

CMM y RUP: Una perspectiva común



Este documento trata de:

- ✓ CMM.
- ✓ RUM
- ✓ Calidad en en Software
- ✓ Desarrollo de Software
- ✓ Ingeniería del Software

El camino a la sabiduría es llano y simple, primero hay que actuar, después error, error y error otra vez, pero menos, menos y cada vez menos [1] "

Piet Hein .

Si usted no está consiguiendo mejorar más rápidamente que los demás, usted está consiguiendo empeorar [2]

Tom Peters

**Manuel Tobarra Narro
Noviembre de 2003
Universidad de Castilla-La Mancha
Escuela Politécnica Superior de Albacete
Asignatura Ingeniería del Software II**

Todas las marcas mencionadas en este documento son propiedad de sus respectivos propietarios.
CMM® is a registered mark of Carnegie Mellon University.
IBM Rational Unified Process® and RUP® are registered marks of International Business Machines Corporation

Contenido

CONTENIDO	3
INTRODUCCIÓN.....	5
ORIGEN	6
CAPABILITY MATURITY MODEL	6
FUNDAMENTOS DEL MODELO	6
CAPACIDAD Y MADUREZ	7
PROCESO MADURO	7
CARACTERÍSTICAS DEL MODELO	8
ESTRUCTURA DEL MODELO	8
ESTRUCTURA EN NIVELES	12
ÁREAS CLAVES DEL PROCESO	13
ÁREA CLAVE DEL PROCESO DEL NIVEL 2 (LEVEL-2 KPA)	13
ÁREA CLAVE DEL PROCESO DEL NIVEL 3 (LEVEL-3 KPA)	14
APLICACIÓN DEL MODELO	15
RATIONAL UNIFIED PROCESS.....	16
MEJORES PRÁCTICAS	16
RUP Y CMM.....	18
UTILIZACIÓN DE RUP PARA SATISFACER EL NIVEL 2 DE CMM	18
REQUISITOS GENERALES DE CMM.....	18
GESTIÓN DE REQUISITOS (REQUIREMENTS MANAGEMENT – RM).....	21
PLANIFICACIÓN DEL PROYECTO DE SOFTWARE (SOFTWARE PROJECT PLANNING - SPP).....	21
SEGUIMIENTO Y SUPERVISIÓN DEL PROYECTO DE SOFTWARE (SOFTWARE PROJECT TRACKING AND OVERSIGHT – PTO)	21
GESTIÓN DE LA SUBCONTRATACIÓN DE SOFTWARE(SOFTWARE SUBCONTRACT MANAGEMENT – SSM)	22
GARANTÍA DE LA CALIDAD DEL SOFTWARE (SOFTWARE QUALITY ASSURANCE – SQA)	23
GESTION DE LA CONFIGURACIÓN DEL SOFTWARE (SOFTWARE CONFIGURATION MANAGEMENT – SCM).....	23
BIBLIOGRAFÍA.....	25

Introducción



Este documento es un trabajo introductorio referente a **Capability Maturity Model (CMM)** y **Rational Unified Process (RUP)**.

Este trabajo forma parte de la asignatura "*Ingeniería del Software II*" (<http://www.info-ab.uclm.es/asignaturas/42551/index.html>) dirigida por la profesora Elena María Navarro Martínez en la Escuela Politécnica Superior de Albacete de la Universidad de Castilla La-Mancha durante el curso 2003-2004.

El modelo de Capacidad de Madurez (CMM) es el modelo más utilizado en la industria del software, no sólo en los Estados Unidos, sino en el mundo entero, por lo que representa el estándar para la industria del software. El modelo mide la capacidad del proceso para desarrollar un software con calidad, incrementando la predictibilidad para terminar los proyectos en coste, tiempo y con la calidad que el cliente espera.

Por otra parte, *Rational Unified Process (RUP)* es conocido como uno de las mejores metodologías para desarrollo de software.

Este documento trata de realizar una introducción en cómo una compañía, organización o proyecto puede utilizar RUP para alcanzar el nivel 2 de CMM.



Capability Maturity Model

Este modelo proporciona a las organizaciones de software una orientación sobre cómo obtener el control de sus procesos de desarrollo y mantenimiento de software, y cómo evolucionar hacia una cultura de ingeniería del software y de gestión por excelencia.

El modelo de CMM esencialmente sigue el concepto “Dí lo que haces, Haz lo que dices, Prueballo”. Esto significa que las organizaciones que desean alcanzar el Nivel 2 de CMM deben documentar varios procedimientos y métodos de Ingeniería de Software, ejecutarlos y ser capaces de probar que están conformes con los métodos y procedimientos mediante registros de calidad y artefactos del proyecto.

Origen

Dada la necesidad de obtener mejores productos, a principios de los años 90 del pasado siglo se pusieron en marcha diversas iniciativas tendentes a obtener mejores prácticas en el sector del desarrollo de software, que culminaron con la publicación de varios modelos y normativa.

La utilización de estos modelos para el desarrollo de software y su orientación casi exclusiva para la mejora de procesos, no permitió tener una visión amplia del resto de actividades que tienen lugar en una organización.

Las empresas, dada la competitividad existente, únicamente se fijaron en la mejora de los procesos operativos, no teniendo en cuenta otras partes de la organización, es decir, de las distintas actividades de las empresas que se encuentran íntimamente relacionadas.

Todo ello dio lugar a cambios en las normas y modelos a finales de los años noventa, que trataban de tener en cuenta toda la organización en su conjunto, donde las distintas actividades están integradas e interrelacionadas. Ello dio lugar, entre otras, a las normas ISO/IEC TR 15504 e ISO 9000:2000).

Fundamentos del modelo.

A principios de los años 80 el Departamento de Defensa de los Estados Unidos enfocó sus tareas a la revisión de los problemas del software y a su mejoramiento. Para contribuir a este programa se creó el Instituto de Ingeniería de Software (SEI) a finales de 1984. Como parte de su trabajo, el Instituto se dio a la tarea de desarrollar el Modelo de Madurez del Proceso de Software y para 1986 se comenzó el Proyecto de Evaluación de la Capacidad del Software. Después de varios años de realizar cuestionarios, evaluaciones, consultas e investigaciones, junto a otras organizaciones, en 1991 SEI produce el Modelo de Madurez de la Capacidad del Proceso de Software.

El Modelo de Madurez de la Capacidad del Proceso de Software (CMM®) permite determinar la capacidad de las organizaciones de desarrollo de software para producir de manera consistente y predecible productos de calidad superior. El modelo brinda guías para seleccionar estrategias de mejoramiento del proceso mediante la determinación de las capacidades

actuales del proceso y la identificación de los puntos críticos para mejorar el proceso y la calidad del software.

Capacidad y madurez

La capacidad del proceso es la habilidad inherente de un proceso para producir los resultados planeados. El principal objetivo de un proceso de software maduro es el de producir productos de calidad que cumplan los requisitos del usuario. Cuando se habla de madurez del proceso se entiende como el crecimiento alcanzado en la capacidad del proceso de software y que se considera como una actividad a largo plazo.

