



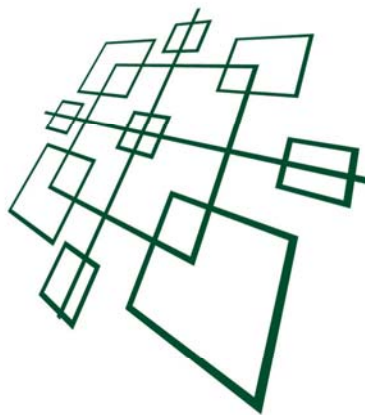
Universidad de las Ciencias
Informáticas

Dirección de Informatización

ANÁLISIS DE LA CALIDAD DEL PROCESO DE DESARROLLO DE SOFTWARE EN EL PROYECTO AKADEMOS

Trabajo de diploma para optar por el título de:

**Ingeniero en
Ciencias Informáticas**



Autor: Grisel Infante Costa

Tutor: Ing. Ramsés Delgado Martínez

Ciudad de la Habana, Junio 2007

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Dirección de Informatización de la Universidad de las Ciencias Informáticas; así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 15 días del mes de junio del año 2007.

Grisel Infante Costa

Ing. Ramsés Delgado Martínez

OPINIÓN DEL TUTOR DEL TRABAJO DE DIPLOMA

Título: **Análisis de la calidad del proceso de desarrollo de software en el proyecto Akademos**

Autor: **Grisel Infante Costa**

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de _____ puntos.

Dado a los 15 días del mes de Junio de 2007

Ing. Ramsés Delgado Martínez

DATOS DE CONTACTO

Tutor: Ing. Ramsés Delgado Martínez

Cargo que ocupa: Especialista Dirección de Calidad de Software, UCI

e-mail: ramsesd@uci.cu

Agradecimientos

A Ramsés Delgado Martínez, por haber aceptado ser mi tutor y por haberme dedicado sus horas...

A Emil Lima Valdés, por darme ánimos en los primeros momentos...

Al Grupo de Desarrollo de Akademos, por su colaboración...

A mis compañeros de estudio, por todo...

Dedicatoria

A mis padres...

A José Antonio Pereda de la Cruz, mi compañero de la vida...

Resumen

En la Universidad de las Ciencias Informáticas, como parte de las tareas para la informatización de los servicios, se implementa el Sistema Automatizado para la Gestión Académica Akademos. Por su importancia, se requiere que el proceso de desarrollo tenga la calidad suficiente como para garantizar la calidad del producto que se obtenga.

Este trabajo, usando como guía el modelo CMMI, un modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software, realiza un estudio del proceso utilizado en el proyecto Akademos con vista a detectar sus principales deficiencias. Este estudio está orientado a las áreas de Planificación y Monitoreo y control del proyecto, Aseguramiento de la calidad, Gestión de configuración, Medición y análisis y Gestión de requisitos. Y como resultado propone una estrategia, que contiene tareas específicas para llevar a cabo, y que contribuyen a elevar la calidad del proceso.

Palabras clave

Proceso de desarrollo de software, Sistema Automatizado para la Gestión Académica, Akademos, estrategia, calidad de proceso.

Abstract

System Automated for the Academic Management Akademos is implemented in the University of Informatics Sciences as one of the tasks for the computerization of the services and process. By his importance, it is required that the development process has the sufficient quality like guaranteeing the quality of the product that is obtained. This investigation proposes a strategy for the development process carried out in the Akademos project, using the Capability Maturity Model Integration, a process improvement maturity model for the development of products and services.

Índice

Introducción	1
Capítulo 1 Fundamentación teórica	5
1.1 Introducción.....	5
1.2 Ingeniería de software.....	5
1.2.1 Proceso de software	7
1.2.2 El proceso unificado de desarrollo de software	7
1.3 Calidad de software	10
1.4 Normas y estándares de calidad de software	11
1.4.1 ISO/IEC 12207	12
1.4.2 ISO/IEC 15504	13
1.4.3 IEEE	15
1.4.4 CMMI	15
1.4.5 Comparación	21
1.5 Conclusiones.....	22
Capítulo 2 Análisis del proceso de desarrollo	23
2.1 Introducción.....	23
2.2 Gestión de requisitos	23
2.2.1 La gestión de los requisitos en el proyecto Akademos.....	25
2.3 Planificación del proyecto.....	26
2.3.1 La planificación en el proyecto Akademos.....	28
2.4 Monitoreo y control del proyecto	29
2.4.1 Monitoreo y control del proyecto Akademos.....	33
2.5 Aseguramiento de la calidad del proceso y el producto	33
2.5.1 Aseguramiento de la calidad del proceso y el producto en el proyecto Akademos	34
2.6 Gestión de configuración.....	35
2.6.1 La gestión de la configuración en el proyecto Akademos.....	37
2.7 Medición y análisis	37
2.8 Conclusiones.....	39
Capítulo 3 Actividades para mejorar los procesos.....	40
3.1 Introducción.....	40
3.2 Tareas para la organización de los procesos.....	40
3.3 Planificación del proyecto y los procesos.....	42
3.4 Aseguramiento de la calidad	46
3.4.1 Definición de estándares	47
3.4.2 Procedimientos para los procesos.....	47
3.4.3 Aseguramiento de la calidad del proceso	48
3.4.4 Aseguramiento de la calidad del producto.....	49

3.5 Monitoreo y control del proyecto	49
3.6 Medición y análisis	51
3.7 Gestión de la configuración.....	53
3.8 Gestión de requisitos	55
3.9 Conclusiones.....	56
Conclusiones generales	58
Recomendaciones	59
Bibliografía.....	60
Glosario	64
Anexos.....	65
Anexo 1	65
Anexo 2.....	66
Anexo 4.....	68
Anexo 5.....	69
Anexo 6.....	70
Anexo 7.....	71
Anexo 8.....	73
Anexo 9.....	74
Anexo 10.....	77
Anexo 11.....	78
Anexo 12.....	79

Índice de figuras y tablas

Figura 1.1 Capas de la IS.....	5
Figura 1.2 Gráfica de RUP en dos dimensiones.....	8
Tabla 1.1 Representaciones del modelo CMMI.....	16
Tabla 1.2 Áreas de proceso de CMMI.....	17

Introducción

En el mundo actual el software se ha convertido en uno de los pilares de la industria. Pero el crecimiento de la complejidad y el volumen de los proyectos, la mala calidad del trabajo, la planificación irreal, el personal inadecuado y los cambios no controlados inciden en la eficiencia y la eficacia de esta industria. No obstante, las exigencias en materia de calidad crecen, ya sea por las necesidades de los clientes de obtener software en menos tiempo y a menor costo, como para garantizar la permanencia en el mercado.

Estos problemas, incluso los que son puramente organizativos, pueden encontrar solución al utilizar los métodos y los procedimientos que ofrece la Ingeniería de software, en actividades como la planificación, la administración y la utilización de metodologías de desarrollo de software. Según Pressman, la Ingeniería de software es una tecnología multicapa en la que se pueden identificar los métodos, el proceso y las herramientas. Los métodos indican cómo construir técnicamente el software. El proceso es el fundamento de la disciplina, la unión que mantiene juntas las capas de la tecnología. Y las herramientas son el soporte automático o semiautomático para el proceso y los métodos. (Pressman 2002)

Cualquier enfoque de Ingeniería de Software que se aplique está orientado hacia la obtención y mantenimiento de software de alta calidad, y para lograrlo es necesario que todos los involucrados en el proceso sean responsables por la calidad del software que se produce. Pero en la práctica, comienzan a preocuparse por esta después de haber escrito el código, o consideran el seguimiento de la calidad a lo largo del proceso una tarea tediosa y agotadora.

La definición de calidad del software dada también por Pressman, enfatiza en la concordancia del software con los requisitos funcionales y de rendimiento que se hayan establecido, que los estándares de desarrollo estén documentados y que cuente con las características implícitas que se esperan de cualquier software. (Pressman 2002)

Otro aspecto definitorio de la calidad del software producido es la calidad del proceso empleado. Un proceso que siga un conjunto de “buenas prácticas” para el desarrollo de software, puede garantizar el éxito del proyecto y la calidad del producto resultante.

Por estas razones, las organizaciones concuerdan en que un software de alta calidad es una meta importante. Y cada vez un mayor número de ellas se preocupa por implantar modelos de calidad en sus procesos, con el objetivo de que sus proyectos sean más predecibles y se reduzcan los riesgos durante el desarrollo. Las normas ISO, los estándares IEEE y CMMI, un modelo de calidad creado específicamente para la industria de software, son de los más usados a nivel mundial.

La industria cubana de software no está aislada de estos problemas y aspiraciones, ya que hoy día se trabaja con muchas deficiencias que de ser superadas, propiciarían el avance a una estadía superior de calidad. En este sentido existen dos elementos que vuelven lento este proceso, en primer lugar, Cuba es un país pionero en la producción de software, por lo que en estos momentos se están dando los primeros pasos y ganando la experiencia necesaria para elevar el nivel de la industria. En segundo lugar, está lo que respecta a la certificación por un modelo de calidad, por ejemplo, CMMI, por ser un modelo norteamericano, posiblemente habría que lidiar con las trabas que impone el bloqueo en aras de lograr una certificación.

La Universidad de Ciencias Informáticas (UCI), es uno de los proyectos de la revolución que permitirá a Cuba abrirse paso en la industria de software. El objetivo de este centro consiste en formar profesionales ingenieros en Ciencias Informáticas, además de acelerar la producción de software en el país.

Con ya cuatro años de creada, en este momento se han visto los primeros avances en materia de calidad del software en la UCI. Se han definido un conjunto de prácticas básicas y una lista de lineamientos mínimos de calidad a cumplir por los proyectos de software que se inician. Estas medidas son el resultado de las experiencias adquiridas en los proyectos anteriormente desarrollados, por lo que no representan un avance en la mejora de la calidad de estos. Este trabajo pretende realizar un estudio acerca de uno de estos proyectos.

La UCI, como parte de las tareas para llevar a cabo, se ha propuesto lograr la automatización de todos los procesos y servicios. Debido a que la Gestión Académica es un proceso vital para cualquier centro de estudios, lo engorrosa que resulta dada la matrícula creciente de la UCI y los varios planes de estudio en que se trabaja, esta se integra en el Sistema Automatizado para la Gestión Académica, Akademos, que ha venido desarrollando un equipo de la Dirección de Informatización. Por su importancia, se requiere que el proceso de desarrollo tenga la calidad suficiente como para garantizar la calidad del producto.

Una revisión a las tareas que está realizando actualmente el equipo de trabajo de Akademos reveló los siguientes aspectos:

- El equipo de desarrollo es inestable.
- No hay claridad en cuanto al estado del proyecto.
- Para comprender la lógica del diseño es necesario analizar todo el código.
- Los miembros del equipo, aunque tienen asignado un rol, no cumplen con sus responsabilidades. En consecuencia, puede que un miembro ejecute tareas que corresponden a otro.
- Si se detectan defectos o el cliente no está conforme con alguna funcionalidad deberá reprogramarse todo, por eso el tiempo de desarrollo es largo, se añaden defectos al software y el equipo debe trabajar mucho.

Estos problemas pueden tener su origen en errores cometidos en el proceso de desarrollo, como una mala gestión del proyecto, o una mala gestión de la calidad del software. De este modo se describe el siguiente **problema de la investigación**:

La no aplicación disciplinada de los procesos de software afecta el desarrollo óptimo del proyecto Akademos.

Para el cual se define como **objeto de estudio**: el proceso de desarrollo de software en el proyecto Akademos, y como **campo de acción**: el proyecto Akademos.

Para resolver este problema se plantea como **objetivo general**:

Definir estrategia para el proceso de desarrollo de software en el proyecto Akademos guiada por un modelo de calidad.

Objetivos específicos.

1. Seleccionar el modelo de calidad apropiado a partir de la comparación de normas, modelos y estándares de calidad de software.
2. Analizar el proceso de desarrollo de Akademos guiado por un modelo de calidad.
3. Proponer estrategia para el proceso de desarrollo de software guiada por un modelo de calidad.

La investigación está estructurada en tres capítulos.

El Capítulo 1 corresponde a la Fundamentación Teórica, donde se exponen los conceptos a tratar en el cuerpo de la investigación. Conceptos asociados a los temas de Ingeniería de software, Calidad del software, y Normas y modelos de calidad del software más usados a nivel mundial.

El Capítulo 2 corresponde al análisis del proceso de desarrollo de software de Akademos guiado por el modelo de calidad CMMI.

El Capítulo 3 corresponde a la serie de actividades que se deben llevar a cabo en aras de mejorar la calidad del proceso en el proyecto Akademos.

Capítulo 1 Fundamentación teórica

1.1 Introducción

En este capítulo se describe el concepto de Ingeniería de Software según Pressman, el cual permite el acercamiento a la definición de proceso de desarrollo hasta llegar a la metodología utilizada en el proyecto Akademos. De manera similar se hace referencia a la calidad del software producido, se estudian los modelos más utilizados a nivel mundial y se escoge uno de ellos para guiar el análisis.

1.2 Ingeniería de software

La Ingeniería de Software (IS) es una tecnología multicapa (Pressman 2002) en la que, según Pressman, se pueden identificar los métodos, que indican cómo construir técnicamente el software; el proceso, que es el fundamento de la IS, es la unión que mantiene juntas las capas de la tecnología; y las herramientas, que forman el soporte automático o semiautomático para el proceso y los métodos. (Figura 1.1) (UCI 2005)

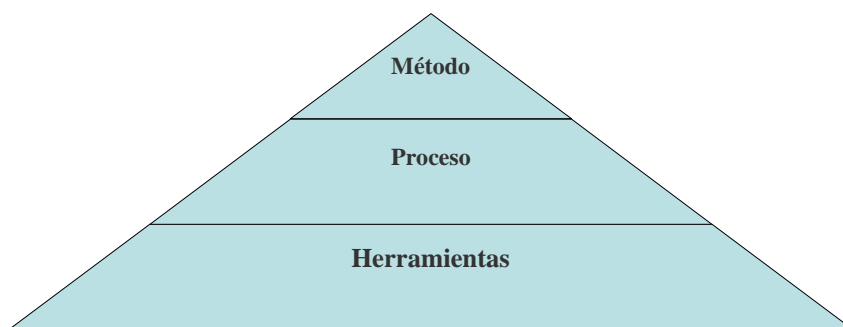


Figura 1.1 Capas de la IS.

