



TESINA DE LICENCIATURA

Título: CMMI aplicado a un proyecto MDD

Autores: María Constanza Miguel

Director: Dra. Claudia Pons

Codirector: Dra. Roxana Giandini

Carrera: Licenciatura en Sistemas

Resumen

El objetivo de esta tesina es definir los puntos planteados por el modelo de madurez CMMI (metas y prácticas específicas de las áreas clave de proceso) que son mapeados directamente con el modelo MDD y para los casos en que no exista un claro y directo mapeo, se presentan pautas generales para que la organización tenga en cuenta y pueda cubrir esas áreas clave con otras actividades que caen fuera de la definición de MDD.

Dado que CMMI no impone una metodología en particular de desarrollo de software para poder realizar una medición de madurez, y tampoco define las condiciones que se deben cumplir para cada tipo de metodología, se definen las pautas a aplicar para MDD dada la poca información existente sobre el tema.

Para lograr el objetivo planteado, se desarrolla una investigación sobre el modelo CMMI-SW describiendo los requisitos que debe cumplir un proyecto de software, independientemente de la metodología de desarrollo que utilice, para aplicar a cada uno de los cinco niveles definidos.

Tomando como base esta investigación, se realiza el correspondiente mapeo con la metodología de desarrollo RUP, tal vez la más conocida y utilizada en el mercado, y luego lo que nos interesa, el mapeo con MDD.

Palabras Claves

CMMI, MDD, Madurez, Calidad, RUP, Mapeo

Conclusiones

El recorrido por los niveles 2 y 3 de CMMI supone que gran parte de las prácticas son realizadas por una empresa, cuyo trabajo consiste en definir el "cómo" se realiza cada KPA definida por CMMI. Una empresa que utilice MDD estará cubriendo muchas KPA de nivel 2 y 3 por el hecho de la utilización misma de los modelos y por su ciclo de vida. Hay KPAs que son independientes de la metodología, en cuyos casos se sugieren maneras de cubrirlas. Esta tesina hace una primera revisión de los puntos de conexión entre CMMI y MDD, animándonos a encontrar otros puntos de conexión de manera evolutiva.

Trabajos Realizados

- *Análisis de las ventajas de la inclusión de calidad.*
- *Análisis de las ventajas de los modelos y MDD.*
- *Investigación del Modelo CMMI nivel 2 y 3.*
- *Descripción de la metodología RUP.*
- *Investigación de puntos de conexión entre CMMI y RUP.*
- *Propuesta de mapeo entre CMMI y MDD.*

Trabajos Futuros

"Evolución de una empresa que implementa MDD a través de los niveles de CMMI": investigación y descripción de una empresa que ha incluido MDD en la ejecución de los proyectos y desea certificarse en CMMI.

"Adaptación de una empresa que implementa RUP y ha certificado en CMMI en la inclusión de MDD": ejemplo evolutivo de una empresa de gran envergadura que ya ha certificado en CMMI e implementa RUP y desea incorporar MDD a su proceso.

CMMI Aplicado a un Proyecto MDD

María Constanza Miguel

Tesina de Grado para obtener el título de

Licenciatura en Sistemas

**Facultad de Informática
de la
Universidad Nacional de La Plata**

Directora: Dra. Claudia Pons

Co-Directora: Dra. Roxana Giandini

Índice de contenido

Índice de contenido.....	2
Índice de tablas.....	4
Índice de ilustraciones.....	5
Capítulo 1: Introducción	6
1.1- Motivación	7
1.2- Objetivos	10
1.3- Organización de la Tesina	11
Capítulo 2: La calidad como objetivo a alcanzar en el desarrollo del software	13
2.1- La mejora de los Productos Software.....	14
¿Mejorar o cambiar?	14
¿A qué nos referimos con calidad?	14
¿Por qué medir la calidad?	15
Aspectos básicos de la calidad en el software	16
¿Qué es la gestión de calidad?	16
2.2 Propuestas sobre calidad	17
¿Cómo se mide la calidad?	17
Objetivos del Instituto de Ingeniería del Software de Estados Unidos (SEI)	21
2.3 Madurez e Inmadurez en las Organizaciones	21
El programa de evaluación del SEI	21
Diferencias entre madurez e Inmadurez	22
Capítulo 3: Integración de procesos y mejora de productos con CMMI.....	25
3.1 Introducción a CMMI	26
¿Qué es CMMI?	26
¿Por qué CMMI?	27
¿Qué cambios introduce la versión 1.2 de CMMI?	28
Integración CMM	28
El alcance del CMMI para el desarrollo	29
CMMI vs. ISO 9000	30
3.2 Representaciones	30
La representación continua	31
La representación por etapas	31
Comparación de las representaciones continua y por etapas	32
¿Qué representación utilizar?	33
3.3 Modelo por Niveles	33
Comprendiendo los niveles	33
Estructuras de las representaciones continua y por etapas	34
Comprendiendo los niveles de capacidad	36
Comprendiendo los niveles de madurez	40
3.4 Componentes del modelo	44
Componentes requeridos, esperados e informativos	44
Componentes informativos de soporte	50
Capítulo 4: Metodologías de Desarrollo y Ciclos de Vida.....	51
4.1- Metodologías de desarrollo	53
¿Cómo se organiza el desarrollo del software?	53
¿Qué es una metodología de desarrollo de software?	54