En una organización de software inmadura el proceso de software es generalmente improvisado, no existen planes rigurosos, sus actividades se enfocan en resolver las crisis que se presentan, carecen de bases objetivas para evaluar la calidad de los productos o para resolver los problemas que surgen. Por lo contrario, cuando la organización alcanza cierto grado de madurez posee una gran habilidad para administrar el proceso de desarrollo y mantenimiento del software, se hacen pruebas y análisis de costo-beneficio para mejorar el proceso, el administrador monitorea la calidad del producto y la satisfacción del cliente, se llevan registros y todos los integrantes están involucrados en el proceso de desarrollo.

Proceso Maduro

Para que un proceso pueda considerarse maduro ha de cumplir las siguientes características:

- **Está definido:** El proceso es claro, sistemático y suficientemente detallado. Además existe acuerdo entre el personal, la gerencia y los proyectos respecto al proceso que se va a utilizar.
- **Está documentado:** Esta escrito en un procedimiento publicado, aprobado y fácilmente accesible. Una de las mejores maneras es a través de una Intranet para apoyar los proyectos de desarrollo (ver página detallada).
- **El personal ha sido formado en el proceso:** Los ingenieros de software y la gerencia han recibido cursos y entrenamiento en cada proceso que aplica a su trabajo.
- **Es practicado:** El proceso definido debe ser usado en las tareas habituales llevadas a cabo por los proyectos. El entrenamiento y la adaptación del proceso a la realidad de la empresa debieran garantizar su aplicación en la vida real.
- **Es apoyado:** La gerencia no sólo debe firmar y promover los procesos definidos, sino que además debe asignar responsabilidad al personal y al los jefes de proyecto por su cumplimiento
- **Es mantenido:** El proceso es revisado regularmente, para asegurarse que está adaptado para satisfacer las necesidades reales de los proyectos
- **Está controlado:** Los cambios y puestas al día del proceso son revisados, aprobados y comunicados oportunamente a todos los usuarios

- **Se verifica:** La gerencia mantiene mecanismos para asegurarse que todos los proyectos siguen el proceso vigente.
- **Se verifica:** Se asegura que el proceso mantiene concordancia con los requerimientos y estándares aplicables.
- **Se mide:** La utilización, los beneficios y el rendimiento resultante del proceso se miden regularmente
- **Puede mejorarse:** Existen mecanismos y apoyo de la gerencia para revisar e introducir cambios en el proceso, de manera de mejorar su eficacia e incorporar nuevas metodologías.

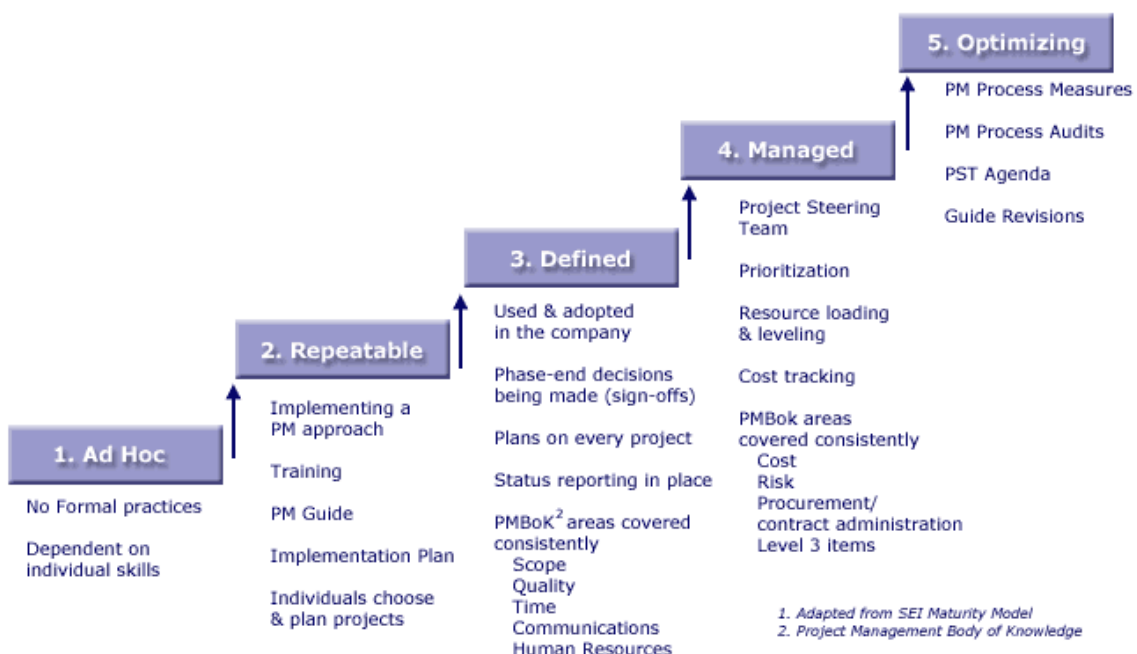
Por contra, un proceso inmaduro se caracteriza por que es fundamentalmente personal, no está documentado, es difícil compartirlo con otros miembros del equipo, no es fácil reproducirlo en nuevos proyectos, no hay entrenamiento, no todo el mundo lo conoce, no se mide, se aplica a veces solamente, es percibido como poco eficiente, es interpretado de manera distinta, etc.

Características del modelo

CMM provee una estructura conceptual para mejorar el manejo y desarrollo de productos de software en una forma disciplinada y consistente. Se enfoca en la capacidad de las organizaciones de software de producir productos de calidad de manera consistente y predecible. Así, cubre prácticas de planeación, ingeniería, desarrollo de software y su mantenimiento, definiendo niveles a los cuales una organización debería de evolucionar para establecer una cultura de excelencia en ingeniería de software.

Estructura del modelo.

El modelo consta de 5 niveles, diseñados de manera que los niveles inferiores proveen las bases para que de forma progresiva se alcancen los superiores. Estas 5 etapas de desarrollo son referidas como niveles de madurez y en cada uno la organización alcanza una capacidad superior del proceso. En la Figura 5 se muestra la estructura de los niveles del modelo.



-
1. **Inicial (Ad Hoc)** : El proceso de software es un **proceso improvisado y caótico**. Pocos procesos están definidos y el éxito que se pueda obtener depende de las habilidades, conocimientos y motivaciones del personal. **No existen calendarios ni estimados de costos y la funcionalidad y calidad del producto es impredecible**. No existe un ambiente estable para el desarrollo y mantenimiento del software. El proceso del software es impredecible por el continuo cambio o modificación a medida que avanza el trabajo.

Una definición amplia es que en este nivel se encuentran todas las empresas *que no han logrado implementar las prácticas básicas de gestión de proyectos e ingeniería de software definidas a partir del nivel 2 o superiores*.

Una empresa está en el nivel caótico cuando sus gerentes y personal afirman que **los proyectos no se pueden planear**, que los requerimientos no se pueden tener bajo control, que no esté siempre en condiciones de controlar las versiones de producto, donde la calidad sea percibida como una burocracia innecesaria, cuando se acepte que los procesos son una cosa personal, cuando no se pueda verificar ni validar el producto, y sobre todo, cuando sus gerentes y personal vivan bajo condiciones de stress y frustración permanentes.