La IS incluye actividades de planificación, administración y el uso de metodologías de desarrollo.

- La actividad de planificación permite estimar razonablemente los recursos, el coste y el tiempo necesarios para el desarrollo del proyecto. (Pressman 2002)
- Con la administración se controla, evalúa y corrige la dirección de acuerdo a las contingencias que se vayan presentando durante el desarrollo.
- Al usar una metodología de desarrollo se logra acoplar los participantes y obtener la garantía de una determinada calidad.

Existe un conjunto de “buenas prácticas” (UCI 2005) o tareas, que están siendo usadas en el desarrollo de software y que pueden garantizar también el éxito de un proyecto de software.

Estas tareas son:

1. Administración de requerimientos: Identificación y representación de las funcionalidades requeridas y otras restricciones y decisiones en forma de requerimientos que puedan ser rastreados durante el desarrollo de software.
2. Usar arquitectura de componentes: Definir una arquitectura robusta y flexible que use componentes ensamblados, o sea, módulos que cumplan una función clara, ya sean nuevos o existentes.
3. Modelar visualmente: Modelar el sistema usando elementos visuales que escondan los detalles, pero que brinden una abstracción adecuada para entender en su totalidad el sistema.
4. Verificar la calidad: Comprobar la calidad a partir de analizar cómo se han implementado los requerimientos durante todo el proceso de desarrollo.
5. Desarrollar iterativamente: Construir la solución a través de refinamientos sucesivos en múltiples iteraciones.
6. Controlar los cambios: A partir de que un cambio es aceptado hay que controlar su cumplimiento, documentar y divulgar al equipo qué cambió y aislar el lugar del cambio mientras se esté cumplimentando.

Con estas prácticas se logran equipos de alto rendimiento que producen proyectos más exitosos porque están en plazo, en presupuesto y satisfacen las necesidades del usuario. (UCI 2005) La IS garantiza el cumplimiento de estas prácticas.

1.2.1 Proceso de software

El proceso de software proporciona las actividades estructurales necesarias para desarrollar un software. En sus tres fases genéricas, definición, desarrollo y soporte, convergen elementos, que pueden ser tareas, hitos, productos de trabajo y puntos de garantía de calidad, que permiten a las actividades estructurales adaptarse a las características del proyecto y a los requisitos del equipo. A lo largo del proceso ocurren también actividades de protección, independientes de las estructurales y que pueden estar orientadas al aseguramiento de la calidad, la gestión de la configuración del software y la medición. Las metodologías de desarrollo de software son procesos de software. (Pressman 2002)

Aunque las metodologías de desarrollo de software son el fundamento de la IS y por ende, mecanismos para alcanzar software de calidad, los aspectos de la gestión de proyectos, que también buscan calidad en el proceso de desarrollo y en el producto final, no deben obviarse.

1.2.2 El proceso unificado de desarrollo de software

El Proceso Unificado de Desarrollo de Software (RUP), utilizado en el proyecto Akademos, es una metodología de desarrollo única debido a sus tres características fundamentales: está dirigido por casos de uso (CU), está centrado en la arquitectura y es iterativo e incremental. (RUP 2003)

Como lo definen sus creadores, RUP es 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. (J. B. R. 1999)

El proceso puede describirse en dos dimensiones, o a lo largo de dos ejes. (Figura 1.2) (RUP 2003)

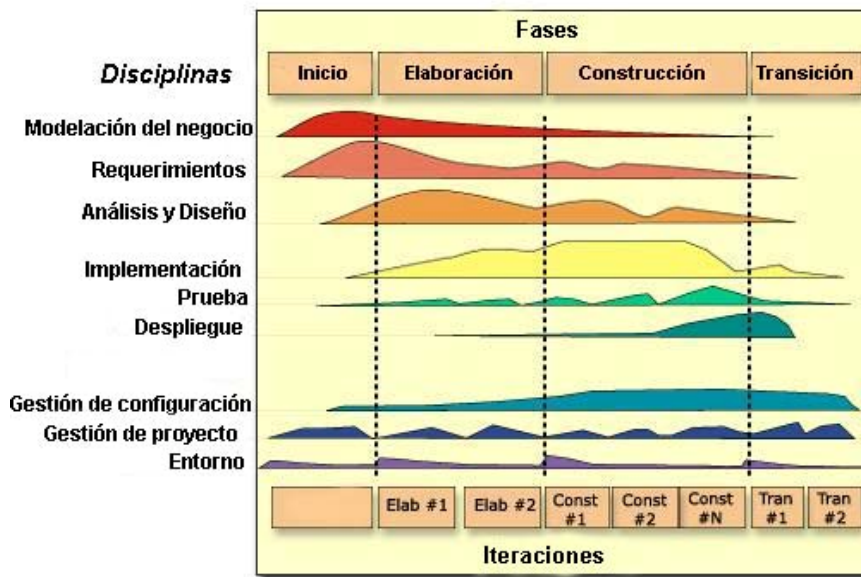


Figura 1.2 Gráfica de RUP en dos dimensiones.

El eje horizontal representa tiempo y muestra el aspecto dinámico del proceso, expresado en términos de ciclos, fases, iteraciones, y metas. El eje vertical representa el aspecto estático del proceso; cómo está descrito en términos de actividades, artefactos, trabajadores y flujos de trabajo. (RUP 2003)

RUP describe cómo utilizar de forma efectiva las buenas prácticas del desarrollo moderno de software en una forma que es aplicable para un amplio rango de proyectos y organizaciones. (J. B. R. 1999)

El desarrollo iterativo, una de las características fundamentales de RUP, permite un entendimiento incremental del problema a través de refinamientos sucesivos; habilita una fácil retroalimentación del usuario; impone metas específicas que propician que el equipo de desarrollo mantenga su atención en producir resultados, a la vez que el progreso es medido

conforme avanzan las implementaciones, obteniendo, por cada iteración, una versión ejecutable del software que se produce.

A través de la Gestión de los Requisitos se determinan, organizan, y documentan las funcionalidades y restricciones en los requisitos. Los que son capturados y comunicados a través de los CU, que dirigen el trabajo desde el análisis hasta las pruebas como otra de las tres características fundamentales de RUP. Además, permite llevar un registro documentado de los cambios efectuados y las decisiones tomadas acerca de estos requisitos.

Cabe señalar que RUP está enfocado hacia la construcción temprana de una arquitectura robusta, derivada de los CU más significativos, como se describe en una de sus características fundamentales. Por lo que pone en práctica el uso de la arquitectura basada en componentes, explotando al máximo las facilidades de reusabilidad.

Usando la modelación visual del software, en Lenguaje Unificado de Modelado (UML), muestra cómo encajan de forma conjunta los elementos del sistema. Propiciando así una comunicación no ambigua debido a que se mantiene la consistencia entre el diseño y la implementación.

RUP verifica la calidad del software con respecto a los requisitos basados en la confiabilidad, funcionalidad, desempeño de la aplicación y del sistema, efectuando pruebas para cada iteración. Estas pruebas, por cada escenario de CU, aseguran que todos los requisitos están propiamente implementados.

Con el objetivo de permitir un desarrollo iterativo, RUP propone controlar, llevar un registro y monitorear los cambios. Para ello se controlan todos los artefactos de software y se establecen espacios de trabajo seguros para cada desarrollador, de manera que puedan aislarse los cambios.

Otras facilidades que incorpora se pueden ver en proveer a cada miembro del equipo de un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas del desarrollo, o que crea y mantiene modelos, en lugar de enfocarse en la producción de una gran cantidad de papeles de documentación. (UCI 2005)

1.3 Calidad de software

La definición de calidad del software de Pressman expone lo siguiente: “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente”. (Pressman 2002) En esta definición se hace hincapié en tres elementos, los requisitos, los estándares y unas características llamadas implícitas.

Los requisitos del software son la base de las medidas de la calidad. Por tanto, la falta de concordancia con los requisitos se considera una falta de calidad.

Los estándares especificados definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la ingeniería del software. Si no se siguen esos criterios, casi siempre habrá falta de calidad.

Existe un conjunto de requisitos implícitos que a menudo no se mencionan, por ejemplo, el deseo por facilitar el uso y un buen mantenimiento. Si el software se ajusta a sus requisitos explícitos pero falla en alcanzar los requisitos implícitos, la calidad del software queda en entredicho.

La calidad del software se ve afectada por un conjunto de factores que McCall categorizó en dos grupos: los factores que pueden ser medidos directamente y los que sólo se pueden medir indirectamente. Pressman, al referirse a estos factores señala que en todos los casos debe incluirse la medición, o sea, se debe comparar el software con una referencia y sólo así se llegará a una conclusión acerca de su calidad. (Pressman 2002)

Por la parte del usuario, un software con calidad debe ser primeramente eficaz, debe realizar las funciones establecidas. Además debe ser amigable, es decir, fácilmente usado por el grupo de usuarios a quien está dirigido. De manera general un usuario debe utilizar el software porque produce resultados confiables y realiza todas las operaciones que se requieren en un tiempo aceptado.

Desde otro punto de vista, un software con calidad debe ser eficiente, o sea, el costo de su desarrollo tomando todos los recursos y el costo de su operación debe ser tal que las organizaciones involucradas en su desarrollo y uso obtengan el máximo beneficio o por lo menos un beneficio aceptable en un período de tiempo establecido.

Para lograr que el producto sea realmente usado por el usuario y que permita además su corrección, ajuste y modificación durante largo tiempo, deben comprometerse en todas las etapas del desarrollo, todos los involucrados en el proceso. Porque “la garantía de calidad del software, es una actividad de protección que se aplica a lo largo de todo el proceso de ingeniería del software”. (Pressman 2002)

1.4 Normas y estándares de calidad de software

En la última década se ha desarrollado gran cantidad de métodos orientados hacia la calidad del software. El principal problema al que se enfrentan las empresas productoras consiste en escoger uno dentro del universo de modelos de calidad, de procesos y de técnicas de trabajo. Sin embargo, el éxito en la selección dependerá de que se tengan en cuenta las características de los proyectos, la visión y la cultura de la organización. La diversidad de métodos obedece a la diversidad de organizaciones y tipos de proyectos posibles. (Palacio 2005)

Entre los métodos más usados en la actualidad se encuentran el estándar ISO/IEC 12207, los estándares de IEEE, el modelo CMMI e ISO/IEC 15504.

El estándar internacional ISO/IEC 12207 determina cuál es el ciclo de vida de un proyecto, y los procesos y tareas que intervienen en cada una de sus fases.

Los estándares del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) son también utilizados porque definen en su conjunto, procesos y productos de trabajo.

ISO/IEC 15504 es un modelo para evaluar los procesos de la organización. Centra el foco en sus capacidades y guía las acciones para mejorarlos.

CMMI es un modelo para la mejora, o evaluación, de los procesos de desarrollo y mantenimiento de sistemas y productos de software. Es el modelo que se encuentra actualmente a la vanguardia en cuanto a la mejora de procesos.

1.4.1 ISO/IEC 12207

ISO/IEC 12207 es un estándar internacional para el establecimiento del ciclo de vida del software, incluyendo procesos y actividades que se aplican desde la definición de requisitos, la adquisición y configuración de los servicios del sistema, hasta la finalización de su uso. (Wikipedia 2007) Su objetivo principal consiste en proporcionar una estructura común para que todos los involucrados en el desarrollo de software usen un lenguaje común, que se traduce en procesos bien definidos.

El estándar está concebido bajo dos principios fundamentales:

- Modularidad
- Responsabilidad

Con la modularidad se pretende conseguir procesos con un mínimo acoplamiento y una máxima cohesión.

En cuanto a la responsabilidad, se busca establecer un responsable para cada proceso, facilitando la aplicación del estándar en proyectos en los que pueden existir distintas personas u organizaciones involucradas.

Los procesos, orientados a objetivos finales, se dividen en actividades y estas a su vez en tareas, y se clasifican en tres tipos:

1. Principales
2. De soporte
3. De la organización

Los procesos de soporte y de organización deben existir independientemente de la organización y del proyecto ejecutado.

Los procesos principales, enumerados a continuación, se instancian de acuerdo con la situación particular.

1. Adquisición
2. Suministro
3. Desarrollo
4. Explotación
5. Mantenimiento
6. Procesos de soporte
7. Documentación
8. Gestión de la configuración
9. Aseguramiento de la calidad
10. Verificación
11. Validación
12. Revisión conjunta
13. Auditoría
14. Resolución de problemas
15. Procesos de la organización
16. Gestión
17. Infraestructura
18. Mejora
19. Formación

1.4.2 ISO/IEC 15504

ISO/IEC 15504 surge de la idea de desarrollar un modelo que fuera la base de un futuro estándar internacional para la evaluación de los procesos del ciclo de vida del software. Inicialmente es

denominado SPICE, y no es un modelo o método, sino un marco de trabajo para métodos de evaluación.

El proyecto SPICE, después de tres fases de trabajo, desde 1993, que incluyeron la convocatoria a diferentes organizaciones para probarlo y valorar sus resultados, en el período 2003- 2005, se publica como el estándar internacional ISO/IEC 15504. Entre sus principales características se puede señalar que comprende evaluación y mejora de procesos, así como determinación de capacidades.