¿Qué metodología se debe utilizar?	54
Aspectos claves de RUP.....	56
Actividades realizadas en cada fase.....	58
Desarrollo de Software Dirigido por Modelos (MDD)	62
¿Qué propone MDD?	63
El ciclo de vida dirigido por modelos	64
Entendiendo las tareas y los roles en el proyecto de desarrollo.....	67
Las tareas.....	67
Los Roles.....	68
Planificación y seguimiento del proyecto de desarrollo.....	69
Evaluación del proyecto.....	70
Administración y reuso de los artefactos.....	70
Reuso de los artefactos.....	71
Reuso de las herramientas.....	72
El rol de UML en MDD.....	72
Conclusión.....	74
Capítulo 5: Una perspectiva común entre RUP, MDD y CMMI.....	76
Mapeo de cada Nivel de CMMI con RUP y MDD.....	77
Nivel 2 – Mapeo con RUP y CMMI.....	78
Nivel 3 – Mapeo con RUP y CMMI.....	104
Capítulo 6: Trabajos Relacionados y Futuros.....	134
¿Cuáles trabajos se relacionan con la presente tesina?.....	134
¿Qué otras investigaciones podrían realizarse tomando como base esta tesina?	134
Capítulo 7: Conclusión.....	136
Sobre la calidad en las empresas.....	136
Sobre los modelos.....	139
Apéndice I: Glosario.....	142
Apéndice II: Abreviaturas.....	144
Bibliografía.....	145

Índice de tablas

Tabla 1: Organizaciones Maduras e Inmaduras.....	24
Tabla 2: ISO vs. CMMI.....	30
Tabla 3: Representación continua y por Etapas	32
Tabla 4: Correspondencia entre niveles de capacidad y de madurez.....	36
Tabla 5: Áreas Claves de Proceso.....	47
Tabla 6: Abreviaturas.....	144

Índice de ilustraciones

Ilustración 1: Representación continua y por Etapas.....	35
Ilustración 2: Componentes del modelo.....	45
Ilustración 3: RUP.....	58
Ilustración 4: Ciclo de Vida Dirigido por Modelos.....	65
Ilustración 5: Los tres pasos principales en el proceso de desarrollo MDD.....	66
Ilustración 6: MDD: Proyecto dentro de otro proyecto.....	68
Ilustración 7: KPAs del Nivel 2.....	78
Ilustración 8: KPAs del Nivel 3.....	104

Capítulo 1: *Introducción*

Este breve capítulo presenta una introducción a la tesina contando la motivación que llevó a la elección de CMMI como modelo para la medición de calidad en una organización que utilice MDD como forma de desarrollar sus proyectos software, el objetivo principal a lograr con su desarrollo y cómo se organizan los capítulos siguientes.

1.1- Motivación

Una empresa de software exitosa es aquella que produce de una manera consistente software de calidad que satisface las necesidades cambiantes de sus usuarios. Una empresa que puede desarrollar este software de forma predecible y puntual, con un uso eficiente y efectivo de recursos, tanto humanos como materiales, tiene un negocio sostenible.

Para producir software que cumpla su propósito, hay que conocer e involucrar a los usuarios de forma disciplinada con el fin de extraer los requisitos reales del sistema. Para desarrollar software de calidad duradera, hay que idear una sólida base arquitectónica que sea flexible al cambio. Para desarrollar software rápida, eficiente y efectivamente, con el mínimo de desechos software y trabajo repetido, hay que tener la gente apropiada, las herramientas apropiadas y el enfoque apropiado. Para hacer todo esto de forma consistente y predecible, con una estimación de los costes del sistema en cada etapa de su vida, hay que disponer de un proceso de desarrollo sólido que pueda adaptarse a las necesidades cambiantes del problema en cuestión y de la tecnología.

[UML, 2da Edición]

Hoy en día no es común que las compañías desarrollen por sí mismas todos los componentes que forman parte de un producto o servicio. Frecuentemente, algunos se construyen en la compañía y otros se adquieren; después todos los componentes se integran en el producto o servicio final. Por ello, las organizaciones deben ser capaces de gestionar y controlar este complejo proceso de desarrollo y mantenimiento.

Los problemas que estas organizaciones encuentran implican soluciones que conciernen a toda la empresa y que requieren una aproximación integrada. La gestión eficaz de los activos de la organización es crítica para el éxito de su actividad. En esencia, estas organizaciones desarrolladoras de productos y servicios necesitan una manera de gestionar una aproximación integrada para sus actividades de desarrollo, como parte para lograr sus objetivos estratégicos.

Uno de los principales problemas que encuentran estas organizaciones, es la falta de calidad no sólo en el producto final entregado, sino en los procesos utilizados para desarrollarlos. Teniendo en cuenta la visión de construcción de software por múltiples entidades para luego ensamblarlas, se torna complicado encontrar una solución que aplique a todas las partes involucradas y que permitan alcanzar el objetivo propuesto.

Si bien en el mercado actual existen modelos de madurez, estándares, metodologías y guías que pueden ayudar a una organización a mejorar su modo de operar, la mayoría de estas aproximaciones de mejora se centran en una parte específica de su actividad y no adoptan una aproximación sistémica a los problemas a los que se enfrentan la mayoría de las organizaciones. Concentrándose en mejorar un área de negocio, estos modelos desafortunadamente han perpetuado los canales y las barreras que existen en el seno de las organizaciones.

