

Universidad de las Ciencias Informáticas

Facultad 15



Título: Propuesta de Mejora de CMMI Nivel 2 de Madurez para Proyectos de Desarrollo Ágil con metodología eXtreme Programming en la Universidad de Ciencias Informáticas.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autores:

Yenia Ramirez Noel.

Alianna Acosta Gispert.

Tutora:

Ing. Ana Marys Garcia Rodríguez.

**Ciudad de La Habana, 2010.
“Año del 50 Aniversario del Triunfo de la Revolución”.**

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Yenia Ramirez Noel

Nombre completo del autor

Alianza Acosta Gispert

Nombre completo del autor

Ing. Ana Marys Garcia Rodríguez.

Nombre completo del Tutor

DATOS DEL CONTACTO

Ing. Ana Marys Garcia Rodríguez

Graduada de Ingeniera en Ciencias Informáticas en Julio del año 2007 en la Universidad de Ciencias Informáticas (UCI). Ha recibido cursos de postgrado: DIU, Negociación y Gestión de Contratación, Técnicas de Dirección, CMMI, Inglés Básico, Curso Básico de Gestión de Proyectos. Presentó dos trabajos en el evento UCIENCIA 2008: Autora y ponente del trabajo “Estrategia para la Gestión de Configuración en Organizaciones Productivas” que se encuentra publicado en las memorias del evento, coautora del trabajo “Apuntes sobre la Disciplina Ambiente” que se encuentra publicado en la Serie Científica de la UCI.

Se encuentra matriculada en la maestría de Gestión de Proyectos.

Se ha desempeñado como revisora en dos revisiones organizadas por la Dirección de Calidad UCI. Se desempeñó como gestora de cambios y analista del proyecto SIGEP, cumpliendo misión en Venezuela en dos ocasiones para la captura de requerimientos de este proyecto. Actualmente es analista del proyecto Aduana.

AGRADECIMIENTOS

AGRADECIMIENTOS

A Dios por darme fuerzas para seguir adelante.

A mi madre, mi tesoro más grande, por creer en mí y darme su amor incondicional.

A mi padre, a mis hermanos, a mi tío William, Raúl, Carlos y a toda mi familia por apoyarme siempre.

A mis dos sobrinitos lindos por llenar mi vida con tantas alegrías.

A todos mis amigos por aceptarme como soy y acompañarme en los momentos difíciles.

A Abigail, mi novio, por ayudarme a crecer y hacerme tan feliz.

A mi compañera de tesis por estar conmigo en las buenas y en las malas.

A mi tutora por habernos guiados y ser una buena amiga y a todos los que de una u otra forma hicieron posible que sea ingeniera y una mejor persona.

Lila.

A Dios y la Virgen de la Caridad del Cobre por darme fuerza para seguir adelante.

A mis abuelos, que son lo más grande que tengo en la vida.

A mis tías y en especial a mi tía Yeni por apoyarme siempre y ser incondicional conmigo.

A mi padre por darme su amor y su cariño. A Pablo mi novio por mantenerse a mi lado en todos estos años.

A mi compañera de tesis que siempre estuvo a mi lado en las buenas y en las malas.

A nuestra tutora que fue mucho más que eso y nos demostró ser una buena amiga.

En fin a todas aquellas personas que de una forma u otra han estado a mi lado en todo este tiempo.

Yenia.

DEDICATORIA

DEDICATORIA

A Juana, la mamita más linda del mundo, la persona más especial e importante de mi vida.

A mis hermanos, a mis sobrinitos y a toda mi familia.

Lila

A mi madre, que a pesar de no estar a mi lado se que está muy orgullosa de mi.

A mis abuelitos lindos Raquel y Alfonso.

A toda mi familia.

Yenia.

RESUMEN

El desarrollo de software se ha expandido a cada uno de los sectores de la vida cotidiana del ser humano logrando facilitar el trabajo de las personas y garantizando una mayor eficiencia y eficacia en el desarrollo de los procesos de negocio de las empresas.

Actualmente se presentan problemas comunes dentro del desarrollo del software que son minimizados y/o evitados por muchas empresas a partir de la aplicación de los objetivos y prácticas que propone el Modelo de Capacidad de Madurez Integrado para Desarrollo (CMMI-DEV) en su versión 1.2. En la Universidad de Ciencias Informáticas (UCI) se está desarrollando un programa de mejora para certificar los proyectos de toda la universidad en el nivel 2 de madurez de CMMI, mas existen diferentes metodologías de desarrollo que se utilizan en los proyectos productivos, entre las que resaltan: Proceso Unificado de Desarrollo (RUP) y eXtreme Programming (XP).

Con este enfoque se realizó una definición de procesos de mejora del nivel 2 de madurez de CMMI-DEV v1.2 para proyectos que utilizan como metodología de desarrollo XP.

Para el desarrollo de la investigación se hizo un estudio acerca de las fases, actividades, roles y artefactos de XP y de los objetivos y prácticas del nivel 2 de madurez de CMMI-DEV v1.2. Se definieron los siete procesos de las áreas de este nivel a partir de las características de XP. Por último se aplicó uno de los procesos definidos a un proyecto de desarrollo con metodología XP para validar la propuesta de la presente investigación.

ÍNDICE

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA..... 5

1. **Introducción.** 5

2. **Características de la Metodología eXtreme Programming (XP).**..... 6

 Fases de desarrollo de XP 8

 Roles de XP 11

 Artefactos de XP 12

3. **Modelo de calidad de software CMMI-DEV v1.2.** 17

 Nivel de madurez 2 (Gestionado) 17

 Áreas de procesos del nivel 2 de madurez 18

Administración de Requerimientos (REQM) 18

Planificación de Proyectos (PP) 19

Monitoreo y Control del Proyecto (PMC) 20

Medición y Análisis (MA) 20

Aseguramiento de la Calidad de Productos y Procesos (PPQA) 21

Administración de la Configuración (CM) 22

Administración de Acuerdos con Proveedores (SAM) 23

4. **Definiciones de Calidad de Software en la UCI.** 24

 Lineamientos de calidad UCI 24

 Auditoría y Revisiones 25

5. **Relación de XP con CMMI-DEV v1.2.** 26

6. **Conclusiones**..... 27

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA 28

1. **Introducción.** 28

2. **Definición de procesos.** 28

Administración de Requerimientos (REQM)..... 28

Planificación del Proyecto (PP)..... 37

3. **Conclusiones**..... 55

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA.....	56
4. Introducción.	56
5. Caso de estudio.....	56
6. Aplicación de la propuesta de mejora del nivel de madurez 2 de CMMI-DEV al caso de estudio.	57
7. Resultados obtenidos.....	67
CONCLUSIONES.....	68
RECOMENDACIONES.....	69
REFERENCIAS BIBLIOGRÁFICAS.....	70
BIBLIOGRAFÍA.....	72

INTRODUCCIÓN

El desarrollo del software desde su surgimiento ha evolucionado a nivel mundial en gran escala. Muchas empresas, instituciones y países han optado por el desarrollo de esta industria para favorecer sus activos económicos. En la actualidad las compañías dependen cada vez más del software para realizar sus labores. Con el desarrollo del software también ha venido el de las Tecnologías de la Información y las Comunicaciones que es un factor decisivo para el desarrollo de las empresas, de la economía y de la sociedad en el mundo. Todo esto trae consigo un proceso que busca lograr más eficacia, eficiencia, satisfacer las necesidades de información y conocimiento de todas las personas y esferas de la sociedad; a la vez agiliza los procesos en las instituciones, incrementa los beneficios y ganancias en el mercado mundial.

Situación problemática.

Por su relativa juventud y complejidad, la industria del software aún se encuentra inmersa en numerosas investigaciones para la mejora de los procesos para el desarrollo del software. El modelo CMMI-DEV propone una mejora a estos procesos para obtener un producto con buena calidad y lograr así la satisfacción del cliente. Muchas empresas a escala mundial han optado por certificarse en CMMI para obtener un reconocimiento a escala internacional y la UCI no se encuentra exenta a esta situación.

Actualmente los procesos de desarrollo del software se enfrentan a las siguientes problemáticas: (Fernández, 2009)

- **Planificación irreal:** Los líderes de desarrollo no son capaces de enfrentar un plan adecuado pues no están entrenados para dicha tarea.
- **Mala calidad del trabajo:** Las prácticas ineficientes de la ingeniería de software y el mal manejo de las técnicas de prueba hacen que los productos tengan muchos defectos y sea necesario la inversión de más tiempo y recursos para la mejora del mismo.
- **Personal inadecuado:** En ocasiones el personal destinado a ciertas tareas se incorpora tarde o no cumplen las necesidades en cuanto a cantidad y calidad, esto hace que el trabajo se demore y se descuide.

- **Cambios no controlados:** Un proyecto de software siempre está en constante cambio, sobre todo en el área de requerimientos. Sin una correcta gestión de la configuración el equipo no puede estimar la magnitud de un cambio lo que trae consigo la pérdida gradual del control del proyecto.

Para dar solución a estos problemas de desarrollo han surgido diferentes estándares y modelos internacionales de calidad que proponen guías para el desarrollo de los productos software y miden la calidad del proceso y/o producto. Entre estos se encuentra CMMI-DEV v1.2 que es un modelo de calidad del software que certifica a las empresas en niveles de madurez. Estos niveles sirven para conocer la madurez de los procesos que se realizan para producir software.

Actualmente en la UCI se desarrolla un proceso de mejora con el objetivo de certificar los proyectos de la UCI en el nivel 2 de madurez de CMMI-DEV v1.2. Para esto se está realizando el proceso de certificación en cinco proyectos pilotos que utilizan la metodología tradicional RUP y de obtenerse resultados satisfactorios resultaría muy práctico y provechoso expandir este programa al resto de los proyectos en la UCI que utilicen esta metodología; sin embargo existen proyectos que utilizan la metodología ágil XP para los cuales no se está llevando a cabo este proceso de mejora, mientras XP como metodología ágil, presenta características que le permiten adecuarse a CMMI-DEV v1.2.

La UCI como gran productora de software que utiliza metodologías de desarrollo de software tanto las tradicionales como ágiles, se encuentra abogando por el éxito de este programa de mejora para ser extendido posteriormente al resto de los proyectos productivos y así obtener una certificación de calidad de procesos y productos que abra las puertas al mercado internacional en gran escala. Sin embargo, a pesar de existir una base con la definición de los procesos del programa de mejora UCI, no se ha trabajado en la adecuación o definición de procesos para la aplicación de este programa a los proyectos que utilizan XP. Esta definición de procesos del programa de mejora UCI no se adapta completamente a las fases, roles y artefactos que están definidos para la metodología XP, por lo que se arriba al siguiente **problema de investigación:** ¿Cómo lograr que los proyectos que utilizan metodología XP en la UCI, estandaricen la gestión de sus procesos según CMMI DEV v1.2 nivel de madurez 2?

En la presente investigación se obtiene como **aporte práctico** una propuesta de definición de procesos de mejora del nivel 2 de madurez de CMMI-DEV v1.2 para proyectos de desarrollo ágil con metodología XP,

que permita alcanzar resultados satisfactorios respecto al tiempo de desarrollo y el costo de los mismos y contribuya a la obtención de productos con mejor calidad, logrando así una mayor satisfacción del cliente.

Constituyó **objeto de estudio** de la investigación: la metodología de desarrollo de software XP y los objetivos y prácticas del nivel 2 de madurez de CMMI-DEV v1.2; siendo el **campo de acción**: el desarrollo de software en proyectos de la UCI que utilizan como metodología de desarrollo XP.

Para dar solución al problema de investigación se define como **objetivo general**: Obtener una propuesta de definición de procesos de mejora que se adapte a los requerimientos del nivel 2 de madurez de CMMI-DEV v1.2 en los proyectos de software de la UCI que utilizan metodología XP.

Como idea a defender se puntualizó que: a partir de una guía de proceso de mejora basado en el nivel 2 de madurez de CMMI-DEV v1.2 para proyectos que utilizan metodología XP en la UCI, se podría lograr que dichos proyectos estandaricen la gestión de sus procesos mejorando de esta manera la calidad del producto final.

Para dar cumplimiento al objetivo de la investigación se acometieron las siguientes tareas de investigación:

1. Caracterizar las fases de desarrollo, roles, artefactos y actividades de la metodología de desarrollo de software XP para su aplicación en la propuesta.
2. Caracterizar el modelo de calidad de software CMMI-DEV v1.2 para determinar los elementos aplicables a la propuesta.
3. Caracterizar los criterios de medida de calidad del software en la UCI para determinar los elementos aplicables a la propuesta.
4. Definición de los procesos de mejora del nivel 2 de madurez de CMMI-DEV v1.2 para proyectos con metodología XP, a partir de las fases, actividades, roles y artefactos de dicha metodología, para la elaboración de la propuesta.
5. Aplicación de uno de los procesos definidos en el presente trabajo, en un proyecto de desarrollo con metodología XP para realizar una validación de la propuesta.

La **estructura del contenido** de la investigación consta de 3 capítulos.

Capítulo 1. Fundamentación Teórica:

En el Capítulo 1 se abordan los principales conceptos relacionados con la metodología de desarrollo de software XP, a través de definiciones de varios autores que han hecho aportes relevantes y significativos en este tema y de las estipulaciones realizadas por el modelo CMMI-DEV v1.2. Además se realiza un estudio de la metodología de desarrollo XP.

Se realiza un análisis del nivel 2 de madurez de CMMI-DEV v 1.2, teniendo en cuenta las características de cada área de proceso de dicho nivel.

Capítulo 2. Diseño de la Solución:

En el Capítulo 2 se describe detalladamente la propuesta de definición de procesos de mejora CMMI-DEV v1.2 nivel 2 de madurez en proyectos de desarrollo ágil con metodología XP. Se definen los procesos a desarrollar a partir de la metodología XP, especificando las actividades, entradas, salidas y responsables de cada actividad.

Capítulo 3. Validación de la Propuesta.

En el Capítulo 3 se realiza la validación de la propuesta a partir de la aplicación de uno de los procesos definidos al proyecto Meñique de la facultad # 5.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1. Introducción.

En el presente capítulo se abordan las principales características de la metodología XP y del nivel 2 de madurez de CMMI-DEV v1.2.

“En la última década la metodología XP constituye una de la mayor tendencia en la industria de desarrollo de software. Es una metodología ágil que surge como una nueva forma o guía para realizar proyectos de software, esencialmente está basada en la simplicidad y agilidad, también en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios que se les realicen al proyecto. La misma se ha definido para proyectos con requisitos muy cambiantes e imprecisos y que exista un alto riesgo técnico.” (Grupo de Calidad de Software de España, 2009).

Actualmente XP es una de las metodologías más exitosas; esto se debe a que está centrada en potenciar las relaciones entre personas como clave para obtener éxito en el desarrollo del software, la misma promueve a el trabajo en equipo, la comunicación con el cliente, proporcionando a cada uno de estos un buen clima de trabajo y un aspecto muy importante es que se preocupa por el aprendizaje de los programadores (Universidad Politécnica de Valencia, 2003).