Con respecto al enfoque de procesos el estándar describe los procesos que una organización puede desarrollar y todas las prácticas genéricas que caracterizan las potencialidades de estos procesos, agrupándolos en cinco categorías.

1. Procesos Cliente
2. Procesos de Ingeniería
3. Procesos de Proyecto
4. Procesos de Soporte
5. Procesos de la Organización

La evolución de la capacidad de los procesos está expresada en términos de niveles de capacidad, características comunes, y prácticas genéricas. Un nivel de capacidad es un conjunto de actividades que trabajan juntas para proveer una mejor ejecución de los procesos. Cada nivel provee una mejor y más compleja ejecución de los procesos que el nivel predecesor.

Nivel 0: Proceso Incompleto

Nivel 1: Proceso Realizado

Nivel 2: Proceso Gestionado

Nivel 3: Proceso Establecido

Nivel 4: Proceso Predecible

Nivel 5: Proceso en Optimización

1.4.3 IEEE

IEEE es una organización internacional dedicada a la estandarización. Su principal objetivo está en promover la creatividad, el desarrollo y la integración, compartir y aplicar los avances en las tecnologías de la información, electrónica y ciencias en general para beneficio de la humanidad y de los mismos profesionales. (Wikipedia 2007)

IEEE posee cerca de 900 estándares activos, con otros 700 más bajo desarrollo, todos sobre ingeniería eléctrica, ingeniería en computación, telecomunicaciones y tecnología de control, entre los que se encuentran estándares para procesos y productos de software.

1.4.4 CMMI

Modelo de Capacidad y Madurez Integrado (CMMI), es un modelo para la mejora o evaluación de los procesos de desarrollo y mantenimiento de sistemas y productos de software, creado por un conjunto de miembros de la industria, el gobierno y el Instituto de Ingeniería de Software (SEI), de los Estados Unidos. La versión más reciente fue liberada en agosto de 2006 como CMMI for Development, Versión 1.2, y es a esta versión a la que se hace referencia en este documento.

El SEI desarrolló varios modelos para la mejora y medición de la madurez específicos para varias áreas, pero era común que una organización necesitara implementar a la vez más de uno de estos modelos, por ejemplo, CMM-SW (CMM para software) y CMM-SE (CMM para ingeniería de sistemas). CMMI se crea con el objetivo de facilitar y simplificar la adopción de varios modelos de forma simultánea, su contenido es el resultado de la integración y la evolución de sus predecesores CMM-SW (CMM para Software), CMM-SE (CMM para ingeniería de sistemas), CMM-IPD (CMM para desarrollo integrado del producto).

CMMI está organizado en áreas de proceso. Un AP es un conjunto de las prácticas relacionadas en un área que, cuando se ejecutan colectivamente, satisfacen un sistema de metas

consideradas importantes para llevar a cabo una mejora en esa área. Los componentes del modelo se dividen en tres grupos, requeridos, esperados e informativos.

Los componentes requeridos describen qué deben lograr las organizaciones para satisfacer un AP. En las evaluaciones, se usan para determinar si la AP está satisfecha. Los objetivos genéricos y específicos son componentes requeridos. Los objetivos genéricos están definidos en múltiples APs, describen las características que deben estar presentes al institucionalizar los procesos y alcanzarlos significa mejorar el control en la ejecución de un área de proceso. Los objetivos específicos son aplicados a una única AP y describen las particularidades que deben implementarse para satisfacerla.

Los componentes esperados describen qué deben implementar las organizaciones para lograr un componente requerido. Incluyen las prácticas genéricas y específicas. Las prácticas genéricas se aplican a cualquier área de proceso porque pueden mejorar el funcionamiento y el control de cualquier proceso, cada una de ellas describe una actividad que se considera importante para el cumplimiento del objetivo genérico asociado. Las prácticas específicas (PEs) describen las actividades esperadas para dar lugar al logro de los objetivos específicos de un área de proceso, o bien del objetivo específico asociado.

Los componentes informativos detallan elementos que pueden ayudar a las organizaciones en el acercamiento de los componentes requeridos y esperados.

Como resultado de la integración, CMMI tiene dos representaciones, continua y escalonada. Ambas representaciones son equivalentes en contenido, por lo que la adopción de una u otra dependerá de las características o las prioridades de mejora de la organización. (CMMI 2006)

Tabla 1.1 Representaciones del modelo CMMI.

Nivel	Representación continua Niveles de capacidad	Representación escalonada Niveles de madurez
Nivel 0	Incompleto	N/A
Nivel 1	Ejecutado	Inicial

Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4	Cuantitativamente gestionado	Cuantitativamente gestionado
Nivel 5	Optimizado	Optimizado

CMMI tiene 22 APs, las cuales se agrupan según la representación del modelo. Vistas desde la representación continua, se agrupan en 4 categorías según su finalidad: Gestión de proyectos, Ingeniería, Gestión de procesos y Soporte. Vistas desde la representación escalonada, se clasifican en 5 niveles que indican la madurez de los procesos.

Tabla 1.2 Áreas de proceso de CMMI.

Área de proceso	Categoría	Nivel de madurez
Análisis y resolución de problemas	Soporte	5
Gestión de configuración	Soporte	2
Análisis y resolución de decisiones	Soporte	3
Gestión integral del proyecto	Gestión de proyectos	3
Medición y análisis	Soporte	2
Innovación y desarrollo organizacional	Gestión de procesos	5
Definición de procesos	Gestión de procesos	3
Desarrollo de procesos orientados a la organización	Gestión de procesos	3
Mejora de los procesos de la organización	Gestión de procesos	4

Formación	Gestión de procesos	3
Integración del producto	Ingeniería	3
Monitoreo y control del proyecto	Gestión de proyectos	2
Planificación de proyecto	Gestión de proyectos	2
Aseguramiento de la calidad del proceso y el producto	Soporte	2
Gestión cuantitativa de proyectos	Gestión de proyectos	4
Desarrollo de requisitos	Ingeniería	3
Gestión de requisitos	Ingeniería	2
Gestión de riesgos	Gestión de proyectos	3
Gestión de acuerdos con proveedores	Gestión de proyectos	2
Solución técnica	Ingeniería	3
Validación	Ingeniería	3
Verificación	Ingeniería	3

Es importante señalar que los niveles de madurez se utilizan para caracterizar la mejora de la organización con respecto a un sistema de APs, mientras que los niveles de capacidad caracterizan la mejora de la organización concerniente a una única área de proceso.

Siguiendo la representación escalonada se describen a continuación los niveles de madurez del modelo.

Nivel 1 o Inicial. En el nivel Inicial los procesos son generalmente caóticos, el éxito depende de la capacidad y la heroicidad de los trabajadores en la organización y no del uso de procesos probados. Aunque a menudo se llegan a terminar los proyectos, se sobrepasan en el

presupuesto y no cumplen con el calendario acordado. En las organizaciones a este nivel es común que los éxitos no puedan llegar a repetirse y que los proyectos sean abandonados en época de crisis.

Nivel 2 o Gestionado. Los procesos a este nivel están planificados, y son ejecutados, controlados y revisados de acuerdo a la planificación. Las salidas de los procesos cumplen con las descripciones, estándares y procedimientos de proceso definidos, además se revisan y controlan. El éxito en los resultados obtenidos puede repetirse. El desarrollo no es opaco y se puede saber el estado del proyecto en todo momento. Las APs que intervienen en este nivel son:

- Gestión de requisitos.
- Planificación del proyecto.
- Monitoreo y control del proyecto.
- Aseguramiento de la calidad del proceso y el producto.
- Gestión de configuración.
- Medición y Análisis.
- Gestión de acuerdos con proveedores.

Nivel 3 o Definido. En el nivel 3 de madurez, los estándares, las descripciones de proceso, y los procedimientos para un proyecto en general se adaptan para satisfacer al proyecto en particular, estos están correctamente definidos y documentados, siendo esta la principal diferencia entre los niveles 2 y 3 de madurez. Los procesos están definidos y son gestionados con mayor rigurosidad, teniendo en cuenta la interrelación entre las actividades del proceso y de las medidas detalladas del proceso, de sus productos de trabajo, y de sus servicios. Los procesos no sólo afectan a los equipos de desarrollo sino a toda la organización. Las APs que intervienen en este nivel son:

- Desarrollo de requisitos.
- Solución Técnica.
- Integración del producto.

- Verificación.
- Validación.
- Desarrollo de procesos orientados a la organización.
- Definición de procesos.
- Gestión integral del proyecto
- Gestión de riesgos.
- Análisis y resolución de decisiones.
- Formación.

Nivel 4 o Cuantitativamente gestionado. En el nivel 4 de madurez, la organización y los proyectos establecen los objetivos cuantitativos para el funcionamiento de la calidad y del proceso y los utilizan como criterios en la gestión de procesos. Los objetivos cuantitativos se basan en las necesidades del cliente, de los usuarios finales, de la organización, y de los ejecutores del proceso. La principal diferencia entre los niveles 3 y 4 de madurez está en que en el nivel 3 los procesos son predecibles cualitativamente y en el nivel 4, al usar técnicas estadísticas para controlar la mejora de los procesos estos son predecibles cuantitativamente. Las APs que intervienen en este nivel son:

- Gestión cuantitativa de proyectos.
- Mejora de los procesos de la organización.

Nivel 5 u Optimizado. En el nivel 5 de madurez, una organización mejora continuamente sus procesos basándose en una comprensión cuantitativa de las causas comunes de la variación de los procesos. Se identifican, evalúan y ponen en práctica mejoras incrementales e innovadoras para optimizar los procesos. Las APs que intervienen en este nivel son:

- Innovación y desarrollo organizacional.
- Análisis y resolución de problemas.

1.4.5 Comparación

Todos los modelos y estándares analizados en los epígrafes anteriores están orientados hacia el logro de la calidad del proceso y el producto de software. De ahí que la selección de uno de ellos para guiar el análisis del proceso de desarrollo y la estrategia propuesta para el proyecto Akademos se centra en los siguientes elementos.

- Temas que abarcan
- Facilidad de aplicación y comprensión
- Adaptabilidad a la UCI

ISO/IEC 12207 define los procesos que intervienen en el ciclo de desarrollo, mantenimiento y operación de software, e ISO/IEC 15504 y CMMI detallan cómo se deben ejecutar estos procesos. IEEE proporciona los estándares para los productos.

ISO/IEC 15504 ha sido un proyecto de lenta maduración, o sea, ha variado mucho desde sus inicios hasta las últimas publicaciones. Su mayor fortaleza está en que es el primer modelo de procesos en dos dimensiones que se ha desarrollado. ISO/IEC 15504 e ISO/IEC 12207, como todas las normas ISO, son muy dependientes entre ellas, y eso es un factor que vuelve abstractos estos modelos, para su comprensión. Sin embargo, CMMI posee una guía paso a paso para la mejora a través de niveles de madurez y capacidad que los supera, e incluye prácticas de institucionalización, que permiten asegurar que los procesos asociados con cada área de proceso serán efectivos, repetibles y duraderos. Estos niveles de madurez y capacidad, en CMMI y en ISO/IEC 15504 permiten el avance del proyecto sin tener que aplicar el modelo completo.

Por otra parte, CMMI inicialmente estuvo dirigido a grandes corporaciones, por lo que es difícil de implementar en organizaciones pequeñas, lo cual es ventajoso en este caso, dadas las aspiraciones de la UCI de convertirse en una gran fábrica de software.

1.5 Conclusiones

Como resultado del estudio efectuado en este capítulo se definieron aspectos indispensables para la comprensión del proceso de desarrollo de software. Del estudio de las normas, estándares y modelos orientados a la calidad, utilizados a nivel mundial, resultó la selección del modelo CMMI, en el nivel 2 de madurez, para guiar el análisis del proceso de desarrollo en el proyecto Akademos, así como los estándares de la IEEE para los productos que se obtienen en la estrategia propuesta en el capítulo 3.

Capítulo 2 Análisis del proceso de desarrollo

2.1 Introducción

En el Capítulo se presenta un análisis de las actividades que se realizan como parte del proceso de desarrollo de software en el proyecto Akademos. Este análisis está guiado por las prácticas que propone el modelo CMMI para el nivel 2 de madurez, siendo este el nivel mínimo en cuanto a madurez de la organización, y está estructurado según las APs que deben estar implementadas a este nivel.

En el nivel 2 de madurez, los procesos están planeados y ejecutados de acuerdo con la política de la organización. El proyecto emplea a personal experto que tiene los recursos adecuados para producir salidas controladas; implica a los stakeholders relevantes; se supervisa, se controla, se repasa y evalúa el ajuste a sus descripciones de proceso.

La disciplina en los procesos lograda en el nivel 2 de madurez asegura que las prácticas existentes se conserven aún durante épocas de tensión. Cuando estas prácticas están correctamente ubicadas, los proyectos se realizan y se manejan según sus planes documentados.

El estado de los productos y la entrega de los servicios son visibles a la administración en los puntos definidos, ya sean hitos importantes o en la terminación de las tareas, debido a que estos son apropiadamente controlados, y satisfacen sus descripciones, estándares, y procedimientos especificados.

2.2 Gestión de requisitos

El objetivo de la gestión de requisitos consiste en gestionar los requisitos del producto y sus componentes en el proyecto, e identificar inconsistencias entre dichos requisitos y los planes y subproductos obtenidos.

En esta AP no se define la captura de requisitos en sí, ya que se supone que la metodología de desarrollo tuvo en cuenta este aspecto, sino que se recomiendan prácticas para asegurar que lo que se está entendiendo es realmente lo que el cliente pide, y que por ende la solución técnica es la más adecuada.

Prácticas específicas (CMMI 2006)

PE 1.1: Llegar a un entendimiento de los requisitos a partir del contacto con los proveedores de requisitos.

PE 1.2: Obtener el compromiso de los participantes del proyecto con los requisitos.

PE 1.3: Gestionar los cambios a los requisitos.