En este tipo de empresas, el software es **virtualmente producto del arte más que de la ingeniería**. Cada "artista" crea su propio proceso personal, el cual es parte de su sello personal. La gerencia ocupa una parte significativa de su tiempo en paliar problemas y enfrentar clientes insatisfechos. Ante una **situación de crisis permanente**, se les hace difícil destinar recursos para definir o documentar procesos, lo que lleva a un círculo sin salida.

Cuando el proyecto se termina, la inversión hecha en desarrollar el proceso es raramente reutilizada en nuevos proyectos. Los desarrolladores de software generalmente tienen que trabajar largas horas y paliar problemas en forma cotidiana, lo cual les disminuye su creatividad y productividad netas. El éxito descansa en los hombros de estos héroes, tal como en una película de acción americana. Su nivel de frustración es elevado y es muy frecuente que, como cualquier "diva", decidan explorar caminos en otras empresas con menor nivel de stress.

El proceso, que no está documentado ni a sido compartido, se va con ellos, dentro de sus cerebros. Los que los reemplazan heredan problemas y dificultades, pero son raramente capaces de recuperar los procesos de desarrollo. Esto obliga a reinventar la rueda, a un alto costo y retrasando los proyectos.

La conclusión a la que podemos llegar es que es demasiado caro o difícil tratar de adivinar lo que el empleado anterior hizo, ya que a las empresas les sale más rentable echar a la basura el desarrollo anterior y empezar todo de nuevo. En casos extremos hay que simplemente terminar el producto para no seguir perdiendo dinero o prestigio frente a los clientes.

En los orígenes de la industria de software el caos predominó y las empresas que sobrevivieron la selección natural llegaron al mercado de nuestros días. La mayoría se extinguió en la gloria del triunfo efímero. Muchos de los actores actuales están predestinados a desaparecer en un futuro próximo. A pesar del talento de su personal y el despliegue de tecnología que puedan sustentar, derrochan su éxito debido a la debilidad de sus procesos de desarrollo.

-
2. **Repetido (Repeatable):** Se establecen **procedimientos de administración del proceso que son básicos para determinar costos, calendarios y funcionalidad**. Se establecen las políticas para la administración del proceso y los procedimientos de implantación. El proceso se basa en repetir éxitos anteriores en proyectos de similares características, por lo que los mayores riesgos se presentan cuando se enfrentan a nuevos proyectos. **Existen problemas de calidad y no hay una adecuada estructura para mejorarla.**

Este nivel no garantiza que todos los proyectos dentro de la empresa estén necesariamente al mismo nivel de madurez. Algunos pueden estar todavía en el nivel inicial.

El mayor beneficio obtenido de la implementación del nivel 2 por la empresa es **la planificación realista de los proyectos**. En ocasiones, si no estamos situados en el nivel 2, la planificación de los proyectos expresan más los deseos de la gerencia que la realidad. Este principio (el mismo en la cual se basa la magia) conduce a una situación de buscar culpables y generar excusas, produciendo al mismo tiempo frustración y desconfianza entre clientes y empleados. Actualmente los cronogramas son cada día más confiables, y mejora a medida que se acumula más información en las bases de datos de los proyectos pasados. El uso generalizado de **métodos de estimación** permite al personal del proyecto de **justificar plazos y recursos**. Aún el "olfato profesional" y la experiencia personal juegan un papel importante en la generación de planes de proyecto, pero ahora son decisiones informadas en vez de simples adivinanzas como en el pasado.

Este nivel todavía permite la proliferación y definición insuficiente de los procesos de ingeniería de software. Los proyectos comparten principalmente sus experiencias en materia de administración de proyectos, pero sus métodos técnicos pueden diferir. Aún existe **incomunicación entre proyectos, grupos y entre personal y gerencia**. Este nivel identifica prácticas de sentido común que son aplicables en todo tipo de organizaciones de desarrollo de software, independientemente de su presupuesto, tamaño o ambiente de desarrollo. La ausencia de cualquiera de sus prácticas simplemente pone en peligro el éxito de la empresa.

3. **Definido (Defined):** El **proceso de software para las actividades administrativas y técnicas está documentado, estandarizado e integrado en un proceso de software estándar dentro de la organización que ayudará a obtener un rendimiento más efectivo**. El grupo que trabaja en el proceso enfoca y guía sus esfuerzos al **mejoramiento del proceso**, facilita la introducción de técnicas y métodos e informa a la administración del estado del proceso. La capacidad del proceso está basada en una amplia comprensión común dentro de la organización de las actividades, roles y responsabilidades definidas en el proceso de software.

La base de datos que reúne estadísticas de los proyectos pasados y en curso, **permite planificar y comparar el rendimiento**. Existen mecanismos de **comunicación entre proyectos y departamentos**, lo que garantiza una visión común del producto y una rápida acción para enfrentar los problemas. He conocido unas pocas empresas a este nivel y la cosa que más me resaltó fue la satisfacción del personal. En empresas de nivel 1 habitualmente se escuchan quejas y acusaciones. A nivel 3 los empleados tienen una alta valoración de los procesos y entienden claramente la manera en que afectan su desempeño habitual. Los gerentes pueden realizar sus verdadera función, administrar.

El hecho de realizar **revisiones tempranas en forma regular** mejora visiblemente la calidad de los productos y minimiza las reiteraciones innecesarias. Curiosamente muchas organizaciones de nivel 1 realizan revisiones de pares, pero lo hacen de manera inconsistente y al primer signo de pánico las suspenden. El nivel 3 ya es un estado avanzado y es percibido por algunos gerentes como un lujo.

4. **Administrado (Managed):** Se **recopilan métricas detalladas del proceso de software y de la calidad del producto**. Ambos son cuantitativamente entendidos y controlados. El ciclo de Shewhart es constantemente utilizado para planear, implementar y registrar las mejoras al proceso. Este nivel de capacidad del proceso permite a la organización predecir las tendencias en el proceso y en la **calidad del producto dentro de los límites establecidos** y además tomar las acciones necesarias en caso que sean excedidos. Los productos tienen “*de partida*” una alta calidad. La empresa es capaz de proponerse metas cuantitativas para la calidad de los productos y de los procesos de software. **Es posible medir la productividad y calidad de los procesos de software a través de todo el proyecto.**

Los proyectos pueden controlar la variación del rendimiento de sus productos y procesos para mantenerla dentro de fronteras cuantitativas aceptables. *Es posible discriminar las variaciones significativas en el rendimiento del proceso de la variación (ruido) al azar*, particularmente dentro de líneas de productos establecidas.

Es necesario aclarar que el hecho de contar con un sistema de métricas de software *no significa que se esté en el nivel 4*. Existen empresas de nivel 1 que miden cuidadosamente el número de defectos detectados durante las pruebas o tests. Es una virtual señal de alarma que les dice cuán graves son sus problemas, pero la inmadurez de sus procesos no les permite hacer nada efectivo, excepto tal vez abortar el producto para evitar un daño mayor que puede resultar de distribuirlo a los clientes.