XP a pesar tener características excepcionales es importante que para obtener un mejor resultado y satisfacer las necesidades del cliente se emplee un modelo de calidad para optimizar eficientemente sus procesos, en este caso se emplea el modelo de calidad CMMI-DEV v 1.2, centrándose específicamente en el nivel 2 de madurez (Gestionado).

CMMI-DEVv1.2 establece una pauta para que las organizaciones mejoren sus procesos y habilidades de gestión para el desarrollo, adquisición y mantenimiento de sus productos o servicios: sustituye los múltiples modelos de Capacidad y Madurez existentes hasta el momento y define tanto el modelo como el método de evaluación. CMMI-DEV v 1 .2 es un modelo descriptivo en el sentido que describe los atributos esenciales que se esperan caractericen una organización dentro de un nivel de madurez en particular. Es un modelo normativo ya que las prácticas detalladas caracterizan el tipo normal de comportamiento que

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

se espera de una organización que realiza proyectos a gran escala. No es prescriptivo ya que no dice a la organización cómo mejorar. CMMI-DEV contempla dos representaciones (continua o escalonada), la representación continua es para una simple área o conjunto de áreas de proceso (niveles de capacidad) y la representación escalonada es para un conjunto definido de áreas de proceso en la organización (niveles de madurez) (Chrissis, 2009).

“Un área de proceso es un grupo de prácticas que se relacionan entre sí y se ayudan mutuamente para alcanzar un conjunto de metas. Son los bloques fundamentales que permiten establecer la capacidad del proceso de una organización. Cada área de proceso reside en un nivel de madurez y pertenece a una categoría determinada. Un nivel de madurez es una etapa bien definida en el camino para convertirse en una organización madura. CMMI-DEV tiene definidos 5 niveles para medir la capacidad de los procesos” (Chrissis, 2009).

En este caso solo se hará referencia al nivel 2 de madurez de CMMI-DEV v1.2 que está centrado en poner en orden las prácticas relacionadas con el manejo elemental de los proyectos. En este nivel los proyectos de cada organización siguen algún tipo de procesos para realizar las actividades relacionadas con la gestión del proyecto (planificación y control), para administrar los requerimientos, las configuraciones, medir y analizar la calidad de los productos y el desempeño de los procesos, así como la gestionar los acuerdos con proveedores. En este nivel lo más importante es que el equipo de desarrollo pueda tener un mínimo de capacidad de gestión de proyectos.

2. Características de la Metodología eXtreme Programming (XP).

El desarrollo bajo XP tiene enumeradas características entre las que se destacan: (Villafuerte, 2009)

- Desarrollo iterativo e incremental.
- Pruebas unitarias continuas.
- Integración del equipo de programación con el usuario.
- Corrección de todos los errores.
- Refactorización del código.
- Propiedad del código compartida.

Valores que inspiran a XP

Comunicación

En XP el cliente y los programadores se comunican directamente y de forma continua. El cliente forma parte del equipo de proyecto para establecer prioridades y resolver dudas en caso que existan. De este modo, puede ver el avance día a día que va teniendo el proyecto esto le posibilita ajustar su agenda y las funcionalidades de forma consecuyente (Palacio, 2009).

Retroalimentación rápida y continua

“Al estar el cliente integrado en el proyecto, su opinión sobre el estado del proyecto se conoce en tiempo real. Al realizarse ciclos muy cortos tras los cuales se muestran resultados, se minimiza el tener que rehacer partes que no cumplen con los requisitos y ayuda a los programadores a centrarse en los que es más importante. Considérense los problemas que derivan de tener ciclos muy largos. El código también es una fuente de retroalimentación gracias a las herramientas de desarrollo. Por ejemplo, las pruebas unitarias informan sobre el estado de salud del código. Ejecutar las pruebas unitarias frecuentemente permite descubrir fallos debidos a cambios recientes en el código”. (Grupo de Calidad de Software de España, 2009)

Simplicidad

La simplicidad consiste en realizar el sistema que realmente el cliente necesita. Esto implica que se resuelvan las necesidades actuales. Con la simplicidad, junto con la comunicación y la retroalimentación resulta más fácil conocer las necesidades reales. (Palacio, 2009)

Coraje o Valentía

El coraje consiste en tomar decisiones difíciles en los momentos que se presenten. En el momento que se detecta un error este debe ser reparado. Cada vez que el código se manifieste susceptible de mejoras, se puede mejorar por parte de cualquier equipo de programación en parejas ya que este se mantiene en un repositorio. Cada vez que exista un desajuste sobre la planificación se trata con el cliente para decidir qué es lo que se le va a entregar y cuándo. (Palacio, 2009)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Fases de desarrollo de XP.

- **Planificación del proyecto.**

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. (Castillo Oswaldo, 2010).

Después de tener ya definidas las historias de usuario es necesario crear un Plan de Publicaciones, donde se indiquen las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicarán estas versiones.

Iteraciones.

Cualquier proyecto que use metodología XP se debe dividir en iteraciones que constan aproximadamente de 3 semanas de duración. Al iniciar cada iteración el cliente debe seleccionar las historias de usuario que se han definido en el Plan de Publicaciones que serán implementadas. Las Historias de Usuario que no pasaron la prueba de aceptación y que se realizaron al terminar la iteración anterior, también son seleccionadas. Las Historias de Usuario son divididas en tareas de entre 1 y 3 días de duración y son asignadas a los programadores. (Castillo, 2010)

Velocidad del proyecto.

“La velocidad del proyecto representa la rapidez con la que los programadores desarrollan el proyecto; es sencillo de estimar, para esto se debe contar el número de historias de usuario que se pueden implementar en cada una de las iteraciones; de modo que se sabrá la cantidad de Historias de Usuarios que se pueden desarrollar en las diferentes iteraciones. Así podemos controlar que todas las tareas se puedan realizar en el tiempo disponible para la iteración” (Castillo, 2010).

Reuniones diarias.

“Diariamente los desarrolladores se reúnen para exponer sus problemas, soluciones e ideas de manera conjunta. En estas reuniones todos los desarrolladores tienen derecho a dar su opinión sobre el proyecto y debe ser una reunión fluida”. (Castillo, 2010)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- **Diseño.**

Diseños simples.

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible y posible de implementar que a la larga costará menos tiempo y esfuerzo desarrollar. (Castillo, 2010).

Riesgos.

En caso que surjan problemas durante la realización de diseño, XP propone utilizar una pareja de desarrolladores para que investiguen la causa del problema y reduzcan a toda costa el riesgo que trae consigo ese problema.

Funcionalidad Extra.

No se debe añadir funcionalidad extra al programa aunque se crea que posteriormente será utilizada, pues puede implicar que el desarrollo de esta funcionalidad extra sea una pérdida de tiempo y recursos.

Refactorizar.

Refactorizar es mejorar y modificar la estructura y codificación de códigos ya creados sin alterar su funcionalidad. Refactorizar supone revisar de nuevo estos códigos para procurar optimizar su funcionamiento. Es muy común reutilizar códigos ya creados que contienen funcionalidades que no serán usadas y diseños obsoletos. Esto es un error porque puede generar código completamente inestable y muy mal diseñado; por este motivo, es necesario refactorizar cuando se va a utilizar código ya creado. (Castillo, 2010).

- **Codificación**

La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad. Crear pruebas que demuestren el funcionamiento de los distintos códigos implementados ayuda a desarrollar dicho código. Crear estas pruebas antes ayuda a saber qué es exactamente lo que tiene que hacer el código a implementar y se sabrá que una vez implementado pasará dichas pruebas sin problemas ya que dicho código ha sido diseñado para ese fin. Se puede dividir la funcionalidad que debe cumplir una tarea a

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

programar en pequeñas unidades, de esta forma, se crearán primero las pruebas para cada unidad y a continuación se desarrollará dicha unidad, así poco a poco se consigue un desarrollo que cumpla todos los requisitos especificados. (Castillo Oswaldo, 2010)

Programación en parejas.

“XP sugiere la programación en parejas ya que aumenta la productividad y la calidad del software desarrollado. El trabajo en pareja consiste en dos programadores trabajando en el mismo equipo; mientras uno codifica insistiendo en la calidad de la función o método que está implementando, el otro analiza si ese método o función se ajusta a lo que realmente se quiere y si está bien diseñado. Los programadores están involucrados en todas las tareas que se realizan, ya que XP propone un modelo de desarrollo colectivo, por lo que cualquier programador puede modificar o ampliar una clase o método de otro programador en caso que sea necesario y subirla al repositorio de código. De este modo, se consigue un código y diseño con gran calidad. Sugiriendo un modelo de trabajo utilizando un repositorio de código para que el mismo se mantenga de forma consistente donde las parejas de programadores publican cada un tiempo determinado los códigos que han implementado y corregido junto a las pruebas que deben pasar. Así los programadores que necesiten utilizar códigos ajenos trabajarán siempre con las últimas versiones.” (Castillo Oswaldo, 2010)

- **Pruebas.**

Uno de los pilares más importantes de esta metodología es el uso de las pruebas, que se utilizan para comprobar el funcionamiento de los códigos que se vayan implementando. (Agustin Yagüe, 2009).

Cada Historia de Usuario lleva asociada criterios de aceptación y para cada criterio de aceptación se definen casos de prueba. Las Pruebas de Aceptación se escriben en las fases tempranas del desarrollo y antes de que el sistema se implemente, para validar las necesidades de los usuarios y para dirigir la implementación. Se puede considerar que las historias de usuario y por extensión las pruebas asociadas, juegan el papel de especificaciones del sistema. (Agustin Yagüe, 2009)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Roles de XP.

A continuación se muestran los roles: (Beck, 1999)

Programador: El programador escribe las Pruebas Unitarias y produce el código del sistema. Debe existir una comunicación y coordinación adecuada entre los programadores y el resto de los miembros del equipo.

Cliente: Escribe las Historias de Usuario y las pruebas funcionales para validar su implementación. Además, asigna la prioridad a las Historias de Usuario y decide cuáles se implementan en cada iteración centrándose en aportar mayor valor al negocio. El cliente es solo uno dentro del proyecto pero puede corresponder a un interlocutor que está representando a varias personas que se verán afectadas por el sistema.

Encargado de pruebas: Ayuda al cliente a escribir las pruebas funcionales. Ejecuta las pruebas regularmente, difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Encargado de seguimiento: Proporciona retroalimentación al equipo. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, para mejorar futuras estimaciones. Realiza el seguimiento del progreso de cada iteración.

Entrenador: Es responsable del proceso global, es el líder del proyecto. Debe proveer guías al equipo de forma que se apliquen las prácticas XP y se siga el proceso correctamente.

Consultor: Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto, en el que puedan surgir problemas.

Gestor: Es el vínculo entre clientes y programadores, ayuda a que el equipo trabaje efectivamente creando las condiciones adecuadas. Su labor esencial es de coordinación.

Rastreador: Es la persona encargada de observar todas las actividades que se realicen en el proyecto y conserva todos los datos referentes al mismo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Artefactos de XP.

Historias de Usuarios.

“La Historia de Usuario es una técnica utilizada para especificar los requisitos del software, se utiliza específicamente en XP. Las mismas son tarjetas de papel donde el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Estas Historias de Usuario tienen un tratamiento muy dinámico y flexible, en cualquier momento pueden romperse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada Historia de Usuario es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas.” (Jeffries, 2001)

Al iniciar un proyecto no es de suma importancia identificar las Historias del Usuario, pero al comenzar cada iteración ya deben estar registrados los cambios realizados en las mismas y los datos que se encuentren grabados en estos se planificarán en la próxima iteración. Estas Historias de Usuario son descompuestas en tareas de programación y se les asignan a los programadores para que las implementen durante una iteración. (Jeffries, 2001)

Historia de Usuario	
Número:	Nombre Historia de Usuario:
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario:	Iteración Asignada:
Prioridad en Negocio: (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: (Alto / Medio / Bajo)	Puntos Reales:
Descripción: Se introducen los datos del artículo (título, fichero adjunto, resumen, tópicos) y de los autores (nombre, correo, afiliación). Uno de los autores debe indicarse como autor de contacto. El sistema confirma la correcta recepción del artículo enviando un correo al autor de contacto con un usuario y contraseña para que el autor pueda posteriormente acceder al artículo.	
Observaciones:	

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Tabla 1: Modelo de Historia de Usuario.

Existen tres aspectos fundamentales que tienen las Historias de Usuario: (Jeffries, 2001)

1. Tarjeta: En esta tarjeta se guarda suficiente información para identificar y detallar la historia.
2. Conversación: El cliente y los programadores se reúnen en caso que sea posible para discutir la historia y ampliar detalles de la misma.
3. Pruebas de Aceptación: Esta prueba permite comprobar que la Historia de Usuario ha sido implementada correctamente.

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nro. y Nombre):
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

Tabla 2: Modelo propuesto para una prueba de aceptación.

Plan de Publicaciones

Un Plan de Publicaciones es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación ideales de las historias de usuario, la prioridad con la que serán implementadas y las historias que serán implementadas en cada versión del programa. Después del desarrollo de un Plan de Publicaciones tienen que estar claros estos cuatro factores: los objetivos que se deben cumplir (que son principalmente las historias que se deben desarrollar en cada versión), el tiempo que tardarán en desarrollarse y publicarse las versiones del programa, el número de personas que trabajarán en el desarrollo y cómo se evaluará la calidad del trabajo realizado. (Castillo Oswaldo, 2010)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Plan de Entrega.

Para crear el Plan de Entrega se utilizan las Historias de Usuarios, para esto se convoca a una reunión para crear este plan. El mismo se usa para crear los planes de iteración para cada iteración. Si cada Historia de Usuario es evaluada en tiempo de desarrollo ideal, el cliente las agrupa según el orden de importancia que tengan.

Se puede trazar el Plan de Entregas en función de los parámetros que se muestran a continuación: tiempo de desarrollo ideal y grado de importancia para el cliente

Glosarios de términos.

Se deben usar los glosarios de términos, se debe especificar correctamente los nombres de métodos y clases, esto ayudará a entender el diseño y facilitará que posteriormente se pueda ampliar y reutilizar el código. (Castillo, 2010).

Tarjetas CRC (Clase, Responsabilidad y Colaboración)

El uso de las tarjetas CRC (Class, Responsibilities and Collaboration) permiten al programador centrarse y apreciar el desarrollo orientado a objetos olvidándose de los malos hábitos de la programación procedural clásica. (Wesley, 2000).

Nombre de la clase.	
Responsabilidades	Colaboradores

Figura 1: Modelo de tarjeta CRC.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

A continuación se muestran los pasos a seguir para llenar las tarjetas:

- Encontrar clases.
- Encontrar responsabilidades.
- Definir colaboradores.
- Disponer las tarjetas.

Tareas de Ingeniería.

Las tareas ingenieriles son la mejora constante de los procesos de producción, con el fin de que estos sean cada vez más productivos. Son tarjetas de papel donde el cliente especifica brevemente aspectos relacionados con la tarea a realizar, ellas al igual que las historias de usuario tienen un tratamiento muy dinámico y flexible pueden ser cambiadas por tareas más específicas o generales y añadirse nuevas o ser modificadas.