PE 1.4: Mantener una trazabilidad bidireccional entre los requisitos y los productos.

PE 1.5: Identificar inconsistencias entre el plan de proyecto, los productos y los requisitos.

Primeramente, los requisitos deben ser evaluados y aprobados. Si un requisito varía, o no está bien definido, tendrá un impacto temporal o económico en el proyecto. Por esta razón, deben ser, además, comprendidos por el cliente y las otras fuentes proveedoras de requisitos apropiadamente escogidas. Los compromisos de los participantes del proyecto, que se encargarán del desarrollo posterior, dependen de la confiabilidad de los requisitos acordados.

Otro aspecto que se incluye es la gestión de los cambios. A lo largo del proyecto pueden generarse nuevos requisitos o necesidades de modificar los ya existentes, estos cambios deben ser gestionados adecuadamente. Para ello será necesario:

- Recopilar la información necesaria.
- Entender el cambio.
- Analizar el impacto, ya sea en fechas, esfuerzo o coste.

- Documentar el cambio para mantener un historial que ayude al seguimiento y control de la volatilidad de los requisitos.

De igual forma se hace necesario revisar en los planes del proyecto, en las actividades y en los productos, la conformidad con los requisitos y los cambios efectuados sobre estos, y así, a partir de las inconsistencias detectadas iniciar las acciones correctivas pertinentes. Cuando los requisitos son gestionados adecuadamente se puede establecer una trazabilidad del requisito original a los productos obtenidos en niveles superiores. La trazabilidad es una evidencia de que los elementos de los niveles superiores provienen de una fuente válida.

2.2.1 La gestión de los requisitos en el proyecto Akademos

En el proyecto Akademos, la captura de requisitos está guiada por la metodología RUP. En la fase de inicio, para garantizar una mayor comprensión de los requisitos, se construyen prototipos de interfaz de usuario para cada requisito. Estos prototipos son mostrados al cliente, usuarios potenciales y los desarrolladores. (Vea en el Anexo 1 el ejemplo de un prototipo de interfaz de usuario efectuado para un caso de uso del sistema) (Collera 2005) De esta forma se logra clarificar ambigüedades y unificar criterios sobre las características del producto. (RUP 2003)

Además, los requisitos, expresados en casos de uso, se utilizan para planificar las iteraciones del ciclo de desarrollo. Una vez establecidos los acuerdos sobre los requisitos, las tareas asociadas a los casos de uso priorizados en cada etapa, se distribuyen entre los miembros del proyecto para su puesta en ejecución.

Durante el desarrollo del producto puede surgir la necesidad de efectuar cambios en los requisitos, sin embargo, estos cambios se realizan al producto pero no sobre la lista de requisitos acordados. En consecuencia, no existe una trazabilidad bidireccional entre los productos y los requisitos porque los cambios sobre estos últimos no se controlan. Estos defectos, generalmente, son detectados en las evaluaciones a la calidad de los productos, las cuales consisten en evaluar

los productos contra la lista de requisitos acordados. Estas evaluaciones se describen más detalladamente en el epígrafe 2.5.

2.3 Planificación del proyecto

El propósito de la planificación de proyecto es establecer y mantener los planes que definen las actividades del proyecto. En la planificación se establecen las estimaciones, se desarrolla un plan como la base de la gestión del proyecto y se obtienen los acuerdos sobre este plan.

La planificación del proyecto incluye el análisis de toda la información necesaria para efectuar la planificación, organización, empleo, dirección, coordinación, reporte y financiamiento. Por tanto, las estimaciones de los parámetros de la planificación deben ser confiables, de manera que cualquier plan efectuado basándose en estas estimaciones tenga la capacidad de soportar los objetivos del proyecto. Al estimar estos parámetros se tienen en cuenta los siguientes factores (CMMI 2006):

- Requisitos del proyecto, que incluyen requisitos del producto, requisitos impuestos por la organización y el cliente, y otros que puedan afectar al proyecto.
- Alcance del proyecto.
- Tareas y productos identificados.
- Propuesta técnica.
- Modelo de ciclo de vida escogido para el proyecto.
- Atributos de las tareas y los productos. Ej. Tamaño o complejidad.
- Calendario.
- Modelos o datos históricos para la conversión de los atributos de las tareas y los productos en horas laborables y costo.
- Metodología usada para determinar el material necesario, las habilidades, las horas laborables y el costo.

Las prácticas que propone CMMI para efectuar las estimaciones (CMMI 2006) son:

PE 1.1: Estimar el alcance del proyecto.

PE 1.2: Establecer estimaciones de los productos y los atributos de las tareas.

PE 1.3: Definir el ciclo de vida del proyecto.

PE 1.4: Determinar estimaciones del esfuerzo y el costo del proyecto.

El alcance del proyecto se estima a partir del establecimiento de un esquema de organización del trabajo basado en la arquitectura del producto. Este análisis permite calcular la fuerza de trabajo necesaria en términos de tareas, roles administrativos y responsabilidades. En él se definen los siguientes elementos:

- Riesgos y tareas de mitigación.
- Tareas correspondientes a las actividades de liberación y soporte.
- Tareas para la adquisición de habilidades y conocimientos.
- Tareas de desarrollo de los planes de soporte necesarios, como los de gestión de configuración, aseguramiento de la calidad y verificación.
- Tareas de integración y gestión de los elementos no desarrollables.

Estos elementos, además de una propuesta técnica para el proyecto, son utilizados para determinar, mediante los métodos adecuados, los atributos de las tareas y los productos, y para definir el ciclo de vida del proyecto. Las estimaciones del esfuerzo y el costo deben estar basadas en modelos o datos históricos, los cuales son usados para transformar los atributos en estimados de horas laborables y costo. La documentación de las estimaciones es necesaria para la revisión y el compromiso de los stakeholders con el plan y el mantenimiento de este a lo largo del desarrollo del proyecto.

El plan de proyecto es un documento formal, usado para gestionar y controlar la ejecución del proyecto. Está basado en los requisitos del proyecto y en las estimaciones establecidas. Define todos los aspectos del esfuerzo relacionados lógicamente: consideraciones acerca del ciclo de vida; tareas técnicas y de gestión; calendario y presupuesto; hitos; gestión de los recursos de datos, identificación de riesgos, requerimientos de habilidades y recursos; y la identificación de

los stakeholders y su interacción (CMMI 2006). A continuación se enumeran las prácticas recomendadas para establecer el plan de proyecto.

PE 2.1: Establecer el presupuesto y el calendario del proyecto.

PE 2.2: Identificar los riesgos del proyecto.

PE 2.3: Planificar la gestión de los datos.

PE 2.4: Planificar los recursos del proyecto.

PE 2.5: Planificar los conocimientos y habilidades del proyecto.

PE 2.6: Planificar la implicación de los stakeholders en el proyecto.

PE 2.7: Establecer el plan de proyecto.

La efectividad del plan depende del compromiso de los responsables de la puesta en práctica y el soporte al plan. El individuo o el equipo que se compromete debe estar seguro de poder llevar a cabo el proyecto dentro del coste y horario establecido. Para lograr que el proyecto sea factible se deben obtener los compromisos con los stakeholders y reconciliar cualquier diferencia entre las estimaciones y los recursos disponibles. En la reconciliación pueden negociarse nuevos recursos, aplicarse formas de incrementar la productividad, o revisar los planes y calendarios. Todos los planes que afectan al proyecto deben ser revisados para asegurar un entendimiento común del alcance, los objetivos, los roles y las relaciones que requiere el éxito del proyecto. Las prácticas para asegurar los compromisos con el plan son las siguientes:

PE 3.1: Revisar los planes que afectan al proyecto.

PE 3.2: Reconsiderar los niveles de trabajo y recursos.

PE 3.3: Obtener los compromisos con el plan.

2.3.1 La planificación en el proyecto Akademos

Las planificaciones realizadas en el proyecto Akademos corresponden a períodos de tiempo en que el equipo de trabajo se compromete a poner en ejecución determinadas funcionalidades del

software. Como resultado se obtiene un documento en que se incluyen las tareas necesarias, a las cuales, según su complejidad, se le asignan los responsables que por sus conocimientos puedan desarrollarla, un tiempo para su entrega, y se distribuyen los recursos. (Vea en el Anexo 2 la versión de un plan efectuado en el proyecto Akademos)

Un miembro designado por el proyecto está encargado de la planificación y del mantenimiento del registro de planes históricos. El plan es revisado periódicamente como parte de las actividades de monitoreo y control para su cumplimiento. Las variaciones en el calendario o las tareas son discutidas y aprobadas entre los integrantes del proyecto, y solamente el planificador está autorizado a efectuar los ajustes pertinentes en el plan.

Las actividades que se incluyen en la planificación pueden ser:

- De ingeniería, son las tareas de diseño, codificación y pruebas.
- De aseguramiento de la calidad del producto, son las evaluaciones efectuadas a los productos de software una vez terminados.
- De gestión de proyecto, pueden ser reuniones del proyecto y cursos de capacitación.

En la fase de inicio no se llevó a cabo la planificación del proyecto. Por tanto, aspectos como los riesgos no se tienen en cuenta; no hay un calendario que seguir y el equipo de desarrollo no puede establecer a ciencia cierta cuánto se ha hecho, y cuánto queda por hacer.

2.4 Monitoreo y control del proyecto

El monitoreo del proyecto según el plan consiste en comparar los valores actuales de los parámetros de la planificación con las estimaciones documentadas en el plan de proyecto. Los valores actuales del progreso y la ejecución, la finalización de las actividades y los hitos, el esfuerzo, el costo, y otros atributos hasta el momento se comparan con el calendario, el presupuesto y las demás estimaciones. Otros aspectos importantes que son monitoreados son los compromisos, los riesgos, la gestión de los datos del proyecto y la participación de los stakeholders.

CMMI recomienda además realizar revisiones al progreso y a los hitos. Estas revisiones son informales, se hacen para mantener al tanto del progreso y ejecución del proyecto a los stakeholders. Por lo que estas no aparecen descritas en los planes del proyecto.

En función de eliminar las desviaciones significativas que la ejecución o los resultados del proyecto tienen con respecto al plan, y que fueron detectadas durante las actividades de monitoreo y las revisiones, el proyecto debe tomar acciones correctivas. CMMI también propone actividades para gestionar la aplicación de estas acciones. A continuación se enumeran las prácticas, con las subprácticas correspondientes para un mejor entendimiento, asociadas a cumplir los objetivos del AP.

PE 1.1: Monitorear los parámetros de la planificación del proyecto.

Subprácticas

1. Monitorear el progreso según el calendario.
2. Monitorear el costo y el esfuerzo consumido.
3. Monitorear los atributos de las tareas y los productos.
4. Monitorear los recursos proporcionados y empleados.
5. Monitorear las habilidades y conocimientos del personal del proyecto.
6. Documentar las desviaciones significativas de los parámetros de la planificación.

PE 1.2: Monitorear los compromisos.

Subprácticas.

1. Revisar regularmente los compromisos.
2. Identificar los compromisos que no estén satisfechos o los que están en riesgo de no satisfacerse.
3. Documentar las revisiones.

PE 1.3: Monitorear los riesgos del proyecto.

Subprácticas.

1. Revisar periódicamente la documentación de los riesgos en el contexto del estado y las circunstancias actuales del proyecto.
2. Revisar la documentación para añadir nueva información disponible o cambiarla.
3. Comunicar a los stakeholders relevantes el estado de los riesgos.

PE 1.4: Monitorear la gestión de los datos.

Subprácticas.

1. Revisar periódicamente las actividades de gestión de los recursos de datos del proyecto según sus descripciones en el plan de proyecto.
2. Identificar y documentar las discrepancias y su impacto.
3. Documentar los resultados de las revisiones.

PE 1.5: Monitorear la participación de los stakeholders.

Subprácticas.

1. Revisar periódicamente el estado de las interacciones de los stakeholders con el proyecto.
2. Identificar y documentar las discrepancias y su impacto.
3. Documentar los resultados de las revisiones.

PE 1.6: Conducir las revisiones al progreso.

Subprácticas.

1. Comunicar regularmente el estado de las actividades asignadas y los productos a los stakeholders relevantes.
2. Revisar los resultados de la colección y análisis de las medidas de control del proyecto.
3. Identificar y documentar las desviaciones y discrepancias con el plan.
4. Documentar las peticiones de cambio y los problemas encontrados en los procesos y los productos.
5. Documentar los resultados de las revisiones.

6. Seguir las peticiones de cambio y los problemas reportados.

PE 1.7: Conducir las revisiones a los hitos.

Subprácticas.

1. Conducir las revisiones a la finalización de determinadas etapas, con los stakeholders relevantes, en puntos significativos del calendario o hitos.
2. Revisar los compromisos, el plan, los riesgos y el estado del proyecto.
3. Identificar y documentar las discrepancias y su impacto.
4. Documentar los resultados de las revisiones, las acciones y las decisiones.
5. Seguir las acciones.

PE 2.1: Analizar las discrepancias.

Subprácticas.

1. Reunir las discrepancias para analizarlas.
2. Analizar las discrepancias y determinar las acciones correctivas necesarias para tratarlas.

PE 2.2: Llevar a cabo las acciones correctivas.

Subprácticas.

1. Determinar y documentar las acciones correctivas necesarias para tratar las discrepancias identificadas.
2. Revisar y acordar con los stakeholders relevantes las acciones que serán ejecutadas.
3. Negociar cambios a los compromisos internos y externos.

PE 2.3: Gestionar las acciones correctivas.

Subprácticas.

1. Monitorear las acciones correctivas hasta su finalización.
2. Analizar los resultados de las acciones correctivas para determinar su efectividad.

3. Determinar y documentar las acciones apropiadas para corregir las desviaciones en los resultados previstos para las acciones correctivas.

