido en la interacción con varios equipos de desarrollo ejecutando tareas según su modo de realizar las actividades. Esto último, al obviarse actividades esenciales del ciclo de desarrollo, trae consigo que los productos lleguen a los clientes con un número significativo de defectos. Se debe sumar además la no utilización de normas internacionales que posibiliten la calidad de los procesos y por consiguiente la calidad de los productos.

Lo planteado anteriormente es producto de la no ejecución de acciones para el control de la calidad (revisiones, pruebas), las que no se pueden realizar de manera efectiva si no existen pautas y normas a ejecutar por los proyectos, lo que permitiría la definición de herramientas para ejecutar las actividades de control de la calidad.

La situación planteada lleva a definir el siguiente **problema de la investigación**:

¿Cómo organizar la documentación y las actividades para el desarrollo de software en los proyectos de la UCI para lograr que el control de la calidad en los proyectos se pueda

realizar de forma efectiva, de manera que se logre estandarizar la documentación de los proyectos en la UCI?

Objeto de estudio

- Gestión de la calidad en proyectos software.

Hipótesis

La estandarización de la documentación y buenas prácticas aplicables al ciclo de desarrollo de un proyecto software contribuye al efectivo control de la calidad y al aumento de la calidad de los productos resultantes.

Campo de acción

- Proyectos software de la Universidad de las Ciencias Informáticas.

Se define como **Objetivo general** definir una estrategia para la estandarización de la documentación y las actividades para el desarrollo de software en la Universidad de las Ciencias Informáticas y como **Objetivos específicos**:

- definir el Expediente de Proyecto
- definir los Lineamientos de Calidad
- definir la estrategia de implantación del expediente de proyecto y los lineamientos de calidad.

Tareas a realizar

- Realizar estudio del estado actual de las normas y estándares internacionales para la documentación de productos software.
- Realizar estudio del estado actual actual de las normas y estándares internacionales para el desarrollo de productos software.
- Identificar documentación necesaria para los proyectos de la UCI.
- Identificar actividades genéricas necesarias para la ejecución de proyectos software.
- Identificar relación con la mejora de procesos de desarrollo de software.
- Evaluar resultados de implantación en la producción.

Métodos de investigación utilizados

El método científico de investigación es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones. En el presente trabajo se emplearán métodos teóricos y empíricos.

Métodos teóricos

Los métodos teóricos que se emplearon en esta investigación son: el histórico-lógico, el hipotético-deductivo y el sistémico.

Se utilizó el **método histórico** pues se realizó una revisión crítica de la bibliografía existente sobre los temas de modelado de procesos para el desarrollo de software y en especial el software educativo y de esta manera lograr un criterio sobre el tema de investigación. También se empleó el método hipotético-deductivo, donde partiendo de la suposición de la existencia de un modelo para la producción de proyectos de software, se puede mejorar la calidad de su proceso y el resultado. Y además, se aplicó el método teórico sistémico pues se vio el problema de la investigación como un sistema y se trató como tal.

Métodos empíricos

Los métodos empíricos que se utilizaron son: la observación y la medición.

Primeramente se empleó la observación con el objetivo de llevar un seguimiento del fenómeno que se trata, y la medición para obtener datos estadísticos que ayuden a evaluar el resultado de la investigación.

Resultados a alcanzar

Como resultado del presente trabajo se obtendrá un Expediente de Proyecto para los proyectos de desarrollo de software para almacenar y organizar todos los artefactos resultantes del ciclo de vida; los lineamientos de calidad basados en las buenas prácticas para el desarrollo que estandarizarán las actividades a ejecutar por los proyectos así como

una estrategia para su implantación en la producción.

La implantación de ambos artefactos permitirá la confección de listas de chequeos tanto para la revisión de los proyectos durante su ciclo de vida como para los procesos de liberación, con lo que se detectarían un mayor número de no conformidades antes de su entrega al cliente.

Estructura del documento

El documento se encuentra estructurado en tres capítulos. En el Capítulo 1 se realiza el estudio del estado del arte de las normas y estándares internacionales, así como de las metodologías de desarrollo más utilizadas en la UCI y algunas propuestas de documentación para productos software. En el Capítulo 2 se define la propuesta de solución del Expediente de proyecto y los Lineamientos de calidad así como su integración con el programa de mejora de procesos y en el Capítulo 3 se realiza un estudio de los resultados de la implantación de los artefactos propuestos.

1 Fundamentación teórica.

1.1 Introducción

En el presente capítulo se presenta un estado de la organización y estandarización de los proyectos en el mundo basado en normas, estándares, modelos y metodologías para el desarrollo de software, así como los artefactos generados durante el desarrollo de los mismos.

1.2 Calidad de software

En la década del 90 se produce un auge a nivel mundial de la calidad, sin importar a que área se quiera aplicar el término: hospitales, universidades, atención al público. La industria como es de esperarse no escapó de este fenómeno y dentro de ella la industria del software [Bedini 2007][Garvin 1984].

Todas las personas aseveran que conocen qué es la calidad del software y que la aplican en todas las tareas que realizan siendo esto una afirmación difícil de realizar si atendemos lo intangible y ambiguo que puede resultar el término [Crosby 1979].

Numerosas entidades han dado su concepto de calidad. Por citar algunos tenemos que para IEEE [IEEE 1990] la calidad es el grado mediante el cual un sistema, componente o procesos satisface las necesidades o expectativas de un cliente o usuario.

Para CMMI calidad es la habilidad de un conjunto de características inherentes a un producto, componente o proceso para cubrir los requerimientos del cliente.

Para ISO calidad es el grado mediante el cual una serie de características inherentes al producto satisfacen los requerimientos [ISO 2005].

Uno de los conceptos más acertados en opinión del autor para calidad de software lo brinda Pressman [Pressmann 1998] donde plantea que es la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentos, y con las características implícitas que se espera de todo software desarrollado profesionalmente.

Como se aprecia en las definiciones anteriores existe un factor común entre todas: requisitos y clientes y no solo en la conformidad de este último con lo solicitado sino las expectativas que se tiene sobre el producto. Lo anterior se puede traducir en que los requisitos son la base para todas las medidas que calidad del software y la calidad de un producto la define el cliente.

1.2.1 Gestión de la calidad

Gestión de la calidad

Según ISO la Gestión de la Calidad es el conjunto de elementos que determinan y aplican la política de la calidad, los objetivos y las responsabilidades y que lo realiza con medios tales como la planificación de la calidad, el control de la calidad, el aseguramiento de la calidad y la mejora.[ISO, 2005]

Aseguramiento de la calidad

IEEE define como aseguramiento de la calidad un patrón de todas las actividades sistemáticas y planeadas para proveer la adecuada certeza que un producto o elemento cumple con los requerimientos técnicos establecidos. [IEEE, 1990a]

Control de la calidad

Así mismo IEEE define como control de la calidad a un conjunto de actividades para evaluar la calidad de los productos desarrollados. [IEEE, 1990a]

1.3 Estándares, normas y modelos de calidad para el desarrollo de software.

Afirmar que un producto software tiene calidad pudiera resultar algo complicado teniendo en cuenta la ambigüedad que trae consigo la palabra “calidad”.

A pesar de esto hay autores y entidades que brindan sus definiciones. Una de las más completas es la de Pressman [Pressmann 1998] donde plantea que calidad de software es “la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente”.

La tendencia a nivel mundial es atacar el tema mediando la utilización de normas, estándares y modelos internacionales para la calidad de software en donde destacan ISO, IEEE, CMMI y SPICE.

1.3.1 ISO

ISO es una red de institutos nacionales de estándares de 156 países que promueve la normalización internacional para facilitar el intercambio de bienes y servicios como de aplicaciones. [ISO 2005a]

En relación con el desarrollo de software existen un conjunto de normas ISO destacándose las que se muestra en la tabla 2.[Tantara 2009]

Norma ISO	Descripción
ISO/IEC 9126:1991	Calidad del producto.
ISO/IEC TR 9294:1990	Guías para la gestión de la documentaciones de software.
ISO/IEC 12207:2005	Ciclo de vida del software.
ISO/IEC 14143:1998	Mediciones de software.
ISO/IEC 15846:1998	Gestión de configuración de software.

Tabla 2. Relación de normas ISO para software.

De las normas antes mencionadas las que tienen un mayor impacto, por el número de actividades del proceso de desarrollo de software, son la ISO 90003:2004, la ISO/IEC 9294:1990 y la ISO/IEC 12207:2005.

1.3.1.1 ISO 90003:2004

Entre los aspectos que tiene en cuenta la Norma ISO 90003:2004 se encuentran:

- Sistema de calidad.
- Especificación de los requisitos del comprador.
- Planificación del desarrollo.
- Planificación de la calidad.

- Pruebas y validaciones.
- Gestión de Configuración de Software.
- Control de documentos .
- Mediciones

Sobre el control de los documentos plantea que deben crearse los mecanismos necesarios para que los documentos sean revisados, aprobados, legibles, se encuentren disponibles cuando sea necesario su uso y los documentos externos se encuentren protegidos.

En este punto no hace alusión a cuáles son los documentos necesarios para la construcción de software.

1.3.1.2 ISO/IEC 12207:2005

Además de la norma antes menciona la ISO cuenta con la Norma ISO/IEC 12207:2005 [ISO 2005b] que describe el ciclo de vida de desarrollo del software. (Ver anexo 1)

Dicha norma divide los procesos de desarrollo de software en tres categorías principales: procesos básicos, procesos de apoyo y procesos organizacionales. En la tabla 3 se resumen los elementos fundamentales de cada una [ISO 2005b].

Procesos Básicos	Procesos de Apoyo	Procesos Organizacionales
<ul style="list-style-type: none"> • Compra • Suministro • Desarrollo • Operación • Mantenimiento 	<ul style="list-style-type: none"> • Documentación • Gestión de configuración • Aseguramiento de calidad • Verificación • Validación • Revisión conjunta • Auditorías • Solución de problemas 	<ul style="list-style-type: none"> • Gestión • Infraestructura • Mejoramiento • Adiestramiento

Tabla 3. Estructura de la norma ISO/IEC 12207:2005

Referente a la documentación la norma plantea [ISO 2005a] que debe existir un proceso para registrar la información producida en las actividades del ciclo de vida de desarrollo del software. El proceso debe contar con las siguientes actividades:

- Implementación del proceso.
- Diseño y desarrollado.
- Producción.
- Mantenimiento.

Implementación del proceso.

La norma plantea que se deben identificar todos los documentos que se irán obteniendo en las fases de desarrollo del software y que cada documento debe contar con:

- a) Título o nombre.
- b) Propósito.
- c) Cantidad prevista de lectores.
- d) Procedimientos y responsabilidades para las entradas, desarrollo, revisión, modificación, aprobación, producción, almacenamiento, distribución, mantenimiento y gestión de la configuración.
- e) Cronograma para las versiones intermedias y finales.

Diseño y desarrollo

En este epígrafe la norma plantea que los documentos generados deben cumplir con el formato previamente establecido para las tablas, tipografía, figuras, encabezados, márgenes, portada, etc. También debe llevarse un control sobre las revisiones y los cambios realizados.

Producción

Para este apartado la norma plantea que los documentos deben generarse según lo planificado sin importar el soporte en que se generen ya sea digital o en papel y los mismos

deben ser almacenados según se haya estipulado atendiendo a la seguridad y mantenibilidad de los mismos.

Además deberán establecerse controles de acuerdo al proceso para al Gestión de la Configuración.

Mantenimiento

Para el mantenimiento la norma plantea que debe llevarse un control de las modificaciones y que aquellos documentos que se encuentren en el Sistema de Gestión de Configuración su modificación sea atendiendo al procedimiento especificado en dicho sistema.

1.3.2 IEEE

En enero de 1963 se crea el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) organización que hoy constituye referente en temas de calidad. Hoy cuenta con más de 375000 miembros alrededor de todo el mundo. Tiene como misión fomentar la innovación tecnológica y excelencia para el beneficio de la humanidad. [ISO 2004]

La IEEE provee estándares y lineamientos para muchas de las ramas de la ingeniería y dentro de ellas la Ingeniería Informática. Para ella en específico proporciona aproximadamente 62 estándares con un elevado precio de adquisición [ISO 2004].

Dentro de todos el más general para los procesos de desarrollo lo constituye el “IEEE Std. 1074-1997 Proceso para el ciclo de vida de desarrollo de software” [IEEE 2007]. En él se enumeran un conjunto de prácticas para llevar a cabo el ciclo de desarrollo del software. Estas actividades se encuentran divididas en 4 secciones: Gestión de proyectos, Pre – desarrollo, Desarrollo y Post – desarrollo y dentro de ellas abarcan las principales áreas para la construcción de software como son la captura de requisitos, gestión de configuración, implementación, análisis y diseño, etc.

Dentro del estándar antes mencionado se encuentra un epígrafe dedicado a la documentación de las actividades de desarrollo. Dichas actividades están encaminadas a planificar, diseñar, implementar, editar, distribuir y mantener la documentación del proyecto.

Consta de dos actividades fundamentales

- a) Implementar la documentación.
- b) Producir y distribuir la documentación.

Implementar la documentación

Incluye el diseño, preparación y mantenimiento de la documentación. Para esto se debe tener en cuenta el formato, estructura de los documentos, formas de presentar el contenido y los gráficos a utilizar para una mayor claridad de la información.

Producir y distribuir la documentación.

Consiste en proveer a los interesados la información necesaria mediante la documentación del proyecto sin importar en qué formato esta se presente.

Similar a la ISO, la IEEE no especifica cuáles son los documentos necesarios para tener en cuenta para la producción de software, ni la información que deben tener, aunque provee un macro proceso para la creación y mantenimiento de dicha documentación.

1.3.3 CMMI

Al inicio de la década de los 90, surgieron una serie de iniciativas para aplicar las mejores prácticas en el ámbito del desarrollo del software que se plasmaron en la realización y publicación de una serie de modelos y normas para el sector.

Uno de estos modelos fue Capability Maturity Model (CMM) desarrollado por el Instituto de Ingeniería de Software (SEI) en el año 1991[Brickey 2004].

CMM para software o SW-CMM no es más que un grupo de lineamientos para el desarrollo de software [Gartner 2001]. Una descripción de los escenarios por los cuales las organizaciones de software evolucionan, cómo ellos definen, implementan, miden, controlan y desarrollan sus procesos de software [CMM 1993][Royce 2006].

Después del lanzamiento del modelo SW-CMM se desarrollaron otros modelos para otras disciplinas. Pronto las empresas se encontrarían con que resultaba muy difícil integrar todos estos modelos, solucionar las lagunas detectadas, resolver las inconsistencias y aclarar las diferentes terminologías [E 2002].