El CMMI (*Capability Maturity Model Integration*) ^[CMMI] proporciona una oportunidad para evitar o eliminar estos canales y barreras, apoyándose en los modelos integrados que trascienden disciplinas. El CMMI para Desarrollo contempla las buenas prácticas relativas a las actividades de desarrollo y mantenimiento aplicadas a productos y servicios. Trata las prácticas que cubren el ciclo de vida del producto desde la concepción hasta la entrega y el mantenimiento. El énfasis lo pone en el trabajo necesario para construir y mantener el producto completo.

CMMI sirve para mejorar procesos, dice *qué* deben hacer los procesos para entregar productos y servicios de calidad pero no especifican *cómo*, quien lo define, es la metodología de desarrollo implantada en la organización.

Sea cual sea la metodología elegida (RUP, SCRUM, etc.), los puntos definidos por el CMMI aplican a cualquiera de ellas alcanzando diferentes niveles de madurez.

Hay un aspecto que es común a todas estas metodologías y es la utilización de modelos. Lo que varía de un método a otro es la clase de modelos que deben construirse, la forma de representarlos y manipularlos. A grandes rasgos podemos distinguir dos tendencias principales: los métodos matemáticos y los métodos diagramáticos. Estos últimos

constituyen una propuesta más amigable, más fáciles de utilizar y comprender que los métodos matemáticos.

El modelado es una parte central de todas las actividades que conducen a la producción de buen software. Construimos modelos para comunicar la estructura deseada y el comportamiento de nuestro sistema. Construimos modelos para visualizar y controlar la arquitectura del sistema. Construimos modelos para comprender mejor el sistema que estamos construyendo, muchas veces descubriendo oportunidades para la simplificación y la reutilización. Construimos modelos para controlar el riesgo. ^{[UML, 2da Edición][CMMI, 2da Edición]}

El MDD (*Model Driven Development o Desarrollo Dirigido por Modelos*) y más concretamente la propuesta MDA (*Model Driven Architecture*) ^[MDA] de OMG (*Object Management Group*) ^[OMG] constituyen una aproximación para el desarrollo de sistemas software, basada en la separación entre la especificación de la funcionalidad esencial del sistema y la implementación de dicha funcionalidad usando plataformas de implementación específicas. MDD plantea una nueva forma de entender el desarrollo y mantenimiento de sistemas de software con el uso de modelos como principales artefactos del proceso de desarrollo. En MDD, los modelos son utilizados para dirigir las tareas de comprensión, diseño, construcción, pruebas, despliegue, operación, administración, mantenimiento y modificación de los sistemas.

El MDD se está volviendo cada vez más importante en el desarrollo de modernas aplicaciones y sistemas empresariales. Las herramientas basadas en MDD proponen un desarrollo de software dirigido por modelos en el cual los lenguajes de modelado visual, como por ejemplo UML, son utilizados para integrar la variedad de tecnologías utilizadas en el desarrollo de sistemas de software. Asimismo, el paradigma de MDD nos facilita una mejor forma de enfrentar y resolver problemas de interoperabilidad en comparación a otros enfoques que no utilizan modelos. El nivel actual de desarrollo del MDD ha sido influenciado de gran forma por las actividades de estandarización de la OMG, centradas en MDA. Su enfoque y los estándares que lo acompañan permiten concretar e integrar un modelo en múltiples modelos objeto-dependientes de cierta plataforma de software.

Este proceso de concreción es alcanzado mediante la aplicación de transformaciones de modelo-a-modelo y de modelo-a-texto.

1.2- Objetivos

El objetivo de esta tesina es definir los puntos planteados por el modelo de madurez CMMI (metas y prácticas específicas de las áreas clave de proceso) que son mapeados directamente con el modelo MDD y para los casos en que no exista un claro y directo mapeo, se presentan pautas generales para que la organización tenga en cuenta y pueda cubrir esas áreas clave con otras actividades que caen fuera de la definición de MDD.

No se trata de definir la evolución de la organización que implementa MDD a través de los niveles de CMMI sino más bien de realizar un recorrido por los niveles de madurez y validar las relaciones existentes con la metodología de desarrollo implementada en la organización.

Dado que CMMI no impone una metodología en particular de desarrollo de software para poder realizar una medición de madurez, y tampoco define las condiciones que se deben cumplir para cada tipo de metodología, se definen las pautas a aplicar para MDD dada la poca información existente sobre el tema.

Para lograr el objetivo, se desarrolla una investigación sobre el modelo CMMI-SW describiendo los requisitos que debe cumplir un proyecto de software, independientemente de la metodología de desarrollo que utilice, para aplicar a cada uno de los cinco niveles definidos.

Tomando como base esta investigación, se realiza el correspondiente mapeo con la metodología de desarrollo RUP y luego lo que nos interesa, el mapeo con MDD.

1.3- Organización de la Tesina

Si se analiza el título de la tesina, vemos que está compuesto por dos conceptos que, tomados aisladamente, tienen su propio protagonismo dentro del área sobre el cual aplican: la medición de la calidad, y el desarrollo de software respectivamente.