Son muy raras las empresas que han decidido implementar este nivel. No son muchos especialistas de procesos que realmente tengan experiencia práctica, o incluso que entiendan bien las áreas claves de proceso del nivel 4. Son solamente 2 prácticas, pero imposibles de alcanzar si no se ha implementado firmemente los 2 niveles de madures anteriores

5. **Optimizado (Optimizing):** El **mejoramiento continuo** del proceso es garantizado por la retroalimentación cuantitativa desde el proceso y desde las pruebas de técnicas y herramientas innovadoras. La organización tiene los medios para **identificar los puntos débiles** del proceso y conocer cómo fortalecerlos. Su actividad clave es el análisis de las causas de defectos y su modo de prevención.

La organización entera se aboca al mejoramiento continuo del proceso. La corporación cuenta con los medios para identificar las debilidades y reforzar en forma proactiva el proceso, con objeto de prevenir la ocurrencia de defectos. Los datos relativos a la eficacia del proceso de software se usan para analizar el costo y el beneficio de usar nuevas tecnologías y de implementar cambios al proceso de software.

Los proyectos de software analizan los defectos para determinar sus causas. Los procesos de software se evalúan para prevenir que los defectos conocidos vuelvan a ocurrir, asimismo las lecciones aprendidas son difundidas a otros proyectos.

No existen más de 10 empresas en el mundo que estén a este nivel (no hay ninguna en países hispano-hablantes). Y las pocas que lo han

logrado no divulgan sus secretos para mantener su ventaja competitiva. Este **nivel es un estado ideal**, una especie de Paraíso o Nirvana, que probablemente nunca será alcanzado por la mayoría de las empresas productoras de software. **Puede considerarse como una hermosa utopía, pero inalcanzable en el mundo normal.**

Cada nivel sirve de base para que los siguiente establezcan una implantación del proceso eficiente y efectiva. La organización puede, sin embargo, de forma provechosa usar procesos descritos en otros niveles. Saltar niveles es contraproducente debido a que cada uno es básico para obtener el siguiente y la capacidad de poder implementar procesos superiores de madurez no implica que se pueda saltar un nivel.

Estructura en niveles.

Cada nivel de madurez está compuesto de varias áreas claves del proceso. Cada una es organizada en 5 secciones definidas como características comunes (figura 2). Éstas especifican las prácticas claves para el cumplimiento de las metas en el área correspondiente. Las divisiones principales del modelo son 5 niveles de madurez y cada nivel es descompuesto a su vez en un conjunto áreas claves del proceso (KPA por sus siglas en inglés Key Process Areas), las cuales están organizadas de acuerdo a sus características.

Las características comunes son:

- **Compromisos.** Describe las acciones que la organización debe tomar para establecer el proceso y que pueda ser soportado. Esta característica está asociada con el establecimiento de políticas y con la responsabilidad de la alta dirección.
- **Habilidades.** Describe las precondiciones que deben existir en el proyecto u organización para implementar un proceso de software de manera competente. Involucra los recursos, estructura de la organización y capacitación requerida.
- **Actividades.** Describe los roles y procedimientos necesarios para implementar las metas de un área clave del proceso. Considera los planes, procedimientos, actividades, revisiones y acciones correctivas que se requieren.
- **Mediciones.** Describe las necesidades de medir el proceso y analizar los resultados.
- **Verificación de la implantación.** Describe los pasos para asegurar que las actividades se desarrollan de acuerdo con lo establecido en el proceso. Generalmente abarca las revisiones y auditorías de la dirección y de los aseguradores de la calidad.

Las prácticas claves describen la infraestructura y actividades que más contribuyen a la efectiva implantación e institucionalización del proceso.

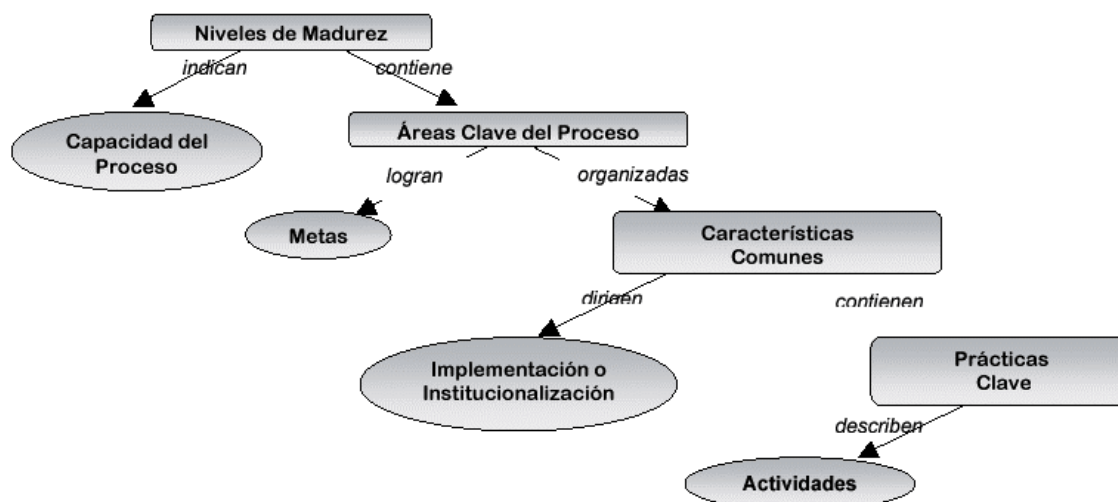


Figura 2. Modelo CMM.

Áreas Claves del Proceso

El conjunto de prácticas de software y el manejo específico de los niveles de madurez son llamadas áreas claves del proceso (o **KPA**: Key Process Areas). Cada una de éstas es descrita en términos de prácticas claves, las cuales describen las actividades o infraestructura que contribuye de manera más efectiva a la implementación e institucionalización de la KPA; son descritas en oraciones sencillas, a menudo seguidas por una explicación detallada [3].

Existen **18 KPA's distribuidas a lo largo de los niveles 2-5** (El nivel 1 no tiene ninguna debido a las características que posee). Cada una de estas tiene distintas metas que cumplir, a continuación se detalla cada una de acuerdo al nivel que pertenece y más adelante se mencionan las metas que les corresponden.

Área clave del proceso del nivel 2 (Level-2 KPA)

Los KPA nivel 2 son:

- **Gestión de requisitos:** Consiste en establecer un entendimiento común entre el cliente y el proyecto de software acerca de los requisitos del usuario que serán cubiertos por el proyecto de software.
- **Gestión de la configuración del software:** Establece y mantiene la integridad de los productos del proyecto de software a través del ciclo de vida del software del proyecto.
- **Aseguramiento de la calidad del software:** Se encarga de proporcionar a la administración una visibilidad apropiada del proceso usado y los productos construidos durante el desarrollo de un proyecto de software.
- **Seguimiento y pruebas del proyecto de software:** Se da la tarea de proporcionar una visibilidad adecuada del progreso actual, de manera que los administradores puedan tomar acciones efectivas

cuando el desempeño del proyecto se desvía .significativamente de lo planeado:

- **Planificación del proyecto de software:** Establece planes razonables para realizar la ingeniería de software y para administrar el proyecto de software.
- **Gestión de subcontratistas de software:** Ésta área clave selecciona a los subcontratistas de software calificados y administrarlos efectivamente.