Para dar solución a estos problemas surge en diciembre del 2001 CMMI. (Capability Maturity Model® Integration) [CMMI 2005] el cual es un enfoque para la mejora de procesos que brinda a las organizaciones los elementos básicos sobre procesos efectivos. Puede ser empleado para guiar la mejora de procesos a lo largo de un proyecto, división u organización. CMMI ayuda a integrar funciones organizacionales tradicionalmente separadas, otorga metas y prioridades en la mejora de procesos, brinda además una guía para los procesos específicos de calidad y un punto de referencia para la valoración de los procesos actuales [CMMI 2005].

Cuenta al igual que CMM con cinco niveles, aunque algunos cambian de nombre[E 2002].

Para poder escalar a un nivel superior es necesario haber cumplido todos los aspectos que plantea el nivel actual, por lo que es posible incluso, cumplir con aspectos de nivel 2 y de nivel 3 simultáneamente y, aun así, mantenerse en un nivel 1.

En la figura 1 se muestran los niveles de CMMI y sus áreas de procesos.

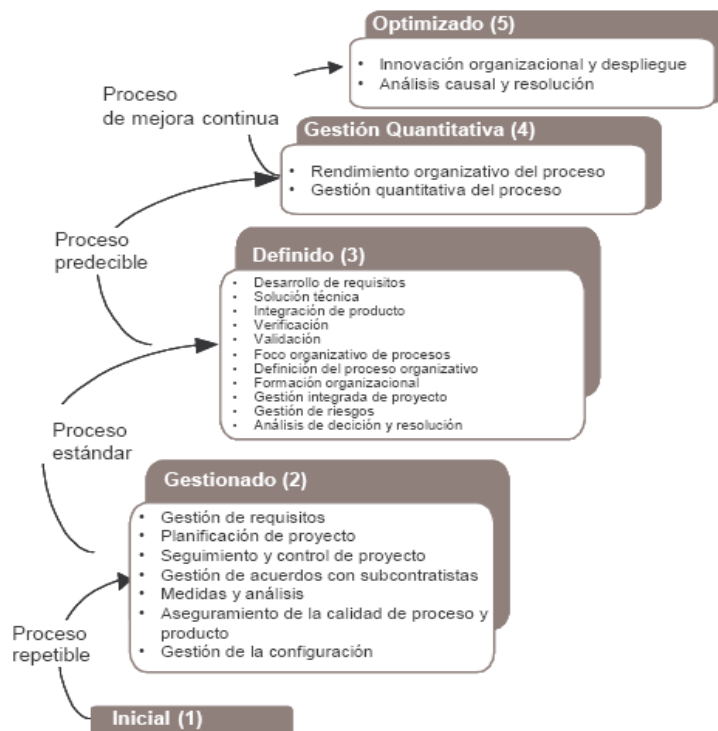


Figura 1. Áreas de procesos por niveles de CMMI.

Para obtener Nivel 2 de CMMI (Gestionado) es necesario cumplir con cada una de las siguiente prácticas áreas de procesos planteadas por el modelo [CMMI 2005]:

- 1) Gestión de los requisitos del producto y del proyecto.
- 2) Planificación de los proyectos.
- 3) Seguimiento y control de los proyectos.
- 4) Gestión de acuerdos con los proveedores de productos y servicios.
- 5) Selección y supervisión de los proveedores.
- 6) Medición y análisis.
- 7) Aseguramiento de la calidad del producto y del proceso.

Cada una de estas áreas de procesos cuentan con prácticas específicas que aseguran se le de cumplimiento al área de proceso.

Además de estas prácticas específicas, todos los procesos para el nivel dos deben cumplir con las 10 prácticas genéricas:

- 1) Crear una política organizacional.
- 2) Planificar el proceso.
- 3) Asignar recursos.
- 4) Asignar responsabilidades.
- 5) Entrenar al personal.
- 6) Gestionar la configuración.
- 7) Identificar los principales involucrados.
- 8) Monitorear y controlar el proceso.
- 9) Evaluar objetivamente la adherencia de los procesos y productos.
- 10) Informar resultados a los niveles superiores de dirección.

A muchas de estas prácticas genéricas se les da cumplimiento con algunos de los procesos descritos para el nivel 2, sin embargo otras, como Entrenar al personal, que se encuentra muy relacionada con el área de proceso de Entrenamiento Organizacional del nivel 3.

También existe un fuerte dependencia entre las áreas de proceso. Resulta imposible definir un proceso para la Gestión de Requisitos (nivel 2 de CMMI) sin revisar Desarrollo de

Requisitos (nivel 3).

Cada área de proceso, además de las prácticas específicas cuenta con información adicional como son las subprácticas y artefactos esperados que son opcionales para el modelo.

Si bien CMMI no propone un ciclo de vida específico para el desarrollo de software si propone un modelo para la definición de procesos organizacionales para el desarrollo de software y aunque no propone todos los artefactos necesarios ni la información que deben tener los mismos, si hace mención a algunos que, con su construcción, pueden generar evidencia de la ejecución del proceso.

Desde sus inicios el SEI se enfrascó en lograr una estandarización en los modelos a utilizar para la construcción de software. Muestra de ello es la creación del Proceso de Software Personal (PSP)[Humphrey 2001], el Proceso de Software de Equipo (TSP)[Humphrey 2000] y en 1991, como ya se había dicho anteriormente, del Modelo de Madurez y Capacidades (CMM)[CMM 1993]. Este último pretendía implantar un modelo de procesos por el cual medir la madurez y la capacidad de una empresa productora de software.

CMMI divide sus Áreas de Procesos (AP) en 4 grupos para representar las interacciones que existen entre las áreas de procesos del mismo grupo y de diferentes grupos. Estos grupos son:

- **Ingeniería.** Las áreas de proceso de Ingeniería cubren las actividades de desarrollo y de mantenimiento que se comparten entre las disciplinas de ingeniería. Estas fueron escritas usando terminología general de ingeniería de tal forma que cualquier disciplina técnica implicada en el proceso de desarrollo del producto pueda usarlas para la mejora de procesos.
- **Soporte.** Las áreas de proceso de Soporte cubren las actividades que dan soporte al desarrollo y al mantenimiento del producto. Estas tratan los procesos que se usan en el contexto de la ejecución de otros procesos. En general, las áreas de proceso de Soporte tratan los procesos que están orientados al proyecto y pueden tratar procesos que se aplican de manera más general a la organización.

- **Gestión de proyectos.** Las áreas de proceso de Gestión de proyectos cubren las actividades de gestión de proyectos relacionadas con la planificación, monitorización y control de proyectos.
- **Gestión de procesos.** Las áreas de proceso de Gestión de procesos contienen las actividades transversales a los proyectos relacionadas con la definición, planificación, despliegue, implementación, monitorización, control, evaluación, medición y mejora de los procesos.

1.3.4 MoProSoft

En diciembre de 2002 se crea a solicitud de la Secretaría de Economía de México el Modelo para la Industria del Software (MoProSoft)[Moprosoft 2008]. Este modelo fue diseñado para medir la capacidad de los procesos que siguen las empresas y para garantizar una calidad constante en el desarrollo y mantenimiento de software. Para esto se tomaron en cuenta estándares internacionales como: ISO 9000, ISO 15504, SW – CMM y CMMI.

MoProSoft se encuentra dividido en tres categorías: Alta Dirección, Gerencia y Operación. Dentro de cada una de estas se encuentran distribuidos los procesos de las siguiente forma:

Alta Dirección

- Gestión de negocio.

Gerencia

- Gestión de procesos.
- Gestión de proyectos.

- Gestión de recursos.
 - Recursos humanos y ambiente de trabajo.
 - Bienes servicios e infraestructura.
 - Conocimiento de la organización.

Operación

- Administración de proyectos específicos.
- Desarrollo y mantenimiento de software.

Cada uno de estos procesos están compuestos por actividades y se definen los roles que dan cumplimiento a las mismas.

Según datos de la Oficina de Normalización y Certificación Electrónica (NYCE) [NYCE 2009] en México existen actualmente 4 empresas con nivel 2 (administrado) de MoProSOFT y 81 con nivel 1 (realizado).

A diferencia de las normas antes mencionadas, MoProSOFT se acerca mucho más al cómo debemos realizar las actividades de desarrollo de software centrado en la Industria Mexicana del Software. Para la descripción de cada uno de los procesos define un conjunto de actividades, objetivos, indicadores, roles, entradas y salidas aunque no realiza aportes significativos en cuanto a las prácticas recomendadas para el desarrollo de software y se limita a apoyarse en las propuestas por CMMI e ISO.

En lo referente a la documentación para los procesos el modelo solo menciona cuáles son los artefactos que se esperan como entradas y como salidas de cada uno de los procesos

con una breve descripción del mismo. Sin embargo, no deja claro cuál será el contenido del documento, ni propone una estructura para la organización de la documentación.

Atendiendo a lo anteriormente planteado se considera que el modelo puede resultar útil en el momento de definir procesos en una organización, pero al nivel de buenas prácticas a aplicar en la industria no realiza aportes significativos ya que se basa en las mismas prácticas propuestas por CMMI aunque realiza otra organización de las mismas. En cuanto a la documentación para los proyectos, pudiera considerar el conjunto de artefactos propuestos como entradas y salidas de los procesos como posibles a estar dentro de un expediente para los proyectos de desarrollo de software.

Los estándares y modelos antes mencionados [Moprosoft 2008][CMMI 2005][IEEE 2007][ISO 2004] proponen un grupo de procesos y una estructura para la organización de los mismos. Según la opinión del autor, esta misma organización resultaría útil para la organización de la documentación asociada a los proyectos ya que estas van a responder directamente a los procesos y teniendo en cuenta lo general de la estructura y su fácil entendimiento se considera que la propuesta por CMMI es la más adecuada.

1.4 Procesos para el desarrollo de software

Un proceso de software se puede caracterizar como un marco común que define un número de actividades comunes que son aplicables a todos los proyectos, con independencia de su tamaño o complejidad. Es decir una colección de tareas de trabajo de ingeniería de software, hitos de proyectos, productos de trabajo y puntos de garantía de calidad, que permiten que las actividades dentro del marco de trabajo se adapten a las características del proyecto del software y a los requisitos del usuario. [Pressman 1998]

La definición de proceso de desarrollo planteada en el proceso de desarrollo de software plantea que: [RUP 2001]

Un proceso de desarrollo de software es una definición del conjunto completo de actividades necesarias para convertir los requisitos de usuario en un conjunto consistente de artefactos que conforman un producto de software y para convertir los cambios sobre esos requisitos en nuevo conjunto consistente de artefactos.

Existen principalmente dos tipos de metodologías: ágiles y robustas (o clásicas). Entre las ágiles las más utilizadas son Extreme Programming (XP) y SCRUM y entre las clásicas RUP y MFS. Cada una de ellas tiene ventajas y desventajas basadas sobre todo en el dominio de la aplicación a desarrollar.

1.4.1 RUP

El Proceso Unificado de Rational es un proceso de desarrollo de software y constituye “un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto”. [RUP 2001]

Dicho proceso utiliza el Lenguaje Unificado de Modelado (UML), para preparar todos los esquemas de un sistema software donde el UML es una parte esencial de RUP. Está basado en tres aspectos definitorios: guiado por casos de uso, centrado en arquitectura, e iterativo e incremental.

En el siguiente gráfico (Figura 2) se presentan los flujos de trabajos y fases de RUP.

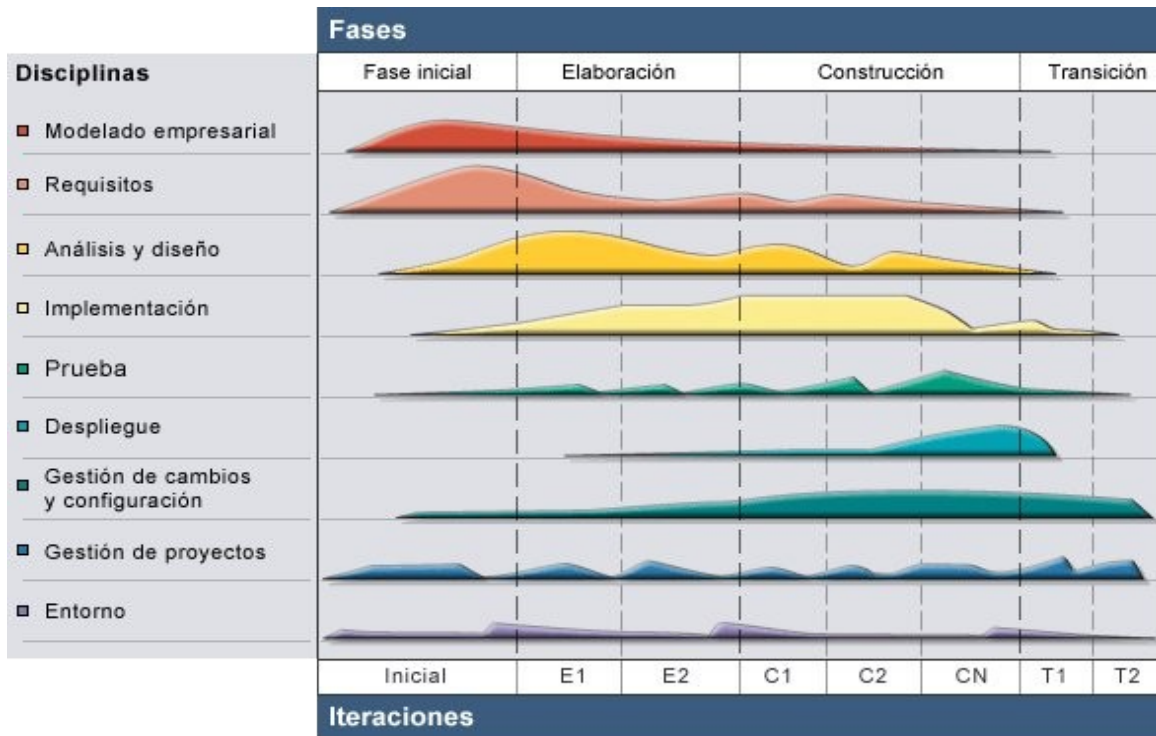


Figura 2. Flujos de trabajo y fases de RUP.