Es por tal motivo que resulta necesario comprender cada uno de estos conceptos separadamente antes de abordar el capítulo 5, en el cual se los vincula y mapea.

La introducción a los temas, el punto de partida de la tesina empieza con el capítulo 2, donde se describen conceptos básicos sobre *calidad*: qué se entiende por este concepto, cómo se aplica a un producto software, los estándares existentes y una breve presentación del SEI, el Instituto que definió los puntos que debía cumplir una organización para alcanzar niveles de calidad y del cual nace CMMI.

Pero, calidad se puede aplicar a muchos ámbitos. En particular, y aquí, interesa la calidad en las organizaciones desarrolladoras de software. La calidad va a ser un parámetro para determinar si tales organizaciones son maduras o inmaduras. Una organización declarada como “inmadura”, para convertirse en una madura lo hace siguiendo una serie de pautas, actividades, normas y trabajo colaborativo de todos los involucrados. La calidad debe estar presente en toda la organización y durante todo el ciclo de vida del software, no se limita a una área en concreto.

Para este trabajo interesan los puntos definidos por el CMMI: un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software, tema desarrollado en el capítulo 3.

La versatilidad de CMMI reside en el hecho de que abarca todos los aspectos y áreas de una organización describiendo en cada caso qué metas deben cumplirse pero no cómo. El “cómo” lo establece cada organización siguiendo sus propias decisiones y actividades. Hay empresas que eligen las normas ISO para certificar la calidad de su trabajo. ISO es conocida en el mercado por su aplicación en múltiples productos y actividades, no solo

los relacionados con el software. Es por eso, que se realiza un análisis entre estos estándares y se justifica la elección por CMMI.

Siguiendo por el camino al tema central de la tesina, se presentan en el capítulo 4 definiciones sobre las metodologías de desarrollo que se aplican en una organización para coordinar, organizar, estandarizar el desarrollo de sus productos software. Existen muchas metodologías, siendo tal vez RUP, la más conocida popular y elegida por las empresas, sobre todo en aquellas de gran magnitud. Es por tal motivo que resulta imprescindible incluir en este capítulo cómo es el proceso de desarrollo RUP. No hace falta tener grandes conocimientos de RUP ni conocer los detalles de la metodología, la idea de incluirlo es para unificar los conocimientos de todo aquel que lee este trabajo.

Y como lo que realmente interesa aquí, es saber cómo aplica a MDD, se describe cómo es el desarrollo en una empresa que aplica MDD. Se van a encontrar varias similitudes con RUP y otros aspectos relevantes se dejan para el capítulo 5.

Con la idea de conocer cómo aplicar CMMI a MDD, se decidió realizar el mapeo en dos fases, la primera de ellas corresponde a su mapeo con RUP, para que quien lea y tenga conocimientos de esta metodología, se encuentre familiarizado con los puntos que plantea CMMI. Comprendida esta fase, se realiza su mapeo con MDD. Este tema se desarrolla en el capítulo 5.

La forma de realizar el mapeo es realizando una descripción de cada área clave de proceso de los niveles 2 y 3 (con sus notas introductorias y listando las metas y prácticas específicas) y luego tomar alguna de esas prácticas específicas y realizar el mapeo con MDD. Para el caso de no encontrar mapeo directo, se toma la meta específica y a grandes rasgos se describen los pasos que debería seguir la organización para no dejar sin cubrir esa área clave de proceso.

Se concluye esta tesina con un capítulo de Conclusión y Apéndices que pueden ayudar a la comprensión de ciertos términos que se mencionan a lo largo del trabajo.

Capítulo 2: *La calidad como objetivo a alcanzar en el desarrollo del software*

Es común decir que un producto, cualquiera sea su naturaleza, es de buena calidad o de mala calidad. Pero, ¿se sabe objetivamente a qué nos estamos refiriendo? ¿qué es lo que estamos juzgando para arribar a esa conclusión?. Podría decirse que uno generaliza el tema "calidad" cuando algún aspecto del producto evaluado no satisface alguna condición esperada por quien observa, sin conocer qué aspectos definen la calidad para ese producto. Estos aspectos se definen generalmente antes de comenzar con la construcción del mismo a modo de objetivo, y lógicamente, éstos varían de producto en producto y de quienes los definen, es decir, las personas interesadas en el resultado del mismo.

Para los productos software, el principal instrumento para garantizar la calidad es contar con un Modelo (o Plan) de Calidad, el cual se basa en normas o estándares genéricos y en procedimientos particulares. Los procedimientos pueden variar en cada organización, pero lo importante es que estén escritos, personalizados, adaptados a los procesos de la organización y que sean cumplidos.

Este capítulo se focaliza en la calidad aplicada a un producto software y a la calidad de toda la organización desarrolladora de software. Es importante comprender este concepto para saber de qué manera se lo considera utilizando una metodología MDD para poder certificar en CMMI.

Al finalizar el capítulo se podrán responder las siguientes preguntas:

- ¿Qué significa calidad?
- ¿Por qué es tan importante incluirla en el desarrollo de un software?
- ¿Cuáles son los estándares existentes para aplicar?
- ¿Qué significa que una organización sea madura/inmadura?
- ¿Cuál es la propuesta del SEI para la medición de la calidad en una organización?

2.1- La mejora de los Productos Software

¿Mejorar o cambiar?