Tarea de Ingeniería	
Número Tarea:	Historia de Usuario (Nro. y Nombre):
Nombre Tarea:	
Tipo de Tarea : Desarrollo / Corrección / Mejora / Otra (especificar)	Puntos Estimados:
Fecha Inicio:	Fecha Fin:
Programador Responsable:	
Descripción:	

Tabla 3: Modelo propuesto para una Tarea de Ingeniería.

Código.

El código debe tener un formato de acuerdo con las normas de codificación que se hayan establecido y mantenerlo de forma consistente y fácil para que todo el equipo pueda entenderlo y refactorizarlo, en este

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

último caso sería reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento.

- **Propiedad del código compartida:** Esto significa que cualquier persona pueda corregir y extender cualquier parte del proyecto. Para esto se realizan frecuentes pruebas de regresión para garantizar que los posibles errores sean detectados. (Cruz, 2008)
- **Simplicidad en el código:** Según la metodología XP es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si es necesario, que realizar algo complicado y tal vez nunca utilizarlo. (Cruz, 2008)

Pruebas Unitarias y de Integración.

“Las Pruebas Unitarias son la escala más pequeña de la prueba, está basada en la funcionalidad de los módulos del programa, como funciones, procedimientos, módulos de clase. En ciertos sistemas también se verifican o se prueban los conductores y el diseño de la arquitectura”. (Pressman, 2005)

La elaboración de código está dirigida a este tipo de prueba, las cuales se establecen antes de escribir el código y se ejecutan constantemente ante cada modificación del sistema. Otros desarrolladores podrán ver cómo utilizar el código contemplando las pruebas.

Para probar los componentes implementados como unidades individuales, se realizan las pruebas de especificación o de caja negra, estas pruebas comprueban el comportamiento de la unidad observable externamente. Las pruebas de estructura o de caja blanca verifican la implementación interna de la unidad.

Pruebas de Aceptación.

Sirven para evaluar las distintas tareas de ingeniería en que se ha dividido una Historia de Usuario y para garantizar su funcionamiento final. El cliente es quien crea y utiliza estas pruebas para comprobar y confirmar que las historias de usuario cumplan su cometido. (Peña, 2008)

Caso de Prueba

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Número Caso de Prueba:	Número Historia de Usuario:
Nombre Caso de Prueba:	
Descripción:	
Condiciones de ejecución:	
Entradas:	
Resultado esperado:	
Evaluación:	

Tabla 4: Caso de Prueba.

Estas pruebas son creadas a partir de las Historias de Usuario. En el transcurso de una iteración la Historia de Usuario seleccionada en la planificación de iteraciones se convertirá en una Prueba de Aceptación. El cliente o usuario debe especificar los aspectos a probar cuando una Historia de Usuario ha sido correctamente implementada. (Peña, 2008)

Para garantizar el correcto funcionamiento de una Historia de Usuario esta puede tener más de una prueba de aceptación y la misma no se considera finalizada hasta que no supera sus pruebas de aceptación. Por lo que debe realizarse una nueva prueba de aceptación para cada iteración o se creará que el equipo de desarrollo no realiza ningún progreso. (Lételier, 2009)

3. Modelo de calidad de software CMMI-DEV v1.2.

Nivel de madurez 2 (Gestionado).

Si una organización ha alcanzado el nivel 2 de madurez entonces se puede decir que existe una disciplina para la gestión de proyectos, donde los recursos humanos están capacitados para hacer su trabajo y existen políticas organizacionales formalmente establecidas, cuyo cumplimiento será monitoreado. Habrá visibilidad de las actividades realizadas y los proyectos se ejecutarán siguiendo un plan y de acuerdo con un proceso formalmente establecido. En este nivel existen procesos definidos en el contexto de cada proyecto, pero también pueden definirse procesos que puedan ser utilizados por más de un proyecto. (Chrissis, 2009)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Objetivo del nivel 2.

Los objetivos genéricos (GG) y las prácticas genéricas (GP) son los siguientes: (Chrissis, 2009)

GG 2 Institucionalizar un proceso administrado

- GP 2.1 Establecer políticas organizacionales
- GP 2.2 Planificar el proceso
- GP 2.3 Proveer Recursos
- GP 2.4 Asignar responsabilidades
- GP 2.5 Entrenar al personal
- GP 2.6 Administrar la configuración
- GP 2.7 Identificar e involucrar a los interesados
- GP 2.8 Monitorear y controlar los procesos
- GP 2.9 Evaluar adhesión objetivamente
- GP 2.10 Revisar el estado con la Alta Gerencia

Áreas de procesos del nivel 2 de madurez.

Administración de Requerimientos (REQM).

Esta área de proceso tiene como propósito mantener bajo control los requerimientos lo que el producto a desarrollar deberá satisfacer. Las prácticas incluidas en este nivel apuntan a que los requerimientos no sólo estén claramente identificados, sino también que todos los involucrados en el proyecto (el cliente, el equipo de proyecto) estén de acuerdo en su significado. Adicionalmente, los requerimientos deben ser la entrada a las actividades de planificación. (Mark C. Paulk, 2002)

“Otro aspecto de suma importancia en esta área es la trazabilidad bidireccional, es decir, cuando los requerimientos son administrados correctamente deberíamos estar en condiciones de relacionar cuál ha sido el origen de los requerimientos, cuál es la relación entre los requerimientos de bajo nivel y los de alto nivel (por ejemplo cuáles son derivados y de cuál requerimiento), cuáles son los artefactos relacionados con los requerimientos (por ejemplo: especificaciones, documentos de diseño o planes) y cuáles componentes del producto implementan cada requerimiento. Ante posibles cambios esta práctica es de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

suma importancia para poder realizar un análisis exhaustivo y además poder determinar si el alcance del proyecto ha sido cubierto o no.” (Mark C. Paulk, 2002)

Esta área de proceso se satisface poniendo en marcha algún tipo de mecanismo o sistema que: (Mark C. Paulk, 2002)

- Identifique quiénes son las fuentes oficiales de requerimientos.
- Identifique cuáles son los canales válidos para ingresar requerimientos al proyecto.
- Controle los cambios (no cualquiera estaría en condiciones de generar un cambio a un requerimiento).
- Permita determinar si todos los involucrados tienen la misma visión respecto al significado de los requerimientos (por ejemplo, mediante la aprobación de algún tipo de documento formal).
- Mantenga la trazabilidad entre los requerimientos y otros artefactos (incluyendo al producto y sus componentes).

En este modelo los requerimientos se clasifican en tres categorías: (Paulk, 1993)

- Los del usuario: aquellos que formula el usuario y que pueden ser documentados en una minuta de reunión.
- Los del producto: derivados a partir de los primeros; generalmente descritos en un lenguaje más cercano al equipo de proyecto.
- Los de las componentes del producto: derivados de los anteriores.

Planificación de Proyectos (PP).

Esta área de proceso tiene como propósito establecer y mantener el plan que será empleado para ejecutar y monitorear el proyecto. El plan se desarrolla sobre la base de los requerimientos administrados por el área REQM. Dentro de esta área de proceso se incluyen todas las actividades necesarias para determinar el alcance del proyecto (funcionalidad a desarrollar, actividades incluidas y excluidas), estimar esfuerzo y costos, establecer el cronograma, identificar riesgos y obtener el compromiso de todos los involucrados respecto al Plan de Proyecto.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Mediante la combinación de varios elementos se implementan las actividades de esta área. Es necesario establecer algún tipo de mecanismo de estimación que emplee como entrada los requerimientos del proyecto. También será necesario formalizar el Plan de Proyecto que no es solamente el cronograma, el ciclo de vida a emplear (por lo menos, fases e hitos) y los mecanismos de aprobación. (Mark C. Paulk, 2002)

Monitoreo y Control del Proyecto (PMC).

“Esta área de proceso es complementaria y una consecuencia de Planificación del Proyecto (PP). Su propósito es monitorear la ejecución del proyecto empleando para esto el plan y gestionar acciones correctivas en el caso de detectarse desvío”. (Carnegie Mellon University, 2010)

“Para gestionar algún proyecto, lo primero es formular los planes ya que no tendría ningún sentido formular planes para algo que no se tiene intenciones de gestionar”. (Carnegie Mellon University, 2010)

“Es necesario implementar prácticas de seguimiento para poder cumplir con estos objetivos, estas prácticas pueden ser: el reporte de horas trabajadas en el proyecto, el informe de avance periódico, revisiones en puntos particulares del proyecto (por ejemplo, al final de cada fase). Teniendo en cuenta que es muy importante identificar personas dentro de la organización que puedan cubrir el rol de líder o gerente de proyecto.” (Mark C. Paulk, 2002)

Medición y Análisis (MA).

“Una premisa presente en todos los movimientos de calidad es que lo que no puede medirse no puede mejorarse. Esta área de proceso apunta, justamente, a desarrollar y mantener capacidades de medición que permitan satisfacer las necesidades de información de la organización” (Mark C. Paulk, 2002).

Estos objetivos de esta área pueden satisfacerse básicamente mediante la puesta en marcha de un programa de métricas que defina: (Mark C. Paulk, 2002)

- Objetivos de la organización (por ejemplo, aumentar la productividad un X% o disminuir los defectos en producción un Y%).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Métricas que permitan determinar si se cumplen o no con los objetivos (por ejemplo, productividad, densidad de defectos).
- Mecanismos de obtención de las métricas (procedimientos manuales, aplicaciones, orígenes de datos).

Los indicadores pueden agruparse en dos grupos: (Mark C. Paulk, 2002)

- Los empleados para monitorear cada proyecto (valor ganado, densidad de defectos).
- Los más generales (porcentaje de proyectos finalizados en fecha, desvío promedio).

Aseguramiento de la Calidad de Productos y Procesos (PPQA).

“Una vez establecidos los procesos y estándares será necesario evaluar su aplicación. El objetivo de esta área es justamente ese: proveer una evaluación objetiva de los procesos y de los artefactos producidos.” (Carnegie Mellon University, 2010)

Las prácticas de esta área implican: (Mark C. Paulk, 2002)

- Evaluar los procesos ejecutados, los artefactos producidos y los servicios provistos contra los estándares y descripciones de proceso aplicables.
- Identificar no conformidades, comunicarlas a los responsables y asegurar su resolución.
- Informar a los interesados (básicamente, el equipo de proyecto y la gerencia) el resultado de las actividades de aseguramiento de la calidad.

“Un tema importante en esta área es la objetividad. Debe garantizarse un nivel apropiado de independencia entre los productores y los evaluadores (aquellos que ejecuten actividades de aseguramiento de la calidad). Un canal de reporte con la gerencia también es importante para comunicar las no conformidades y garantizar que se resuelvan.” (Mark C. Paulk, 2002)

En esta área las prácticas específicas pueden implementarse mediante la creación de un rol encargado de estas actividades y de los procesos y estándares correspondientes. Algunas actividades de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Aseguramiento de la Calidad ¹podrán estar en el proceso de producción (por ejemplo, como una actividad en el Plan de Proyecto), mientras que otras podrán ejecutarse independientemente y tener su propio plan (más en el estilo de una auditoría).

Administración de la Configuración (CM).

“Esta área de proceso tiene como propósito mantener la integridad de todos los artefactos (entregables o no) producidos por el proyecto, lo cual implica identificar los elementos de configuración, realizar sobre estos, cambios de manera controlada, generar y mantener la Línea Base y proveer información precisa acerca del estado de la configuración a todos los interesados”. (Mark C. Paulk, 2002)

“Es probable que esta sea una de las áreas de procesos más difíciles de implementar, además de tener problemas para planificar y ejecutar sus actividades de acuerdo con esos planes. Estas prácticas se enfocan a resolver este tipo de problemas”. (Mark C. Paulk, 2002)

Para la Administración de la Configuración será necesario contar con algún tipo de sistema (preferiblemente, total o parcialmente automatizado) que permita realizar las actividades típicas: (Mark C. Paulk, 2002)

- Identificar elementos de configuración ²y mantener sus relaciones con otros elementos;

¹ **Aseguramiento de la Calidad:** Todas aquellas acciones planeadas y sistemáticas necesarias para proveer una adecuada confianza de que un producto o servicio satisfará determinados requerimientos de calidad (ISO 8402).

² Se identifica como **elemento de configuración** cualquier artefacto o documento que es utilizado por el equipo o afecta el desarrollo de actividades posteriores. Al administrarlo y comunicar su estado podemos asegurar que todos estamos trabajando sobre la misma versión del mismo. Entre estos elementos están: el plan de desarrollo, los requerimientos, el plan de pruebas, el plan de calidad o el plan de gestión de configuración.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Crear, extraer (checkout) e ingresar (checkin) elementos de configuración del sistema de administración de la configuración.
- Generar líneas base en determinados hitos del proyecto.
- Auditar elementos de configuración, líneas base y el sistema de administración de la configuración.
- Elevar, analizar y aprobar solicitudes de cambio.

Administración de Acuerdos con Proveedores (SAM).

“Esta área de proceso apunta a resolver otro de los problemas habituales en muchas organizaciones, el de la tercerización. Esta área originalmente fue pensada para todo lo relacionado con la adquisición de productos que vayan a ser incorporados en la solución a entregar al cliente, las prácticas incluidas aquí también son útiles para tener en cuenta si es necesario comprar algo que no será finalmente entregado al cliente, como son las herramientas de desarrollo”. (Juan Van Bom, 2007)

Es necesario definir los mecanismos mediante los cuales se pondrá en práctica esta área de proceso, estos se muestran a continuación: (Mark C. Paulk, 2002)

- Se seleccionarán los proveedores potenciales.
- Se elegirán los proveedores que suministrarán al proyecto los productos/servicios necesarios.
- Se formalizarán los acuerdos con los proveedores (por ejemplo, mediante un contrato o mediante un acuerdo de nivel de servicio).
- Se formalizará la entrega de los requerimientos al proveedor.
- Se formalizará la recepción del producto/servicio solicitado al proveedor y los correspondientes criterios de aceptación.

Es importante aclarar dos aspectos sumamente importantes: (Mark C. Paulk, 2002)

1. Los proveedores no necesariamente pertenecen a otra empresa u organización; todo grupo que deba suministrar productos y servicios al proyecto e inclusive a otros proyectos pueden ser considerado como un proveedor. Entonces es necesario realizar los acuerdos con estos de manera formalizada.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

2. En caso que la solución que se le entregue al cliente llegara a necesitar algún tipo de producto COTS³ (por ejemplo: un motor de base de datos o algún tipo de componente puntual), también es necesario formalizar el acuerdo con el proveedor correspondiente a este caso.