RUP plantea cuatro fases en las que se divide el desarrollo de software: Inicio, Elaboración, Construcción y Transición. Dentro de las mismas plantea un conjunto de flujos que deben realizarse de manera iterativa e incremental a lo largo de los flujos antes mencionados. Cada uno de estos flujos proponen un conjunto de actividades y roles para realizarlas que al ejecutarse garantizan un desarrollo de software ordenado y fluido aumentando así la probabilidad de un producto final con calidad y por consiguiente una mayor satisfacción del cliente.

Además de los básicos (Captura de requisitos, Análisis y diseño, Implementación y Pruebas) dedicados a la parte ingenieril del desarrollo RUP incluyó otros dedicados al soporte como son los flujos de Gestión de proyectos, Gestión de ambiente y Configuración y Gestión de Cambio.

RUP pudiera considerarse una de las metodologías más completas de las existentes. Describe un número significativo de actividades del ciclo de desarrollo de un software, pero aplicarlo en su totalidad puede ser engorroso e improductivo, por lo que lo más recomendable constituye adaptarlo a las características propias del proyecto.

RUP propone una serie de artefactos para satisfacer las actividades que ellos proponen como parte de sus flujos de trabajo. Ese aspecto será tratado con más profundidad en el epígrafe 1.6.1.

1.4.2 Metodologías ágiles

Las metodologías ágiles presentan un modelo de desarrollo incremental, con pequeñas entregas con ciclos rápidos, cooperativos donde los desarrolladores y usuarios trabajan juntos en estrecha comunicación, y donde es capaz de adaptarse a los cambios, siendo un método muy simple y fácil de aprender. Tienen como características el ser adaptativas en vez de predictivas, centradas en la gente o en los equipos, iterativas, orientadas hacia prestaciones y hacia la entrega, de comunicación intensiva, y que requieren que el negocio se involucre en forma directa.

Entre estas metodologías se destacan XP y SCRUM. En la figura 3 [Ambler 2008] se muestra el ciclo de vida clásico de las metodologías ágiles.

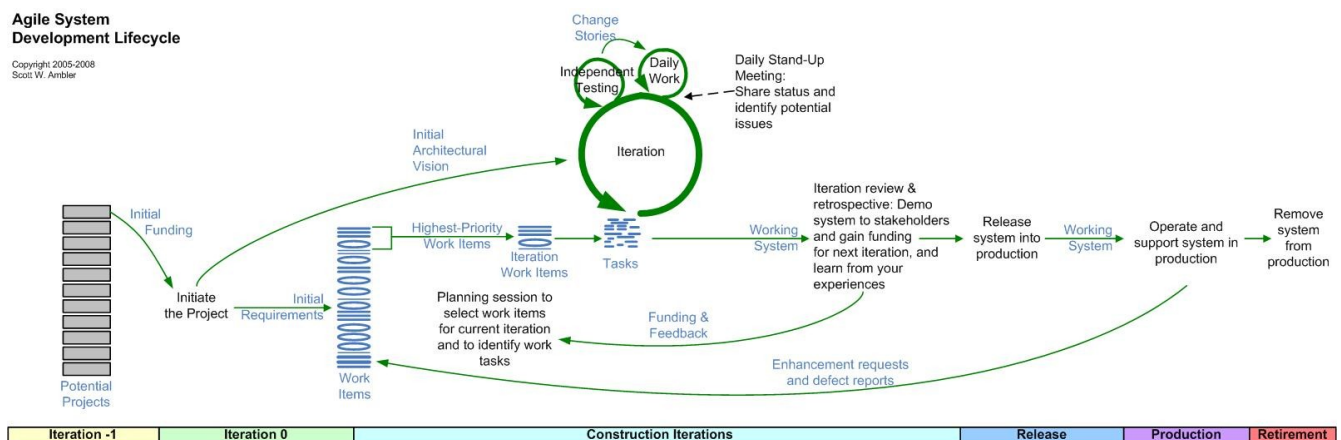


Figura 3. Ciclo de vida básico para metodologías ágiles

De manera general se aprecia que básicamente cumple con las mismas actividades de las metodologías clásicas aunque si difieren en el tiempo que le dedican a cada una de ellas. Entre los artefactos más comunes que se generan en los proyectos desarrollados con este tipo de metodología se encuentran: los documentos de planificación que incluyen los de estimación, documento de requisitos, diagramas de clases a diferentes niveles, prototipos, código fuente, casos de pruebas y algún otro que sea requerido.

Comparando los artefactos utilizados en el desarrollo con metodologías ágiles con los utilizados por metodologías de las llamadas convencionales se observa que las primeras utilizan un número más reducido de artefactos, dado sobre todo por la premisa de que para desarrollar con este tipo de metodologías se necesitan desarrolladores expertos y con equipos de trabajo pequeños, por lo tanto se reducen los canales de comunicación.

Las metodologías ágiles suelen ser muy efectivas cuando se trata del desarrollo de proyectos sin un gran número de funcionalidades, sean de ciclo corto, se desarrollen con equipos de trabajos pequeños y los desarrolladores del proyecto sean experimentados. Estas metodologías si bien, como su nombre lo indica, agiliza el trabajo, presenta carencias, algunas de ellas de impacto en una Universidad como la UCI. Una de las más significativas a nivel organizacional es que al documentarse muy poco sobre el proyecto, mantiene el conocimiento generado durante la construcción en las personas que lo generaron y frena la Gestión del Conocimiento en la organización. Otro aspecto significativo es la poca importancia que le brindan a las actividades de soporte como el Aseguramiento de la Calidad, la Gestión de Proyecto y al Gestión de la Configuración, aunque en esta última si manejan el versionado de los artefactos. Estas tres últimas actividades mencionadas se encuentran presentes en los modelos y estándares de calidad consultados [CMMI 2005] [IEEE 2007] [ISO 2005a] como fundamentales para el desarrollo de software.

1.5 Buenas prácticas para el desarrollo de software.

Las mejores prácticas para el desarrollo de software son aquellas que han demostrado que con su aplicación se reducen los tiempos de desarrollo, disminuyen los costos y aumenta la calidad [CONTRUX 2008]. Muchas prácticas existen para dar soporte a un mejor desarrollo de software y algunas de ellas han probado su efectividad a través de décadas y a pesar de

esto no todas las organizaciones hacen uso de ellas.

Muchas de estas prácticas pueden verse recogidas en estándares, normas y modelos internacionales como los presentados hasta el momento [CMMI 2005][IEEE 2007][ISO 2005a]. En ellos apreciamos que de una forma u otra coinciden en varios aspectos del ciclo de vida y que se presentan independientemente de la metodología a utilizar. Entre ellos vemos: la definición de una metodología apropiada, gestionar los requisitos, realizar la gestión de la configuración a los elementos que componen el sistema, realizar una planificación de las actividades, realizar actividades para el aseguramiento de la calidad, validar los resultados con el cliente y realizar revisiones periódicas a los artefactos.

También International Business Machine (IBM), empresa líder en el mercado, fundada en 1981 y dueña del Rational Unified Process, ha publicado una lista de lo que considera buenas prácticas para el desarrollo de software apoyada a sus años de experiencia en el ramo.

IBM ha publicado un conjunto de buenas prácticas para el desarrollo de aplicaciones software [IBM 2008] agrupadas en 16 secciones:

1. Procesos de desarrollo
2. Requerimientos
3. Arquitectura
4. Diseño
5. Diseño de aplicaciones WebSphere (para aplicaciones muy específicas)
6. Implementación
7. Pruebas por pares
8. Pruebas
9. Pruebas de rendimiento
10. Gestión de configuración

11. Calidad y gestión de defectos
12. Despliegue
13. Operación de sistemas y soporte
14. Migración de datos
15. Gestión de proyectos
16. Mediciones

Dichas prácticas constituyen aspectos vitales para el desarrollo de software como la definición de una metodología apropiada para el tipo de aplicación de desarrollar, aceptar todos los artefactos con el cliente, definir la arquitectura del sistema, realizar pruebas al código, planificar las pruebas al sistema, gestionar la configuración de los artefactos del sistema, establecer criterios de calidad y realizar mediciones.

Actualmente IBM es la empresa propietario de RUP, por lo que para la documentación y el ciclo de vida para los proyectos software se basa en los artefactos propuestos por dicho proceso.

1.6 Documentación de proyectos software

Cuando se menciona la palabra software la primera idea es cientos de líneas de código con el fin de crear un ejecutable y no la documentación que debe llevar consigo la construcción del software.

Al no conceder toda la importancia a documentar todas las actividades del desarrollo de aplicaciones, suele quedar en un segundo plano y a realizarse por obligación.

En la revisión de las normas y estándares antes mencionados [IEEE 2007] [ISO 2005a] se aprecia la importancia que le asignan a la documentación de software. Si bien no dicen que se debe documentar si mencionan un conjunto de criterios básicos a seguir para realizar dicha actividad. Todo esto va a estar muy ligado a las evidencias que se necesitan para la ejecución de los procesos del proyecto así como del ciclo de vida elegido, de ahí que para la construcción de la propuesta se analizaran diferentes ciclos de vida y proyectos con

diferentes organizaciones.

1.6.1 Artefactos del Proceso Unificado de Rational (RUP)

Como se ha mencionado con anterioridad RUP propone un proceso para el desarrollo de software adaptable para diversos tipos de proyectos. Dentro del mismo plantea un conjunto de actividades que debe desarrollar un proyecto a lo largo del ciclo de vida de desarrollo de un software [IBM 2007]. Para cada una de estas actividades el proceso propone un conjunto de artefactos que deben ser construidos durante su ejecución y facilita una serie de plantillas que dan respuesta a estos artefactos.

En total propone 35 plantillas sin una organización predeterminada. Esta se basa en cómo se defina el proceso de desarrollo de software a ejecutar. Las plantillas propuestas se encuentran muy bien documentadas y como característica esencial tienen su apego a la propia metodología y su cobertura a todas las actividades.

Según los resultados del diagnóstico realizado en el 2008 a los proyectos de la Universidad el 63% de los proyectos revisados usan RUP para definir su ciclo de desarrollo [DCS 2008b] lo que presupone una alta compatibilidad con las plantillas propuesta por el proceso.

Un elemento que apoya el uso de la documentación propuesta por RUP, es la utilización de su suite ya que la misma es capaz de generar gran parte de la documentación de un proyecto de forma automática. Sin embargo en la actualidad solo el 10% de los productos de la Universidad utilizan herramientas de la Suite de Rational en sus proyecto [DCS 2008b], número con tendencia a disminuir debido a la migración hacia sistemas operativos linux y el uso de herramientas libres en la producción, por lo que es un factor que carece de peso a la hora de seleccionar la documentación a utilizar.

1.6.2 Artefactos NASA

La Administración Nacional de Aeronáutica y del Espacio (NASA) fue fundada en el año 1958. Hoy en día es una de las empresas que más ha aportado al desarrollo de la humanidad, no solo en el plano aeroespacial y tecnológico sino también en la rama del software.

Actualmente cuenta con un parque tecnológico formado por más de 20 empresas [NASA

2008] de software que se dedican a la investigación y desarrollo de aplicaciones que van desde la inteligencia artificial, a software médicos y aplicativos para la aviación.

Además de lo antes mencionado cuenta con el Goddards Space Flight Center Software Process Improvement [NASA 2008] con el objetivo de mejorar la calidad, estabilidad y seguridad de los productos desarrollados por la NASA.

Entre las actividades que realiza el GSFC se encuentran la construcción de procedimientos, lineamientos y plantillas para las empresas de desarrollo de software pertenecientes a la NASA.[NASA 2008]

Todos estos artefactos se encuentran organizados según cuatro criterios: Gestión de proyectos, Gestión de desarrollo, Soporte organizacional y Adquisición.

El Centro propone un conjunto de plantillas que cubren los procesos propuestos por el mismo aunque muchas de estas plantillas no se encuentran disponibles y las que lo están no muestran todo su contenido.

1.6.3 Ready Set

Ready Set es un proyecto de código abierto perteneciente a Tigris.org creado con el objetivo de producir y mantener un conjunto de documentos reutilizables para la Ingeniería de Software.[Readysset 2010]

Los documentos propuestos se encuentran agrupados en siete categorías:

- Planificación de proyecto.
- Requerimientos y especificación.
- Arquitectura y diseño.
- Implementación y pruebas.
- Despliegue e instalación.

- Operaciones y soporte.
- Continuidad o final.

En su conjunto cuenta con 33 documentos que brindan solución a la mayoría de las áreas técnicas del desarrollo de software y en ellos aparecen propuestas de contenidos a documentar. No se encuentra enmarcado a ninguna metodología en específico y se enfoca en actividades generales.

Como desventajas de la utilización de esta propuesta pudiera citarse la falta de artefactos claves para el desarrollo como la plantilla para el Plan de Gestión de Configuración y Modelado de Negocio. Para cada uno de los artefactos que se proponen se presenta una información general de lo que debe contener pero no se encuentra estructurado en un formato específico, además de esto la propuesta se encuentra totalmente en formato HTML por lo que el primer paso sería llevarlo a un formato editable.

La tabla 4 muestra una comparación entre las diferentes propuestas para la documentación de proyectos.

Criterio	RUP	Ready Set	NASA	IEEE
Organización de la documentación	No propone una organización para la documentación. Se organiza de acuerdo a la implementación del ciclo de vida.	Proponen una estructura para la documentación.	No propone una organización para la documentación.	No propone una organización para la documentación.
Costo	Se cobra como parte de la Suite de Rational.	No tiene costos asociados	Algunas están disponibles para su descarga total, otras para su descarga parcial y otras	Tienen un costo elevado de adquisición.

			son privadas para el uso de la organización.	
Dependencia de la metodología	Es dependiente de la metodología que se decida seguir para el desarrollo en este caso RUP.	No son dependientes de la metodología.	Adecuadas a los ciclos de vida de los proyectos de la NASA.	No son dependientes de la metodología.
Complejidad	Al estar estrechamente vincula a una metodología en particular no propone artefactos necesarios para el desarrollo debido a que no se encuentran en su propuesta.	Incompleto.	Incompleto.	Contiene un número significativo de plantillas.
Complejidad	La mayoría de las plantillas exigen un volumen elevado de información y en ocasiones la misma se encuentra repetida.	Bastantes sencillas	Bastantes sencillas	Son complejas, pero sin llegar a ser inmanejables.

Tabla 4. Comparación de propuesta para documentación del software.

1.7 Conclusiones

Después de analizadas un conjunto de normas y estándares para el desarrollo de software, diferentes ciclos de vida y estándares existentes para la documentación de un proyecto se concluye que las organizaciones deben seguir aquellos estándares que más se adecuen a

sus objetivos y estructura. Para el caso particular de la UCI se escoge el modelo CMMI [CMMI 2005] atendiendo a un conjunto de factores como la estructuración del modelo, su fácil comprensión, precio y el hecho de contar con otra serie de modelos y constelaciones que pueden llegar a cubrir en un futuro las necesidades de la Universidad.