Es una pregunta fundamental que surge al momento de detectar que algo no está funcionando como se planificó, por lo tanto, para saber qué camino tomar, es importante comprender ambos significados. Por definición, se entiende como "cambio" al dejar una cosa o situación para tomar otra, y "mejora" como el hecho de acrecentar algo, haciéndolo pasar a un estado mejor. ^[RAE]

En referencia a un producto software, uno de los tantos objetivos planteados, es lograr cierto nivel de calidad. Cuando éste no se logra, entonces hablamos de mejorar el desarrollo (la construcción) del software, estableciendo pautas para que la calidad sea incluida en todos los pasos del desarrollo.

¿A qué nos referimos con *calidad*?

Se presentan dos definiciones estandarizadas, publicadas y conocidas sobre este término. La primera corresponde al estándar IEEE, Std. 610-1990 y dice lo siguiente:

El grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario.

Esta definición bien podría aplicarse a cualquier otro producto de naturaleza diferente ya que por lo general, cualquier producto que se quiera construir deberá cumplir con requerimientos, necesidades o expectativas de usuario que se proponen cuando se analizan los requerimientos del mismo. Una definición un tanto más completa y más acorde al tema de la tesina es la publicada en el año 1998 por Pressman y dice lo siguiente:

La Calidad del Software es la concordancia con los requerimientos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo

documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente.

Teniendo en cuenta la definición anterior, se puede decir que los requisitos del software son la base de las medidas de calidad y que la falta de concordancia con los requisitos es una falta de calidad. Los estándares o metodologías definen un conjunto de criterios de desarrollo que guían la forma en que se aplica la Ingeniería del Software. Si no se sigue ninguna metodología siempre habrá falta de calidad. Todas las metodologías y herramientas tienen un objetivo común: producir software de alta calidad.

A la hora de definir la calidad del software se debe diferenciar entre la calidad del producto de software y la calidad del proceso de desarrollo. No obstante, las metas que se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo, ya que la calidad del producto va a estar en función de la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo es casi imposible obtener un buen producto. No basta con tener en cuenta la calidad del producto una vez finalizado, cuando los problemas de mala calidad ya no tienen solución o la solución es muy costosa.

¿Por qué medir la calidad?

Hoy en día las compañías que producen software para satisfacer a un mercado creciente que reclama este tipo de soluciones tecnológicas deben buscar continuamente alternativas que les permitan mejorar su *performance* y calidad de productos para poder seguir compitiendo en un escenario cada vez más globalizado y agresivo. Esta situación se torna más obvia cuando analizamos las necesidades de la producción global de software:

- Mejor performance
- Más eficiencia
- Evitar pérdidas de mercado
- Recursos humanos mejor preparados
- Productos que faciliten la integración de diferentes tecnologías.

Si visualizamos estos puntos de una manera más general podríamos afirmar que la industria tiene una fuerte necesidad de optimización de todas las facetas del desarrollo de productos. Reforzando esta afirmación es hoy mundialmente aceptado que la calidad del proceso utilizado para desarrollar un determinado producto impacta fuertemente en la calidad final de éste. Por lo tanto, el mejoramiento de los procesos de desarrollo deja de ser una posible opción para convertirse en una estrategia de supervivencia.

Aspectos básicos de la calidad en el software

Para lograr calidad en el software, es necesario desarrollar los siguientes aspectos básicos:

- Las expectativas del cliente deben ser traducidas en requisitos;
- Desarrollar un sistema para cumplir con dichos requisitos de forma previsible;
- Establecer una metodología de actuación para llegar a "cero defectos" en el trabajo;
- Disponer de un procedimiento de medida para el control de la satisfacción del cliente. Previamente a ello, será preciso conocer perfectamente y a fondo las capacidades de la empresa: su organización y procedimientos internos de gestión, sus métodos estandarizados, su sistema de control y seguimiento de los procesos de desarrollo y sus herramientas de soporte.

Una organización no preparada para dar calidad, en el sentido esperado por sus clientes, caerá muy rápidamente en la situación de expectativas no cumplidas o no alcanzadas.

¿Qué es la gestión de calidad?

Se puede entender como el conjunto de actividades y medios necesarios para definir e implantar un sistema de calidad, por una parte, y responsabilizarse de su control, aseguramiento y mejora continua, por otra. En este sentido, la gestión de la calidad en

cualquier organización (y, por supuesto, en las dedicadas al desarrollo y mantenimiento de software) cuenta con dos niveles de trabajo:

- El nivel de entidad u organización, donde se trata de crear y gestionar una infraestructura que fomente la calidad de los productos software mediante la adecuación y mejora de las actividades y procesos involucrados en su producción e incluso, en su comercialización y en la interacción con los clientes.
- El nivel de proyecto, donde las guías que la infraestructura organizativa prevé para las distintas actividades de desarrollo y mantenimiento de software deben ser adaptadas a las características concretas del proyecto y de su entorno para ser aplicadas en la práctica.

Dentro del primer nivel de acción, la gestión de la calidad en organizaciones de software ha seguido dos líneas que pueden ser complementarias entre sí:

Por una parte, se ha seguido la línea marcada por las entidades internacionales de estandarización para todas las organizaciones de producción o servicios. Principalmente, se ha impuesto en la práctica las directrices marcadas por ISO (*Organization for International Standardization*) ^[ISO] a través de su serie de normas ISO 9000 para la gestión de calidad.