4. Definiciones de Calidad de Software en la UCI.

“La UCI es un centro productivo, cuya misión es producir software y servicios informáticos a partir de la vinculación estudio–trabajo como modelo de formación. Es considerado la mayor organización productora de software en el país.” (CaliSoft, 2003)

Los proyectos de la UCI se rigen por el Aseguramiento de la Calidad de los modelos ISO/IEC 15504 y CMMI-DEV, cada proyecto define su estrategia de Aseguramiento de la Calidad en dependencia del tipo de producto que desarrolla y las metodologías y estándares de desarrollo que son definidos para la realización de dichos productos. (CaliSoft, 2003)

Lineamientos de calidad UCI.

Los lineamientos de calidad son una guía la cual deben seguir los proyectos de desarrollo para la producción de software atendiendo los artefactos que deben generar y procesos que se deben realizar para que el producto final obtenga la calidad que el cliente requiere. Debido a la metodología que se describe en este documento es ágil, no todos los lineamientos son aplicables a ella. (CaliSoft, 2003)

Generales

- Planificar capacitación para el personal del proyecto.
- Establecer registro de resultados de investigación del proyecto.
- Definir roles y responsabilidades.
- Definir las competencias.
- Definir equipos de proyecto.

³ **COT**: Commercial-of-the-Shelf. Productos de software disponibles para ser comprados y usados sin necesidad de realizar actividades de desarrollo.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Cuidar los bienes del cliente.

Ingeniería

- Gestionar requisitos.
- Desarrollar requisitos.
- Definir modelo de diseño.
- Definir estándares para el desarrollo del proyecto.
- Planificar y ejecutar pruebas del proyecto.
- Efectuar seguimiento de las No Conformidades
- Diseñar artefactos de prueba.

Gestión de proyecto.

- Estimar costo y esfuerzo del proyecto.
- Desarrollar plan del proyecto.
- Definir los resultados del proyecto.
- Gestión de recursos.

Soporte

- Definir glosario de términos.
- Definir configuración de la metodología a utilizar.

Auditoría y Revisiones.

Las auditorías y revisiones en la UCI se realizan mediante listas de chequeos, que permiten saber el lineamiento, las No Conformidades de la Prueba de Aceptación, la evidencia, el impacto, la ubicación, procedimientos, respuesta, descripción, fase y metodología de los proyectos.

5. Relación de XP con CMMI-DEV v1.2.

El área de Administración de Requerimientos es viable. Aunque XP no define un proceso completo para la gestión de requisitos, es posible construir una integración de elementos. Usando un artefacto de base (la Historia de Usuario) una organización pequeña pueda definir un modelo para la gestión de requisitos. XP proporciona elementos básicos para la gestión de requisitos. XP lleva mediante las Historias de Usuario un control sobre la trazabilidad de requisitos, se especifica cuál Historia de Usuario modifica a la Historia de Usuario que será modificada. El Cliente forma parte del equipo de desarrollo por lo que la comunicación es más fluida y es una ventaja. (Julio Ariel Hurtado Alegría, 2006)

El área de Planificación de Proyecto está cubierta por la metodología XP, ya que la planificación se aplica a pequeñas iteraciones y no es un documento necesario. La planificación es la base para el trabajo, pero los planes pueden cambiar, el objetivo es desarrollar un producto que satisfaga al cliente. Los planes se ajustan a lo largo del proyecto, se tiene en cuenta los riesgos dándole prioridad a las actividades y todos los participantes del proyecto están involucrados en estas actividades. (Julio Ariel Hurtado Alegría, 2006)

El área de Monitoreo y Control del Proyecto ejerce como parte de XP que es una metodología muy formal por lo que es posible llevar a cabo esta área. Se hacen reuniones diarias que permiten establecer un buen control y monitoreo sobre el avance del proyecto

El área de Medición y Análisis no se definen por ninguna metodología en particular, es necesario un catálogo activo de proceso que incluye técnicas, prácticas, elementos de proceso (roles, disciplinas, fases, ciclos de vida) que hacen posible la aplicación de los requisitos de una manera práctica.

Una estrategia a tener en cuenta es la velocidad del proyecto definido en la metodología XP, esta estrategia puede apoyar la calidad y la medición de la productividad.

El área de Aseguramiento de la Calidad del Producto y Procesos PPQA XP ofrece prácticas que hacen posibles un buen manejo de la calidad del producto y se realizan pruebas que forman parte de PPQA.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El área de Administración de la Configuración es la más débil ya que no llega ni al nivel 1 utilizando técnicas de las metodologías ágiles .Pero XP propone una técnica centrada principalmente en la gestión de configuración. Es posible extender estas técnicas para manejar los elementos de configuración y utilizar herramientas de apoyo en algunas de las tareas rutinarias.

El área de Administración de Acuerdos con Proveedores está presente en XP, pues selecciona sus proveedores teniendo en cuenta el potencial de estos, ya que considera que es de vital importancia que los productos o servicios suministrados por los proveedores al proyecto sean los mejores y tengan la calidad requerida.

6. Conclusiones

Si lo más importante es mejorar la forma de hacer el producto, entonces se puede decir que CMMI-DEV v 1.2 es un modelo que puede llevar al éxito a una empresa cualquiera, ya que apunta a la eficiencia operativa. Pero también es bueno aclarar que un determinado grado de madurez no conlleva el éxito a una organización, para ello está la estrategia de negocio y la capacidad que tenga el personal para ponerlo en práctica, por eso es importante seleccionar una metodología que se adapte de forma favorable a los requisitos del proyecto, como también es importante lograr un equilibrio entre la formalidad de nuestros procesos y la flexibilidad que necesitan quienes harán uso de esto finalmente.

XP es una metodología que pasa por alto la utilización de casos de uso y la producción de una extensa documentación. Permite entregar el software al cliente en el menor tiempo reduciendo los costos y garantizando la calidad dl producto. Las características de XP como metodología ágil le permite alcanzar el nivel 2 de madurez de CMMI v 1.2 para la mejora de la gestión de sus procesos.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

1. Introducción.

La propuesta de solución es la definición de un conjunto de procesos de mejora de CMMI-DEV v1.2 para el nivel 2 de madurez en proyectos de desarrollo que utilizan XP en la UCI, basada en el Programa de Mejora que se está realizando en la universidad y adecuada a los proyectos que utilizan metodología XP.

2. Definición de procesos.

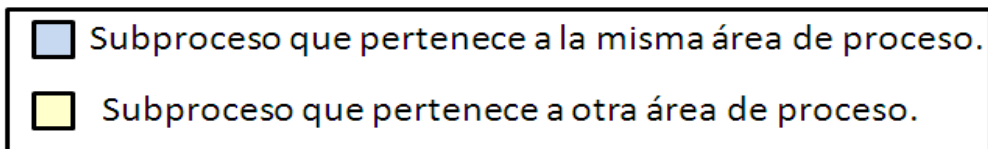


Figura 2 Leyenda para la descripción de los subprocesos.

Administración de Requerimientos (REQM).

Políticas: (Chrissis, 2009)

SG 1 Gestionar los requerimientos.

SP 1.1 Obtener una comprensión de los requerimientos.

Obtener y desarrollar con el cliente el entendimiento del significado de los requisitos de tal forma que los miembros del proyecto puedan comprometerse con estos.

SP 1.2 Obtener el compromiso sobre los requerimientos.

Negociar y registrar el compromiso de todos los involucrados (cliente, equipo de trabajo) con los requisitos que fueron especificados en el proyecto.

SP 1.3 Gestionar los cambios de los requerimientos.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

Administrar los cambios a los requisitos conforme vayan evolucionando en el proyecto, con el fin de mantenerlos documentados para garantizar el historial, justificación y evaluación del impacto de los cambios sucedidos a los requisitos desde el punto de vista de los afectados.

SP 1.4 Mantener la trazabilidad bidireccional de los requerimientos.

Mantener la trazabilidad de los requisitos con sus requisitos derivados y los productos de trabajo de forma bidireccional, asegurando la realización de todos los requisitos y la documentación de los mismos.

SP 1.5 Identificar las inconsistencias entre el trabajo del proyecto y los requerimientos.

Deben realizarse revisiones con el fin de identificar las inconsistencias entre el Plan de Publicaciones, los productos de trabajo y los requisitos y sus cambios. Debe identificarse la fuente de la inconsistencia, los cambios a realizar en los productos de trabajo y deben iniciarse las acciones correctivas pertinentes.

Recursos para ejecutar el proceso.

- Recursos humanos.
- Historias de Usuario para la especificación de requisitos funcionales y no funcionales.
- Herramienta Open Source Requirements Management Tool (OSRMT) para la trazabilidad.
- Computadoras.

Asignación de responsabilidades.

Rol	Responsabilidades
Cliente	Registra en la Historia de Usuario los requerimientos para que el equipo de proyecto los obtenga y los desarrolle. Proveer los requisitos a los miembros del proyecto. Participar en los encuentros coordinados por los miembros del proyecto. Define las prioridades, costo, tiempo y alcance de los requisitos de software. Validar requisitos del producto asignado.
Entrenador	Planificar la administración de los requisitos (conjuntamente con el programador y el cliente actualiza las Historias de Usuario y el Plan de Publicaciones con las actividades y recursos humanos referentes a requisitos). Participa en la validación de los proveedores de requisitos asegurándose que el cliente está de acuerdo con la selección del proveedor responsable.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

	<p>Coordina las reuniones con el cliente y la logística necesaria para realizar estos encuentros.</p> <p>Realizar reuniones de aceptación con el cliente.</p> <p>Analizar el efecto de los cambios en los elementos de la matriz.</p> <p>Asigna a los responsables de las inconsistencias para que las resuelvan y monitorea el cierre de estas inconsistencias.</p> <p>Revisar las inconsistencias entre los requisitos, los productos de trabajo y los planes del proyecto.</p>
Programador	<p>Analizar los requerimientos para asegurar que se cumplen los criterios establecidos.</p> <p>Alcanzar una comprensión de los requerimientos con el cliente para poder comprometerse con esto.</p> <p>Analizar el efecto de los cambios en los elementos de la matriz.</p> <p>Actualizar la matriz de trazabilidad.</p>

Involucrados relevantes
Internos.
Cliente.
Programador.
Entrenador.

Relación con otros procesos	
Administración de Requisitos (REQM)	
Planificación de Proyectos (PP).	Se describe la planeación de las actividades de administración de los requisitos.
Monitoreo y Control de Proyecto (PMC).	Se describe el proceso de seguimiento y control a las actividades y productos de trabajo de administración de requisitos y toma las acciones correctivas apropiadas.
Aseguramiento de la Calidad del Procesos y el Producto (PPQA).	Se realiza el aseguramiento de la calidad del producto y del proceso a la Administración de Requisitos.
Gestión de Configuración (CM).	Se describe el proceso de mantenimiento de líneas base y del control de cambios a la configuración de la documentación de los requisitos.

Subprocesos principales.

- Administración de Requisitos.
- Entendimiento y Compromiso.
- Traceo.
- Control de Inconsistencias.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

Diagrama de interacción entre los subprocesos principales.



Figura 3: Interacción entre los subprocesos principales.

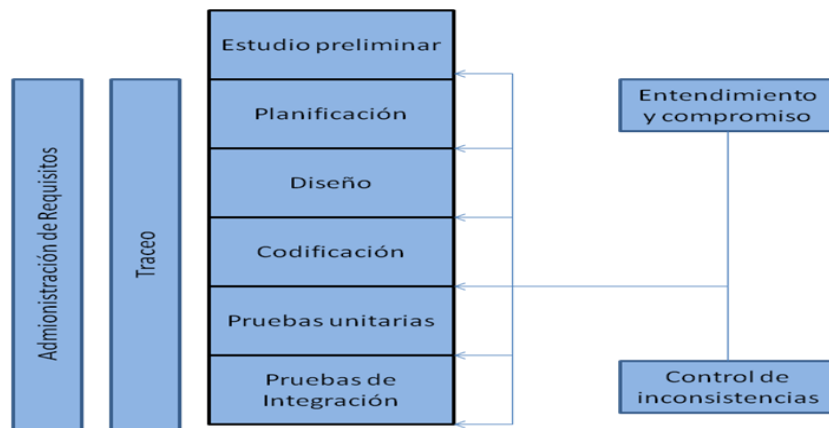


Figura 4: Relación con el ciclo de vida.

Descripción textual del subproceso Administración de Requisitos.

Administración de Requisitos				
Criterios de entrada.		Inicio del proyecto.		
Criterios de Salida.		Terminación del proyecto.		
No	Actividades.	Descripción	Entrada	Salida
1	Planificar Administración de Requisitos.	<p>Actualizar el Glosario de Términos (entrenador, programador, cliente).</p> <p>Actualizar las actividades concernientes a la administración de requisitos en el WBS y el Calendario (entrenador,</p>	Proyecto Técnico.	<p>Glosario de Términos.</p> <p>WBS y el Calendario.</p>

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

		encargado de seguimiento).		
2	Subproceso Traceo.	<i>Ver descripción del subproceso.</i>	Proyecto Técnico.	Matriz de Trazabilidad (actualizada).
3	Determinar proveedor de requisitos.	3.1 Facilitar los requisitos (cliente).		
4	Preparar entorno de administración de requisitos.	4.1 Coordinar reuniones con el equipo de proyecto (entrenador, cliente). 4.2 Preparar las herramientas de trabajo (rastreador).	Plan de Proyecto.	Herramientas Instaladas y Configuradas.
5	¿Si es necesario modelar el negocio?	Si el proyecto necesita modelar el negocio pasar a la actividad 6. En caso que el proyecto tenga un negocio definido ir a la actividad 8.		
6	Modelar el negocio.	6.1 Recibir documentación del negocio (programador, cliente). 6.2 Obtener el Modelo del Negocio con BPM (programador). 6.3 Documentar el Glosario de Términos (rastreador).	Glosario de Término.	Historias de Usuario. Modelo del Negocio con BPM. Glosario de Términos (actualizado).
7	Entendimiento y Compromiso.	<i>Ver descripción del subproceso.</i>	Modelo de Proceso de Negocio con BPM	
8	Obtener y especificar requisitos.	8.1 Identificar requisitos funcionales y no funcionales (programador, cliente). 8.2 Describir requisitos funcionales y no funcionales (programador, cliente). 8.3 Prioridad en negocio: Dar una prioridad a cada uno alta, media, baja (programador, cliente). 8.4 Pasar a la actividad 9.	Historia de Usuario.	Historia de Usuario (definida).
9	Entendimiento y	<i>Ver descripción del subproceso.</i>	Historia de	

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

	Compromiso.		Usuario.	
10	¿Si existen requisitos no entendidos?	10.1 En caso que algún requisito no sea entendido ir a la actividad 8. 10.2 En caso que sean entendidos ir a la actividad 11.	Historia de Usuario.	Historia de Usuario.
11	Traceo.	<i>Ver descripción del subproceso.</i>		
12	Actualizar Planes.	12.1 Actualizar el Plan de Proyecto con las nuevas actividades (entrenador, encargado de seguimiento).	Historia de Usuario. Plan de Proyecto.	Plan de Proyecto (actualizado).
13	Especificar requisitos del producto.	13.1 Definir artefactos a obtener en cada fase del proyecto (entrenador, todos los involucrados).	Historia de Usuario Matriz de Trazabilidad.	Historias de Usuarios Plan de Entrega. Plan de Publicaciones. Glosario de Términos. Iteraciones. Tarjetas CRC. Pruebas de Aceptación.
14	Entendimiento y Compromiso.	<i>Ver descripción del subproceso.</i>	Historia de Usuario.	
15	¿Existen requisitos no entendidos?	15.1 En caso que algún requisito no sea entendido ir a la actividad 13. 15.2 En caso que sean entendidos ir a la actividad 16.	Historia de Usuario	
16	Traceo.	<i>Ver descripción del subproceso.</i>	Historia de Usuario.	Matriz de Trazabilidad (actualizada).
17	Actualizar planes.	17.1 Actualizar el Plan de Proyecto (entrenador, encargado de seguimiento).	Historia de Usuario. Plan proyecto.	Plan de Proyecto (actualizado).