En cuanto al ciclo de vida lo importante es identificar qué tipos de proyectos se van a realizar y en función de esto seleccionar cuál proceso de desarrollo se utilizará. La tendencia se ha convertido en, a partir de los ya existentes, tanto ágiles como clásicos, tomar lo que se considere apropiado y adaptarlo según las necesidades, por lo que pocas veces se aprecia la utilización pura de alguna metodología. Complementando esto, entidades con un largo historial en el desarrollo de aplicaciones informáticas proponen un conjunto de buenas prácticas a aplicar independientemente del ciclo de vida que se decida utilizar.

La documentación en las aplicaciones informáticas juega un papel importante en la comprensión del producto a desarrollar, el diseño de la solución, la generación y gestión del conocimiento y en la comunicación con el cliente. A pesar de que en el mundo existen algunas propuestas para la misma ninguna satisface enteramente las necesidades de la producción en la Universidad.

2 Propuesta de solución

2.1 Introducción

En el presente capítulo se realiza la propuesta del Expediente de proyecto para la estandarización de los proyectos de software en la UCI y de un conjunto de buenas prácticas para el ciclo de vida de desarrollo. Se presenta el basamento que llevó a la concepción de ambos artefactos, su proceso de obtención y su evolución.

Atendiendo a lo planteado en los epígrafes 1.2 y 1.2.1 del capítulo anterior, la presente propuesta tiene influencia en ambas áreas de la Gestión de la calidad. La solución se localiza en el área que abarca el aseguramiento de la calidad ya que se propone un conjunto actividades sistemáticas y planeadas para proveer la adecuada certeza que un producto cuenta con cierto nivel de calidad basado en la aplicación de normas y estándares impactando así en el control de la calidad, permitiendo la confección de herramientas y métodos que permitan evaluar los productos desarrollados, todo esto en su conjunto posibilitará un aumento en la percepción de la calidad del producto una vez entregado al cliente.

2.2 Estrategia

Muchos autores han dado su definición de estrategia asociadas a varios contextos, uno de los más comunes es el que se refiere a la estrategia militar. Este concepto a medida que ha avanzado la humanidad se ha ido adaptando al plano empresarial en donde nos encontramos definiciones como la expuesta por Burgelman [Burgelman 2002] donde plantea que estrategia “es la teoría que la alta dirección tiene sobre la base para sus éxitos pasados y futuros. ”. Según el portal web de BusinessDictionary [Business, 2010] estrategia es “una alternativa tomada para que ocurra un futuro deseado, como el cumplimiento de un objetivo o solución de un problema”.

Una estrategia está compuesta por diferentes actividades a realizar:

1. Definir fines, objetivos y priorizar en el tiempo

2. Transformar objetivos en metas medibles
3. Diseñar pautas o guías de acción que orienten a todos los involucrados
4. Identificar medios y recursos
5. Diseñar el conjunto de acciones necesarias para lograr esos fines

2.3 Estandarización de la documentación. Expediente de proyecto.

La documentación asociada a los proyectos de software y sistemas debe cumplir con algunos requisitos como son:

- Servir como medio de comunicación entre los miembros del equipo
- Servir de repositorio de información que pueda ser utilizado por los ingenieros de sistemas
- Proveer información para el control de los planes, cronogramas e hitos en el proceso de desarrollo de software
- Definir quién hace y cómo hace las actividades específicas del desarrollo.

Satisfacer estos requerimientos requiere de un grupo de documentos que no son solamente las documentaciones técnicas, asociadas al producto.

No es posible organizar la documentación de los proyectos sin definir un esquema genérico para la organización que norme cómo usar y evaluar la documentación según el tipo de proyectos. Por esta razón y para lograr estandarizar las documentaciones de los proyectos en la UCI se define en esta tesis un expediente de proyectos.

Para este trabajo de investigación un **Expediente de Proyecto** es la herramienta que agrupa y organiza todos los artefactos que se generan durante el desarrollo de software [Delgado 2009].

Para lograr esta organización en los proyectos de desarrollo de software en la UCI se decidió definir una estructura para el Expediente de proyectos y un grupo de plantillas para este expediente. Para esto fueron analizados varios modelos y estándares que norman la

documentación de software como el Modelo de Madurez y Capacidades (CMMI) [CMMI 2005] en su versión 1.2, las normas ISO y los estándares de la IEEE [IEEE 1990].

Otro aspecto a tener en cuenta es la compatibilidad que debía tener dicho expediente con el modelo CMMI v1.2, modelo seleccionado para los procesos de desarrollo de software en la Universidad.

La confección del expediente se dividió en varias etapas [Delgado 2009]:

1. Definir estructura
2. Identificar las necesidades propias de documentación de los proyectos de la UCI
3. Revisar plantillas propuestas por normas y estándares y adaptarlas al expediente definido para la UCI
4. Adicionar las plantillas que por las características de la UCI no se encuentran presentes en los modelos seleccionados

2.3.1 El expediente de proyectos, su estructura general.

En el capítulo 1 se mencionaron las 4 clasificaciones utilizadas en CMMI para la organización de sus procesos. Estas mismas clasificaciones fueron las utilizadas para agrupar las plantillas en el Expediente de Proyecto (EP) agregando una cuarta categoría para la documentación legal que, dada su naturaleza y lo delicado de la misma no se consideró conveniente mezclar con documentación técnica.

El contenido de esta categoría legal no han sido definidos por el autor de este trabajo sino por especialistas en esta área, por esta razón no se incluirá en el presente trabajo.

La definición e institucionalización del expediente se realizó siguiendo el modelo escalonado definido CMMI por cada una de sus niveles de madurez iniciando por el nivel 2 incluyendo algunas exigencias del nivel 3.

La estructura completa del expediente se muestra en la figura 4.

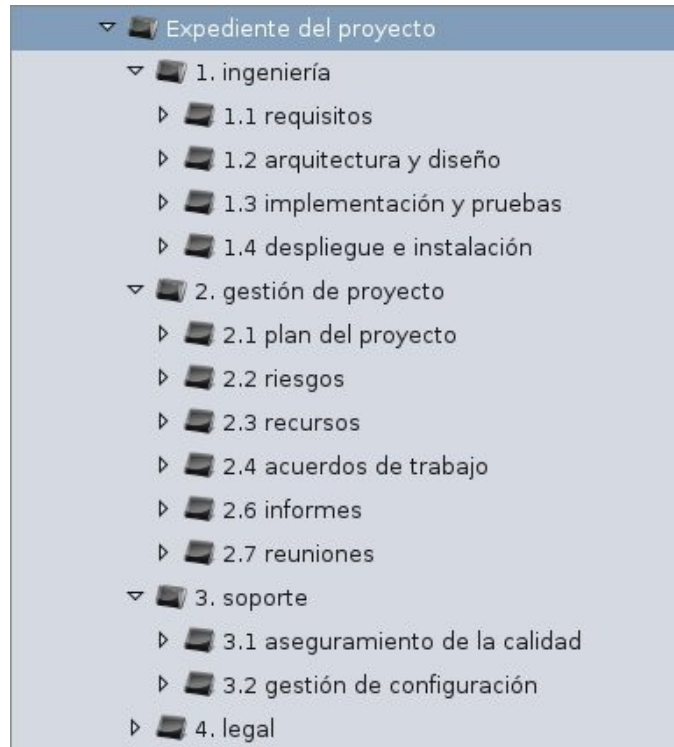


Figura 4. Estructura del Expediente de Proyecto.

2.3.1.1 Ingeniería

En la categoría de ingeniería se encuentran todos aquellos documentos relacionados con los procesos de la Ingeniería de Software. Su construcción no está basada en ninguna metodología en específico, aunque si se tienen en cuenta actividades básicas para la construcción de software.

En la definición de los artefactos para esta categoría se tuvo en cuenta el criterio de que estos debían ser lo más genérico posibles para que pudieran ser utilizados por proyectos con diferentes metodologías y ciclos de vida. Este carácter genérico propicia que haya ciclos de vida de algunos tipos de proyectos que no se satisfacen con la propuesta como es el caso de las aplicaciones multimedia y proyectos que se desarrollen bajo un enfoque ágil utilizando Extreme Programming o similares, sin embargo es un aspecto que solo afecta a la categoría de ingeniería.

Una característica presente en todos los estándares y normas revisados es su no adherencia a una metodología de software específica, por lo que suelen quedar poco específicas las secciones que abordan las áreas de ingeniería.

Para esto se tomó como referencia el Proceso Unificado de Rational (RUP) [RUP 2004] dada su adaptabilidad a casi cualquier proceso de desarrollo que se quiera utilizar y adaptable a una gran variedad de tipos de proyectos. Como se planteó en el epígrafe 1.3 se seleccionó la estructura para la organización de procesos propuesta por CMMI para estructurar el Expediente de Proyecto. Las AP de CMMI contempladas en la categoría de ingeniería se ubican mayormente en el nivel 3 del modelo y responden a actividades genéricas que puedan ser adaptadas al ciclo de vida de los proyectos de las organizaciones. En el caso de RUP si se basa en el ciclo de vida propuesto por el modelo por lo que se realizó una integración entre las estructuras propuestas por RUP y CMMI.

Otra área que se vio necesario reforzar fue la referente a la arquitectura, en dos aspectos: la arquitectura del software y la arquitectura de información.

Finalmente quedó estructurada de la siguiente forma el acápite de ingeniería:

1 Ingeniería

1.1 Requisitos

Se agrupan los estándares necesarios para el levantamiento, descripción y gestión de los requisitos atendiendo a lo planteado por varias metodologías.

1.2 Arquitectura y diseño

Se agrupan los estándares para el modelaje de la aplicación como son los documentos de arquitectura, diseño del modelo de datos, arquitectura de información, etc.

1.3 Implementación y prueba

Incluye los estándares necesarios para la implementación, los derivados de la misma (código fuente) y los necesarios para probarlos.

1.4 Despliegue e instalación

Incluye los estándares necesarios la instalación y el despliegue de las soluciones como manual de instalación, modelo de despliegue.

2.3.1.2 Gestión de proyecto

Este grupo recoge los estándares necesarios para la administración efectiva de un proyecto software. Esto se traduce en los planes para la ejecución de cada uno de los procesos presentes en el desarrollo de software así como aquellos aspectos generales del proyecto (cronogramas, recursos, presupuestos...).

La categoría de gestión de proyectos quedó de la siguiente forma:

2 Gestión de proyectos

2.1 Plan de proyecto

Incluye los estándares para llevar a cabo la planificación del proyecto incluyendo el presupuesto.

2.2 Riesgos

Incluye los estándares para la identificación y mitigación de los riesgos del proyecto.

2.3 Recursos

Se definen los estándares referentes a los recursos del proyecto, tanto humanos como materiales. Se incluyen los documentos de roles y responsabilidades y plan de capacitación.

2.4 Acuerdos de trabajo

Se incluyen los acuerdos tomados entre las partes involucradas en el

proyecto como pueden ser el documento visión (si es necesario) y el proyecto técnico.

2.5 Informes

Se incluyen los estándares para la ejecución de informes, entre ellos los que tienen que ver con diagnósticos a las organizaciones.

2.6 Reuniones

Espacio destinado para las minutas de las reuniones efectuadas en el proyecto.

La UCI, como prospecto de la nueva universidad cubana tiene un modelo compuesto por tres procesos fundamentales: Producción – Formación – Investigación, y este modelo ha sido uno de los retos de este expediente, lograr sincronizar estos tres procesos. En esto tiene gran impacto el área de Entrenamiento Organizacional (OT) de CMMI. Esta es un área de proceso de Nivel 3 de CMMI, pero a pesar de que este expediente está basado en el Nivel 2 de dicho modelo, se tuvieron en cuenta algunos elementos dentro del Plan de Capacitación. Teniendo en cuenta la importancia de la integración de estos tres procesos para la UCI, esta área de proceso se verá reforzada en la próxima versión.

El resto de las áreas de procesos eran perfectamente compatibles por lo que se tomó la decisión que dejarlas tal y como estaban propuestas dentro del expediente.

2.3.1.3 Soporte

En este grupo se encuentran los documentos que garantizan el soporte al software en construcción y se encuentra formado por dos subgrupos: Gestión de Configuración de Software y Aseguramiento de la Calidad.

La categoría de soporte se definió de la siguiente manera:

3 Soporte

3.1 Aseguramiento de la calidad

Se incluyen los estándares para asegurar la calidad de los proyectos como plan de mediciones, listas de chequeo, el Plan de aseguramiento de la calidad, entre otros.

3.2 Gestión de configuración

Se incluyen los estándares relacionados con la gestión de configuración del proyecto con son lo el plan y las solicitudes de cambio.

2.3.2 Necesidades organizacionales

Para identificar las necesidades de la organización se tuvieron en cuenta varios criterios. El primero de ellos fue recogido de la primera versión del expediente en reuniones, encuentros y capacitaciones efectuadas. Otro elemento fueron las opiniones recogidas a los líderes de proyectos de más experiencia en la universidad mediante entrevistas realizadas.

Entre las principales necesidades identificadas se encontraron la de incluir estándares relacionados con la arquitectura, revisar la duplicación de información, reducción del número de artefactos, inclusión de algunos artefactos para otras metodologías de desarrollo e incluir temas relacionados con el presupuesto.

2.3.3 Integración con estándares

Para cada área tenida en cuenta en el Expediente se definieron plantillas que permitirán estandarizar la documentación de los proyectos de la UCI. La selección de las plantillas se realizó a partir de la revisión de las propuestas por la IEEE [IEEE 1990] [IEEE 1998], la NASA en sus Sistema de Gestión de la Calidad, las normas ISO relacionadas con la producción de software, DoD y RUP.

Partiendo de este análisis y de la experiencia personal de líderes de proyectos y desarrolladores el expediente quedó conformado por las plantillas que se muestra en la tabla 5.