Área clave del proceso del nivel 3 (Level-3 KPA)

En el Nivel Definido, el proceso estándar para desarrollo y mantenimiento de software a través de la organización es documentado, incluyendo la Ingeniería de Software y la Gestión de Procesos, y estos procesos se integran en un todo coherente. Los procesos establecidos en el nivel 3 son utilizados (y cambiados, cuando sea necesario) para ayudar a los directores del software y al equipo técnico a ser mas eficientes. La organización saca provecho efectivo de la Ingeniería de Software cuando se estandariza sus procesos de software. Existe un grupo que es responsable de las actividades del proceso de software de la organización; por ejemplo, SEPG o software engineering. Se implementa un programa de formación que asegura que el equipo y directores tienen el conocimiento y habilidades necesarias para cumplir sus roles.

La capacidad de los procesos de software en el las organizaciones que alcanzan el Nivel 3 puede resumirse como estándar y consistente porque tanto la Ingeniería de Software como las actividades de gestión son estables y repetibles.

Los KPA nivel 3 son:

- **Administración de subcontratistas de software:** Ésta área clave selecciona a los subcontratistas de software calificados y administrarlos efectivamente.
- **Enfoque del proceso de la organización:** Establece la responsabilidad organizacional de las actividades del proceso de software que mejoran la capacidad total del proceso de software de la organización.
- **Definición del proceso de software:** Desarrolla y mantiene un conjunto útil de activos de software que mejoren el desempeño de proceso entre proyectos y proporcionen una base de beneficios acumulables y de largo plazo para la organización.
- **Programa de capacitación:** Se encarga de desarrollar las habilidades y conocimiento de individuos de tal forma que puedan desempeñar sus roles efectiva y eficientemente.
- **Administración de la integración del software:** Sirve para integrar las actividades de ingeniería y administración del software en un proceso de software coherente y definido, que es adaptado del proceso estándar de software de la organización y de sus activos de proceso relacionados.
- **Ingeniería del producto de software:** Desempeña consistentemente un proceso de ingeniería bien definido que integra todas las

actividades de ingeniería de software para producir, de manera efectiva y eficiente, productos correctos y consistentes de software

- **Coordinación intergrupala:** Establece los medios para que el grupo de ingeniería de software participe activamente con otros grupos de ingeniería, de tal manera que el proyecto tenga mayor capacidad de satisfacer las necesidades del cliente de manera efectiva y eficiente.
- **Revisiones intergrupales:** Esta área clave elimina defectos de los productos de software en etapas tempranas y de manera eficiente. Un efecto colateral muy importante es desarrollar un mejor entendimiento de los productos de software y de los defectos que pueden prevenirse.

Aplicación del Modelo.

El modelo describe los **principios y prácticas relacionadas con la madurez** del proceso de software y propone ayudar a las organizaciones dedicadas al desarrollo del software a alcanzar la madurez de su proceso de software en términos del tránsito evolutivo desde un proceso improvisado y caótico a uno maduro con una adecuada disciplina y mayor capacidad. CMM® 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 se espera de una organización que realiza proyectos a gran escala. No es **prescriptivo ya que no dice a la organización cómo mejorar**. Excepto para el nivel 1, cada nivel de madurez es dividido en varias áreas claves que indican el área en la organización hacia la cual debe enfocarse el mejoramiento del proceso de software, identifican las políticas que se deben seguir para obtener un nivel de madurez y describen como la organización puede madurar. Cada área identifica un grupo de actividades relacionadas que, cuando se desarrollan de forma colectiva, permiten lograr una serie de objetivos considerados importantes para ampliar la capacidad del proceso. Cuando las metas que propone el área son cumplidas, la organización puede afirmar que se ha institucionalizado la capacidad del proceso caracterizada por ésta. Las metas indican el alcance, las fronteras y la intención para cada una.



Rational Unified Process

El Rational Unified Process™ (RUP) es un proceso de ingeniería de software que mejora la productividad del equipo de trabajo y entrega las mejores prácticas del software a todos los miembros del mismo. Los contenidos específicos para e-business del RUP™ proporciona una guía específica en áreas tales como la de Modelamiento de Negocios, Arquitecturas Web, Pruebas y Calidad.

Mejores Prácticas

La creación sólida de software de calidad requiere el conocimiento específico de las tareas que deben llevarse a cabo en cada entorno. Ahí radica la importancia de aplicar un proceso de desarrollo flexible y adaptado a cada objetivo de desarrollo.

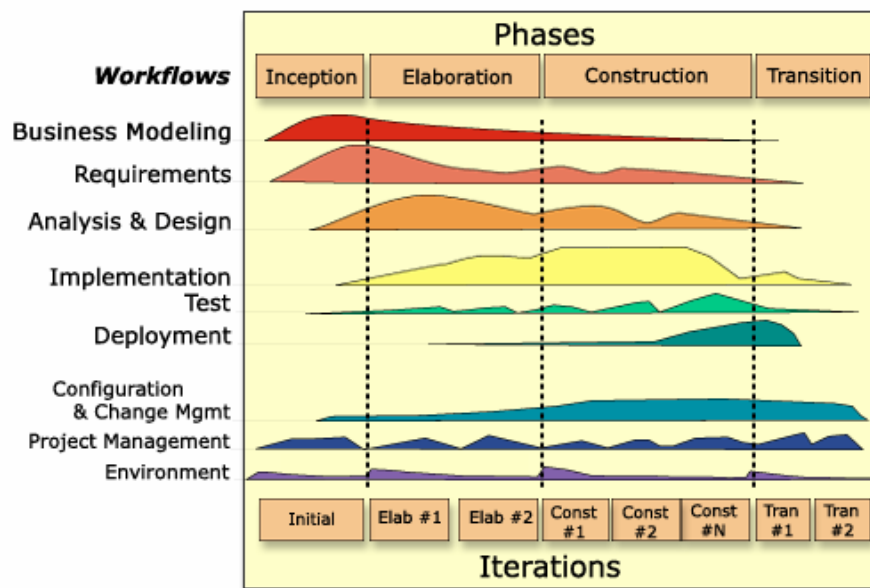


Figura 3. RUP.

El proceso RUP (Rational Unified Process) (figura 4) combina un **conjunto básico de mejores prácticas** aprobadas por el sector con una serie de complementos opcionales del proceso a fin de dar cabida y soporte a proyectos de cualquier envergadura o alcance.

Cualquier tipo de proyecto (incluidos los pequeños, los basados en Web, aquéllos fundamentales para un proyecto y los proyectos integrados) permiten obtener unos resultados más acordes con las previsiones gracias a la aplicación del proceso RUP.

Los **puntos clave en el RUP** son:

- ✓ Guiado por casos de uso.
- ✓ Centrado en la Arquitectura

✓ Iterativo e Incremental



RUP y CMM

A fin de que el nivel 2 de CMM pueda aplicarse a RUP tal como se indica en este documento, partimos de las siguientes suposiciones:

1. La organización ha adoptado Rational Unified Process (RUP)
2. La organización está utilizando RUP como se documenta en el RUP
3. La organización está creando los artefactos requeridos por RUP.