2.4.1 Monitoreo y control del proyecto Akademos

El proceso de Monitoreo y control de proyecto depende totalmente de la planificación del proyecto. Por tanto, su ejecución se ve afectada en el proyecto Akademos. Los planes efectuados son revisados de forma periódica, por la dirección del proyecto, para chequear el cumplimiento del calendario trazado y los hitos. Como resultado se obtiene un documento denominado Acta de la revisión. Las no-conformidades detectadas en estas revisiones son discutidas entre los miembros del equipo, quienes acuerdan las acciones correctivas a tomar, y designan un responsable para su ejecución.

2.5 Aseguramiento de la calidad del proceso y el producto

El objeto del Aseguramiento de la calidad del proceso y el producto es proporcionar al equipo de trabajo y la dirección del proyecto una visibilidad objetiva del estado de los productos que se obtienen y los procesos implementados.

Las actividades de aseguramiento de la calidad del proceso y el producto consisten en efectuar evaluaciones a los procesos implementados en el proyecto, y los productos que se obtienen como resultado de estos procesos, contra los estándares y procedimientos definidos. Dos aspectos que añaden importancia a estas evaluaciones son que de ellas se obtienen en primer lugar, elementos que pueden usarse para la mejora de los procesos en futuros productos y servicios, y en segundo lugar, los reportes de inconformidades o no-cumplimientos, problemas identificados durante las evaluaciones, que reflejan desajustes de los procesos o productos con respecto a los estándares y procedimientos definidos, y que constituyen un indicador del estado de la calidad. Por esta razón será necesario propiciar un ambiente que anime la participación de

los trabajadores del proyecto en estas actividades, y registrarlas con suficiente detalle para que el estado y los resultados puedan ser conocidos en todo momento.

CMMI propone las siguientes prácticas para el AP:

PE 1.1: Evaluar objetivamente los procesos.

PE 1.2: Evaluar objetivamente los productos.

CMMI recomienda evaluar los productos antes de ser entregados al cliente, o en los puntos definidos para las evaluaciones durante el desarrollo, así como realizar evaluaciones en curso o incrementales.

PE 2.1: Comunicar y asegurar la resolución de los no-cumplimientos.

PE 2.2: Establecer registros de los resultados de las actividades de aseguramiento de la calidad.

Los no-cumplimientos se resuelven con el miembro apropiado del equipo de trabajo mientras es posible. De otro modo deberán elevarse a los niveles de la administración designados para recibir y actuar en función de estos y seguirlos hasta su resolución. En este proceso se hace necesario mantener actualizados a los stakeholders sobre los resultados de las evaluaciones y el estado de la calidad.

2.5.1 Aseguramiento de la calidad del proceso y el producto en el proyecto

Akademos

En el proyecto Akademos no existen estándares definidos para los productos, ni descripciones o procedimientos establecidos para los procesos. Por lo que no se llevan a cabo evaluaciones de la adherencia. Las evaluaciones efectuadas consisten en revisar la lista de requisitos funcionales y no funcionales del software, y chequear que el producto los cumpla. Los procesos ni siquiera están descritos, por tanto, no son evaluados. Las no-conformidades resultantes de estas evaluaciones se comunican al equipo de trabajo y su resolución es analizada con todo el

proyecto. Sólo algunas de estas evaluaciones se documentan y el documento que se genera es denominado Acta de la revisión.

En general se promueve la planificación y ejecución de estas actividades, aunque no está definido un plan de calidad, sino que junto a la planificación de entrega de un producto de software se planifica la respectiva evaluación. De esta forma ningún producto puede ser entregado antes de ser evaluado. El proyecto ha designado un grupo de calidad que se encarga solamente de las evaluaciones de los productos una vez terminados.

2.6 Gestión de configuración

El AP Gestión de configuración persigue los siguientes objetivos:

1. Establecer las líneas base.
2. Garantizar el mantenimiento de las líneas base a través del seguimiento y control de los cambios.
3. Establecer la integridad de las líneas base documentando y realizando auditorías a la configuración.

En las prácticas enunciadas a continuación para cumplir estos objetivos se utilizan las subprácticas como elemento de detalle para una mejor comprensión del contenido.

PE 1.1: Identificar los elementos de configuración.

PE 1.2: Establecer un sistema de gestión de configuración.

PE 1.3: Crear o liberar las líneas base.

Subprácticas.

1. Obtener autorización antes de crear o liberar líneas base.
2. Crear o liberar solamente las líneas base de los elementos de configuración.
3. Documentar la serie de elementos de configuración que están contenidos en una línea base.

4. Garantizar la disponibilidad de las líneas base.

PE 2.1: Seguir las solicitudes de cambios.

PE 2.2: Controlar los elementos de configuración.

PE 3.1: Establecer los registros de la gestión de configuración.

PE 3.2: Ejecutar auditorías a la configuración.

Subprácticas.

1. Evaluar la integridad de las líneas base.
2. Confirmar, según los registros, que los elementos de configuración fueron correctamente identificados.
3. Revisar la estructura y la integridad de los elementos en el sistema de gestión de configuración.
4. Confirmar que los elementos en el sistema de gestión de configuración son correctos y están completados.
5. Confirmar el seguimiento de los estándares y procedimientos aplicables a la gestión de configuración.
6. Seguir las acciones desde la auditoría hasta su clausura.

Un sistema de gestión de configuración debe gestionar múltiples niveles de control y almacenar y establecer las versiones archivadas de los elementos de configuración. Los elementos bajo la gestión de configuración pueden incluir especificaciones y documentos de interfaz que definen requerimientos del producto. Otros documentos, como resultados de pruebas, pueden estar incluidos también dependiendo de su implicación en la definición del producto. La selección de los productos para la gestión de configuración está basada en criterios establecidos durante la planificación.

El análisis de las solicitudes de cambio permite determinar el impacto del cambio sobre el producto y los productos relacionados, el presupuesto y el calendario. Por ese motivo, es muy

importante controlar los cambios sobre los elementos de configuración durante toda la vida del producto, obtener autorización antes de cambiar los elementos que estén bajo el sistema de gestión de configuración y chequear que los cambios no afecten su integridad o provoquen efectos no deseados en las líneas base.

2.6.1 La gestión de la configuración en el proyecto Akademos

El proyecto Akademos mantiene el código fuente bajo un sistema de gestión de configuración. (Vea en el Anexo 3 la interfaz del Sistema de Gestión de Configuración utilizado en el proyecto Akademos) Una política de confianza establecida entre los miembros del equipo, limita el acceso de los desarrolladores a los elementos bajo el sistema, o sea, un programador sólo podrá acceder al módulo en el cual trabaja. Esta política garantiza que los cambios a la configuración solamente sean efectuados por la persona autorizada, aunque en algunos casos los desarrolladores puedan llegar a acuerdos sobre los cambios. Además de tener la obligación de informar al resto del equipo cuando algún elemento haya sido modificado. Estas actividades y acciones no generan documento alguno.

Muy esporádicamente, y por decisión propia, un miembro de la dirección del proyecto efectúa una revisión a la configuración para determinar errores y depurarlos. Por lo que se podría decir que no está definido un procedimiento que permita controlar que el código fuente bajo el sistema de gestión de configuración no esté corrupto.

2.7 Medición y análisis

El AP de Medición y análisis persigue desarrollar y sostener una capacidad de medición que sea utilizada para apoyar las necesidades de información sobre la gestión. En este nivel se comienza a tratar el concepto de que “solo lo que se mide se puede controlar” (CMMI 2006). Sus objetivos, claramente definidos son los siguientes:

1. Alinear los objetivos y actividades de medición y análisis con los objetivos y necesidades de información identificados.
2. Proporcionar los resultados de las mediciones dirigidos a los objetivos y necesidades de información.

La información necesaria y los objetivos para la medición pueden extraerse de:

- Planes de proyecto.
- Monitoreo del desarrollo del proyecto.
- Entrevistas con los administradores u otras personas.
- Objetivos de gestión establecidos.
- Planes estratégicos y de negocio.
- Requerimientos formales u obligaciones impuestas por contrato.

Si la organización tiene problemas con los plazos de entrega, entonces este aspecto debe ser medido. Otros parámetros de definición pueden ser “lo que más preocupa a la dirección” o “lo que piden los clientes”. CMMI propone las siguientes prácticas:

PE 1.1: Establecer los objetivos de la medición.

PE 1.2: Especificar las métricas.

PE 1.3: Especificar los procedimientos para la colección y almacenamiento de los datos.

PE 1.4: Especificar los procedimientos para el análisis.

Una vez que se haya definido lo que es necesario medir, el proceso también pide que se defina la forma de medirlo. Para especificar cómo la información medible será analizada y reportada, deberán definirse los procedimientos necesarios para el análisis de los datos y la comunicación de los resultados a nivel administrativo.

Los resultados de las mediciones basadas en información real del proyecto pueden ayudar a monitorear la ejecución, cumplir con las obligaciones del contrato, tomar decisiones técnicas y habilitar acciones correctivas para llevar a cabo. Las prácticas correspondientes se enuncian a continuación:

PE 2.1: Colectar los datos para la medición.

PE 2.2: Analizar los datos de la medición.

PE 2.3: Almacenar los datos y los resultados.

PE 2.4: Comunicar los resultados.

La información almacenada puede estar compuesta por planes de medición, especificaciones de las métricas, datos colectados, reportes de los análisis y presentaciones. Todos estos datos deben ser revisados para asegurar su integridad y precisión, y almacenados según los procedimientos definidos. Para prevenir el uso inapropiado, el acceso a estos datos debe ser controlado.

Esta AP no está implementada en el proyecto Akademos.

2.8 Conclusiones

El análisis del proceso de desarrollo en el proyecto Akademos, guiado por las APs recomendadas por CMMI para desarrollar en el nivel 2 de madurez, permitió encontrar deficiencias en procesos claves para la gestión del proyecto.

Capítulo 3 Actividades para mejorar los procesos

3.1 Introducción

Este capítulo se deriva de los resultados obtenidos en el análisis del proceso de software del proyecto Akademos. En él se describe una propuesta de tareas que de aplicarse podrían propiciar una mejora de los procesos existentes basada en las prácticas específicas de CMMI y soportada en el Expediente del proyecto de la Dirección de Calidad de la UCI que utiliza estándares IEEE.

3.2 Tareas para la organización de los procesos

Para llevar a cabo las tareas que se proponen para los procesos, será necesario que el proyecto esté organizado en función de los roles participantes. En este epígrafe se describen las actividades necesarias para la organización del proyecto.

Tarea 1.1 Definir los roles a jugar por los integrantes del proyecto.

El líder de proyecto, auxiliándose de otros expertos, asigna las responsabilidades. Deberán aplicarse exámenes para determinar las competencias de los trabajadores, y encuestas para determinar sus preferencias. A partir de las preferencias y las competencias determinadas entonces se asignarán los roles. Como resultado de esta tarea se obtiene un documento de roles y responsabilidades. (Vea en el Anexo 3 el Documento de Roles y Responsabilidades) a continuación se mencionan algunos roles que intervienen en los procesos que se proponen.

Líder de proyecto: Es el encargado de toda la organización del proyecto y el personal, asegura la mitigación de los riesgos del proyecto, define las actividades del monitoreo, las mediciones, y el aseguramiento de la calidad.

Planificador: Es el encargado del mantenimiento de los planes asociados al proyecto, interviene en la planificación de las actividades concretas del desarrollo.

Responsable de la configuración: Es el encargado del mantenimiento del sistema de gestión de configuración.

Será necesario también definir un Grupo de calidad, integrado por al menos dos miembros que deben desempeñar los siguientes roles.

Responsable de calidad: Es el responsable del grupo de calidad. Participa en la definición y aplicación de los procesos de Monitoreo y control de proyecto, Medición y análisis y Aseguramiento de la calidad.

Revisor: Participa en las actividades de evaluación, supervisión y monitoreo de los procesos de Monitoreo y control de proyecto y Aseguramiento de la calidad.

Corrector: Analiza y propone acciones correctivas para los problemas encontrados en las actividades de control y aseguramiento de la calidad.

Analista de medición: Es el encargado de recolectar y analizar los datos de las mediciones, e informar los resultados.

Tarea 1.2 Garantizar un plan de entrenamiento para los integrantes del proyecto en cada uno de los procesos que serán implementados.

La organización, el líder de proyecto y otros expertos, planifican y ejecutan el entrenamiento a los integrantes del proyecto, en los roles que llevarán a cabo y en los procesos que serán implementados.

3.3 Planificación del proyecto y los procesos

Las actividades de Planificación descritas en este epígrafe sólo se usarán en caso que el proyecto decida iniciar un nuevo ciclo de desarrollo. La planificación está dividida en tres etapas, la primera determinará la viabilidad del proyecto, la segunda asegurará la organización del proyecto y la tercera el detalle de las tareas a realizar en el desarrollo. Antes de describir las tareas por etapas se hace necesario mencionar tres aspectos importantes.

Los requisitos del software, expresados en casos de uso, constituyen la base de las estimaciones, y los planes, por ende, cada vez que un requisito cambia, deben efectuarse los ajustes correspondientes en los planes del proyecto. Estos ajustes pueden ser incluir nuevas tareas derivadas del cambio, o reasignar tiempo y recursos a las tareas planificadas.

Otro punto a señalar es que los resultados del proceso de Medición y análisis son utilizados en la planificación, ya sea porque se tiene en cuenta la necesidad de replanificar si los parámetros del proyecto analizados lo requieren, o porque el proceso en sí utiliza métricas, como una de las que se propone utilizar para priorizar los riesgos a la hora de planificar las tareas de mitigación de riesgos.

Además, la planificación es un proceso engorroso pero necesario, por esta razón debe estudiarse una herramienta que permita efectuar la planificación del proyecto de forma automatizada.