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

18	¿Es el fin del proyecto?	18.1 En caso que el proyecto haya finalizado ir a la actividad 19. 18.2 Si el proyecto no ha finalizado ir a la actividad 5.		
19	Terminación del Proyecto.	<i>Ver descripción del proceso Planeación del proyecto.</i>		

Descripción textual del subproceso Entendimiento y Compromiso

Entendimiento y Compromiso				
Criterios de entrada		Modelo de proceso de negocio con BPM, Historia de Usuario		
Criterios de Salida				
No	Actividades	Descripción	Entrada	Salida
1	¿Es el fin de la fase Planificación?	1.1 En caso que sea el fin de la fase de Planificación ir a la actividad 2. 1.2 Si no es el fin de la fase ir a la actividad 5.		
2	Revisar elementos del negocio con el cliente.	2.1 Realizar una reunión con el cliente y revisar el modelado del negocio (entrenador, programador, cliente). 2.2 Especificar en la reunión si hay cambios sugeridos por el cliente (entrenador, cliente, programador).	Modelo de Negocio con BPM.	
3	¿Hay cambios sugeridos?	3.1 Si se sugieren cambios a los documentos revisados ir a la actividad 4. 3.2 Si no ir a fin.		
4	Actualizar elementos del negocio.	4.1 Se modifican los elementos del negocio a los que se le hayan sugerido cambios.	Modelo de Negocio con BPM.	Modelo de Negocio con BPM (actualizado).
5	Analizar requisitos.	5.1 Estudiar la documentación obtenida (entrenador, programador)	Historia de Usuario.	
6	Revisar con el cliente.	6.1 Realizar una reunión con él cliente para realizar la aceptación de las sugerencias de los programadores con respecto a los requisitos no entendidos que se especificarían	Historia de Usuario.	

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

		nuevamente o serian eliminados del ciclo de vida del software (entrenador, programador).		
7	¿Se aceptan los requisitos sugeridos?	7.1 Si se aceptan los requisitos sugeridos ir a la actividad 8. 7.2 En caso que no se acepten salir del proceso.		
8	Actualizar la Historia de Usuario.	8.1 Actualización de la Historia de Usuario con los requisitos aceptados (a sugerencia del programador) ir a fin.	Historia de Usuario (aceptada)	Historia de Usuario (actualizada)
9	Aceptar requisitos.	9.1 Los miembros del proyecto se comprometen aceptando lo especificado en la Historia de Usuario (todos los involucrados en el proyecto).	Historia de Usuario (actualizada).	Historia de Usuario (aceptada).

Descripción textual del subproceso Traceo.

Traceo				
Criterios de entrada		Solicitud de cambio: La Historia de Usuario		
Criterios de Salida		Matriz de Trazabilidad		
No	Actividades	Descripción	Entrada	Salida
1	¿Hay una solución de cambio o es un nuevo elemento?	Si existe una solicitud de cambio o la inserción de un nuevo elemento ir a la actividad 2. Si no existe solicitud de cambio ni la inserción de un nuevo elemento salir del proceso (ir a fin).		
2	Analizar el efecto de la trazabilidad.	2.1 Identificar los requisitos alterados por la trazabilidad (entrenador, rastreador, programador). 2.2 Analizar el efecto de los cambios en los elementos de la matriz (entrenador, rastreador, programador).		
3	¿El cambio o inserción esta aprobado?	3.1 Si la solicitud de cambio o la inserción de un nuevo elemento está aprobada ir a la actividad 4. 3.2 En caso que no se haya aprobado la solicitud de cambio o la inserción de un nuevo elemento salir del proceso (fin).		

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

4	¿Es un cambio?	4.1 si es un cambio ir a la actividad 5. 4.2 si es la inserción de un nuevo elemento ir a la actividad 6.		
5	Actualizar matriz de trazabilidad.	5.1 Actualizar los atributos de los elementos y la dirección de trazabilidad entre estos (programador).		Matriz de Trazabilidad (actualizada).
6	Inserción de un nuevo elemento.	6.1 Si es una inserción definir los atributos de trazabilidad de cada elemento, definir la dependencia de los elementos y generar la matriz de trazabilidad (programador).	Historia de Usuario.	Matriz de Trazabilidad (actualizada).

Descripción textual del subproceso Control de Inconsistencias.

Control de Inconsistencias				
Criterios de entrada		Matriz de Trazabilidad.		
Criterios de Salida		Productos de trabajo consistentes.		
No	Actividades	Descripción	Entrada	Salida
1	¿Es el fin de una fase?	Si es el fin de una fase ir a la actividad 2. Si no es el fin de una fase salir del proceso (ir a fin).		
2	Revisar las inconsistencias.	2.1 Revisar si el producto de trabajo es consistente con los requisitos que fueron especificados en la matriz y los planes (encargado de pruebas).	Matriz de Trazabilidad. Planes.	
3	¿Existen inconsistencias?	3.1 Si se encuentran inconsistencias durante la revisión ir a la actividad 5. 3.2 En caso que no existan inconsistencias salir de la fase.		
4	Asignar responsables para resolver las inconsistencias.	4.1 El entrenador asigna a los responsables del producto de trabajo las inconsistencias identificadas y orienta que deben definir las acciones correctivas a ejecutar para resolverlas.		
5	Resolver inconsistencias.	5.1 Los responsables resuelven las inconsistencias y se lo reportan al entrenador (todos los involucrados).		Planes (actualizados).
6	Monitorear inconsistencias y aceptar cambios.	6.1 El responsable de resolver las inconsistencias cuando las soluciona se lo comunica al entrenador y al encargado de pruebas y les solicita autorización para	Matriz de trazabilidad. Planes	

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

		actualizar la Matriz de Trazabilidad con los nuevos cambios, este chequea que las inconsistencias fueron resueltas.	(actualizados).	
7	¿Hay inconsistencias sin resolver o cambios no aceptados?	7.1 En caso que hayan inconsistencias sin resolver o cambios no aceptados regresar a la actividad 5. 7.2 En caso que se hayan resuelto todas las inconsistencias o los cambios hayan sido aceptado ir a la actividad 8.		
8	Traceo.			Matriz de Trazabilidad.

Planificación del Proyecto (PP).

Políticas: (Chrissis, 2009)

SG 1 Establecer estimaciones.

SP 1.1 Estimar el alcance del proyecto.

Establecer un cronograma (Plan de Proyecto) para estimar el alcance que tendrá el proyecto, identificar las tareas del cronograma con suficiente detalle para poder estimar tiempo, esfuerzo y los responsables de realizarla.

SP 1.2 Establecer las estimaciones de los atributos del producto de trabajo y de las tareas.

Establecer y mantener los estimados de los atributos de los productos de trabajo y de las tareas en cuanto a complejidad, tamaño, costo y tiempo.

SP 1.3 Definir el ciclo de vida del proyecto.

Definir las fases del Ciclo de Vida del proyecto a considerar en la planeación, estas fases serán definidas de acuerdo al alcance de los requisitos, estimado de recursos y la naturaleza del proyecto.

SP 1.4 Determinar las estimaciones de esfuerzo y de coste.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

Estimar con fundamentos (utilizando un método de estimación) el costo y esfuerzo de los productos de trabajo y de las tareas del proyecto.

SG 2 Desarrollar un Plan de proyecto.

SP 2.1 Establecer el presupuesto y el calendario.

Establecer y mantener el presupuesto y Plan de Publicaciones, identificar los hitos principales, suposiciones, restricciones, dependencias entre tareas.

SP 2.2 Identificar los riesgos del proyecto.

Identificar, documentar y analizar los riesgos del proyecto.

SP 2.3 Planificar la gestión de los datos.

Planear la gestión de los datos del proyecto (Los datos son cualquier forma de documentación: manuales, presentaciones, correo electrónico, medios impresos, etc.), estos incluyen tanto los entregables como los datos de uso interno.

SP 2.4 Planificar los recursos del proyecto.

Planificar los recursos tanto humanos y materiales necesarios para la ejecución del proyecto.

SP 2.5 Planificar el conocimiento y las habilidades necesarias.

Planificar las capacitaciones necesarias a todo el personal del proyecto para obtener experiencia y conocimientos apropiados para el proyecto (tener en cuenta tanto la capacitación del personal actual como la subcontratación o reclutamiento de personal experimentado). Incorporar las actividades de contratación y capacitación al plan del proyecto.

SP 2.6 Planificar la involucración de las partes interesadas.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

Planificar la participación de los involucrados relevantes, identificar las actividades principales, en cuáles son necesarios y en cuáles son afectados.

SP 2.7 Establecer el Plan de Proyecto.

Establecer y mantener el Plan de Proyecto.

SG 3 Obtener el compromiso con el plan.

SP 3.1 Revisar los planes que afectan al proyecto.

Revisar todos los planes que afectan el proyecto con los niveles establecidos (equipo de trabajo, alta gerencia y cliente) para poder establecer un compromiso con el plan.

SP 3.2 Reconciliar los niveles de trabajo y de recursos.

Conciliar las diferencias entre los recursos estimados y los recursos disponibles para el proyecto.

SP 3.3 Obtener el compromiso con el plan.

Obtener el compromiso con los involucrados relevantes sobre la ejecución y soporte al Plan de Publicaciones. Se deben revisar los compromisos internos y externos con la gerencia y documentarlos.

Asignación de responsabilidades

Rol	Responsabilidades
Rastreador	Establecer el Plan de Administración de la Configuración.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

Rol	Responsabilidades
Alta gerencia	<p>Legalizar el proyecto.</p> <p>Firmar el acta de inicio del proyecto.</p> <p>Suministrar al cliente información relevante para desarrollar el Proyecto Técnico.</p> <p>Asignar recursos al proyecto.</p> <p>Presentar al cliente acta de terminación.</p> <p>Establecer el Plan de Acuerdos con los Proveedores.</p> <p>Revisar los planes de los proyectos con el proyecto y clientes.</p>
Programador	<p>Definir propuesta de solución del software para el Proyecto Técnico.</p> <p>Definir características técnicas de la solución para el Proyecto Técnico.</p> <p>Definir arquitectura de la solución para el Plan de Entregas.</p>
Cliente	<p>Revisar y aprobar entregables del proyecto (Plan de Entregas).</p>
Gestor	<p>Realizar la Capacitación del personal.</p> <p>Acordar fechas de realización de las revisiones de PPQA.</p>
Grupo revisor	<p>Revisar y aprobar el Proyecto Técnico (Plan de Proyecto, Plan de Entregas).</p>
Entrenador	<p>Elaborar el proyecto Técnico.</p> <p>Establecer el cronograma de las actividades del proyecto.</p> <p>Identificar restricciones y dependencias en el cronograma.</p> <p>Identificar riesgos.</p> <p>Realizar las estimaciones del proyecto.</p> <p>Desarrollar el Plan de Proyecto.</p> <p>Revisión de los planes del proyecto.</p> <p>Recoger compromiso de los involucrados con los planes del proyecto.</p>

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

Rol	Responsabilidades
Encargado de seguimiento	Elabora el plan de recursos materiales. Establecer el Plan de Monitoreo. Establecer el Plan de Mediciones. Establecer el Plan de Administración de la Calidad. Establecer el Plan de Administración de la Configuración. Establecer el Plan de Administración de Requisito. Establecer el Plan de PPQA.

Involucrados relevantes.

Involucrados internos:	Involucrados externos:
Cliente	Alta gerencia
Encargado de seguimiento	Grupo revisor
Programador	
Gestor	
Entrenador	
Rastreador	

Relación con otros procesos.

	Planeación de Proyectos (PP)
Administración de Requisitos (REQM).	Se describe la planeación de las actividades de administración de requisitos y REQM brinda información acerca de los requisitos necesarios para la planeación y re planeación.
Monitoreo y Control de Proyecto (PMC).	Se describe la planeación de las actividades de monitoreo y control.
Aseguramiento de la Calidad del Proceso y el Producto (PPQA).	Se describe la planeación de las actividades de aseguramiento de la calidad.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

Administración de la Configuración (CM).	Se describe la planeación de las actividades de administración de la configuración.
Medición y Análisis (MA).	Se describe la planeación de las actividades de medición y análisis.
Administración de Acuerdo con Proveedores (SAM).	Se describe la planeación de las actividades de administración de acuerdo con proveedores.

Herramientas para llevar a cabo este proceso.

- Recursos humanos.
- Herramientas Redmine o Project.net para la gestión del proyecto.
- Herramienta Método de Estimación UCI para la estimación del proyecto.
- Computadoras.

Principales subprocesos.

- Planeación del Proyecto.
- Estudio Preliminar.
- Estimación.
- Elaboración del Proyecto Técnico.
- Definición del Plan del Proyecto.
- Planeación de otros Planes.
- Compromiso al Plan.

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

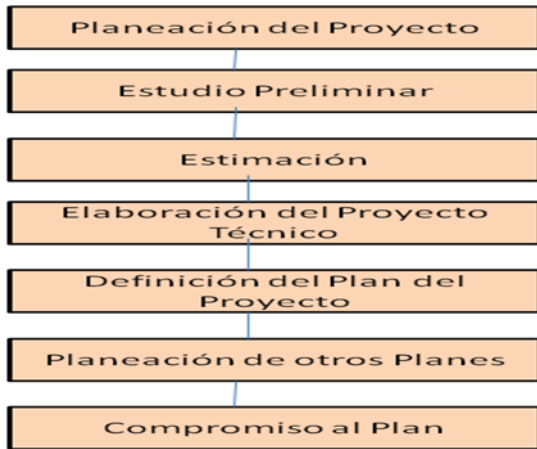


Figura 5 Interacción entre los subprocesos principales.