Por otra parte, el SEI (*Software Engineering Institute*) ^[SEI] propone un modelo de clasificación y mejora de los procesos empleados por las organizaciones de software denominado CMMI. Su trabajo se centra en el estudio y clasificación de los distintos procesos involucrados en la producción de software bajo el enfoque de una serie de niveles de madurez.

2.2 Propuestas sobre calidad

¿Cómo se mide la calidad?

Como se mencionó anteriormente, la medición de calidad se realiza sobre dos enfoques: la calidad del producto y la calidad del proceso empleado en la construcción del mismo.

Para saber cómo realizar una medición, es necesario contar con algún modelo/guía que nos sirva de base para saber qué aspectos son los que se deben evaluar.

Construir un Modelo de Calidad es bastante complejo y es usual que estos modelos descompongan la calidad del producto software jerárquicamente en una serie de características y subcaracterísticas que pueden usarse como una lista de comprobación de aspectos relacionados con la calidad.

A nivel producto, algunos de los modelos propuestos son:

- **Modelo de Gilb:** plantea la creación de una especificación de requisitos de calidad para cada proyecto que deben escribir conjuntamente el usuario y el analista. Es un modelo que permite determinar una lista de características que definen la calidad de la aplicación.
- **Modelo GQM (*Goal – Question - Metric*):** es una propuesta de objetivos/metas orientado a la definición de modelos de calidad. Plantea el enfoque de medición para evaluar la calidad del software basado en la identificación de objetivos a lograr.
- **Modelo FURPS:** cuenta con cinco características de calidad del software: (1) Funcionalidad, (2) Facilidad de uso, (3) Confiabilidad, (4) *Performance* y (5) Facilidad de soporte. Además plantea 2 categorías de requerimientos, las cuales son:
 - Requerimientos funcionales (F): especifican funciones que el sistema debe ser capaz de realizar, sin tomar restricciones físicas a consideración, y se definen a través de las entradas y salidas esperadas.
 - Requerimientos no funcionales (URPS): *Usability* (Facilidad de uso), *Reliability* (Confiabilidad), *Performance* y *Supportability* (Facilidad de soporte). describen atributos del sistema o atributos del ambiente del sistema.
- **Modelo de McCall:** organiza los factores en tres ejes o puntos de vista desde los cuales el usuario puede contemplar la calidad de un producto:
 - Características operativas
 - Corrección. ¿Hace lo que quiero?

- Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- Seguridad (Integridad). ¿Es seguro?
- Facilidad de uso. ¿Está diseñado para ser usado?
- Capacidad de soportar los cambios
- Facilidad de mantenimiento. ¿Puedo corregirlo?
- Flexibilidad. ¿Puedo cambiarlo?
- Facilidad de prueba. ¿Puedo probarlo?
- Adaptabilidad a nuevos entornos
 - Portabilidad. ¿Podré usarlo en otra máquina?
 - Reusabilidad. ¿Podré reutilizar alguna parte del software
 - Interoperabilidad. ¿Podré hacerlo interactuar con otro sistema?

A nivel proceso, la ISO ha publicado los siguientes estándares:

ISO-9000. Dentro de la familia ISO-9000 existen tres estándares: ISO-9001, ISO-9002 e ISO-9003, que se deben considerar dependiendo del alcance de las actividades de la empresa. En el caso del desarrollo de software, ISO-9001 es la más apropiada, ya que incluye actividades de diseño y desarrollo. En cierta manera, ISO-9000 es comparable con CMMI por ser una familia de estándares, mientras que ISO-9001 es similar a CMM. De tal manera, ISO-9001 es un estándar aplicable a la administración de la calidad en el desarrollo de software. A diferencia de CMM, el modelo ISO-9001 no incluye múltiples niveles de calidad. La certificación es equivalente al nivel tres de la escala de SEI.

ISO-12207. Es una familia de normas internacionales relacionadas con la administración de la calidad en los procesos del ciclo de vida del software. Esta norma está dirigida a lograr acuerdos o contratos entre los desarrolladores y clientes donde se requiere el desarrollo, mantenimiento u operación de un sistema de software. Es una norma de alto

nivel que deja sin especificar los detalles de cómo llevar a cabo las actividades y tareas de los procesos. Se describen cinco procesos primarios: adquisición, suministro, desarrollo, mantenimiento y operación. Los cinco procesos se dividen en actividades y las actividades en tareas, con lo cual se agregan requisitos para su ejecución. Así, se especifican ocho procesos: de apoyo, documentación, administración de configuración, aseguramiento de calidad, verificación, validación, auditoría y resolución de problemas. Además, se consideran cuatro procesos organizacionales: administración, infraestructura, mejora y entrenamiento. Esta norma busca que las organizaciones adapten estos procesos según el alcance de los proyectos particulares, eliminando actividades que no se aplican.