1era Etapa. Esta etapa de planificación está basada en los riesgos de llevar a cabo el proyecto y el alcance, y necesita de la aprobación de todos los participantes en el proyecto. Debe efectuarse después de haber concluido la modelación del negocio y la captura de requisitos del software.

Tarea 2.1 Descripción del alcance del proyecto.

El líder de proyecto describe las tareas que se llevarán a cabo durante el desarrollo, y define los puntos que determinarán su finalización. En esta actividad se genera el documento Alcance del proyecto.

Tarea 2.2 Identificación de los riesgos.

El líder de proyecto identifica los riesgos que pueden afectar al desarrollo del proyecto y los documenta en una lista de riesgos, con su impacto y probabilidad de ocurrencia. En esta actividad se genera la Lista de Riesgos.

Tarea 2.3 Aprobación de la ejecución del proyecto.

El líder de proyecto presenta a los trabajadores y stakeholders el alcance y los riesgos de llevar a cabo el proyecto, quienes aprueban o no su ejecución.

2da Etapa. Después de la aprobación de los involucrados en la ejecución del proyecto se efectúa la segunda etapa, correspondiente a la planificación inicial. Las tareas se enumeran a continuación.

Tarea 2.4 Estimar el esfuerzo, coste y tiempo de desarrollo del proyecto.

El líder de proyecto realiza las estimaciones del esfuerzo, coste y tiempo de desarrollo del proyecto usando los métodos adecuados.

Para efectuar las estimaciones del esfuerzo el proyecto debe utilizar la técnica del Análisis de los Puntos de Función y el método COCOMO II. El Análisis de los Puntos de Función permite cuantificar el tamaño de un sistema en unidades independientes denominadas Puntos de Función. A partir de los cuales se estima el esfuerzo usando COCOMO II, que es el método más utilizado en la actualidad cuando no se tiene información histórica a la cual recurrir. Consiste básicamente en la aplicación de ecuaciones matemáticas sobre los Puntos de Función sin ajustar

o la cantidad de líneas de código estimados para un proyecto. Los Puntos de Función se obtienen después de la captura de los requisitos del software y a partir de los CU. Usando el tamaño y el esfuerzo calculados, el proyecto podrá estimar el coste y tiempo necesarios para llevar a cabo el desarrollo.

Tarea 2.5 Establecer el presupuesto y el calendario del proyecto.

Usando las estimaciones, el líder de proyecto establece el presupuesto y el calendario del proyecto.

Tarea 2.6 Definición del ciclo de vida.

El líder de proyecto define el ciclo de vida del proyecto a partir de las tareas que implica el desarrollo de los requisitos del software, sus fases y las iteraciones necesarias.

Tarea 2.7 Planificar la mitigación de los riesgos.

El líder de proyecto, evalúa el impacto, la probabilidad de ocurrencia y prioriza los riesgos del proyecto. Según la prioridad planifica las tareas para evitarlos, mitigarlos y de contingencia en caso que estos ocurran. Los resultados de esta actividad se documentan en el Plan de mitigación de riesgos. (Vea en el Anexo 5 el Plan de mitigación de riesgos).

Tarea 2.8 Planificar la gestión de los datos.

El líder de proyecto identifica los recursos de datos, determina los que será necesario controlar y planifica las actividades orientadas al almacenamiento y las políticas de acceso. Los resultados de esta actividad se documentan en el Plan de Gestión de la configuración. (Vea en el Anexo 6 el Plan de Gestión de la configuración).

Tarea 2.9 Planificar los recursos.

El líder de proyecto determina cuál es el equipamiento, el personal y los procesos disponibles para llevar a cabo el proyecto. Luego los distribuye según las actividades a desarrollar.

Tarea 2.10 Planificar las habilidades y conocimientos.

El líder de proyecto analiza cuáles son los conocimientos y habilidades disponibles entre los miembros del proyecto y planifica las tareas para la adquisición de los necesarios para llevar a cabo el proyecto.

Tarea 2.11 Planificar la implicación de los stakeholders.

El líder de proyecto identifica los stakeholders relevantes y planifica las interacciones que estos deben mantener con el proyecto.

Tarea 2.12 Definir el proceso de Monitoreo y control del proyecto.

El líder de proyecto, con los stakeholders y el responsable de calidad, definen cuáles son los puntos significativos del proyecto que deben ser monitoreados.

Algunos de estos puntos significativos pueden ser:

- Monitorear el estado del proyecto
- Monitorear los riesgos
- Monitorear los conocimientos y habilidades del personal
- Monitorear atributos de las tareas y productos de trabajo
- Monitorear la gestión de los datos
- Monitorear los recursos

Tarea 2.13 Planificar el proceso de Medición y análisis.

El líder de proyecto y el responsable de calidad, definen las necesidades de medición del proyecto y planifican las mediciones. Como resultado de esta actividad se obtiene el Plan de Mediciones. (Vea en el Anexo 7 el Plan de Mediciones)

Tarea 2.14 Planificar el proceso de Aseguramiento de la calidad.

El líder de proyecto y el responsable de calidad, planifican los momentos en que se efectuarán las evaluaciones a los productos y los procesos durante el desarrollo. Como resultado de esta tarea se genera el Plan de Aseguramiento de la calidad. (Vea en el Anexo 8 el Plan de Aseguramiento de la calidad)

Tarea 2.15 Obtener los compromisos con el Plan de proyecto.

En reunión con todos los integrantes del proyecto y los stakeholders se acuerda la ejecución del plan trazado. El proyecto debe estar seguro de poder llevar a cabo el desarrollo dentro del coste y tiempo establecidos.

Como resultado de las actividades de las dos primeras etapas se obtiene el Plan de desarrollo del proyecto, que contiene toda la información recolectada y referencia los planes efectuados. (Vea en el Anexo 9 el Plan de Desarrollo)

3era Etapa. Al inicio de cada iteración se debe revisar y actualizar la planificación inicial. En esta revisión se especifican las actividades concretas a realizar.

Tarea 2.16 Planificación de la iteración.

El líder de proyecto, el planificador y los stakeholders se reúnen para planificar las actividades correspondientes a la iteración que se inicia, actualizar la planificación general y los cronogramas de las tareas. En esta actividad se genera el Plan de Iteración.

3.4 Aseguramiento de la calidad

La planificación de este proceso se lleva a cabo durante la segunda etapa de la planificación del proyecto.

Uno de los objetivos que se pretende lograr con el Aseguramiento de la calidad consiste en garantizar la existencia de los estándares y los procedimientos que establecen los métodos para desarrollar el software. Por tanto, son estas las primeras tareas propuestas.

3.4.1 Definición de estándares

Tarea 3.1 Definir los estándares para los productos.

Los estándares son criterios previamente establecidos contra los cuales se comparan los productos en las evaluaciones de la calidad. El líder de proyecto y los desarrolladores definen los estándares para los productos.

Algunos tipos de estándares que pueden ser definidos para el proyecto son:

- Estándares de documentación: especifican la forma y el contenido para la planificación, el control, y la documentación del producto.
- Estándares de código: especifican la lengua en la cual el código debe ser escrito y definen cualquier restricción en el uso de las características de la lengua. Definen las estructuras de la lengua legal, las convenciones del estilo, las reglas para las estructuras de datos y las interfaces, así como la documentación interna del código.

3.4.2 Procedimientos para los procesos

Teniendo en cuenta que Akademos no es un proyecto que se inicia, o sea, ya tiene implementado un número de procesos, el proyecto deberá documentar sus procedimientos. A continuación se propone una serie de tareas que deberán incluirse en estos procedimientos.

Tarea 3.2 Definir los procesos y sus procedimientos.

Los procedimientos son los criterios contra los cuales se comparan los procesos de desarrollo y de control. El líder de proyecto define los procesos que deberán implementarse en el proyecto y describe los procedimientos o pasos explícitos que se seguirán para realizar cada uno de ellos auxiliándose de expertos en estos procesos.

Los procesos que deben ser implementados y que se describen en esta estrategia son:

1. Planificación del proyecto
2. Aseguramiento de la calidad
3. Monitoreo y control del proyecto
4. Medición y análisis
5. Gestión de la configuración
6. Gestión de requisitos

3.4.3 Aseguramiento de la calidad del proceso

Tarea 3.3 Supervisar los procesos.

La supervisión debe extenderse a todos los procesos implementados en el proyecto y consiste en efectuar un monitoreo para asegurar que se están siguiendo los pasos adecuados. En el monitoreo, el revisor compara los pasos reales realizados con los que están en los procedimientos documentados. De esta supervisión se obtiene un documento de no-conformidades. (Vea en el Anexo 10 el Documento de No-Conformidades)

El propio proceso de Aseguramiento de la calidad también debe supervisarse, es una manera de garantizar que se están siguiendo los procedimientos apropiados de control y que se mantiene la documentación requerida.

3.4.4 Aseguramiento de la calidad del producto

Tarea 4.2 Evaluar los productos.

Estas evaluaciones garantizan que se están siguiendo los estándares respectivos, son efectuadas por el revisor y pueden ser pruebas o revisiones técnicas. Incluyen todos los productos resultantes de los procesos. De estas evaluaciones se obtiene un documento de no-conformidades. (Vea en el anexo 10 el Documento de No-Conformidades)

Los primeros productos para evaluar deberán ser los propios estándares y procedimientos definidos para el proyecto.

Tarea 4.3 Comunicar y resolver los no-cumplimientos

El revisor comunica los no-cumplimientos y los resuelve con el miembro apropiado del proyecto. En caso que este no pueda dar solución, el revisor deberá comunicarlos a los niveles superiores de la dirección del proyecto hasta su resolución.

3.5 Monitoreo y control del proyecto

Para el proceso de Monitoreo y control se recomiendan tres actividades fundamentales, planificación, para la cual se tienen en cuenta los puntos significativos definidos durante la segunda etapa de planificación del proyecto; revisiones y acciones correctivas.

Tarea 5.1 Planificación

Consiste en la elaboración del Plan de monitoreo y control del proyecto por el responsable de calidad. Este plan contiene, entre otros elementos, las revisiones que se realizarán, en dependencia de los puntos significativos escogidos para monitorear. (Vea en el Anexo 9 el Plan de Desarrollo)

Algunas de las revisiones y los aspectos que se miden en ellas pueden ser:

1. Revisiones del estado del proyecto
 - Atrasos en las actividades
 - Adelantos en las actividades
 - Estado de las actividades en desarrollo
 - Estado del cumplimiento del proyecto
2. Control de los riesgos
 - Estado de riesgo actuales del proyecto
 - Efecto de los riesgos para el proyecto
 - Riesgos eliminados
 - Nuevos riesgos detectados
3. Monitoreo de los conocimientos y habilidades
 - Conocimientos adquiridos en el período
 - Estado de la adquisición de conocimientos
4. Revisiones a las tareas
 - Estado de las tareas
 - Responsable de las tareas
 - Resultados y productos esperados
 - Resultados y productos obtenidos
5. Control de los datos
 - Cumplimiento del plan de gestión de configuración
 - Problemas de los elementos de configuración
6. Monitoreo de los recursos
 - Recursos disponibles
 - Utilización de los recursos asignados al personal

Tarea 5.2 Efectuar revisiones.

El revisor realiza las revisiones para detectar las desviaciones significativas y los problemas existentes, valora el impacto de estos problemas y asigna un orden de prioridad por el que se deben ir corrigiendo.

Tarea 5.3 Acciones correctivas

El corrector debe realizar un análisis de los problemas identificados y desviaciones significativas teniendo en cuenta su nivel de importancia para proponer las acciones correctivas apropiadas. Estas decisiones deben ser presentadas al grupo de calidad para su aprobación.

El proceso de Medición y análisis es un proceso de apoyo a las actividades de Monitoreo y control del proyecto, por lo que en este último deberán usarse las capacidades de medición creadas en el proyecto.

3.6 Medición y análisis

A continuación se enumeran las tareas para la implementación del proceso de Medición y análisis.

Tarea 6.1 Identificación de las necesidades de medición.

El líder de proyecto y el responsable de calidad definen los objetivos que deben cumplir las mediciones, e identifican a partir de los planes del proyecto y las actividades de monitoreo y control del desarrollo los elementos que deben ser medidos.

Tarea 6.2 Definición de métricas.

El responsable de calidad recolecta o define las métricas para satisfacer las necesidades de medición del proyecto.

Tarea 6.3 Definición de los procedimientos para llevar a cabo las mediciones.

El responsable de calidad define:

- los mecanismos para recolectar los datos que serán usados en las mediciones,
- los mecanismos de almacenamiento de los datos resultantes de las mediciones,
- los métodos de información de los resultados de las mediciones a los interesados.

Los resultados de estas actividades se registran en el Plan de Mediciones. (Vea en el Anexo 7 el Plan de Mediciones).

A continuación se relacionan algunas métricas que servirán para el monitoreo del proyecto.

Desviación del tiempo (D)

La métrica D será usada para determinar la desviación existente del tiempo real respecto al tiempo planificado. Este resultado servirá para decidir si es necesario replanificar.

$$TD = TR + TP$$

$$D = TD - TDI$$

Donde

D: Desviación del tiempo total de desarrollo respecto al tiempo total planificado.

TD: Tiempo total de desarrollo.

TR: Tiempo usado de desarrollo del proyecto hasta el momento.

TP: Tiempo planificado para las actividades por realizar en el proyecto.

TDI: Tiempo total planificado inicialmente.

El objetivo es obtener $D = 0$, lo que significaría que se está cumpliendo con las actividades en el tiempo planificado.

Si $D > 0$ entonces es necesario replanificar porque la duración de las actividades planificadas ha excedido el tiempo planificado.

Si $D < 0$ significa que se tiene holgura en el tiempo de desarrollo, es decir, el tiempo en que se cumplieron las actividades fue menor al planificado y esta diferencia de tiempo se le puede dedicar a otras actividades a tenerla de reserva por si ocurre algún retraso.