En este documento se van a identificar los diversos roles que aparecen especificados en RUP, disciplinas, plantillas y actividades que podrían aplicarse para satisfacer las diversas prácticas claves (key practices) del Nivel 2 de CMM.

Las tablas que aparecen en la sección xxx referentes a Key Process Area (KPA) indican cada KPA, su descripción, el rol de RUP que podría ser responsable de satisfacer esa práctica clave, qué proceso de RUP y procedimiento o plantilla podría ayudar a satisfacer la práctica clave, y notas aclaratorias. En los casos donde RUP no se aplica completamente con la práctica clave, se indican sugerencias de cómo podría lograrse cumplir esas prácticas clave.

Utilización de RUP para satisfacer el nivel 2 de CMM

Como ya se ha indicado arriba, el nivel 2 de CMM consta de seis KPA (Key Process Areas) de las más de 133 que existen en el modelo CMM.

Estas seis KPA son:

- **Gestión de requisitos.**
- **Gestión de la configuración del software.**
- **Aseguramiento de la calidad del software**
- **Seguimiento y pruebas del proyecto de software**
- **Planificación del proyecto de software**
- **Gestión de subcontratistas de software**

Primero se va a indicar cómo aplicar RUP a través de todas las KPA, posteriormente se indica cómo utilizar RUP para satisfacer cada KPA.

Requisitos generales de CMM

Tenemos que tener en cuenta que lo largo de los seis KPA del Nivel 2 de CMM existen varios conceptos:

- **Políticas de la Organización**

- Recursos y Financiación
- Aprendizaje y Orientación
- Planes y Procedimientos
- Medidas
- Revisiones

Para cada KPA, el modelo CMM requiere que exista una política organizacional escrita y que sea seguida por la organización a través del Compromiso a Realizar características comunes. RUP no trata de la necesidad de políticas, aunque el Software Engineering Process Authority (SEPA) descrito en el RUP podría ser la entidad apropiada para ayudar a desarrollar las políticas.

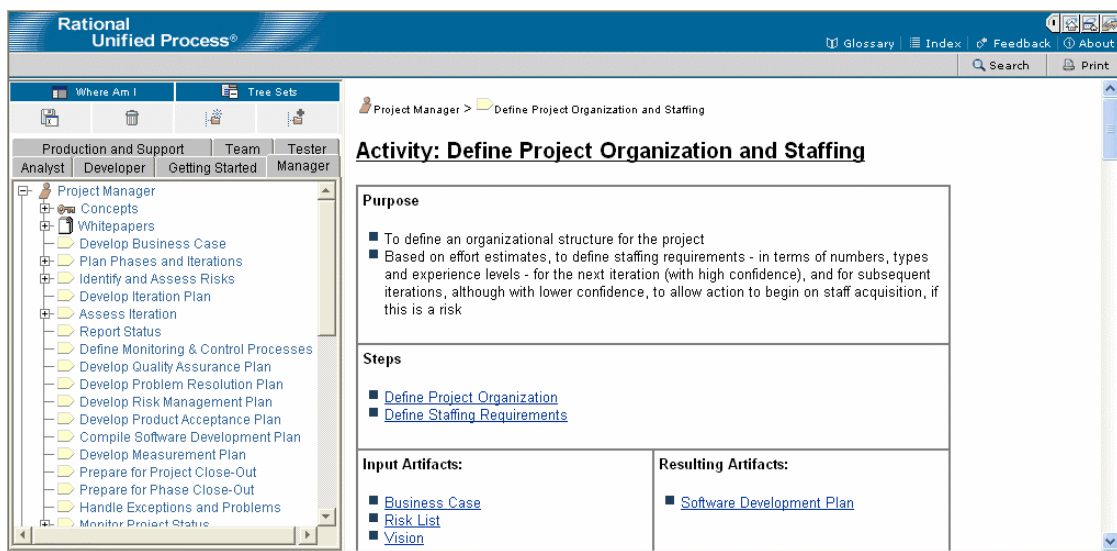


Figura 4. Activity: Define Project Organization an Staffing.

Los **Recursos y Financiación por CMM**. Las organizaciones pueden equipararse los recursos requeridos por CMM a los roles de RUP, teniendo en mente las actividades de CMM para los roles que son responsables de tener que realizar, incluso si RUP no los establece. Por supuesto, la financiación debe ser proporcionada por los roles para ser capaces de realizar de forma efectiva sus responsabilidades. La actividad RUP (Project Management -> Develop Software Development Plan->Define Project Organización & Staffing) es el lugar apropiado para definir y planificar los papeles necesarios (figura 4).

Puede realizarse un aprendizaje u orientación de manera informal por medio de RUP, dado que está basado en web y puede estar disponible para todos los miembros del equipo del proyecto a través del navegador Web.

El modelo CMM requiere que se documenten varios planes y procedimientos y que estén disponibles para la organización. RUP proporciona muchas plantillas *basadas en IEEE*, incluyendo el plan de desarrollo de software, plan de gestión de la configuración, y *plan de control de la calidad (QA, Quality Assurance)*. Dado que estas plantillas proporcionan instrucciones de cómo completarlas, se consideran procedimientos por CMM. Si son completadas completa y correctamente, los planes resultantes podrían utilizarse para documentar otros procedimientos requeridos por CMM que no específicamente establecidos por el propio RUP.

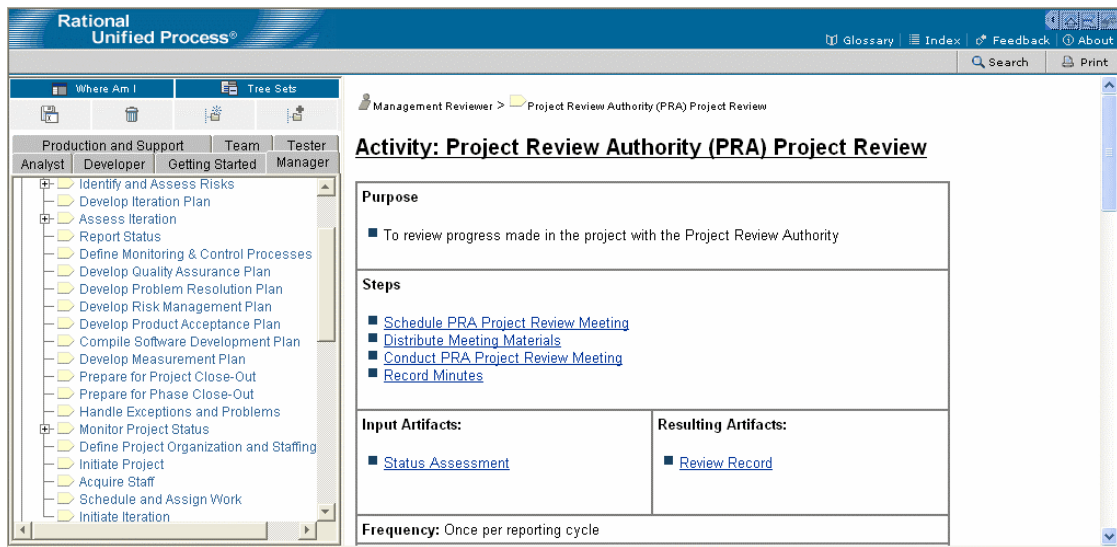


Figura 5. Project Review Authority (PRA) Project Review.