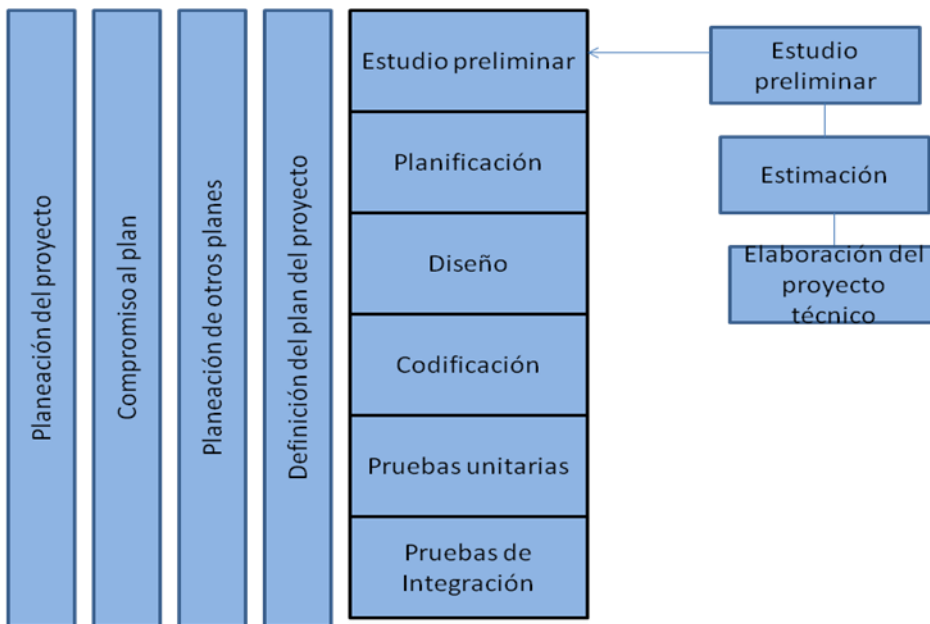


Figura 6 Relación con el ciclo de vida.

Descripción textual del subproceso Planeación del Proyecto.

Planeación del proyecto		
Criterios entrada	de	Inicia el proyecto para satisfacer las necesidades del cliente

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

Criterios de salida		Se generó el Plan de Proyecto, incluyendo otros planes para obtener de esta manera el compromiso de todos los involucrados al mismo tiempo.		
No	Actividades	Descripción	Entrada	Salida
1	Planear la planeación.	<p>1.1 El entrenador define un calendario para las principales actividades del proceso de planeación del proyecto. Debe incluirse:</p> <p>Planear la planeación.</p> <p>La realización del estudio preliminar.</p> <p>La planeación y asignación de recursos del proyecto.</p> <p>La planeación del levantamiento de requisitos.</p> <p>La definición del Plan del Proyecto.</p> <p>El establecimiento de compromisos.</p> <p>La actualización del plan.</p> <p>1.2 Se definen los responsables para cada actividad y a los mismos se les asigna un tiempo de realización (entrenador).</p>	Ficha Técnica.	<p>WBS</p> <p>Calendario (inicial con las actividades de planeación).</p>
2	Estudio Preliminar.	Ver descripción del subproceso.		
3	¿El Proyecto Técnico es aceptado?	<p>3.1 Si se acepta el proyecto técnico ir a la actividad 4.</p> <p>3.2 Si no es ir a fin.</p>	Ficha Técnica.	Proyecto Técnico.
4	¿Se inicia la ejecución del proyecto?	<p>4.1 Si se inicia el proyecto pasar a la actividad 6.</p> <p>4.2 Si no mantenerse en la actividad 5 hasta que el proyecto inicie.</p>		

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

5	Planear y asignar recursos al proyecto.	5.1 Se solicita a la alta gerencia los recursos.	Plan de Proyecto. WBS y Calendario (detallado). Proyecto Técnico (aprobado).	WBS y Calendario (actualizado). Plan de Proyecto (Plan de Personal). Planes y Registro de Monitoreo (Plan de Recursos Materiales actualizado).
6	Planear y levantamiento de negocio y requisitos.	Se actualiza el WBS y Calendario con las actividades del negocio y requisitos (entrenador).	Proyecto Técnico (aprobado) WBS y Calendario (alto nivel)	WBS y Calendario (actualizado con las actividades del negocio y requisitos)
7	Modelar el negocio.	La descripción de esta actividad pertenece al procesos de Desarrollo de Requisitos del nivel 3		
8	Levantamiento de los requisitos.	La descripción de esta actividad pertenece al procesos de Desarrollo de Requisitos del nivel 3		
9	Definición del Plan de Proyecto.	Ver descripción de esta actividad.	Historia de Usuario. Proyecto Técnico.	Plan de Proyecto.
10	Compromiso al plan.	Ver descripción de esta actividad.	Plan de Proyecto.	
11	¿Se realizó el compromiso?	11.1 Si se ha realizado el compromiso ir a la actividad 12. 11.2 De no haberse realizado ir a la actividad 9.		
12	¿Hay algún cambio?	12.1 Si se ha realizado algún cambio ir a la actividad 13. 12.2 De no haberse realizado ir a la actividad 14		

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

13	Actualizar plan.	13.1 Se actualizan los elementos del Plan de Proyecto dependientemente de los cambios ocurridos (entrenador, encargado de seguimiento).	Plan de Proyecto.	Plan de Proyecto (actualizado).
14	¿Es el fin del proyecto?	14.1 Si es el fin del proyecto para la actividad 14.2 Si no regresar a la actividad 9.		

Descripción textual del subprocesos Estudio Preliminar.

Estudio Preliminar				
Criterios de entrada		Plan de Publicaciones		
Criterios de salida		Proyecto Técnico		
No	Actividades	Descripción	Entrada	Salida
1	Planificar Estudio Preliminar.	<p>1.1 La alta gerencia conjuntamente con el entrenador, planifican las actividades necesarias para realizar el estudio preliminar.</p> <p>1.3 Se establecen los tiempos para la elaboración y aceptación del Proyecto Técnico (alta gerencia, entrenador).</p> <p>1.4 Si es necesario realizar viajes al extranjero se planifica la tramitación correspondiente (alta gerencia).</p>	<p>Ficha Técnica.</p> <p>WBS y Calendario (inicial).</p>	WBS y Calendario.
2	Elaboración del Proyecto Técnico	Ver descripción de esta actividad.	Ficha Técnica.	Proyecto Técnico.
3	Estimación.	Ver descripción de esta actividad.	Proyecto Técnico.	Reporte de estimación de tamaño, tiempo, costo y esfuerzo.
4	Revisión del Proyecto Técnico	<p>El grupo revisor analiza el Proyecto Técnico y determina si es conveniente ejecutar el proyecto en los términos establecidos.</p> <p>2.2 Revisa la calidad de los documentos, su completitud y profundidad (grupo revisor).</p>	Proyecto Técnico.	Proyecto Técnico (revisado).
5	¿Se aprobó la revisión del	5.1 Si se aprueba el Proyecto Técnico ir a la		

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

	proyecto técnico?	actividad 7. 5.2 En caso de no ser aprobado ir a la actividad 6.		
6	¿Redefinir el Proyecto Técnico?	6.1 Si es necesario redefinir elementos del proyecto técnico retornar a la actividad 2. 6.2 Si no se redefinirá el proyecto técnico e ir a fin.		
7	Revisión con el cliente.	7.1 Se revisa el proyecto técnico con el cliente, el mismo chequea que el proyecto técnico se ajusta a sus necesidades, cumpla sus expectativas y este dentro del presupuesto (entrenador, cliente, alta gerencia).	Proyecto Técnico.	Proyecto Técnico (revisado por el cliente).
8	¿Se acepto el Proyecto técnico?	8.1 Si se acepta el proyecto técnico ir a la actividad 8.2 De no ser aceptado ir a la actividad 6.		
9	Aceptación del Proyecto Técnico.	9.1 ambas partes aceptan el proyecto técnico (entrenador, cliente, alta gerencia)	Proyecto Técnico.	Proyecto Técnico (aceptado)

Descripción textual del subproceso Estimación.

Estimación				
Criterios de entrada		Proyecto Técnico		
Criterios de salida		Reporte de estimación de tamaño, tiempo, costo y esfuerzo		
No	Actividades	Descripción	Entrada	Salida
1	Determinar paquetes funcionales.	El entrenador se entrevista con el cliente para definir los paquetes funcionales del sistema. 1.2 El entrenador registra en la herramienta del Método de Estimación el listado de los paquetes funcionales.	Proyecto Técnico.	Lista de Paquetes Funcionales.
2	Determinar complejidad de paquetes funcionales.	2.1 El entrenador mediante la entrevista con el cliente determina la complejidad de los paquetes funcionales (grande, mediano o pequeño). 2.2 El entrenador registra en la herramienta del Método de Estimación la complejidad de los paquetes funcionales (grande, mediano o pequeño).	Lista de Paquetes Funcionales.	Lista de Paquetes Funcionales (con la estimación del atributo complejidad).
3	Determinar cantidad y complejidad	3.1 El entrenador en la entrevista con el cliente hace una explicación de los factores de complejidad (técnica, valor agregado, ambiente)		Lista de Factores (con el atributo

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

	de los factores.	que pueden incidir en las estimaciones, así como la necesidad de evaluarlos (muy alto, alto, medio, medio bajo, bajo) (sí, no) y cuantificarlos en los casos que lo requiera. 3.2 El entrenador obtiene las informaciones necesarias en conjunto con el cliente para determinar la complejidad de los factores. 3.3 El entrenador registra en la herramienta del Método de Estimación la complejidad y cantidad de los factores de complejidad en los casos que lo requiera.		cantidad y/o complejidad).
4	Obtener estimaciones de tamaño, tiempo, esfuerzo y costo basado en datos históricos.	4.1 El entrenador obtiene de la Herramienta del Método de Estimación el reporte de las estimaciones de tamaño, tiempo, costo y esfuerzo. 4.2 El entrenador muestra al cliente el reporte de las estimaciones de tamaño, tiempo, costo y esfuerzo.		Reporte de tamaño del sistema, tiempo de las actividades, esfuerzo del personal del proyecto y costo del proyecto.

Descripción textual del subproceso Elaboración del proyecto técnico.

Elaboración del proyecto técnico				
Criterios de entrada		Ficha Técnica		
Criterios de salida		Proyecto Técnico		
No	Actividades	Descripción	Entrada	Salida
1	Establecer marco de referencia.	1.1 El entrenador describe los antecedentes del proyecto, el problema a resolver, enuncia de modo general la solución propuesta y describe los beneficiarios y los impactos esperados de la solución.	Ficha Técnica.	Proyecto Técnico (actualizado).
2	Caracterizar entorno de aplicación.	2.1 El entrenador describe el entorno para el cual se desarrolla la solución. Para esto obtiene del cliente información respecto a las características del negocio. Si es una institución describe la base reglamentaria que debe acatar la solución. 2.2 El entrenador describe además el entorno donde se explotará la solución. Con esta información puede posteriormente identificar riesgos.	Proyecto Técnico.	Proyecto Técnico (actualizado).

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

3	Definir la solución.	<p>3.1 El entrenador de conjunto con el programador define la solución y a partir de un conocimiento profundo del problema a resolver y del entorno de la aplicación, define el tipo de solución tecnológica y sus componentes (si es una solución tecnológica integral desarrollada a la medida, los componentes que la integran, si es un producto terminado que se implanta).</p> <p>3.2 El entrenador y el programador describen además los objetivos y el alcance de la solución, las áreas del negocio involucradas y define los criterios de éxito.</p>	Proyecto Técnico.	Proyecto Técnico (actualizado).
4	Definir requisitos técnicos.	<p>El entrenador y el programador partiendo de las funcionalidades que debe ofrecer el sistema determinan las características técnicas.</p> <p>Las características técnicas de la solución podrían enunciarse a partir de los siguientes elementos, según aplique para cada solución particular:</p> <ul style="list-style-type: none"> • Desempeño. • Plataforma de desarrollo. • Características de la información que se maneja. • Requisitos sobre la interfaz de usuario. • Aspectos legales que se deben respetar. • Tamaño de la solución. • Necesidad de conexión con sistemas legados. • Necesidad de interacción con dispositivos externos. • Necesidad de migrar datos de un sistema anterior. • Nivel de definición de los procesos a informatizar. • Modelo de distribución de los servicios del sistema. • Estimación del volumen de datos a manejar. • Nivel de demanda de servicios concurrentes. • Nivel de parametrización del sistema mecanismos de aseguramiento de la integridad de la información. 	Proyecto Técnico.	Proyecto Técnico (actualizado).

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

6	Definir modelo de desarrollo.	<p>6.1 El entrenador y el programador teniendo en cuenta el tipo de aplicación, las características del entorno de aplicación y la complejidad de la solución, usando además metodologías de desarrollo como referencia y acatando las políticas de calidad pautadas por la organización, definen el modelo de desarrollo a emplear.</p> <p>De manera general debe hacerse énfasis en los siguientes elementos según aplique:</p> <ul style="list-style-type: none"> • Metodología de referencia. • Entregables por disciplina. • Objetivos y alcances por etapas. • Tiempos para las disciplinas del proceso de desarrollo. • Estrategia de validación, pruebas y aceptación de la solución. <p>6.2 El entrenador elabora un Plan de Publicaciones con las actividades generales del proceso de desarrollo.</p>	Proyecto Técnico.	<p>Proyecto Técnico (actualizado).</p> <p>WBS y Calendario (alto nivel).</p>
7	Identificar Riesgosa.	<p>7.1 El entrenador identifica los riesgos que pueden obstaculizar la correcta ejecución de cada una de las disciplinas del desarrollo. Toma en cuenta los factores de complejidad de la solución, las características del negocio y el entorno de aplicación.</p> <p>7.2 Se realiza una reunión con todos los miembros del proyecto para asignar a las personas que mitigaran los riesgos.</p>	Proyecto Técnico.	<p>Proyecto Técnico (actualizado).</p> <p>Plan de Riesgo (actualizado)</p>
8	Definir organización del proyecto.	<p>8.1 El entrenador define la estructura organizativa del proyecto.</p> <p>8.2 El entrenador define como se realizarían las reuniones con los involucrados para garantizar la comunicación entre estos y los compromisos que cada parte asume para la ejecución de la solución.</p>	Proyecto Técnico.	<p>Proyecto Técnico (actualizado con la estructura organizativa del proyecto).</p>
9	Elaborar Plan de Gestión de Datos.	<p>9.1 El entrenador define la política a seguir para garantizar la confidencialidad y protección de los datos e información suministrados por el cliente, necesarios para la construcción y prueba de la solución.</p>	Proyecto Técnico.	<p>Proyecto Técnico (actualizado con el Plan.</p>

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

					de Gestión de Datos).
--	--	--	--	--	-----------------------

Descripción textual del subproceso Definición del Plan del Proyecto.