ISO-15504. El modelo de mejora del proceso de software y determinación de capacidad (SPICE, *Software Process Improvement and Capability Determination*), es un modelo de evaluación o determinación de la capacidad de los procesos, conocido también como la norma ISO-15504. Es una familia de normas internacionales que tiene como objetivo el desarrollo de sistemas de calidad en el software. Combina los enfoques de CMM con los de ISO-9000, incorporando al marco de referencia de ISO 9000 con la evaluación de capacidad y madurez de proceso de CMM. Su objetivo es lograr ganancias significativas en productividad y calidad, además de ayudar a los compradores de productos de software a obtener un mayor retorno para su inversión y reducir el riesgo asociado con los grandes proyectos de software. Este modelo busca mejorar la calidad del producto mediante una evaluación comprobada, consistente y confiable del estado de los procesos de software de una organización y usar los resultados de estas evaluaciones como parte de programas coherentes de mejora. La norma establece un denominador común para una evaluación uniforme de los procesos de desarrollo; ya que la tecnología evoluciona, ISO-15504 hace énfasis en la calidad, actualización y vigencia del producto.

Objetivos del Instituto de Ingeniería del Software de Estados Unidos (SEI)

El SEI es un centro de investigación y desarrollo perteneciente a la Universidad de Carnegie Mellon, fundado y financiado por el Departamento de Defensa de los Estados Unidos, a través de la Oficina de la Subsecretaría de Defensa para Adquisición, Tecnología y Logística. Su meta es proporcionar a las organizaciones las pautas de actuación necesarias para obtener mejoras observables en su proceso del software, de manera que desarrollen productos sin defectos respetando requerimientos, fechas y costes. Esto se consigue mediante el cumplimiento de cuatro objetivos:

- Acelerar la introducción en las organizaciones de producción de software de las prácticas y técnicas de ingeniería del software más eficaces y eficientes, identificando, evaluando y mejorando aquellas que se consideren útiles.
- Mantener a largo plazo la competencia en ingeniería del software y en la gestión del cambio tecnológico.
- Habilitar a organizaciones privadas y públicas, trabajando con ellas, para que hagan mejoras en sus prácticas de ingeniería del software.
- Fomentar la adopción y uso continuo de estándares de excelencia en prácticas de ingeniería del software.

2.3 Madurez e Inmadurez en las Organizaciones

El programa de evaluación del SEI

Con la finalidad de determinar la madurez de los procesos de las organizaciones y de hacer más efectivos sus procesos de mejora, el SEI ha diseñado un Programa de Evaluación en el que se especifican los procedimientos necesarios para alcanzar y mantener los mayores niveles de calidad en la utilización de su tecnología de evaluación, y hacer así posible la coordinación y coherencia de las distintas mediciones obtenidas.

Aquellas organizaciones o profesionales independientes que cumplan con determinados requisitos de solvencia técnica, experiencia y conocimientos (según formación reglada por el SEI) pueden ser autorizadas por el SEI, mediante la firma de un acuerdo por el que la organización se convierte en asociada al SEI para realizar evaluaciones, a llevar a cabo proyectos de evaluación según los modelos y métodos de ambas áreas.

Los asociados al SEI están obligados a transmitirle los resultados de las evaluaciones que lleven a cabo. Estos resultados se almacenan para uso exclusivo de empleados del SEI a los que puedan ser útiles para tareas de investigación y desarrollo, sin mantenerse referencia alguna a proyectos o empresas. Por tanto, el SEI no certifica resultados de evaluación ni nivel de madurez de una organización, ya que ello le obligaría a realizar un control continuo de la calidad de los procesos, ni confirma ni desmiente que se haya realizado un proyecto de evaluación. Los resultados de una evaluación son propiedad de quien la financie, que tendrá la potestad de darles la publicidad que estime oportuna, pero nunca utilizando el emblema del SEI. No obstante, el SEI publica los niveles alcanzados por los clientes de los evaluadores asociados cuando dispone de la autorización de quien haya financiado la evaluación.

Diferencias entre madurez e Inmadurez

Para entender cuáles son las mejoras más críticas en una organización, es necesario comprender las diferencias entre madurez e inmadurez.

En una organización de software inmadura, los procesos software son generalmente improvisados por profesionales y sus respectivos directores en todo el curso del proyecto. Incluso si un proceso ha sido especificado, puede que no se siga ni cumpla rigurosamente. Las organizaciones de software inmaduras son reaccionarias, y sus gerentes usualmente están focalizados en solucionar crisis inmediatas. Estas organizaciones exceden rutinariamente tiempos y presupuestos porque no se basan en

estimaciones realistas. Cuando se imponen los plazos, pueden comprometerse tanto la funcionalidad como la calidad del producto para cumplir los tiempos.

En una organización inmadura, no existen bases objetivas para juzgar la calidad del producto, o para solucionar problemas de procesos o productos. Poco se sabe sobre cómo los pasos en un proceso de software afectan a la calidad, y la calidad de un producto es difícil de predecir. Por otro lado, las actividades previstas para realzar la calidad, tal como revisiones (*reviews*) y *testing*, a menudo son acortadas o eliminadas cuando los proyectos tienen demoras en la entrega del proyecto. El cliente poco conoce del proyecto sino hasta el momento de la entrega.

Una organización de software madura, en cambio, posee una amplia habilidad para dirigir el desarrollo de software y mantener los procesos. El SEI ha identificado varias dimensiones, sobre las que una organización puede enfocarse para mejorar su actividad: las personas, los métodos y procedimientos, y las herramientas y equipamiento.