Ingeniería	Gestión de proyecto	Soporte	Legales
Diagrama de Proceso - Nombre del Proceso v1.0	Plan Desarrollo de Software v1.0	Glosario de términos	Acta de Aceptación
Especificación de Requisitos v1.0	Presupuesto v1.0	Listas de chequeo	Acta de Entrega
IDEF0 Diagram Shapes	Plantilla DCS - Lista de riesgos v1.0	Plantilla DCS - No Conformidades (ampliada) v1.0	Plantilla ALBET - Acta de inicio de proyecto
Modelo de Casos de uso del sistema v1.0	Plan Mitigación de Riesgos v1.0	No Conformidades (reducida) v1.0	Acta de Terminación de Proyecto
Plan de gestión de requisitos v1.0	Ambiente de desarrollo v1.0	Plan aseguramiento de la calidad v1.0	Carta
Especificación de Requisitos v1.0	Plan de capacitación	Plantilla DCS - Plan de mediciones v1.0	Indefiniciones
Modelo del Negocio v1.0	Roles y responsabilidad v1.0	Respuestas a No Conformidades v1.0	Informe Técnico
Modelo del Dominio v1.0	Documento Visión v1.0	Solicitud de cambio (reducida) v1.0	Minuta de reuniones
Arquitectura de Información v1.0	Diagnóstico v1.0	Pedido de cambio v1.0	Proyectos Técnicos
Documento de Arquitectura de Software v1.0	Minuta de reunión v1.0	Plan Gestión de Configuración v1.0	
Informe del Levantamiento de Información para la Arquitectura de			

Información v1.0			
Modelo de Diseño v1.0			
Diseño casos de prueba v1.0			
Plan de pruebas v1.0			
Modelo de Despliegue v1.0			

Tabla 5. Plantillas del Expediente de Proyecto.

Un paso importante en la confección del EP lo fue la adaptación de las plantillas seleccionadas al entorno de la Universidad. Para esto se utilizó la terminología manejada por los proyectos, se describieron los epígrafes y algunos nombres fueron adaptados a los artefactos manejados por los proyectos.

2.3.4 Adiciones

Como se mencionó en epígrafes anteriores, en la Universidad se construyen artefactos que no se encuentran formalmente en las normas y estándares internacionales consultados, por lo que fue necesario realizar una revisión de las variantes de los mismos utilizadas por diferentes proyectos para así llegar a una plantilla que fuera consenso y fácil de asimilar por todos los proyectos.

Ejemplos de estos documentos son:

- Plantilla DCS - Evaluación de Áreas de la Organización
- Plantilla DCS - Arquitectura de Información
- Plantilla DCS - Informe del Levantamiento de Información para la Arquitectura de Información v1.0
- Plantilla DCS – Presupuesto

2.4 Estandarización de la producción. Lineamientos de calidad.

Independientemente del ciclo de vida que se seleccione para la construcción de un determinado producto, existen actividades genéricas que deben ser ejecutadas para obtener un producto con la calidad deseada.

Estas actividades están basadas en la experiencia en el desarrollo de proyectos y muchas de ellas han sido recogidas en estándares, normas y modelos internacionales. Podría decirse que estas actividades constituyen un conjunto de *buenas prácticas*.

La UCI en apenas 4 años ha aumentado considerablemente el número de proyectos. Actualmente esta cifra se calcula en 160 proyectos [DCS 2008b]. Esto ha derivado a su vez en una mayor heterogeneidad de los mismos.

Atendiendo a la necesidad de tener una guía de actividades a realizar por los proyectos independientemente del tipo de proyecto se crean los Lineamientos de Calidad v1.0.

2.4.1 Antecedentes

Los Lineamientos de Calidad (LC) constituyen un conjunto de buenas prácticas encaminadas a estandarizar la producción de software en la Universidad. Su base fundamental se encuentra en las Prácticas Específicas (SP) del CMMI v1.2 para el nivel dos de madurez y en la experiencia en el desarrollo de software alcanzado por la Universidad en estos cinco años de construcción de software.

En el año 2005, atendiendo a las necesidades crecientes de la producción y a la necesidad de estandarizar las actividades del proceso de desarrollo se definieron los Lineamientos Mínimos de Calidad. Estos primeros proyectos tenían en común su ciclo de vida basado en RUP [DCS 2008a]. Dado este criterio los lineamientos tomaron las actividades propuestas por el proceso y que debían estar presentes en todos los proyectos independientemente de la configuración del ciclo de vida.

En el período 2006 – 2007 el número de proyectos tanto como su diversidad aumenta

considerablemente en la Universidad lo que hizo que aumentaran los ciclos de vida y las diferencias entre los mismo, por lo que aparece la necesidad de enfocar más los lineamientos hacia los procesos de desarrollo que a la utilización de una metodología específica, de ahí la selección de un modelo de procesos para el desarrollo de software.

Para la selección de norma o modelo que iba a sustentar los LC se realizó un estudio entre varios de ellos como CMMI [CMMI 2005], SPICE [SPICE 2010], IEEE [IEEE 2007], ISO [ISO 2005b] y MoProSoft [Moprosoft 2008] sin olvidar lo que planteaban algunas metodologías como RUP [IBM 2007] y XP [Ambler 2008].

Cada una de los estándares y normas analizados propones un conjunto de actividades que se deben realizar para cumplir con ellas. En el estudio realizado se identificaron cuáles eran los aspectos en común entre ellas para llegar a una única propuesta.

Otro aspecto que se tuvo en cuenta fueron los resultados arrojados por las revisiones y auditorías ejecutadas a los proyectos. Durante la ejecución de estas actividades se fueron identificando un conjunto de elementos que se encontraban presentes en los Lineamientos Mínimos de Calidad y que, después de un análisis de tendencias, se arribó a la conclusión que solamente eran aplicables a un tipo muy específico de proyectos y debía considerarse su derogación.

2.4.2 Lineamientos de calidad.

Los LC lo conforman un total de 32 actividades divididas en 4 secciones: General, Ingeniería, Gestión de Proyecto y Soporte. Esta estructura tributa, al ser similar a la del EP, a una mejor organización de los mismos y a una mejor comprensión por parte de los proyectos.

Cada actividad cuenta con una descripción, una o varias plantillas del Expediente de proyecto que deben ser utilizadas para el cumplimiento de la misma, cuál es el artefacto que queda como evidencia de la misma, así como le referencia a guías para su ejecución.

Las actividades incluidas en los lineamientos son los siguientes:

Generales

1. Planificar capacitación para el personal del proyecto.
2. Establecer registro de resultados de investigación del proyecto.
3. Definir roles y responsabilidades.
4. Definir equipos de proyecto.

Ingeniería

5. Gestionar requerimientos.
6. Desarrollar requisitos.
7. Definir arquitectura de software.
8. Definir arquitectura de información.
9. Definir modelo de diseño.
10. Definir estándares para el desarrollo del proyecto.
11. Planificar y ejecutar pruebas del proyecto.
12. Efectuar seguimiento de las No Conformidades.
13. Diseñar artefactos de prueba.

Gestión de proyecto

14. Definir proyecto técnico para el proyecto.
15. Definir visión del proyecto.
16. Estimar costo y esfuerzo del proyecto.
17. Desarrollar plan del proyecto.
18. Definir plan de resultados del proyecto.
19. Establecer horarios de trabajo de los miembros del proyecto.
20. Identificar riesgos del proyecto y establecer plan de mitigación de los mismos.
21. Gestionar recursos.
22. Registrar los acuerdos de trabajo y las minutas de las reuniones.

Soporte

23. Definir plan de resultados del proyecto.
24. Establecer horarios de trabajo de los miembros del proyecto.
25. Identificar riesgos del proyecto y establecer plan de mitigación de los mismos.
26. Gestión de recursos.

- 27.Registrar los acuerdos de trabajo y las minutas de las reuniones.
- 28.Plan de aseguramiento de la calidad.
- 29.Definir Plan de Gestión de la Configuración de Software.
- 30.Definir glosario de términos.
- 31.Definir configuración de la metodología a utilizar.
- 32.Utilizar herramienta CASE.

2.5 Estrategia de continuidad. Programa de mejora.

2.5.1 Integración del Expediente de proyecto al programa de mejora

En el año 2008, atendiendo a las necesidades de mejoras en el proceso productivo de la Universidad se inició un Proyecto para la mejora de procesos de desarrollo de software. Para esto se seleccionó el modelo CMMI. Dicho modelo para su versión 1.2 está formado por 22 áreas de procesos y en su representación escalonada se encuentra agrupado en 5 niveles. El nivel 2 llamado de Gestionado lo conforman 7 áreas de procesos básicas para la Gestión de proyectos:

- Gestión de requisitos
- Planificación de proyectos
- Monitoreo y control de proyectos
- Gestión de acuerdos con proveedores
- Medición y análisis
- Aseguramiento de la calidad de procesos y productos
- Gestión de configuración

Como parte de la ejecución del proyecto de mejora se irán obteniendo una serie de artefactos que van a servir como evidencia de la ejecución de los procesos.

La mejora de proceso no se puede ver aislada del trabajo de la Universidad. Uno de los principios que debe regir la misma es la integración con los procesos, artefactos y normativas

implantadas. Esto no implica que lo que ya se encuentre institucionalizado no deba ser cambiado, todo lo contrario; durante la ejecución del programa de mejora todo debe ser revisado para lograr una compatibilización y que fluya de manera armónica el proceso de desarrollo de software.

Como se ha descrito en el epígrafe anterior durante las diferentes etapas del programa de mejora se irán generando artefactos que pueden afectar de alguna forma los ya existentes, pueden aparecer nuevos artefactos necesarios para cumplir con las necesidades de los procesos y con las prácticas del modelo y pueden desaparecer algunos que ya no sean necesarios.

Cada uno de estos artefactos estarán contenidos en el Expediente de proyecto. Para garantizar una integración efectiva en cada equipo de trabajo estará presente una persona capacitada en el trabajo con el Expediente y los lineamientos.

El proceso de integración se describe en la figura 5

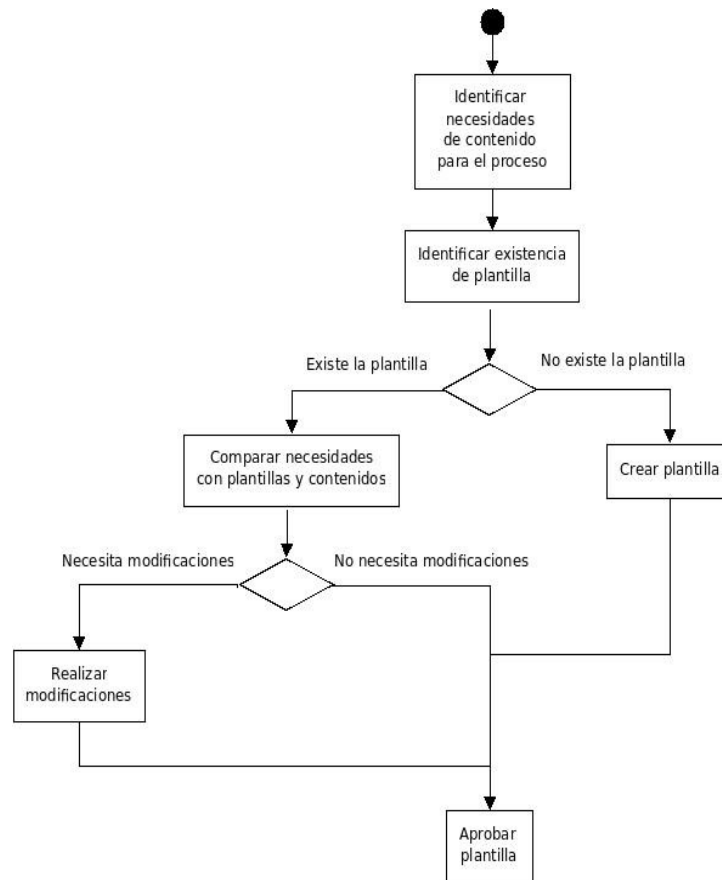


Figura 5. Proceso de integración del Expediente de Proyecto.

Identificar necesidades de contenidos para el proceso.

Identificar las necesidades de contenidos para el proceso que se esté describiendo, o sea, que información es necesaria quede registrada para el cumplimiento exitoso de las actividades del proceso, así como lo que servirá de evidencia para el chequeo del mismo. El responsable de la actividad es el TWG y como artefacto se obtiene una lista de necesidades.

Identificar existencia de plantilla.

Identificar la existencia de la información necesaria para el proceso dentro de las plantillas que existen en el Expediente de proyecto.

El responsable de estas actividades es el experto en el Expediente de proyecto.

Comparar necesidades con plantillas y contenido.

De identificarse plantillas que aborden las necesidades de los procesos, se revisará si satisfacen todas las necesidades indicadas.

El responsable de estas actividades es el experto en el Expediente de proyecto y como artefacto se obtiene una lista de necesidades no satisfechas.

Realizar modificaciones.

Si los documentos identificados en el Expediente de proyecto no satisfacen todas las necesidades del proceso, se realizarán los cambios necesarios a las plantillas identificadas.

El responsable de la actividad es el TWG y como artefacto se obtiene la plantilla modificada.

Crear plantilla.

De no existir una plantilla o plantillas que satisfagan las necesidades del proceso, se creará una o varias nuevas plantillas atendiendo a los criterios identificados con anterioridad. El responsable de esta actividad es el TWG y como artefacto se obtiene la nueva o nuevas plantillas.

Aprobar plantilla.

Una vez definida la platilla ya sea por existir en el Expediente de proyecto, por modificarse una existente o por crearla completamente, se procederá a su aprobación. El responsable de esta actividad es el TWG.

2.5.2 Integración lineamientos de calidad

Al igual que el EP, los lineamientos de calidad se verían afectados a medida que se lleve a cabo el programa de mejora. El modelo seleccionado obliga a cada uno de los procesos que se definan a que incluyan cuales serán políticas que seguirá para su institucionalización.

Algunas de estas políticas y los procesos en si irán suplantando algunos de los lineamientos existentes, por lo que se hace necesario crear un flujo de trabajo que permita mantener una

coherencia entre los procesos y los LC.

A diferencia del Expediente de proyecto para el caso de los lineamientos el impacto del cambio es menor debido a varios factores:

1. Las áreas de procesos del alcance del programa de mejora pertenecen a la categoría de Gestión de Proyecto por lo que solo se verían afectados los lineamientos pertenecientes a esa categoría.
2. Algunos de los lineamientos que se ven afectados pasan a formar parte de la política de determinado proceso.
3. Los lineamientos de calidad se encuentran basados en buenas prácticas para el desarrollo de software, entre ellas CMMI.

2.5.3 Evolución

Resulta evidente que para que algo perdure en el tiempo debe evolucionar adaptándose a los nuevos tiempos. Ni el Expediente ni los Lineamientos escapan de esta realidad. La producción en la Universidad debe ir ganando cada vez más en madurez lo que implica avanzar en la estandarización de los procesos y artefactos. De no evolucionar ambos artefactos en vez de apoyar a la producción se convertirían en un estorbo para la misma, por lo que los próximos pasos a dar son:

- Mantenimiento de los artefactos según se avance con el Programa de Mejora para el nivel 2 de CMMI.
- Introducción de artefactos y lineamientos que apoyen el avance a nivel 3 de CMMI de manera paulatina y según se vayan creando las condiciones.
- Mantener monitoreo sobre las metodologías más utilizadas en la Universidad para mantener la compatibilidad de los artefactos son las mismas.
- Mantener monitoreo sobre nuevas versiones de estándares y normas internacionales

para incorporar las mejoras necesarias.