Definición del Plan del Proyecto					
Criterios de entrada					
Criterios de salida					
No	Actividades	Descripción	Entrada	Salida	
1	Identificar dependencias y restricciones.	<p>1.1 En el WBS se determinan las dependencias entre las actividades del mismo (entrenador, encargado de seguimiento).</p> <p>1.2 Se definen las restricciones del proyecto en el Plan de Proyecto (entrenador, encargado de seguimiento).</p>	WBS y Calendario (con las actividades de los requisitos).	<p>Plan de Proyecto (inicial con las restricciones).</p> <p>WBS y Calendario (dependencias identificadas).</p>	
2	Desarrollar Plan de Iteraciones.	<p>2.1 El entrenador elabora el Plan de la Iteración (a partir de las dependencias obtenidas)</p> <p>2.2 El entrenador junto al planificador actualiza el Plan de Proyecto con las características principales de las iteraciones.</p> <p>2.3 El entrenador detalla el Plan de entrega con las actividades de la iteración y los responsables de ejecutarlas.</p>	<p>Plan de Proyecto</p> <p>WBS y Calendario (dependencia y restricciones identificadas).</p>	<p>Plan de Proyecto (con las características principales de las iteraciones)</p> <p>Plan de Iteraciones WBS y Calendario (detallado)</p>	
3	Planeación de Otros Planes.	Ver descripción del subproceso.	<p>WBS y Calendario (detallado).</p> <p>Plan de Proyecto.</p>	Planes.	
4	Identificar riesgos.	3.1 El entrenador identifica los riesgos que pueden afectar al desarrollo del proyecto y se lo comunica a todos los involucrados en el proyecto.	Plan de Proyecto.	Planes y Registro de Monitoreo (Plan de Riesgo).	
5	Refinar WBS y Calendario.	4.1 El entrenador junto al encargado de seguimiento refina el WBS y Calendario teniendo en cuenta informaciones como los riesgos, las dependencias entre las	WBS y Calendario (detallado).	WBS y Calendario (refinado).	

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

	actividades entre otras.		
--	--------------------------	--	--

Descripción textual del subproceso Planeación de Otros Planes.

Planeación de Otros Planes				
Criterios de entrada				
Criterios de salida				
No	Entrada	Descripción	Entrada	Salida
1	Elaborar o realizar el Plan de Capacitación.	1.1 Si aún no existe el Plan de Capacitación se elabora y si ya existe se actualiza en dependencia de las necesidades de la iteración (gestor, entrenador).	Plan de Proyecto. WBS y Calendario (detallado).	Plan de proyecto. (Plan de capacitación).
2	Elaborar o actualizar Plan de Monitoreo.	2.1 Si aún no existe el Plan de Monitoreo se elabora, teniendo en cuenta las fechas de las entregas, de las iteraciones para el monitoreo y control de las principales entregas. Si ya existen se actualizan en dependencia de las necesidades de cada iteración (entrenador).	Plan de Proyecto WBS y Calendario (detallado)	Plan de Proyecto (Plan de Monitoreo)
3	Elaborar o actualizar Plan de Mediciones.	3.1 Si no existe el Plan de Mediciones se elabora teniendo en cuenta las métricas de la organización y del proyecto y si ya existe se actualiza en dependencia de las necesidades de la iteración (entrenador, encargado de pruebas, encargado de seguimiento).	Plan de Proyecto WBS y Calendario (detallado).	Plan de Mediciones.
4	Elaborar o actualizar Plan de Gestión de la Calidad.	4.1 Si aún no existe el Plan de Gestión de la Calidad se elabora y si ya existe se actualiza en dependencia de las necesidades de la iteración (entrenador, encargado de pruebas, gestor).	Plan de Proyecto WBS y Calendario (detallado).	Plan de Proyecto (Plan de Gestión de la Calidad).
5	Elaborar o actualizar Plan de Acuerdo con Proveedores.	5.1 Si aún no existe el Plan de Administración de Acuerdos con Proveedores se elabora y si ya existe se actualiza con los nuevos datos de la iteración en caso que sea necesario (entrenador, alta gerencia).	Plan de Proyecto WBS y Calendario (detallado).	Plan de Proyecto (Plan de Administración de Acuerdos con Proveedores).
6	Elaborar o actualizar Plan de Administración de la	6.1 Si aún no existe el Plan de Administración de la Configuración se elabora y si ya existe se actualiza con los nuevos datos de la iteración, en caso que sea necesario (entrenador, rastreador).	Plan de Proyecto. WBS y Calendario	Plan de Proyecto (Plan de Administración de la

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

	Configuración		.(detallado)	configuración).
7	Elaborar o actualizar Plan de Administración de Requisitos.	7.1 Si aún no existe el Plan de Administración de Requisitos se elabora y si ya existe se actualiza con los nuevos datos de la iteración, en caso que sea necesario (entrenador, programador).	Plan de proyecto. WBS y Calendario (detallado).	WBS y Calendario (actualizado con las actividades de requisitos, tiempo de duración y los responsables).
8	Elaborar o actualizar Plan de Gestión de Datos	8.1 Si aún no existe el Plan Gestión de Datos se elabora y si ya existe se actualiza con los nuevos datos de la iteración, en caso que sea necesario (entrenador, rastreador).	Plan de Proyecto WBS y Calendario (detallado).	Planes y Registro de Monitoreo (Plan de Gestión de Datos).
9	Elaborar o actualizar Plan de Involucrados.	9.1 Si aún no existe el Plan de Involucrados se elabora y si ya existe se actualiza con los nuevos datos de la iteración, en caso que sea necesario (entrenador, alta gerencia).	Plan de Proyecto WBS y Calendario (detallado)	Plan de Proyecto (Plan de Involucrados).

Descripción textual del subproceso Compromiso al Plan.

Compromiso al plan				
Criterios de entrada				
Criterios de salida				
No	Actividades	Descripción	Entrada	Salida
1	Revisión de Planes por el equipo y entrenador.	1.1 El entrenador revisa los Planes con el equipo de trabajo para obtener un compromiso con estos.	Planes: Plan de Proyecto. WBS y Calendario. Planes y Registro de Monitoreo	Planes: (revisados por el equipo de trabajo).

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

2	Revisión de Planes por la alta gerencia y Conciliación de recursos estimados.	<p>2.1 El entrenador revisa los planes con la alta gerencia.</p> <p>2.2 El entrenador realiza una conciliación de los recursos con que cuenta y los que necesita.</p>	Planes (revisado por equipo de trabajo).	Planes (revisado por alta gerencia).
3	¿Hay diferencias entre los asignados y los estimados?	<p>3.1 Si la gerencia no asigna al proyecto los recursos que necesita y por tanto hay diferencias entre los recursos estimados y asignados, ir a la actividad 4.</p> <p>3.2 Si la gerencia asigna al proyecto todos los recursos que necesita y no hay diferencia entre los recursos estimados y asignados entonces se ejecuta la actividad 5.</p>		
4	Realizar acciones consecuentes a los recursos asignados.	<p>4.1 El entrenador realiza acciones consecuentes con los recursos actuales.</p> <p>4.2 Actualiza los Planes con las acciones a tomar.</p>	Planes (revisado por alta gerencia).	Planes (actualizado acorde a los recursos asignados).
5	Revisión de compromisos por cliente	5.1 El entrenador en una reunión con el cliente realiza una revisión de los compromisos que tiene con el cliente, las fechas pactadas y entregables y le comunica el estado del proyecto.		
6	¿Se realizó el compromiso?	<p>6.1 En caso que el cliente esté de acuerdo con lo que el entrenador le plantea acerca de los planes del proyecto, lo acepte y se comprometa a esto, ir a la actividad 7.</p> <p>6.2 En caso que los miembros del proyecto se comprometan a los planes ir a la actividad 8.</p> <p>6.3 En caso que la alta gerencia esté de acuerdo con lo planteado en los planes del proyecto y se comprometa ir a la actividad 9.</p> <p>6.4 En caso que no se realice cualquiera de los compromisos anteriores ir a fin.</p>		
7	Compromiso por parte del cliente.	7.1 El cliente e involucrados aceptan los compromisos contraídos con el proyecto. Estos acuerdos y compromisos se establecen en la reunión.		

CAPÍTULO 2: DESARROLLO DE LA PROPUESTA

8	Comprometerse a los planes del proyecto por parte del equipo de trabajo.	8.1 El entrenador convoca una reunión donde informa a los miembros del equipo de trabajo que se ven afectados por los Planes. 8.2 Los miembros del equipo de trabajo se comprometen con los Planes ya revisados.	Planes (revisados por equipo de trabajo).	
9	Comprometerse a los planes del proyecto por parte de la alta gerencia.	9.1 La alta gerencia se compromete con los Planes revisados en la reunión.		

3. Conclusiones.

Cada proceso en particular es único ya que depende de los objetivos y la estructura de la organización donde se aplica. Es importante definir los procesos de una manera estructurada y organizada y la prestación de una serie de directrices, de modo que una pequeña organización pueda adoptarlas y adaptarlas para poder alcanzar una certificación del nivel 2 de madurez de CMMI v 1.2.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

4. Introducción.

El principal objetivo de la propuesta de la presente investigación, es lograr la optimización permanente en los procesos relacionados a través de un camino de varios estados de evolución/aprendizaje llamados: “Niveles de Madurez”.

En este capítulo se aplica la propuesta de mejora del nivel 2 de madurez de CMMI-DEV v 1.2 a un caso de estudio, que corresponde a una aplicación web que permitirá crear y editar mapas isométricos para el desarrollo de juegos en línea.

5. Caso de estudio.

En la UCI se hace engorroso el desarrollo de entornos tridimensionales y su vinculación a los navegadores, el tiempo de implementación de los entornos es bastante amplio dando lugar a que los proyectos no se entreguen a tiempo, esto trae como consecuencia un retraso de manera general en el proyecto.

La integración de los videojuegos a la plataforma web, especialmente la creación de mapas y entornos se hace muy compleja. En la Facultad 5 de la UCI específicamente en el proyecto Meñique se está realizando un Editor de Mapas Isométricos para juegos en líneas: IsoMap3D.

El sistema constará con un editor gráfico en el cual se podrá realizar mapas, seleccionar los tipos de superficies deseados, ver las propiedades y descripción de las mismas, así como una vista previa de la selección. Contará con un panel que muestra la vista previa de la selección realizada. Seguidamente un panel desplegable que contiene los distintos tipos de superficies deseadas con sus propiedades y descripciones incluidas:

- Tipo de superficie: Selección del tipo de superficie deseada para aplicar al mapa.
- Propiedades: Propiedades de la superficie.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Además, la aplicación contiene distintos modelos que pueden ser aplicados al mapa teniendo en cuenta las propiedades y dimensiones de los objetos, entre estos se encuentran:

- Casa: Selección de los diferentes modelos de casas existentes.
- Edificios: Selección de los diferentes modelos de edificios existentes.
- Propiedades: Propiedades y dimensiones del objeto seleccionado.

6. Aplicación de la propuesta de mejora del nivel de madurez 2 de CMMI-DEV al caso de estudio.

En el nivel 2 de madurez los proyectos deben asegurar que los procesos se planifiquen y realicen de acuerdo con las políticas.

Para darle cumplimiento a las políticas lo más importante es:

- La Institucionalización: Es cambiar los hábitos y la rutina diaria de trabajo de la organización.

La aplicación de la propuesta de mejora del nivel de madurez 2 de CMMI-DEV v 1. 2 se realizará al proceso Administración de Requerimientos.

Primer día de trabajo:

Actividad: Planificar la administración de requisitos y se obtuvo como resultado el WBS y el Calendario el Glosario de términos actualizado.

Paso 1: Actualizar el Glosario de Términos (entrenador).

Paso 2: Actualizar las actividades concernientes a la administración de requisitos en el WBS y el Calendario (entrenador, encargado de seguimiento).

Segundo día de trabajo:

Actividad: Preparar entorno de administración de requisitos y se obtuvo como resultado las herramientas instaladas y configuradas.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Paso 1: Coordinar reuniones con el equipo de proyecto (entrenador, cliente).

Paso 2: Preparar las herramientas de trabajo (rastreador).

Tercer día de trabajo:

Como el proyecto tenía un negocio definido, se realiza la siguiente actividad:

Actividad: Obtener y especificar requisitos y se obtuvo como resultado las Historias de Usuario definidas.

Paso 1: Identificar requisitos funcionales y no funcionales (programador, cliente).

Paso 2: Describir requisitos funcionales y no funcionales (programador, cliente).

Paso 3: Se le da a la Historia de Usuario una prioridad, o sea, alta, media, baja (programador, cliente).

Historias de Usuarios:

HU1: Insertar objetos en el mapa.

HU1.1 Seleccionar la categoría Superficie.

HU1.2 Insertar las superficies de terreno seleccionada.

HU1.3 Seleccionar la categoría Edificios.

HU1.2 Insertar las edificaciones seleccionada.

HU2. Utilizar un núcleo gráfico propio para la creación de mapas u objetos.

HU3. Editar la posición de los objetos.

HU3.1 Editar la posición de la superficie seleccionada.

HU3.1.1 Rotar la superficie seleccionada.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

HU3.2 Editar la posición de la edificación seleccionada.

HU3.2.1 Rotar la edificación seleccionada.

HU4. Ajustar la visión de la cámara respecto al mapa.

HU4.1 Mover la cámara hacia la derecha.

HU4.2 Mover la cámara hacia la izquierda.

HU4.3 Mover la cámara hacia la arriba.

HU4.4 Mover la cámara hacia la abajo.

HU4.5 Centrar la cámara.

HU5. Mostrar una vista previa del objeto seleccionado.

HU6. Rotación del mapa.

Historia de Usuario	
Número: 1	Nombre Historia de Usuario: Crear elementos gráficos utilizando un núcleo gráfico propio de la aplicación.
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Todos	Iteración Asignada: 1
Prioridad en Negocio: Baja (Alta / Media / Baja)	Puntos Estimados: 3
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 3
Descripción: El usuario puede crear cualquier tipo de gráfico tridimensional utilizando un framework o núcleo gráfico de ActionScript 3.0.	

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Observaciones:

Da cumplimiento a la historia HU2 Utilizar un núcleo gráfico propio para la creación de mapas”. El uso del núcleo gráfico es independiente al entorno de desarrollo.

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Ubicar objetos en el mapa
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Todos	Iteración Asignada: 2
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 3
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 3
Descripción: El usuario selecciona 1 de los 4 objetos de superficie o 1 edificación que desea insertar. Posteriormente inserta el objeto para la conformación del mapa ideado.	
Observaciones: Da cumplimiento a la historia HU1 Insertar objetos en el mapa”.	

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Edición de posición de objetos
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Todos	Iteración Asignada: 3
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 3
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 3

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Descripción: El usuario selecciona el objeto que se encuentra en el mapa y desea editar su posición (superficies o edificaciones). Posteriormente edita la posición el objeto que se encuentra en el mapa en el sentido de las manecillas del reloj.
Observaciones: Da cumplimiento a la historia HU3. "Editar la posición de los objetos".

Historia de Usuario	
Número: 4	Nombre Historia de Usuario: Ajuste de visión de cámara.
Modificación (o extensión) de Historia de Usuario (Nro. Y Nombre):	
Usuario: Todos	Iteración Asignada: 3
Prioridad en Negocio: Medio (Alta / Media / Baja)	Puntos Estimados: 3
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 3
Descripción: El usuario mediante un componente desplaza la posición de la cámara en la dirección deseada (arriba, abajo, derecha, izquierda), además de centrarla de cualquier posición que se encuentre la misma respecto al plano.	
Observaciones: Da cumplimiento a la historia HU4. "Ajustar la visión de la cámara respecto al mapa"	

Historia de Usuario	
Número: 5	Nombre Historia de Usuario: Vista previa.
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Usuario: Todos	Iteración Asignada: 4
Prioridad en Negocio: Baja (Alta / Media / Baja)	Puntos Estimados: 2
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 2
Descripción: El usuario puede visualizar una vista del objeto que selecciona.	
Observaciones: Da cumplimiento a la historia HU5 “Mostrar una vista previa del objeto seleccionado”.	