Las Medidas y Análisis en CMM requieren que la organización **identifique, tome, y utilice medidas** relacionadas con procesos que permitirán a la organización valorar y mejorar su proceso de software y calidad. RUP recomienda que se complete un plan de medida para especificar las medidas que deben ser realizadas y utilizadas. Posteriormente, estas medidas pueden ser revisadas y analizadas durante las revisiones del proyecto del PRA (Project Review Authority) (Figura 5), o durante las reuniones periódicas del Software Engineering Process Authority (SEPA). Muchas de las herramientas que de Rational, tales como Rational ClearCase, ClearQuest, y RequisitePro proporcionan medidas más que adecuadas para obtener estos objetivos de una forma automática.

La característica común Verificación de la Implementación de CMM requiere tres niveles de revisiones: *senior management*, *project*, y *software quality assurance*. RUP incluye varias revisiones:

- Project Approval Review
- Project Planning Review
- Iteration Plan Review
- PRA Project Review
- Iteration Criteria Evaluation Review
- Iteration Acceptance Review
- Lifecycle Milestone Review
- Project Acceptance Review

La revisión *Senior Management* puede ser llevada a cabo por *PRA project review*. Esta revisión necesita que la revisión de la gestión senior (*senior management*) del proyecto se haya sobre las actividades e hitos del proyecto. Un resultado de esta revisión es el Registro de Revisión (*Review Record*).

Las revisiones del proyecto puede ser llevada a cabo a través de la Actividad *Monitor Project Status* de RUP. Esta actividad necesita que los diferentes miembros del equipo del proyecto envíen informes de estado al director del proyecto. Esta actividad puede ser más efectiva si el equipo del proyecto actualmente cumple con el estado de revisión.

Las revisiones de Garantía de la Calidad del Software (*Software Quality Assurance*) de CMM pueden ser fácilmente satisfechas teniendo un grupo SQA representado en las reuniones de estado del proyecto y las revisiones de la gestión serior (*senior management reviews*).

Gestión de Requisitos (Requirements Management – RM)

Cuando son seguidas, las disciplinas RUP, Gestión de Requisitos y Gestión del Proyecto, cumplirán las características comunes de las Actividades Realizadas (*Activities Performed*) en el KPA Gestión de Requisitos (*Requirements Management*).

La organización tendrá que identificar el rol que sea responsable de analizar los requisitos del sistema y asignarlos al hardware, software, y otros componentes del sistema. Este rol es típicamente realizado por el ingeniero del sistema, y no es establecido específicamente en RUP.

Las herramientas de Rational RequisitePro y ClearQuest proporcionan una ayuda excelente en la automatización de la Gestión de Requisitos en la organización.

Planificación del Proyecto de Software (Software Project Planning - SPP)

La disciplina de Gestión de Proyecto de RUP cumple la mayoría de la SPP, incluyendo el Plan de Desarrollo de Software (*Software Development Plan – SDP*), el establecimiento de un ciclo de vida de desarrollo, y la integración de la planificación del proyecto de forma inicial en el ciclo de vida.

Aunque RUP necesita estimaciones para ser preparado, no especifica ningún método de estimación. Por tanto, la organización o el proyecto debe seleccionar y documentar el procedimiento de estimación del tamaño del producto, esfuerzo y coste, así como también los requisitos de computación críticos necesarios para el proyecto. Similarmente, se necesita un procedimiento detallando cómo el proyecto crea y actualiza su programación.

RUP no contiene enlaces al método de estimación COCOMO II. Esto puede ser un buen punto de partida para organizaciones que estudian cuando investigar los métodos de estimación apropiados.

La identificación de riesgos del proyecto, estimación, y documentación pueden ser automatizadas utilizando el producto de Rational ClearQuest.

Seguimiento y Supervisión del Proyecto de Software (Software Project Tracking and Oversight – PTO)

La Gestión del Proyecto de RUP es muy similar a la KPA *Software Project Planning*, y cumple la mayoría de las Actividades de la PTO, particularmente en el seguimiento continuo del esfuerzo de desarrollo a través de la actividad *Monitor Project Status*.

El principal aspecto a tener en cuenta de la PTO es el seguimiento continuo de los riesgos del proyecto, tamaño del producto, esfuerzo del proyecto y costes, y recursos críticos de computación. Durante el seguimiento, si se producen desviaciones significativas de las estimaciones iniciales, es necesario realizar acciones correctivas mediante la actividad de *Handle Exceptions & Problems* de RUP (Figura 6).

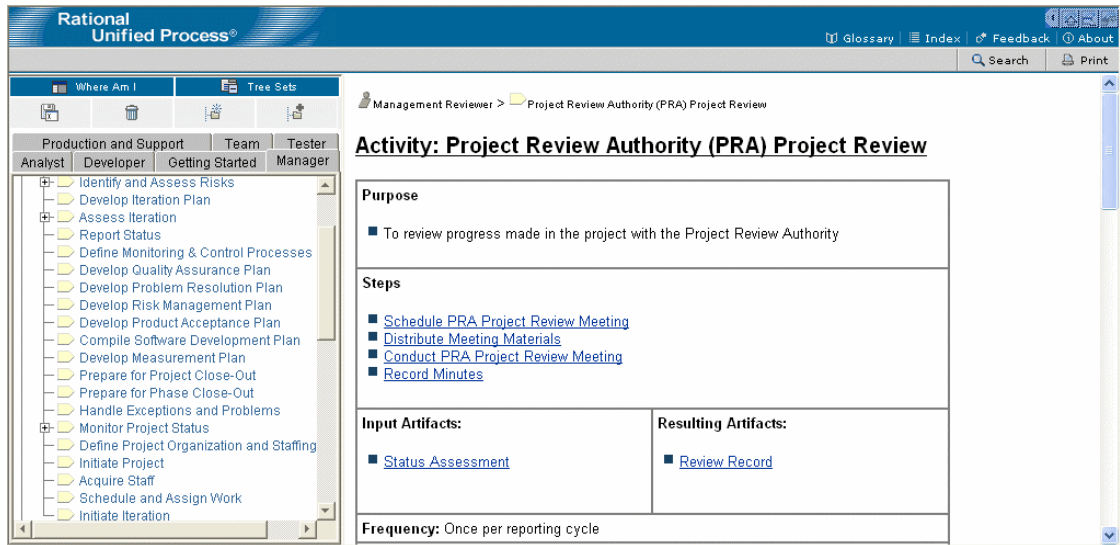


Figura 6. Handle Exceptions & Problems.

La herramienta ClearQuest de Rational puede ayudar a la automatización del seguimiento de los riesgos del proyecto.