2.6 Conclusiones

De este capítulo se concluye lo siguiente:

1. Se definió una nueva versión para el estándar de la documentación de los proyectos de desarrollo de software denominado Expediente de Proyecto v2 que definen una estructura y un conjunto de plantillas.
2. Se definieron un conjunto de lineamientos a cumplir por los proyectos de desarrollo de software denominado Lineamientos de Calidad.
3. Ambas soluciones no se encuentran ligadas a una metodología en específico.
4. Se definió cómo se va a ejecutar la integración de ambas soluciones con los resultados del programa de mejora de procesos.

3 Validación de la hipótesis y presentación de resultados

3.1 Introducción

En el presente capítulo se realizará un estudio de los resultados alcanzados con la utilización del expediente y los lineamientos en los proyectos de la UCI. También se describe el proceso utilizado para la implantación de las soluciones.

El estudio está basado en los resultados obtenidos mediante las revisiones realizadas por el Grupo de Auditorías y Revisiones y el Laboratorio Industrial de Pruebas, ambos pertenecientes a la Dirección de Calidad de Software de la UCI.

Atendiendo a los resultados obtenidos se presentarán los cambios realizados y que dieron origen a la nueva versión del Expediente de Proyecto.

3.2 Estrategia de implantación.

La definición tanto del Expediente como de los Lineamientos no es suficiente y se vería incompleta si no estuviera aparejada a una estrategia de implantación en los proyectos de desarrollo. La estrategia definida se concibió para que pudiera ser ejecutada en el menor tiempo y que fuera lo menos traumática para los proyectos, sobre todo para aquellos que ya se encontraban en un momento adelantado de su desarrollo.

3.2.1 Reglas para la implantación

Para la definición de la estrategia de implantación tanto del Expediente de proyecto como de los lineamientos se definieron un grupo de principios que se convirtieron en reglas del proceso de implantación:

1. La implantación del Expediente de proyecto no puede implicar un motivo de retraso para el desarrollo de los proyectos.
 - Los proyectos normalmente se encuentran ajustados a un cronograma de desarrollo pactado con el cliente. Para aquellos que son internos y que no se encuentran presionados por un cronograma este elemento puede carecer de sentido, pero para aquellos que están siguiendo un plan de entregas introducir

un elemento que retrase dicho cronograma puede significar en elemento de rechazo.

2. La implantación debe realizarse en el menor tiempo posible.

- El tiempo de implantación debe ser el menor posible para no alejar al personal del proyecto de las tareas que tienen planificadas por el proyecto.

3. Es necesaria la retroalimentación de los proyectos.

- Tanto para la confección de EP como de los LC como para su mantenimiento y perfeccionamiento e constituye un elemento fundamental la retroalimentación que se tenga de las personas que se encuentran utilizándolo con el fin de realizar las revisiones y corrección y de identificar los elementos a tener en cuenta para las próximas versiones.

4. Los líderes de proyectos son los máximos responsables por la implantación de los mismos en los proyectos.

- Los líderes de proyectos, como máximos responsable de la ejecución de los proyectos y de garantizar la calidad de los mismos son los responsables de la aplicación de los estándares de calidad y por ende de la implantación del EP y los LC.

5. Los asesores de calidad son los responsables de velar por la implantación de ambos artefactos.

- Los asesores de calidad son la representación del Centro de Calidad en las entidades productivas, por lo tanto son los responsables de velar por la correcta implantación de ambos artefactos.

Para una implantación eficiente se cuentan con una serie de antecedentes que facilitarán esta actividad:

- Existe una experiencia de trabajo con algunas de las plantillas por parte de los primeros proyectos de la Universidad, sobre todo los de exportación.
- Para su creación se ha tenido en cuenta el criterio de todas las facultades mediante el asesor de calidad.
- En el año 2005 se definieron los lineamientos mínimos de calidad teniendo en cuenta

las necesidades y los tipos de proyectos que se ejecutaban en ese momento.

3.2.2 Acciones

Para la implantación del Expediente de proyecto y los Lineamientos de Calidad se hace necesario llevar a cabo un conjunto de acciones a los diferentes niveles de calidad de la Universidad.

Nivel Dirección Calidad de Software

- Enviar propuesta a los Asesores de Calidad para realizar un levantamiento en sus facultades y ver si la propuesta es aplicable a sus proyectos.
- Revisar los señalamientos realizados por los proyectos y realizar los cambios pertinentes
- Presentar propuesta a Consejo de Calidad, Consejo de Dirección y Consejo de Producción para su aprobación.

Nivel de proyecto

- Presentar expediente al proyecto.
- Realizar levantamiento del estado que tiene su expediente actual con respecto a la nueva propuesta.
- Comenzar migración hacia el nuevo expediente.
- Guardar copia de los artefactos anteriores y discontinuar su uso una vez migrados a la nueva estructura.
- Revisar las prácticas previstas en los lineamientos y aquellas a las que no se le da cumplimiento tomar acciones correctivas para su ejecución atendiendo a la etapa del proyecto.

Nivel UCI

- Institucionalizar el Expediente de proyecto y los Lineamientos de Calidad.

3.2.3 Cronograma

Acción	Semana							Responsable
	1	2	3	4	5	6	7	
DCS								

Enviar propuesta a los Asesores de Calidad para realizar un levantamiento en sus facultades y ver si la propuesta es aplicable a sus proyectos.										DCS
Revisar los señalamientos realizados por los proyectos y realizar los cambios pertinentes										DCS
Presentar propuesta a Consejo de Calidad, Consejo de Dirección y Consejo de Producción para su aprobación.										DCS
Proyecto										
Presentar expediente al proyecto.										Líder de SW
Realizar levantamiento del estado que tiene su expediente actual con respecto a la nueva propuesta.										Grupo calidad de proyecto
Comenzar migración hacia el nuevo expediente.										Equipo de proyecto
Guardar copia de los artefactos anteriores y discontinuar su uso una vez migrados a la nueva estructura										Grupo calidad de proyecto
Revisar las prácticas previstas en los lineamientos y aquellas a las que no se le da cumplimiento tomar acciones correctivas para su ejecución atendiendo a la etapa del proyecto.										
Proyecto										
Institucionalizar el Expediente de proyecto y los Lineamientos de Calidad.										Rector

3.2.4 Evaluación

En función de llevar a cabo el seguimiento de la implantación y la calidad en la construcción de los artefactos se realizaron revisiones periódicas a los proyectos en la Universidad.

Cada una de estas revisiones fueron ejecutadas por equipos de cuatro personas mediante la

utilización de una guía exploratoria construida a partir del Expediente de proyecto y los lineamientos con el objetivo de:

- Evaluar la definición y cumplimiento del cronograma.
- Evaluar la definición y cumplimiento del plan de resultados.
- Contribuir a mejorar la aplicación de los elementos claves de la ingeniería y gestión de software en los proyectos productivos.
- Evaluar la aplicación de los procedimientos y disposiciones establecidas para la producción. (comienza con los LMC, EP).

En la Guía Exploratoria aparecen recogidos los criterios de evaluación que permitan verificar el cumplimiento de cada de las actividades prevista en los lineamientos mediante la evidencia plasmada en los artefactos del expediente de proyecto.

Una vez culminada la revisión, el proyecto fue informado de las deficiencias encontradas y las recomendaciones pertinentes por el equipo revisor.

3.3 Diagnóstico, revisiones y pruebas.

Entre las tareas a desarrollar por la Dirección de Calidad de Software de la UCI se encuentran las ejecuciones de revisiones a proyectos y la ejecución de pruebas y revisiones a artefactos.

Las revisiones a proyectos se ejecutan mediante la aplicación de una guía exploratoria desarrollada de conjunto con el grupo de normalización, donde se verifica, entre otros aspectos, la conformidad del proyecto con los lineamientos de calidad y el Expediente de proyecto. (ver Anexo 2)

La guía exploratoria incluye otros aspectos que no son de interés analizar en el marco de la presente tesis, pero que constituyen información necesaria para la organización e incluye una serie de criterios a evaluar en un proyecto que son medida de la utilización del expediente y los lineamientos.

Los criterios seleccionados son:

- Establecimiento del glosario de término de los documentos
- Gestión de los requisitos
- Completamiento del expediente de proyecto
- Definición y cumplimiento de los resultados del proyecto.
- Establecimiento y cumplimiento del cronograma del proyecto
- Definición del proyecto
- Establecimiento del alcance del proyecto
- Gestión de los riesgos del proyecto
- Asignación de los roles y las tareas
- Empleo de la metodología seleccionada y de la notación UML
- Establecimiento de la gestión de configuración
- Definición y empleo de herramientas
- Requisitos de hardware
- Definición general de arquitectura

Estos criterios se encuentran asociados tanto a la utilización de los artefactos del expediente de proyecto como a la aplicación de los lineamientos de calidad.

El períodos utilizado para el análisis es desde enero de 2008 a diciembre de 2008.

La planificación de las revisiones se muestra en la siguiente tabla (tabla 6):

Tipo	Cantidad de proyectos	% a revisar	Previsto a revisar	Planificadas	Ejecutadas	Aplazadas	Aborta	% de cumplimiento
Exportación	29	100	29	31	24	4	3	77.4
Nacional	69	45	31	33	26	3	4	78.8
UCI	39	45	18	18	10	6	2	55.6
Total	137		78	82	60	13	9	73.2

Tabla 6. Planificación de revisiones según tipo de proyecto.

3.3.1 Revisión 1

La primera revisión se realizó en febrero de 2008 y revisaron 9 proyectos. Los principales problemas detectados se muestra en la figura 6.

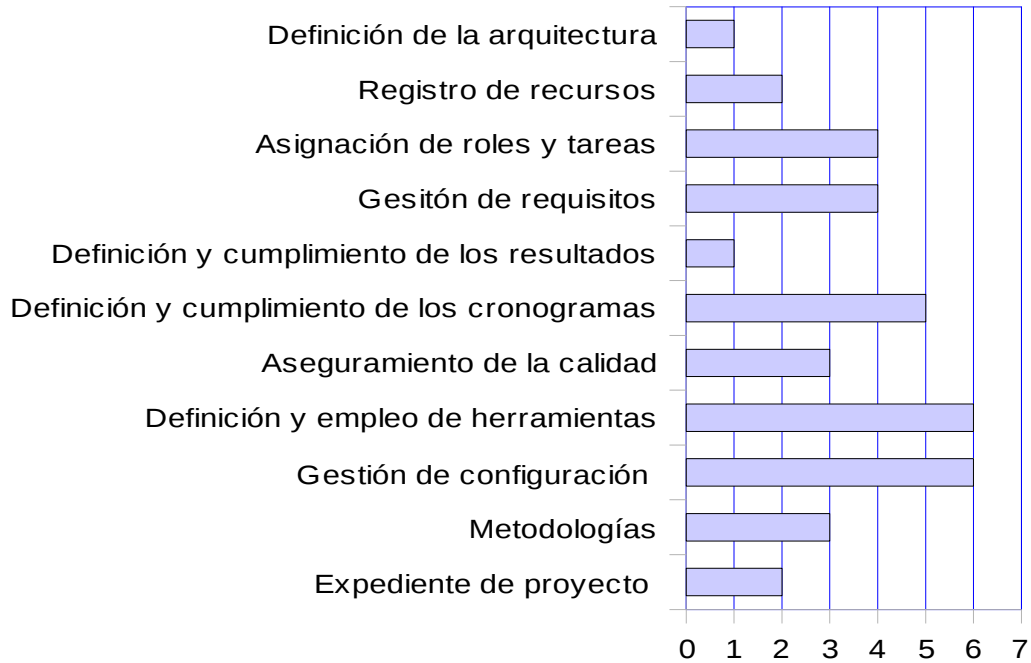


Figura 6. Principales problemas revisión de febrero.

3.3.2 Revisión 2

En la revisión realizada en el mes de marzo estuvieron involucrados 14 proyectos. Las principales deficiencias identificadas se muestran en la figura 7.

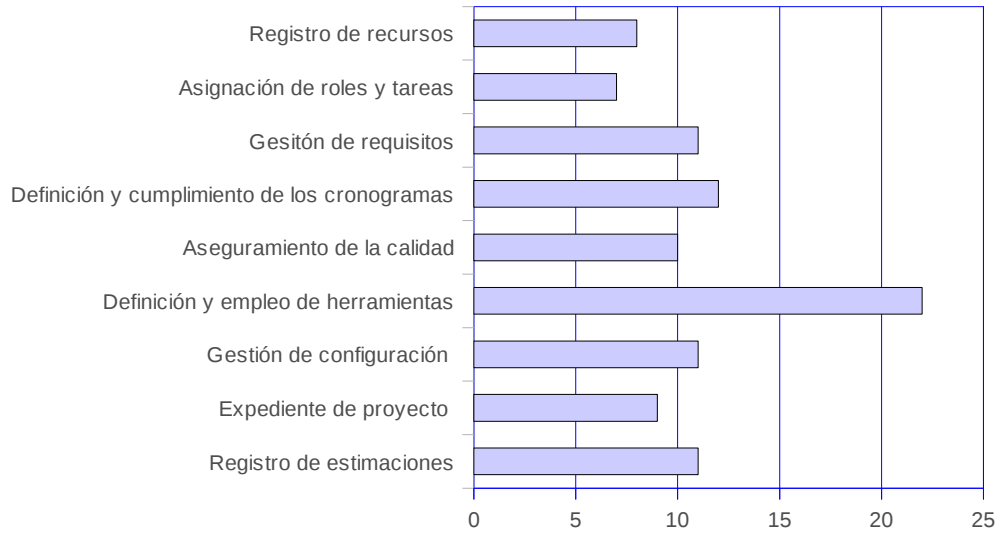


Figura 7. Principales problemas revisión de marzo.

3.3.3 Revisión 3

En la revisión realizada en el mes de mayo estuvieron involucrados 12 proyectos. Las principales deficiencias identificadas se muestran en la figura 8.