La métrica D debe usarse en el proceso de Monitoreo y control para monitorear el estado del proyecto.

Severidad (S)

La métrica S será usada para determinar cuán problemático puede ser un riesgo y otorgar un nivel de prioridad a su mitigación.

$$S = I * P$$

Donde:

S: Severidad o exposición al riesgo.

I: Impacto sobre el proyecto en una escala de 1 a 5 teniendo en cuenta el valor de P.

P: Probabilidad de que ocurra.

Mientras mayor sea la severidad, mayor es la prioridad que se le da a cada riesgo.

La métrica S debe usarse en la 2da etapa de Planificación para priorizar los riesgos.

Tarea 6.4 Efectuar mediciones.

El analista de medición recolecta y analiza los datos de las mediciones.

Tarea 6.5 Informar los resultados.

El analista de medición divulga los resultados de las mediciones a los interesados.

3.7 Gestión de la configuración

En el proceso de Gestión de la configuración deben cumplirse las siguientes tareas:

Tarea 7.1 Definición de los elementos de la configuración.

El responsable de la configuración define, de los recursos de datos del proyecto, cuáles será necesario mantener bajo el sistema de gestión de configuración, y determina las relaciones entre estos elementos.

Por su importancia para la estabilidad de las planificaciones debe mantenerse bajo el sistema de gestión de configuración la lista de los requisitos del software acordados.

Tarea 7.2 Establecimiento de las líneas base.

El responsable de la configuración, el líder de proyecto y los desarrolladores establecen las líneas base de los elementos de configuración.

Tarea 7.3 Establecimiento de una política para controlar los cambios.

El responsable de la configuración define los niveles de acceso y los procedimientos que deberán llevarse a cabo para modificar la línea base y los elementos de configuración. Estos elementos se registran en el Plan de Gestión de la configuración. (Vea en el Anexo 6 el Plan de Gestión de la configuración)

Como elemento de control, debe existir una política para limitar el acceso a la línea base, de manera que esta sólo pueda ser modificada con análisis y autorización previa. Si un elemento de la línea base es modificado deben analizarse cuáles son sus dependencias y proceder a la actualización de los elementos relacionados para garantizar así el mantenimiento de la integridad de la configuración.

Tarea 7.4 Controlar los cambios.

El responsable de la configuración autoriza el acceso a la línea base y los elementos de configuración para efectuar los cambios. Después de efectuado, revisa que todos los elementos relacionados hayan sido actualizados también.

Tarea 7.5 Registrar las acciones efectuadas sobre la configuración.

El responsable de la configuración mantiene un registro actualizado de los cambios efectuados sobre los elementos de configuración y otras acciones.

Tarea 7.6 Efectuar auditorías a la configuración.

Estas auditorías forman parte de las actividades de Aseguramiento de la calidad del producto. El revisor supervisa la integridad de las líneas base y que los elementos de configuración estén correctos y completados.

3.8 Gestión de requisitos

La Gestión de requisitos incluye actividades de planificación, gestión de cambios y las evaluaciones de los planes y los productos contra los requisitos.

Tarea 8.1 Planificación.

El líder de proyecto y el responsable de calidad definen las políticas y los procedimientos que se usarán para asegurar el entendimiento de los requisitos por parte de los stakeholders y los trabajadores del proyecto, y planifican las revisiones de los planes y productos contra los requisitos. El líder de proyecto y el responsable de la configuración establecen un mecanismo de gestión de los cambios a los requisitos. Como resultado de esta actividad se obtiene el Plan de Gestión de los requisitos. (Vea en el Anexo 11 el Plan de Gestión de los requisitos).

Tarea 8.2 Gestión de cambios.

El responsable de la configuración mantiene la línea base de los requisitos, autoriza el acceso a la línea base y revisa que los cambios hayan sido efectuados de manera correcta.

Siendo esta la actividad más importante de la Gestión de requisitos por la repercusión que estos pueden tener en las planificaciones del proyecto y los procesos. De ella se derivan algunas actividades como el establecimiento de la línea base de los requisitos, el mantenimiento de la trazabilidad entre las dependencias y el control de los cambios, que se realizan como parte del proceso de Gestión de la configuración.

La línea base está constituida por los requisitos que ya fueron analizados, evaluados, comprendidos y acordados por el cliente y los usuarios, conjuntamente con los participantes del proyecto, y de la cual se han determinado las dependencias.

Los cambios a los requisitos, además de ser implementados en los productos involucrados, deben ser actualizados en la línea base de los requisitos siguiendo las políticas establecidas por el proceso de Gestión de configuración.

Tarea 8.3 Revisar los planes y productos de trabajo contra los requisitos.

Estas revisiones consisten en chequear que los planes y los productos generados concuerden con los requisitos. Son efectuadas por el revisor y de ellas se genera un documento de no-conformidades. (Vea en el Anexo 10 el Documento de No-Conformidades)

3.9 Conclusiones

Las tareas propuestas en este capítulo incluyen gran parte de las prácticas de CMMI que no se cumplen en los procesos implementados en el proyecto Akademos. Están referidas sólo a los

procesos que intervienen en el nivel 2 de madurez y están orientadas a la mejora de la calidad del proceso de desarrollo en general.

Conclusiones generales

Como resultado del estudio efectuado de normas y estándares de calidad de software, y del análisis del proceso de desarrollo de software en el proyecto Akademos se concluye con lo siguiente:

- El modelo CMMI se ajusta a las características del proyecto Akademos y de la UCI en general.
- La utilización del modelo CMMI en el análisis del proceso de desarrollo de software en el proyecto Akademos arrojó las deficiencias existentes en el mismo.

En la estrategia planteada quedaron definidas las actividades a ejecutar para elevar la calidad del proceso de desarrollo de software en el proyecto Akademos.

Recomendaciones

- Iniciar un nuevo ciclo de desarrollo en el proyecto Akademos que permita la implementación de la estrategia propuesta.
- Generalizar el estudio realizado en este trabajo a otros proyectos de la UCI.

Bibliografía

ASSOCIATION, I. S. News and Information 2006. [Disponible en: <http://standards.ieee.org/stdsdev/index.html>]

CABRERA, Y. V. Akademos: Sistema Automatizado para la Gestión Académica, Subsistema Gestión de Seguridad. Dirección de Informatización. Ciudad de La Habana, Universidad de las Ciencias Informáticas, 2005. p.

CMMI, P. T. CMMI For Development, Version 1.2. 2006. p.

University of Southern California, Center for Software Engineering, COCOMO II. [Disponible en <http://sunset.usc.edu>]

COLLERA, T. A. Akademos: Sistema Automatizado para la Gestión Académica, Módulo de Control Docente. Facultad de Ingeniería Industrial. Ciudad de La Habana, Instituto Superior Politécnico José Antonio Echeverría, 2005. p.

GARCÍA, J. CMM y CMMI, 2005. [Disponible en: <http://www.ingenierosoftware.com/calidad/cmm-cmmi.php>]

HUMPHREY, W. Introduction to the Personal Software Process. 1997. p.

HUMPHREY, W. S. Introduction to the Team Software Process. 1999. p.

INSIGHT. Implementación de tecnologías Rational 2007. [Disponible en: http://www.insight.com.uy/casos_implementacion_de_tecnologias_rational.html]

J.B.R. El Proceso Unificado de Desarrollo de Software. 1999. p.

KAN, S. H.; V. R. BASILI, et al. Software quality: an overview from the perspective of total quality management. 1994

MARÍN, Y. D. C. Akademos: Sistema Automatizado para la Gestión Académica, Módulo de Gestión de Profesores. Facultad de Ingeniería Industrial. Ciudad de La Habana, Instituto Superior Politécnico José Antonio Echevarría, 2005. p.

MEDINA, D. D. A. Akademos: Sistema Automatizado para la Gestión Académica, Módulo de Diseño y Distribución de Reportes. Facultad de Ingeniería Industrial. Ciudad de La Habana, Instituto Superior Politécnico José Antonio Echeverría, 2005. p.

NAVEGAPOLIS.NET. Sinopsis de los modelos CMM y CMMI, 2006. [Disponible en: <http://www.navegapolis.net/>

NOA, V. C. Akademos: Sistema Automatizado para la Gestión Académica, Subsistema Expediente Virtual. Facultad de Matemática- Computación. Santiago de Cuba, Universidad de Oriente, 2004. p.

PALACIO, J. Gestión y Procesos en empresas de software, 2005. [Disponible en: www.navegapolis.net

PERAL, C. H. G. L. E. Akademos: Sistema automatizado para la gestión académica. Subsistema de Gestión de Matrícula. Facultad de Matemática- Computación. Ciudad de La Habana, Universidad de La Habana, 2004. p.

Pino, F. J., Garcia, F., Ruiz, Piattini, F. M. Adaptación de las normas ISO/IEC 12207:2002 e ISO/IEC 15504:2003 para la evaluación de la madurez de procesos software en países en desarrollo, 2006. [Disponible en: <http://www.ewh.ieee.org>]

PRESSMAN, R. S. Ingeniería del Software. Un enfoque práctico. 5ta. 2002. p.

RISCO, A. D. Akademos: Sistema Automatizado para la Gestión Académica, Subsistema Generador de Reportes. Facultad de Matemática -Computación. Ciudad de La Habana, Universidad de La Habana, 2004. p.

ROJAS, D. A. G. Akademos: Sistema Automatizado para la Gestión Académica. Subsistema de Registro Virtual. Facultad de Matemática- Computación. Ciudad de La Habana, Universidad de La Habana, 2004. p.

ROSALES, Y. Z. Akademos: Sistema Automatizado para la Gestión Académica, Subsistema Planes de Estudio. Facultad de Matemática- Computación. Ciudad de La Habana, Universidad de La Habana, 2004. p.

RUP. Ayuda de Rational Unified Process. 2003. p.

TYRRELL, S. The many dimensions of the Software Process, 2004. [Disponible en: www.acm.org/crossroads/xrds6-4/software.html]

UCI. Introducción a la Ingeniería de Software. Conferencia 1, Dpto de Ingeniería y Gestión de Software, 2005.

WIKIPEDIA. Capability Maturity Model Integration, 2007. [Disponible en: http://en.wikipedia.org/wiki/Capability_Maturity_Model_Integration#Evaluation]

WIKIPEDIA, ISO/IEC 12207, 2007 [Disponible en: http://es.wikipedia.org/wiki/ISO/IEC_12207]

Glosario

Área de proceso: Conjunto de prácticas relacionadas, que ejecutadas colectivamente, satisfacen un conjunto de metas consideradas importantes para hacer mejoras significativas en esa área.

Práctica específica: Es una actividad que es considerada importante en la meta específica asociada. Describe las actividades esperadas para conseguir las metas específicas de un área de procesos.

Proceso: Un conjunto de actividades interrelacionadas que transforma las entradas en salidas.

Stakeholder: Involucrados con el proyecto, puede utilizarse para referirse a los clientes y proveedores.

Anexos

Anexo 1

Prototipo de interfaz de usuario.

Caso de Uso: Actualizar grupo académico. Sección “Adicionar Estudiante”

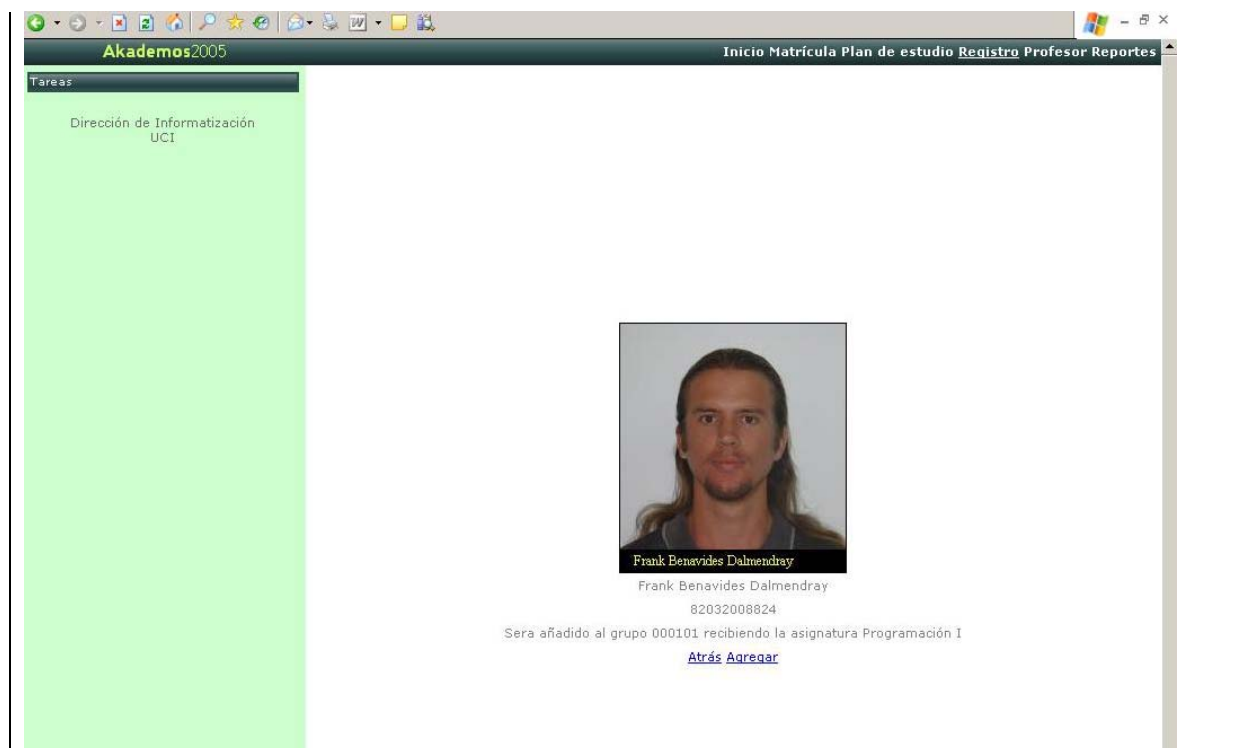
Prototipo de interfaz de usuario 1

The screenshot displays the 'Akademos2005' web application interface. The browser window title is 'Inicio Matrícula Plan de estudio Registro Profesor Reportes'. The main content area is titled 'RESULTADOS DE LA BÚSQUEDA' and shows 'Página 1 de 1'. A search bar on the left contains the text 'frank' and a 'Buscar' button. Below the search bar is a section for 'Estructuras' (Facultad 1) with a grid of icons representing different academic structures. At the bottom left, there is a legend for 'Estados' (Status) with color-coded boxes for: Prematriculado (blue), Matriculado (black), Baja (orange), Baja académica (green), Egresado (dark blue), Eliminado (yellow), Repitente (red), En ingreso (dark green), Licencia (dark red), and Licencia Mat. (purple). The search results table on the right contains the following data:

CI	Nombre	Grupo	
82032008824	Frank Benavides Dalmendray	01301	+
82021006843	Frank Puig Placeres	01402	+
83071325628	Frank Yunier Estevez Rodriguez	01206	+
86072306187	FRANKIEL HERNÁNDEZ DEL PINO	01104	+
84110111901	Franklin Yannier Garmendia Lopez	01202	+

Dirección de Informatización
UCI

Prototipo de interfaz de usuario 2



Anexo 2

Plan efectuado en el proyecto Akademos, Tareas Semana del 15-17 de enero de 2007.