Gestión de la Subcontratación de Software (Software Subcontract Management – SSM)

RUP está diseñado para ser utilizado por una organización que desarrolla productos de software o sistemas directamente para una organización cliente. El modelo CMM tiene en cuenta el hecho de que muchas veces existen organizaciones que colaboran en el trabajo de piezas o tareas del esfuerzo de desarrollo del sistema software. CMM asume que una de estas organizaciones es el “principal contratista” (*prime contracto*), y los otros son “subcontratistas” (*subcontractors*). Los subcontratistas típicamente utilizan su propio proceso de desarrollo de software, procedimientos, y herramientas para realizar sus respectivas partes del sistema. Por tanto, muchas de las prácticas clave necesitadas por CMM en la *KPA Software subcontract Management (SSM)* no son específicamente cumplidas por RUP.

Sin embargo, una organización puede cumplir esta KPA utilizando el estándar IEE Std 1061 *Recommended Practice for Software Acquisition*. Este estándar esboza los pasos recomendados que una organización debería seguir cuando sufre un esfuerzo de adquisición. Si se utiliza correctamente el estándar, muchos de los procedimientos del SSM requeridos por el CMM pueden ser producidos.

Por supuesto, el contratista principal podría ser animado a utilizar RUP también.

Garantía de la Calidad del Software (Software Quality Assurance – SQA)

La SQA de CMM llega más allá de la calidad del producto software. Se centra más en la calidad del proceso software. La filosofía que está debajo de CMM con respecto a la garantía de la calidad de software es que “un proceso de calidad producirá un producto de calidad”.

El rol de “Revisor” participa en muchos de las revisiones que requiere CMM como la revisión SDP. Mediante la inclusión de un “Revisor” de RUP en el proyecto y reuniones del la Gestión del Proyecto, muchos de las revisiones del proceso requeridas por CMM pueden ser cumplidas.

RUP apoya la terminación de un Plan SQA. Esto se basa en la plantilla IEEE, y cuando es completada correctamente se cumplirán muchos de los requisitos del SQA de CMM, tales como los requisitos de revisio del producto y proceso, una auditoría y procedimientos de escalado de la desviación.

La herramienta ClearQuest puede ser utilizada de forma efectiva para documentar y seguir las desviaciones identificadas durante las auditorías y revisiones del SQA.

Gestion de la Configuración del Software (Software Configuration Management – SCM)

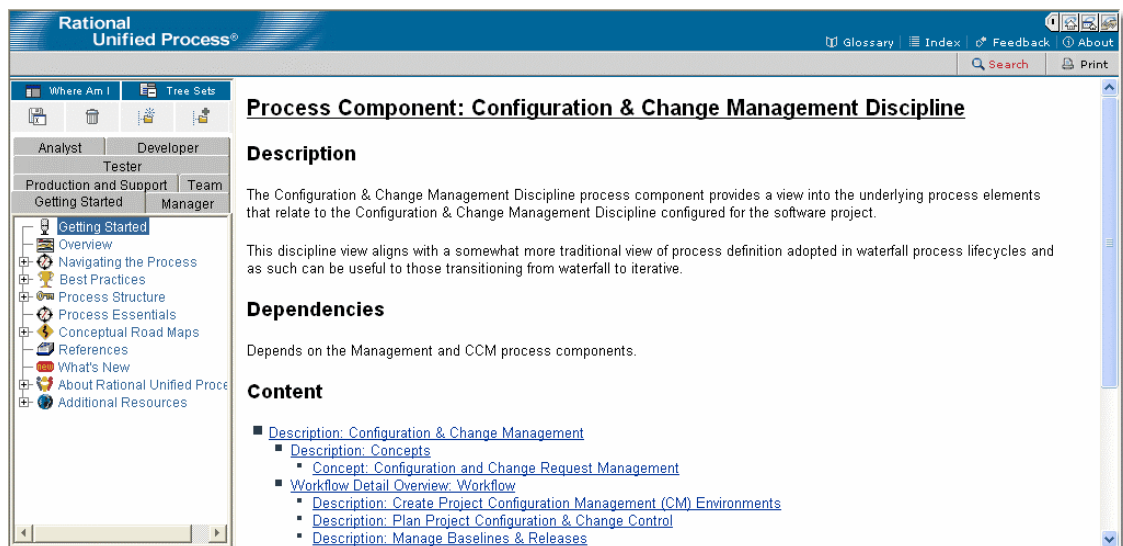


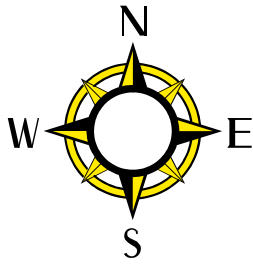
Figura 7. Configuration and Change Management.

La disciplina de RUP, Configuration and Change Management (figura 7) cumple muchos de los requisitos SCM de CMM.

RUP necesita que el Plan de la Gestión de la Configuración del Software (*Software Configuration Management Plan*) se complete y documente. Este plan detallará muchos de los procedimientos específicos del proyecto necesarios para completar las tareas SCM. Un plan de alto nivel SCM, en

conjunción con la documentación del producto o herramientas tutoriales de RUP podría no ser suficiente para el Nivel 2 de CMM ya que proporcionan una guía general, no producidos específicos del proyecto.

Muchos de los informes, auditorías, estatutos de configuración, y medidas pueden ser automatizadas utilizando ClearCase y ClearQuest. De hecho, estas herramientas se construyeron específicamente para ayudar a las organizaciones a realizar sus tareas de SCM mejor y más eficientemente.



Bibliografía

[1] COORDINACIÓN DE INNOVACIÓN EDUCATIVA /QFB-UMSNH Última. *¿Cómo lograr un trabajo bien hecho?*

(http://dieumsnh.qfb.umich.mx/un_buen_trabajo.htm)

[2] Tom Peters, Thomas J. Peters. *The Pursuit of Wow*. Vintage Books; 1st edition (January 15, 1994).

[3] Capability Maturity Model for Software (CMM) Software Engineering Institute. *The Capability Maturity Model: Guidelines for Improving the Software Proces*. Addison Wesley, EE.UU, Julio 2000 14a Edición.

[4] Cognence Inc. *Achieving Capability Maturity Model Level 2 Using the Rational Unified Process*. (<http://www.cognence.com/>)

[5] IBM Rational software. *Reaching CMM Leves 2 and 3 with the Rational Unified Process*. (<http://www.rational.com>)

[7] JACOBSON, I, BOOCH, G., RUMBAUGH, J. *El Proceso Unificado de Desarrollo de Software*. Addison-Wesley. 2000.

[8] Gabriela Juárez Sánchez. *Estudio y aplicación de métodos de valoración de CMM*. Universidad de las Américas-Puebla.

[9] SEI Capability Maturity Model

(<http://www2.umassd.edu/SWPI/processframework/cmm/cmm.html>)

[10] Defense Finance and Accounting Service, DoD of USA, *DFAS - PAL - CMM Maturity Levels*.

(<http://www.dfas.mil/technology/pal/cmm/index.htm>)

[11] Per Kroll, Philippe Kruchten. *The Rational Unified Process Made Easy: A Practitioner's Guide to Rational Unified Process*. Addison-Wesley Pub Co; 1st edition (April 8, 2003).

[12] Philippe Kruchten. *The Rational Unified Process: An Introduction (2nd Edition)*. Addison-Wesley Pub Co; 2nd edition (March 14, 2000).