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Rotar Mapa.
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre):	
Usuario: Todos	Iteración Asignada: 4
Prioridad en Negocio: Media (Alta / Media / Baja)	Puntos Estimados: 2
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 2
Descripción: El usuario puede rotar el mapa en sentido de las manecillas del reloj o contrario a las manecillas del reloj.	
Observaciones: Da cumplimiento a la historia HU5 “Rotar mapa”.	

Cuarto día de trabajo:

Actividad: Entendimiento y Compromiso.

Como no es el fin de la fase de Planificación:

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Paso 1: Analizar requisitos, se estudia la documentación obtenida (Historia de Usuario), (entrenador, programador).

Paso 2: Revisar con el cliente, se realiza una reunión con él cliente para realizar la aceptación de las sugerencias de los programadores con respecto a los requisitos no entendidos que se especificarían nuevamente o serían eliminados del ciclo de vida del software (entrenador, programador).

Paso 3: Como se aceptaron los requisitos sugeridos por el programador se actualiza la Historia de Usuario con los requisitos aceptados y se obtiene la Historia de Usuario actualizada.

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Crear elementos gráficos.
Modificación (o extensión) de Historia de Usuario (Nro. y Nombre): HU-1 Crear elementos gráficos utilizando un núcleo gráfico propio de la aplicación.	
Usuario: Todos	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados: 3 semanas
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales: 3 semanas
Descripción: El usuario puede crear cualquier tipo de elemento gráfico tridimensional utilizando un framework o núcleo gráfico de ActionScript 3.0	
Observaciones: Da cumplimiento al requisito R2. "Utilizar un núcleo gráfico propio para la creación de mapas". El uso del núcleo gráfico es independiente al entorno de desarrollo. Puede ser utilizado por cualquier usuario que lo importe a un proyecto de ActionScript 3.0 o Flex. El entorno de desarrollo centra su funcionamiento en él.	

Paso 3: Aceptar requisitos y se obtiene la Historia de Usuario aceptada, para esto los miembros del proyecto se comprometen aceptando lo especificado en la Historia de Usuario (todos los involucrados en el proyecto).

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Quinto día de trabajo:

Como los requisitos fueron entendidos se realiza la siguiente actividad:

Actividad: Traceo.

Paso 1: Como existe una solicitud de cambio se analiza el efecto de la trazabilidad.

Paso 1.1: Se identifican los requisitos alterados por la trazabilidad (entrenador, rastreador, programador).

Paso 1.2 Analizar el efecto de los cambios en los elementos de la matriz (entrenador, rastreador, programador).

Paso 2: Como es un cambio se actualiza la matriz de trazabilidad y se obtuvo como resultado la Matriz de Trazabilidad actualizada.

Matriz de Trazabilidad actualizada

Historias de Usuarios						
Requerimientos	HU 1	HU 2	HU 3	HU 4	HU 5	HU 6
1	x					
2		x				
3			x			
4				x		
5					x	
6						x
7	x					

Sexto día de trabajo:

Actividad: Actualizar Planes.

Paso 1: Se actualiza el Plan de Proyecto con las nuevas actividades y se obtiene como resultado Plan de Proyecto actualizado (entrenador, encargado de seguimiento).

Pasó 2: Se especifican los requisitos del producto para esto se definen los artefactos a obtener en cada fase del proyecto y se obtiene como resultado los siguientes artefactos: Diseño simple, Tarjetas CRC,

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Pruebas de aceptación, Historias de Usuarios, Plan de Entrega, Plan de Publicaciones, Glosario de Términos, Iteraciones (entrenador, todos los involucrados).

Historias de Usuarios	Actividades	Versión	F. de publicación de versión.	Prioridad	Tiempo de desarrollo	No Trabajadores	Evaluación de calidad
HU1		---	29/01/10	Alta	3	2	P. aceptación
HU2		0.1	19/02/10	Media	3	2	P. aceptación
HU3		0.3	23/04/10	Media	3	2	P. aceptación
HU4		0.3	23/04/10	Baja	3	2	P. aceptación
HU5		0.5	14/05/10	Media	2	2	P. aceptación
HU6		0.5	14/05/10	Media	2	2	P. aceptación

Tabla 5 Plan de Publicaciones.

Iteraciones	Orden de las Historias de Usuario a implementar.	Cantidad de tiempo de Trabajo
Iteración 1	1- Crear elementos gráficos utilizando un núcleo gráfico propio de la aplicación.	3 semanas
Iteración 2	2- Ubicar objetos en el mapa.	3 semanas
Iteración 3	3- Edición de posición de objetos 4- Ajuste de visión de cámara.	3 semanas
Iteración 4	5- Vista previa. 6- Rotar Mapa.	2 semanas

Tabla 6 Iteraciones.

Entrega	Final 1ra iteración 4ta semana de enero del 2010	Final 2da iteración 3ra semana de febrero del 2010	Final 3ra iteración 3da semana de abril del 2010	Final 4ta iteración 2da semana de mayo del 2010
IsoMap3D	----	Versión 0.1	Versión 0.3	Versión 0.5

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

Tabla 7 Plan de Entrega.

Caso de Prueba de Aceptación	
Código:	Historia de Usuario (Nro. y Nombre):
Nombre:	
Descripción:	
Condiciones de Ejecución:	
Entrada / Pasos de ejecución:	
Resultado Esperado:	
Evaluación de la Prueba:	

Tabla 8 Prueba de aceptación.

Nombre de la clase: GridMap	
Responsabilidades: Une todas las superficies como un mapa.	Colaboradores: Bulding: Todas las propiedades de los edificios y las dimensiones de los objetos 3D que se puedan aplicar. Surface: Todas las propiedades de las superficies y las dimensiones de los objetos 3D que se puedan aplicar. ViewPorts: Mostrar objetos 3D creados por las clases Bulding y Surface.

Tabla 9 Tarjeta CRC.

Paso 3: Entendimiento y Compromiso.

Se estudia nuevamente la documentación y como no hubo sugerencias por parte de los programadores se mantienen los requisitos especificados.

Paso 1: Traceo, como no existe solicitud de cambio ni la inserción de un nuevo elemento se sale del proceso.

7. Resultados obtenidos.

Después de realizar una profunda investigación de la propuesta de mejora en el nivel 2 de madurez de CMMI-DEV v 1.2 para proyectos que utilicen la metodología XP, se han obtenido resultados favorables que avalan la utilización de la misma en el desarrollo de software.

La propuesta de mejora del nivel 2 de madurez de CMMI-DEV v 1.2 para proyectos que utilizan la metodología XP proporciona mayor efectividad en la detección temprana de errores a lo largo del ciclo de vida del proyecto, reduciendo drásticamente el número de errores que afecta directamente a los clientes y usuarios. Mayor tolerancia al cambio e incremento de la capacidad de adopción y adaptación de nuevas tecnologías. Mejora en la rapidez y efectividad de respuesta ante exigencias del negocio. Mejora en la colaboración y comunicación efectiva con involucrados internos y externos. Se obtienen resultados predecibles en los proyectos. Permite mitigar los riesgos que afectan a los proyectos.

CONCLUSIONES

En el presente trabajo se realizó la definición de los procesos asociados a las siete áreas del nivel 2 de madurez de CMMI-DEV v1.2 para proyectos que utilizan como metodología de desarrollo XP. Para el desarrollo de la propuesta se realizó un estudio de la metodología XP y del modelo de calidad del software CMMI-DEV v 1.2 nivel 2 de madurez, también se tuvieron en cuenta los criterios de medida de calidad de software de la UCI y se tomó como base para elaborar la propuesta el programa de mejora que está realizando la UCI.

La propuesta obtenida será un avance para mejorar la calidad, los procesos y la habilidad de gestión para el desarrollo, adquisición y mantenimiento del producto o servicio final.

RECOMENDACIONES

Una vez concluido el desarrollo de la presente investigación se recomienda:

- La aplicación de todos los procesos definidos en la presente investigación para realizar un análisis de los resultados que arroja.
- Aplicar los procesos que no fueron aplicados durante la validación de la propuesta a un proyecto que utilice metodología XP.

REFERENCIAS BIBLIOGRÁFICAS

(Agustin Yagüe, 2009) **Agustin Yagüe, Juan Garbajosa. 2009.** Sistedes Sociedad de Ingeniería del Software y Técnicas de Desarrollo del Software. *Sistedes Sociedad de Ingeniería del Software y Técnicas de Desarrollo del Software*. [En línea] 2009. [Citado el: 15 de febrero de 2010.] <http://www.sistedes.es/>. ISBN.

(Agustin Yagüe, 2009) **Agustin Yagüe, Juan Garbajosa. 2009.** Sistedes. *Sistedes*. [En línea] 2009. [Citado el: 16 de febrero de 2010.] <http://www.sistedes.es/>. ISSN 1988-3455.

(Beck, 1999) **Beck, Kent. 1999.** *Extreme Programming Explained. Embrace Change*. s.l: Pearson Education, 1999.

(Calisoft, 2003) CaliSoft. (2003). *Calisoft centro para la excelencia*. Recuperado el 16 de febrero de 2010, de Calisoft centro para la excelencia: http://calidadsoft.prod.uci.cu/index.php?option=com_content&view=article&id=34&Itemid=27

(Carnegie Mellon University. 2010) Carnegie Mellon University. (2010). *Software Engineering Institute*. Recuperado el 28 de enero de 2010, de Software Engineering Institute: <http://www.sei.cmu.edu/>

(Castillo, 2010) **Castillo Oswaldo, Figueroa Daniel, Sevilla Héctor. 2010.** Fases de la Programación Extrema. *Programación Extrema*. [En línea] 25 de enero de 2010. <http://programacionextrema.tripod.com/fases.htm>.

(Chrissis, 2009) **Chrissis, Mary Beth. 2009.** *Guía para la integración de procesos y la mejora de productos*. España, Universidad Politécnica de Madrid: Pearson Educación, S.A, 2009. ISBN: 9788478290963.

(España, 2009) España, L. n. (marzo de 2009). *Inteco*. Recuperado el 2 de febrero de 2010, de Inteco: www.inteco.es

(Fernández,2009) **Pérez, Yudit Fernandez.** Curso gratis de Enfoque administrativo de la Ingeniería de Software. *Curso gratis de Enfoque administrativo de la Ingeniería de Software*. [En línea] http://www.canal cursos.com/cursos-gratis/curso-gratis-de-enfoque-administrativo-de-la-ingenieria-de-software_f113313.htm. http://www.canal cursos.com/cursos-gratis/curso-gratis-de-enfoque-administrativo-de-la-ingenieria-de-software_f113313.htm.

(Jeffries, 2001) **Jeffries, R., Anderson, A., Hendrickson, 2001.** *Extreme Programming Installed*. s.l.: Addison-Wesley, 2001.

(Mark C. Paulk, 2002) Mark C. Paulk, B. C. (2002). *Capability Maturity Model Integration: Staged Representation*. Pittsburgh, Pennsylvania.

REFERENCIAS BIBLIOGRÁFICAS

(Martin, 2005) **Martin, Fowler. 2005.** The new Metodology. *The new Metodology*. [En línea] Copyright Martin Fowler, 13 de diciembre de 2005. [Citado el: 10 de febrero de 2010.] <http://www.martinfowler.com/articles/newMethodology.html>.

(Palacio, 2009) palacio, J. (2009). *Navegapolis*. Recuperado el 25 de enero de 2010, de Navegapolis: <http://www.navegapolis.net/content/view/807/62/>

(Presman, 2005) **Pressman, Roger. S. 2005..** *Ingeniería de software. Un enfoque práctico*. 2005.

(Reynoso, 2008) **Reynoso, Carlos Billy. 2008.** Página de Carlos Reynoso. *Investigación, Publicaciones y Cursos de Antropología, Ciencia Cognitiva y Complejidad*. [En línea] 2008. [Citado él: 27 de enero de 2010.] <http://carlosreynoso.com.ar/>.

(Universidad Politécnica de Valencia, 2003)**Valencia, Universidad Politécnica de. 2003.** ISSI Ingeniería del Software y Sistema de Información. *ISSI Ingeniería del Software y Sistema de Información*. [En línea] 10 de noviembre de 2003. <http://issi.dsic.upv.es/publications/archives/f-1069167248521/actas.pdf>.

(Villafuerte, 2009) Villafuerte, V. (2009). *Extreme Programing*. Recuperado el 20 de enero de 2010, de Extreme Programing: <http://extremeprogramming.host56.com/>

(Wesley,2000) Wesley, A. (2000). *Una explicación de la programación extrema. Aceptar el cambio*.

BIBLIOGRAFÍA

Alfonso Romero, D. L. (2007). *Scielo*. Recuperado el 2010, de Scielo: <http://www.scielo.org.pe/pdf/iigeo/v10n19/a05v10n19.pdf>

Beck, K. (2001). *Manifiesto for Agile Software Development*. Recuperado el 2010, de Manifiesto for Agile Software Development: <http://agilemanifesto.org/>

Calisoft. (2009). *Calisoft Centro de Calidad para Soluciones Tecnológicas*. Recuperado el 2010, de Calisoft Centro de Calidad para Soluciones Tecnológicas: http://calisoft.uci.cu/index.php?option=com_content&view=article&id=34&Itemid=27

Eliecer Herrera Uribe, L. E. (2007). *Redalyc* . Recuperado el 2010, de Redalyc : <http://redalyc.uaemex.mx/redalyc/pdf/849/84934064.pdf>

España, A. d. (9 de marzo de 2009). *ati Asociación de Técnicos de Informática*. Recuperado el 2010, de ati Asociación de Técnicos de Informática: <http://www.ati.es/spip.php?article1135>

Ferer, J. (2005). *AGILE Spain* . Recuperado el 2010, de AGILE Spain : <http://www.agile-spain.com/>

Hsieh, T. (2010). *Agile CMMI*. Recuperado el 2010, de Agile CMMI: <http://www.agilecmmi.com/index.php>

Jeffries, R. E. (2010). *XProgramming*. Recuperado el 2010, de XProgramming: <http://xprogramming.com/index.php>

Mary Beth Chrissis, M. K. (2009). *Software Engineering Institute*. Recuperado el 2010, de Software Engineering Institute: <http://www.sei.cmu.edu/library/abstracts/whitepapers/cmmi-dev-v12-spanish.cfm>

Stephens, M. (mayo de 2004). *Software Reality*. Recuperado el 2010, de Software Reality: <http://www.softwarereality.com>

Thomas, D. (2007). *agile software product strategist*. Recuperado el 20010, de agile software product strategist: <http://www.davethomas.net/>