Tarea	Fecha	Responsable	Chequea
Entregar casos de prueba de los módulos de Reporte y Matrícula	Martes 16 10:00pm	Responsables de los módulos Matrícula y Reportes	Líder de proyecto y Planificador
Prueba de los módulos Matrícula y Reportes	Miércoles 17 8:30am	Grupo de desarrollo	Líder de proyecto
Probar las ayudas publicadas	Miércoles 17 10:00pm	Responsables de los módulos	Líder de proyecto y

					Arquitecto
Reunión con los estudiantes de tercer año	Jueves 18 a las 9:30am	Líder de proyecto, Planificador y Arquitecto	Líder de proyecto y Arquitecto		
Cierre de las clases a documentar	Viernes 19 9:00am	Todos	Responsable de calidad y Planificador		

Anexo 3

Interfaz del sistema de gestión de configuración utilizado por el proyecto Akademos.

The screenshot shows the FORGE UCI web interface. At the top, there is a navigation bar with links for INICIO, MI PÁGINA, PROYECTOS, CÓDIGO, and AYUDA. Below this is a secondary navigation bar with links for RESUMEN, FOROS, REGISTRO, LISTAS, TAREAS, DOCUMENTOS, ENCUESTAS, NOTICIAS, SCM, and FICHEROS. The main content area displays a file list with the following columns: File, Rev., Age, Author, and Last log entry. The file list includes various folders and files, such as CU/, WinApp/, accesodatos/, akademos/, ayuda/, cnegocio/, componentes/, cursos/, doc/, estudiante/, gaUtiles/, matricula/, planestudio/, profesor/, registro/, reportes/, seguridad/, servicios/, akademos.sln, akademos.suo, and default.aspx. At the bottom of the interface, there is a message: "No admin address has been configured" and "Powered by ViewVC 1.0.3". There are also logos for FORGE, W3C, and XHTML 1.0.

File	Rev.	Age	Author	Last log entry
CU/	971	2 months	pepe	
WinApp/	1054	5 weeks	fbenavides	
accesodatos/	1075	2 weeks	nmino	
akademoss/	919	2 months	fbenavides	
ayuda/	769	3 months	pepe	
cnegocio/	1072	3 weeks	fbenavides	
componentes/	697	5 months	dceperor	Subida Inicial
cursos/	923	2 months	pepe	
doc/	697	5 months	dceperor	Subida Inicial
estudiante/	1073	2 weeks	nmino	
gaUtiles/	1064	3 weeks	fbenavides	
matricula/	1071	3 weeks	fbenavides	
planestudio/	697	5 months	dceperor	Subida Inicial
profesor/	697	5 months	dceperor	Subida Inicial
registro/	1069	3 weeks	fbenavides	
reportes/	992	2 months	dsantiler	
seguridad/	1052	6 weeks	pepe	
servicios/	983	2 months	obarriel	
akademoss.sln	847	3 months	obarriel	
akademoss.suo	697	5 months	dceperor	Subida Inicial
default.aspx	697	5 months	dceperor	Subida Inicial

Anexo 4

Documento de Roles y Responsabilidades

	Roles y responsabilidades
	<Proyecto>
<Versión>	
Roles y responsabilidades	

Rol	Responsabilidad

Equipos de trabajo

Fase:

Equipo:

Nombre	Rol

Anexo 5

Plan de mitigación de riesgos

Riesgo	Tipo de Riesgo	Impacto	Descripción	Probabilidad	Efectos

Plan de Mitigación de Riesgos
<Nombre del Proyecto>

<Versión>

1. Introducción
 - 1.1 Alcance
 - 1.2 Definiciones, acrónimos y abreviaturas
 - 1.3 Referencias
2. Riesgos
 - 2.1 <Identificador de riesgo—un nombre o número descriptivo>
 - 2.1.1 Indicadores
 - 2.1.2 Estrategia de Mitigación
 - 2.1.3 Plan de Contingencia

2.2 <Próximo Identificador de riesgo—un nombre o número descriptivo>

3. Gestión de Riesgos

Riesgo	Probabilidad	Impacto	Mitigación del riesgo	Monitoreo del riesgo	Administración del riesgo

4. Tareas para la Gestión de Riesgos

5. Organización y Responsabilidades

6. Presupuesto

7. Herramientas y Técnicas

8. Elementos de Riesgos a Gestionar

Anexo 6

Plan de gestión de la configuración

Plan de Gestión de Configuración

IEEE 828-1998

<Nombre del Proyecto>

<Versión>

1. Introducción

1.1 Propósito

1.2 Alcance

1.3 Definiciones, acrónimos y abreviaturas

1.4 Referencias

1.5 Resumen

2. Gestión de Configuración de Software

2.1 Organización de la Gestión de Configuración de Software

- 2.2 Responsabilidades
- 2.3 Relación de la Gestión de Configuración con el ciclo de vida del proyecto
 - 2.3.1 Interfaces con otras organizaciones dentro del proyecto
 - 2.3.2 Responsabilidades con otras organizaciones dentro del proyecto
- 3. Actividades de Gestión de Configuración de Software
 - 3.1 Identificación de la configuración
 - 3.1.1 Especificación de la identificación
 - 3.1.2 Identificación para el formulario de control de cambios
 - 3.1.3 Líneas base del proyecto
 - 3.1.4 Bibliotecas
 - 3.2 Control de la configuración
 - 3.2.1 Procedimientos para cambiar una línea base
 - 3.2.2 Procedimiento para procesar pedidos de cambios y su aceptación
 - 3.2.3 Comité de Control de Cambios
 - 3.2.4 Revisión de documentos
 - 3.2.5 Herramientas automatizadas para el Control de Cambios
 - 3.3 Estado de la configuración
 - 3.3.1 Almacenamiento, manipulación y entregables del proyecto
 - 3.3.2 Reportes
 - 3.3.3 Proceso de entregas
 - 3.4 Auditorías a la configuración
 - 3.4.1 Número de auditorías a realizar y cuándo serán llevadas a cabo
- 4. Hitos
- 5. Entrenamiento

Anexo 7

Plan de Mediciones

Plan de Mediciones

<Proyecto>

<Versión>

1. Introducción

1.1 Propósito

1.2 Alcance

1.3 Definiciones, Acrónimos y Abreviaturas

1.4 Referencias

1.5 Resumen

2. Gestión de las metas y submetas

3. Métricas

Categoría	Conceptos a medir	Métrica

3.1 Plantilla para cada métrica

Nombre	
Definición	
Metas	
Procedimiento de análisis	
Responsabilidades	

4. Métricas Primitivas

4.1 Plantilla para las métricas primitivas:

Nombre	
Definición	
Procedimiento de recolección	
Responsabilidades	

5. Anexos

Anexo 8

Plan de Aseguramiento de la calidad

Plan de Aseguramiento de la Calidad

IEEE 730-1998

<Proyecto>

<Versión>

1. Introducción

1.1 Propósito

1.2 Alcance

1.3 Definiciones, Acrónimos y Abreviaturas

1.4 Referencias

1.5 Resumen

2. Objetivos de Calidad

3. Gestión

3.1 Organización

3.2 Tareas y responsabilidades

Tarea de Aseguramiento de calidad	Precondición Al finalizar la fase:	Poscondición Antes de la fase:	Responsable	Comentarios

4. Documentación

5. Métricas

6. Estándares y Guías

Estándar	Ubicación	Comentarios			
7. Plan de Revisiones y Auditorías					
7.1 Tareas generales de Revisiones y Auditorías					
7.2 Cronograma					
7.3 Organización y Responsabilidades					
7.4 Resolución de problemas y actividades de corrección					
7.5 Herramientas, técnicas y Metodologías					
No. de Revisión	Tipo	Objetivos	Descripción (Iteración)	Fase del Proyecto	Responsable
8. Pruebas y Evaluación					
9. Herramientas, Técnicas y Metodologías					
10. Resolución de Problema y Acción Correctiva					
11. Gestión de Configuración					
13. Entrenamiento					

Anexo 9

Plan de Desarrollo

Plan de Desarrollo de Software
<Proyecto>
<Versión>
1 Introducción
1.1 Alcance

- 1.2 Definiciones, Acrónimos y Abreviaturas
- 1.3 Referencias
- 2. Visión del Proyecto
 - 2.1 Propósito, alcance y objetivos del proyecto
 - 2.2 Suposiciones y Restricciones
 - 2.3 Entregables del Proyecto
 - 2.4 Evolución del Plan de Desarrollo de Software
- 3. Organización del Proyecto
 - 3.1 Estructura Organizativa
 - 3.2 Interfaces Externas
 - 3.3 Roles y Responsabilidades

Rol	Responsabilidad

- 4. Procesos de Control
 - 4.1 Estimación del Proyecto
 - 4.2 Plan del Proyecto
 - 4.2.1 Plan de la fase

Tabla: Fases del proyecto y sus iteraciones

Fase	Número de Iteraciones	Duración

Tabla: Hitos de las Fases

Fase	Hito
Inicio	
Elaboración	
Construcción	
Transición	

4.2.2 Objetivos de las Iteraciones

4.2.3 Liberaciones

4.2.4 Cronograma del Proyecto

4.2.5 Recursos del Proyecto

4.2.5.1 Plan de selección del personal

4.2.5.2 Plan de capacitación

4.2.6 Planes de Iteraciones

Fase	Iteración	Casos de Uso a describir	Justificación	Observaciones

4.3 Monitoreo y control del proyecto

4.3.1 Plan de Gestión de Requisitos

4.3.2 Plan de control del cronograma

4.3.3 Plan de Control del Presupuesto

4.3.4 Plan de Control de Calidad

4.3.5 Plan de Reportes

Reporte	Frecuencia	A quien	Fichero	Observaciones

4.3.6 Plan de Mediciones

4.4 Plan de control de riesgos

- 5. Planes Técnicos
 - 5.1 Casos de desarrollo
 - 5.2 Métodos, herramientas y técnicas
 - 5.3 Plan de la Infraestructura
 - 5.4 Plan de aceptación del proyecto
- 6. Planes de procesos de soporte
 - 6.1 Plan de Gestión de Configuración
 - 6.2 Plan de evaluación
 - 6.3 Plan de documentación
 - 6.4 Plan de aseguramiento de la calidad
 - 6.5 Plan de resolución de problemas
 - 6.6 Plan de gestión de subcontratación
- 7. Planes adicionales
- 8. Anexos

Anexo 10

Documento de No-Conformidades

No Conformidades Detectadas

<Nombre del Proyecto>

Módulo <Nombre del Módulo>

<Versión>

Elemento de Configuración: <Nombre del Elemento>

- 1. Aspectos generales
 - 1.1 Elementos Probados
 - 1.2 Elementos no Probados y causas.
- 2. Tabla de No Conformidades Detectadas

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección	Importancia	Recomendación

3.Recomendaciones

Elemento	No	No conformidad	Aspecto correspondiente	Etapa de detección

4. Anexos

Anexo 11

Plan de gestión de los requisitos

Plan de Gestión de Requisitos

<Nombre del Proyecto>

<Versión>

1. Introducción

1.1 Alcance

1.2 Definiciones, Acrónimos y Abreviaturas

1.3 Referencias

2. Gestión de Requisitos

2.1 Organización, responsabilidades e interfaces

2.2 Herramientas, ambientes e infraestructura

3. Programa de Gestión de Requisitos

3.1 Identificación de requisitos

Artefacto (Tipo de documento)	Elemento de trazabilidad	Descripción

3.2 Seguimiento

3.2.1 Criterio de <elemento de seguimiento>

3.3 Atributos

3.3.1 Atributo para <elemento de seguimiento >

3.4 Entregables del Proyecto

3.5 Gestión de cambios a los requisitos

Anexo 12

Tabla de prácticas genéricas para las Aps del nivel 2 de madurez de CMMI.

PG 2.1: Establecer una política organizacional para el proceso.

PG 2.2: Planificar el proceso.

PG 2.3: Proveer los recursos necesarios en el proceso.

PG 2.4: Asignar responsabilidades.

PG 2.5: Entrenar al personal.

PG 2.6: Gestionar la configuración del proceso.

PG 2.7: Identificar e implicar a los stakeholders relevantes en el proceso.

PG 2.8: Monitorear y controlar el proceso.

PG 2.9: Evaluar la adherencia del proceso.

PG 2.10: Revisar el estado del proceso.