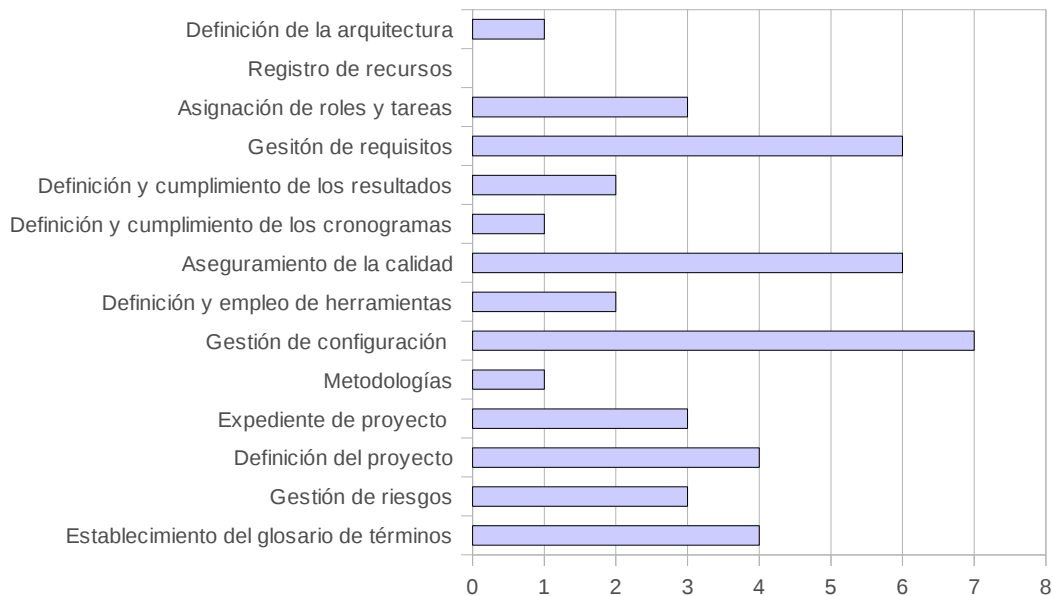


Figura 8. Principales problemas revisión mayo.

3.3.4 Revisión 4

En la revisión realizada en el mes de junio estuvieron involucrados 14 proyectos. Las principales deficiencias identificadas se muestran en la figura 9.

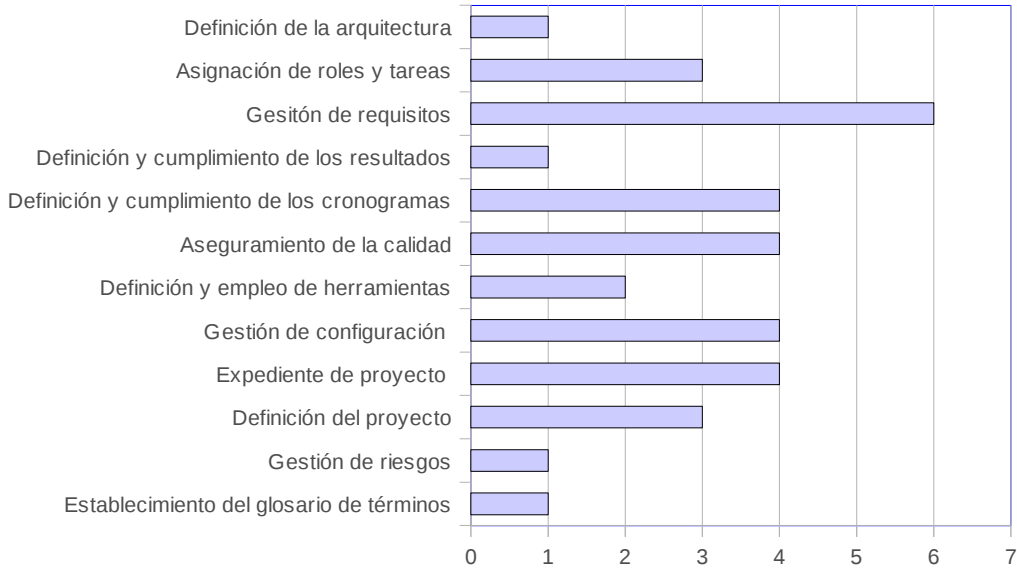


Figura 9. Principales problemas revisión junio.

3.3.5 Revisión 5.

Para la quinta revisión se tomó en cuenta el período octubre – diciembre de 2008 en el que se revisaron 11 proyectos. Las principales deficiencias detectadas se muestran en la figura 10.

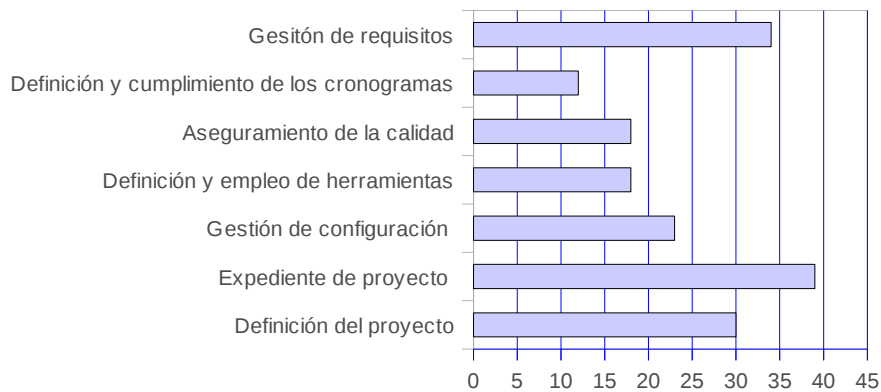


Figura 10. Principales problemas revisión octubre – noviembre – diciembre.

3.4 Pruebas de software

Las pruebas de software se ejecutan en la UCI desde el año 2005. Su objetivo actual es revisar técnicamente todos los artefactos que serán entregados a los clientes por parte de los proyectos lo que incluye tanto documentación como aplicación.

Las estadísticas demuestran que el coste de la corrección de los defectos detectados en etapas tempranas del desarrollo es menos que cuando estos son detectados en las últimas etapas del ciclo de vida (ver figura 11) [Humphrey 2001]. De ahí de que la Dirección de Calidad de Software se haya puesto como objetivo revisar también los artefactos que se generan en las etapas iniciales del desarrollo.

Al inicio esta tarea fue prácticamente imposible de ejecutar atendiendo que no existía ni un lineamiento que obligara a los proyectos a desarrollar ciertos artefactos básicos para el desarrollo y por otro lado no existía una documentación estándar esta situación impedía que este objetivo pudiera cumplirse como se espera dado que no podían generarse listas de chequeos estandarizadas por lo que carecían de objetividad las mismas.

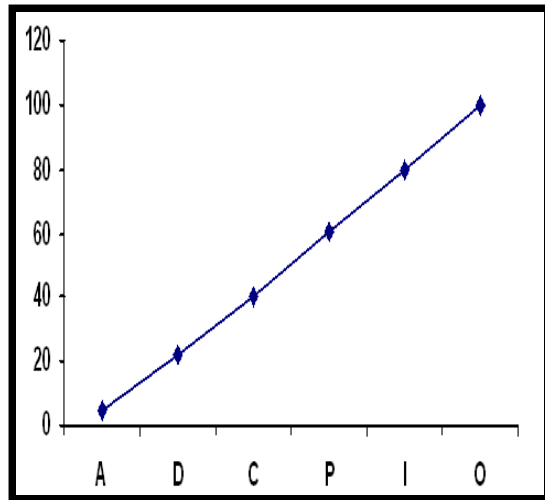


Figura 11. Costo de defectos por etapa de detección

En las pruebas ejecutas a un determinado proyecto de la Universidad en los artefactos correspondientes a etapas iniciales del desarrollo como el Documento de Especificación de Requisitos y el Documento de Arquitectura de Información se detectaron 811 No

Conformidades (NC) y la aplicación correspondiente a esos artefactos 432 NC. Atendiendo a lo anteriormente planteado de no haber sido identificadas y corregidas las NC de los artefactos iniciales, muchas de estas pudieron haber pasado a la aplicación y aumentar el tiempo promedio de corrección de las NC que, para el proyecto analizado tuvo una media de 5 días, hubiera aumentado aún más.

3.5 Mejoras

Atendiendo a los resultados obtenidos como parte de las revisiones y las opiniones de los usuarios se han realizado estudios sobre posibles cambios a realizar en el Expediente y los Lineamientos de Calidad de cara a sus próximas versiones. Estos cambios se han pensado teniendo en la cuenta los resultados que ha arrojado el programa de mejora. Para esto se ha propuesto una estrategia a corto plazo y otra a mediano plazo con el fin de satisfacer las necesidades de la producción en el menor tiempo posible.

El primer punto identificado se corresponde con que los artefactos propuestos, si bien cubría un amplio conjunto de actividades, aún no cubrían todas las necesidades de los proyectos debido a que habían ciclos de vida que no quedaban cubiertos en su totalidad principalmente en el área de ingeniería.

Para resolver este problema se definieron cuáles serían los ciclos de vida a completar para una primera etapa y en función de estos trabajar de conjunto con proyectos que implementaran estos ciclos para el completamiento.

En estos momentos se encuentra cubierto en su totalidad la documentación para los proyectos con ciclos de vida ágiles. Se trabaja en los artefactos para los proyectos multimedias.

En una segunda etapa, cuando se realice la implantación de los procesos que se obtengan como resultado de la mejora de procesos, se habrán abarcado todos los ciclos de vida que se sigan para la producción tratando de alinearlos, se podrá implantar un único expediente de proyecto con guías de adaptación para cada uno de los ciclos de vida identificados.

Otro aspecto identificado fue la necesidad de contar con una ayuda más específica para la construcción de la documentación, esto debido principalmente a la inexperiencia de las personas vinculadas a los proyectos que un número significativo son estudiantes.

En este sentido se revisaron cada una de las guías presentes dentro de cada uno de los documentos y se hicieron más detalladas y les fueron incorporados ejemplos en muchos de los casos. Otra acción realizada fue el completamiento de los casos de estudios para cada una de las plantillas del expediente.

Se detectaron un conjunto de artefactos que no satisfacían del todo las necesidades de los proyectos, en algunos casos por poseer información redundante o innecesaria y en otros por carecer de algún aspecto. Los artefactos detectados fueron:

- Evaluación de áreas de la organización
- Arquitectura de Software
- Diseño casos de prueba
- No Conformidades
- Listas de chequeo
- Plan aseguramiento de la calidad
- Plan de Gestión de Configuración
- Plan Desarrollo de Software
- Plan de Gestión de riesgos

En cuanto a la organización de los documentos, atendiendo a recomendaciones de ALBET, la empresa comercializadora de la UCI, se tomó la decisión de eliminar la carpeta para la documentación legal del expediente.

Algunos de los documentos pertenecientes a esta categoría han sido identificados en los procesos del programa de mejora y se integrarán a la organización existente.

En cuanto al formato de los documentos se pasó a utilizar el formato propuesto por la Infraestructura Productiva de la Universidad que se corresponde con el mismo a utilizar en los documentos de la mejora de procesos. También se realizaron modificaciones en los nombres de los documentos.

Siendo consecuentes con objetivos de la producción en la Universidad de que la misma se vea integrada con la investigación se agregó un espacio para el registro de las investigaciones asociadas al proyecto.

Siguiendo las políticas de migración hacia software libre todas las plantillas fueron llevadas a Open Document Format.

El Expediente de Proyecto en su versión 2.0, atendiendo a las modificaciones realizadas se muestra en la tabla 7 y la figura 12.





















-
- ▼  EP v2.0
 - ▼  1. ingeniería
 - ▶  1.1 requisitos
 - ▶  1.2 arquitectura y diseño
 - ▼  1.3 implementación y prueba
 - ▶  Código fuente
 - ▶  Manual de usuario
 - ▶  1.4 despliegue e instalación
 - ▼  2. gestión de proyecto
 - ▶  2.1 plan de proyecto
 - ▶  2.2 riesgos
 - ▶  2.3 recursos
 - ▶  2.4 acuerdos de trabajo
 - ▶  2.5 información del cliente
 - ▶  2.6 informes
 - ▶  2.7 reuniones
 - ▶  2.8 Investigaciones
 - ▼  3. soporte
 - ▶  3.1 aseguramiento de la calidad
 - ▶  3.2 gestión de configuración

Figura 12. Estructura Expediente de Proyecto v2.0

Ingeniería	Gestión de proyecto	Soporte
Especificación de requisitos de software v2.0	Diario de Actividades v1.0	Glosario de Términos v2.0
Evaluación de áreas de la organización v2	Pila de Sprint (Sprint Backlog) v1.0	Listas de chequeo v2.0
Historias de usuario v1.0	Plan Desarrollo de Software v2.0	Plan aseguramiento de la calidad v2.0
Modelo de dominio v2.0	Presupuesto v1.0	Plan de mediciones v1.0
Modelo del negocio v2.0	Registro Tarea de ingeniería v 1.0	Pedido de Cambio v2.0
Modelo de negocio con BPM v1.0	Plan Mitigación de Riesgos v2.0	Plan de Gestión de Configuración v2.0
Plan de gestión de requisitos v2.0	Ambiente de desarrollo v2.0	Solicitud de Cambio v2.0
Arquitectura de información v2.0	Plan de capacitación v2.0	
Arquitectura de software v2.0	Roles y responsabilidades v2.0	
Informe del Levantamiento de Información para la Arquitectura de Información v2.0	Documento Visión v2.0	
Modelo de diseño v2.0	Proyecto Técnico v1.0	
Casos de prueba v2.0	Diagnóstico v2.0	
Diseño de Caso de Pruebas funcionales basados en CU v1.0	Minuta de reunión v2.0	
Diseño de Caso de Pruebas funcionales basados en CU v1.0	Diseño de investigación v1.0	
No Conformidades v2.0	Estado del arte del producto a desarrollar v1.0	
Plan de pruebas v2.0	Reporte de Investigación v1.0	
Registro de Prueba Unitaria o Integración v1.0		
Modelo de despliegue v2.0		

Tabla 7. Artefactos del Expediente de Proyecto.

3.6 Conclusiones

Del presente capítulo se concluye lo siguiente:

1. Se implantó el Expediente de proyecto prácticamente en el 100% de los proyectos de

la UCI.

2. Se pudieron ejecutar revisiones a los proyectos basadas en lo estandarizado tanto en el Expediente de proyecto como en los lineamientos de calidad.
3. En las revisiones realizadas se detectaron un conjunto de oportunidades de mejora tanto para el Expediente de proyecto como para los Lineamientos de calidad.
4. En las pruebas de software se pudieron detectar no conformidades en etapas tempranas del desarrollo que si una estandarización de los artefactos no hubiera sido posible.

4 Conclusiones

1. Se definió un expediente para los proyectos de desarrollo de software basado en estándares y normas internacionales, que facilita la estandarización de los artefactos, así como el intercambio de información y la reducción del número de no conformidades.
2. Se logró estandarizar y unificar el formato de las documentaciones para todos los clientes.
3. Se definieron los lineamientos de calidad basados en las prácticas recomendadas para el desarrollo de software, así como en las metodologías más utilizadas en la Universidad, lo que permitirá la homogeneización del proceso de desarrollo.
4. Se definió una estrategia para la implantación de ambos artefactos que ha permitido tener retroalimentación de los artefactos para su mejora continua.
5. Los artefactos y el proceso de implantación de los mismos se encuentran en concordancia con los resultados del proceso de mejora que se está ejecutando en la Universidad.
6. Se cumplió con el objetivo planteado al definirse una estrategia para la estandarización de la documentación y las actividades para el desarrollo de software en la UCI.
7. Se demostró la hipótesis de que la estandarización de la documentación y buenas prácticas aplicables al ciclo e desarrollo de un proyecto software contribuye al efectivo control de la calidad y al aumento de la calidad de los productos resultantes.

5 Recomendaciones

Teniendo en cuenta los principios que guiaron el desarrollo de los artefactos y los resultados de la implantación se recomienda:

1. ampliar los artefactos del Expediente de Proyecto para los tipos de desarrollos no contemplados en la realización del mismo,
2. actualizar el Expediente y los Lineamientos con los resultados que se vayan obteniendo como parte del Programa de Mejora de Procesos,
3. insertar artefactos y lineamientos correspondientes a niveles de madurez superiores de CMMI para ir creando las bases para su posterior implantación.

6 Referencias bibliográficas

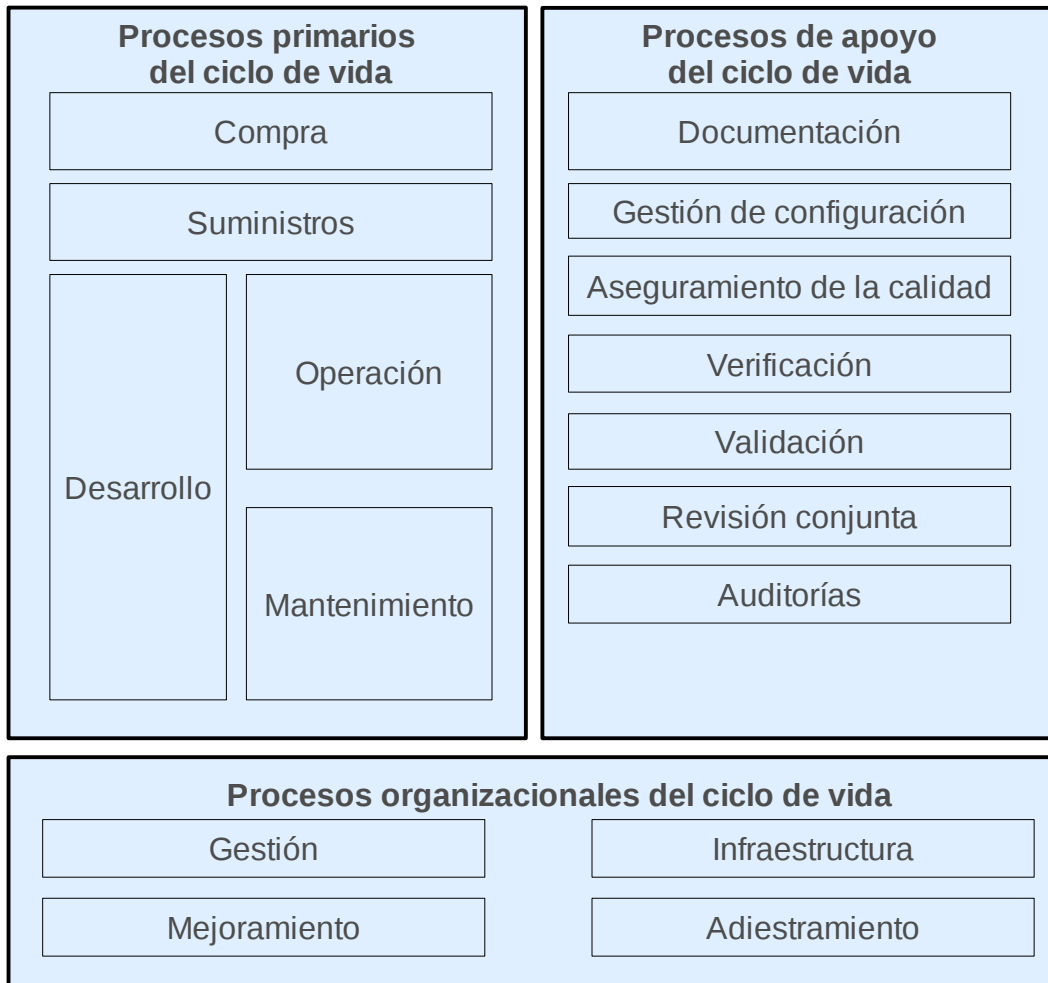
1. [Ambler 2008] Ambler, Scott. The Agile System Development Lifecycle (SDLC). 2008. Available from world wide web: <<http://www.ambysoft.com/essays/agileLifecyle.html>>.
2. [Avangate 2006] Avangate, Software product documentation, http://www.avangate.com/articles/Software-product-documentation_22.htm, 2010.
3. [de Antonio 2001] de Antonio, Angelica. *Gestión de configuración*. [Chile]: SPIN, 2001.
4. [Bedini 2007] Bedini, Alejandro. Extracto del libro en formato digital “calidad tradicional y de software, 2007.
5. [Brickey 2004] Brickey, Cheryl. Getting from Here (SW-CMM) to There (CMMI) in a Large Organization, SEPG 2004, 2004
6. [Bulgerman 2002] Burgelman, R.A., Strategy is Destiny: How strategy-making shapes a company's future, New York: The Free Press, 2002.
7. [Business 2010] Business Dictionary. <http://www.businessdictionary.com/definition/strategy.html>, 2010.
8. [CMM 1993] CMM. The Capability Maturity Model. 1993.
9. [CMMI 2005] CMMI for Development. SEI, EEUU. 2005.
10. [CMMI 2007] CMMI for Adquisition. SEI, EEUU. 2005.
11. [CMMI 2009] CMMI for Services. SEI, EEUU. 2005.
12. [CONTRUX 2008] CONTRUX. Software Development Resources. 2008. Available from world wide web: <<http://www.construx.com/Page.aspx?nid=14>>.
13. [Crosby 1979] Crosby, P.B.: Quality is free. Ed. McGraw Hill, Nueva York, 1979.
14. [Cueva 1999] Cueva, Juan M. Calidad de Software, Universidad de Oviedo, España, 1999.
15. [DCS 2008a] DCS. Diagnóstico 2008.
16. [DCS 2008b] DCS. Diagnóstico 2008. Resultados del levantamiento de información. UCI, 2008.
17. [Delgado 2009], Delgado Martinez, Ramses, Expediente de proyecto en la UCI, Memorias XIII Feria Internacional Informática 2009, 2009.
18. [E 2002] E, Emma. La evolución desde la gestión convencional a la gestión moderna del software (CMM y CMMI). 2002.
19. [Fernandez 2002] Fernández, H. “La Mejora Continua de los Resultados”, www.buscarportal.com/articulos/iso_9001_enfoque_procesos.html, Asturias, España, 2002.
20. [Ferrer 2003] Ferrer, Jorge. Metodologías ágiles. Germinus, España, 2003.
21. [Gartner 2001] Gartner. Describing the Capability Maturity Model. 2001.
22. [Garvin 1984] Garvin, D.A., “What Does “Product Quality Really Mean?”. Sloan Management Review. Vol. 26, No.1, p.25-43, 1984.
23. [Gibson 2006] Gibson, Diane. Performance Results of CMMI®-Based Process, Agosto 2006.
24. [Grundmann 2005] Grundmann, Markus. A CMMI Maturity Level 2 assessmen of RUP,

- IBM, 2005.
25. [Heredia 1985] Heredia, R. "Dirección Integrada de Proyecto" Ed. Alianza, Segunda edición, Madrid, 1985.
 26. [Humphrey 2001] Humphrey, Watts. Introducción al Proceso de Software Personal. [Madrid]: Addison Wesley, 2001.
 27. [Humphrey 2000] Humphrey, Watts. Introduction to the Team Software Process. Assison Wesley, 2000.
 28. [IBM 2008] IBM. Best practices for software development projects. 2008.
 29. [IBM 2007] IBM. Rational Unified Procces. 2007.
 30. [IEEE 2007] IEEE. IEEE Standard for Developing Software Life Cycle Processes. 2007.
 31. [IEEE 1990] IEEE. IEEE Standard for Software Configuration Management. 1990.
 32. [IEEE 1990a] IEEE Std 610. "IEEE Standard Glossary of Software Engineering Terminology", 1990
 33. [IEEE 1998] IEEE. IEEE Standard for Software Quality Assurance Plans. 1998.
 34. [IEEE 1990] IEEE. IEEE Standard Glossary of Software Engineering Terminology. 1990.
 35. [ISO 2005a] ISO. International Organization for Standarization. <http://www.iso.org/iso/home.htm>, 2005.
 36. [ISO 2005b] ISO. ISO/IEC 12207:2005. 2005b.
 37. [ISO 2004] ISO. Software engineering — Guidelines for the application of ISO 9001:2000 to computer software. 2004.
 38. [ISO 2005] ISO/ FDIS 9001:2000. Sistema de Gestión de la Calidad — Requisitos.
 39. [ISO 2005c] ISO / FDIS 9000: 2005. Sistemas de Gestión de la Calida — Fundamentos y Vocablos.
 40. [ISO 2000] ISO/ FDIS 9004:2000. Sistema de Gestión de la Calidad — Recomendaciones para la mejora del desempeño., 2000.
 41. [Juran 1990] Juran, J. "Juran y la planificación para la calidad. Ed. Díaz de Santos, Madrid, 1990.
 42. [Juran 1993] Juran, J y Gryna, F "Manual de control de la calidad", 4ta edición, Mc Graw-Hill, 1993.
 43. [Moprosoft 2008] MoProsoft. Comunidad Moprosoft. <http://www.comunidadmoprosoft.org.mx/>, 2008.
 44. [NASA 2008] NASA. GSFC Software Process Improvement. <http://software.gsfc.nasa.gov/index.cfm>, 2008.
 45. [NASA 2008a] NASA. Software Assurance Guidebook, 2008.
 46. [NASA 2008b] NASA. Software Assurance Standards, 2008.
 47. [NYCE 2009] NYCE. Unidad de Verificación de T.I. <http://www.nyce.org.mx/dictamenes.htm>, 2009.
 48. [Oktaba 2005] Oktaba, Hanna. Historia de una norma. Revista Software Guru, Junio

- de 2005.
49. [Pleue 2007] Pleue, Brandon. CMMI in Latin America. 2007.
 50. [Pressman 1998] Pressman, Roger. Ingeniería de Software. Un enfoque práctico. 4ta edición 1998.
 51. [RAE 2001] Real Academia Española, "Diccionario de la lengua española", 22ª ed., 2001.
 52. [Readysset 2010] Tigris.org. <http://readysset.tigris.org/>, 2010.
 53. [Reeves 1994] Reeves, C.A. y Bednar, D.A.: "Defining Quality: Alternatives and Implications". Academy of Management Review. Vol.19, No.3,p.419-445, 1994.
 54. [Royce 2006] Royce, Walker. CMM vs. CMMI: From Conventional to Modern Software Management, 2008
 55. [RUP 2004] RUP. El procesos unificadode desarrollo de software. Editorial Felix Varela, 2004.
 56. [RUP 2001] RUP. Rational Unified Procces. 2001.
 57. [Pollice 2001] Pollice, Gary. RUP and XP, finding commond ground. The Rational Edge, IBM, 2001.
 58. [Sintesis 2005] Revista Síntesis. <http://www.revistasintesis.cl/>, 2005
 59. [Sommerville 2001] Sommerville, Ian. Software documentation, Lancaster University, UK, 2001.
 60. [SPICE 2010] ISO/IEC 15504 Standard, <http://www.isospice.com/categories/ISO{47}IEC-15504-Standard/>, 2010.
 61. [Stylus 2010] Software Development Life Cycle, <http://www.stylusinc.com/Common/Concerns/SoftwareDevtPhilosophy.php>, 2010.
 62. [SWEBOK 2004] IEEE, Software Engineering Body of Knowledge , 2004.
 63. [Tantara 2009] Tantara. Tantara: ISO standards for the software industry. http://www.tantara.ab.ca/iso_std.htm, 2009.
 64. [Taguchi 1979] Taguchi, G. Introduction to Off-line Quality Control, Japanese Standards Association, Tokyo, 1979.
 65. [UCI 2008] Planeación estratégica, UCI, 2008.
 66. [Urquiaga 2006] Urquiaga, I y otros. "Cultura por la calidad", Instituto de Investigación y Normalización, Habana, 2006.
 67. [Williams 2004] Williams, Ralph. Useable CMMI documentation, Cooliemon, 2006.

7 Anexos

7.1 Anexo 1. Ciclo de vida del software según la ISO 12207.



7.2 Anexo 2. Guía exploratoria para revisiones

Guía exploratoria			
Criterio de evaluación	Referencia	Evidencia	Procedimientos
Establecimiento del glosario de término de los documentos			
Gestión de los requisitos			
Conformidad del producto con los requisitos			
Conformidad con el esquema del expediente de proyecto			
Completamiento del expediente de proyecto			

Registro de actividades conjuntas y visitas de los diferentes factores del proyecto			
Distribución de las PCs y asignación de los horarios de producción			
Establecimiento y cumplimiento del cronograma del proyecto			
Definición del proyecto			
Establecimiento del alcance del proyecto			
Gestión de los riesgos del proyecto			
Establecimiento de la estructura organizacional, los roles y las responsabilidades			
Asignación de los roles y las tareas			
Vinculación con los procesos docentes e investigativos			
Empleo de la metodología seleccionada y de la notación UML			
Registro del capital humano, los recursos materiales y los resultados el proyecto			
Registro del tiempo de trabajo de los desarrolladores y de los defectos detectados			
Registro de las estimaciones realizadas			
Establecimiento de la gestión de configuración			
Definición y empleo de herramientas horizontales			
Definición y empleo de tecnología y herramientas específicas			
Requisitos de hardware			
Definición general de arquitectura			