Pero, ¿qué es lo que sustenta todo el conjunto? Los procesos utilizados en su organización. Éstos le permiten alinear el modo de operar de su organización, evolucionar e incorporar los conocimientos de cómo hacer mejor las cosas. Los procesos le permiten también explotar mejor sus recursos y comprender las tendencias de su actividad.

El proceso acordado es documentado, utilizado, y consistente con la manera de trabajo realizado hasta el momento por los desarrolladores. Las definiciones de los procesos son actualizadas cuando es necesario, y las mejoras son desarrolladas a través de pruebas piloto y/o análisis de costo-beneficio. Los papeles y las responsabilidades dentro del proceso están claros a través del proyecto y de la organización. En una organización madura, los gerentes monitorean la calidad de los productos software y los procesos que producen. Esta es una base objetiva y cuantitativa para juzgar la calidad de los productos y analizar los problemas con el producto y los procesos. Los tiempos y presupuestos se basan en funcionamientos históricos y son realistas; los resultados

esperados para los costos, tiempos, funcionalidad y calidad del producto son usualmente alcanzados. Generalmente, la organización madura sigue un proceso disciplinado constantemente porque todos los participantes entienden el valor de lo que se está haciendo, y la infraestructura necesaria para soportar el proceso.

El siguiente cuadro resume las diferencias entre madurez e inmadurez de las organizaciones:

Organizaciones con procesos inmaduros	Organizaciones con procesos maduros
Procesos improvisados por los gerentes y desarrolladores.	Procesos documentados.
Cada uno posee sus propios procesos.	Procesos seguidos consistentemente.
Procesos comprometidos en orden a cumplir los costos y las fechas acordadas.	El rendimiento de los procesos es medido, seguido y entendido.
Calidad difícil de predecir.	La calidad es predecible porque los procesos están bajo control.
Los procesos “viven” mientras viven los desarrolladores.	Los procesos “viven” por sí solos y son mejorados continuamente.
Las nuevas tecnologías corren riesgos de caer en desuso.	Las nuevas tecnologías son incorporadas de manera disciplinada.

Tabla 1: Organizaciones Maduras e Inmaduras

Capítulo 3: Integración de procesos y mejora de productos con CMMI

Una vez comprendido el concepto de calidad, su importancia en el software y cuáles son las propuestas existentes en el mercado para que puedan ser adoptadas por las organizaciones, este capítulo se centra en CMMI, el modelo elegido en el presente trabajo para su aplicación en una organización que desarrolla software utilizando el modelo MDD, la justificación de su elección, los tipos de representaciones y las componentes del modelo.

No se realiza un análisis exhaustivo del modelo, sino que se da una versión simplificada del mismo para conocer los conceptos que interesan en el trabajo, los suficientes para relacionar con metodologías de desarrollo. El objetivo entonces, es conocer las partes principales de CMMI que puedan llegar a tener relación con las metodologías, tema que se desarrolla en el capítulo siguiente.

Al finalizar el capítulo se podrán responder las siguientes preguntas:

- ¿Qué es CMMI?
- ¿Cuáles son las componentes del modelo?
- ¿Cuáles son los niveles que define CMMI según las representaciones que define?
- ¿Cuáles son las diferencias entre CMMI y las normas ISO?

3.1 Introducción a CMMI

El principio de CMMI es: “la calidad de un sistema o de un producto está muy influenciada por la calidad del proceso empleado para desarrollarlo y para mantenerlo”

¿Qué es CMMI?

CMMI es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios.

No es una metodología de desarrollo de software ni de gestión de proyectos. Tampoco asume el modelo en cascada ni el iterativo.

Consiste en las mejores prácticas que tratan las actividades de desarrollo y de mantenimiento que cubren el ciclo de vida del producto, desde la concepción a la entrega y el mantenimiento.

El CMMI para Desarrollo contempla las buenas prácticas relativas a las actividades de desarrollo y mantenimiento aplicadas a productos y servicios. Trata las prácticas que cubren el ciclo de vida del producto desde la concepción hasta la entrega y el mantenimiento. El énfasis lo pone en el trabajo necesario para construir y mantener el producto completo.

CMM es una aplicación de sentido común de los conceptos de gestión de procesos y mejora de la calidad al desarrollo y mantenimiento del software.

Algunas características son:

- Estudia los procesos de desarrollo de software de una organización y produce una evaluación de la madurez de la organización según una escala de cinco niveles.
- La madurez de un proceso es un indicador de la capacidad para construir un software de calidad.
- Es un modelo para la mejora de las organizaciones.
- Obliga a una revisión constante.

El modelo de CMM esencialmente sigue el concepto “*Dí lo que haces, Haz lo que dices, Pruébalo*”:

- Di lo que Haces: Documentar.
- Haz lo que Dices: Desarrollar.
- Pruébalo: Registros de Calidad y artefactos del proyecto.

¿Por qué CMMI?

En el mercado actual, existen modelos de madurez, estándares, metodologías y guías que pueden ayudar a una organización a mejorar su modo de operar. Sin embargo, la mayoría de las aproximaciones de mejora disponibles se centran en una parte específica de su actividad y no adoptan una aproximación sistémica a los problemas a los que se enfrentan la mayoría de las organizaciones. Concentrándose en mejorar un área de negocio, estos modelos desafortunadamente han perpetuado los canales y las barreras que existen en el seno de las organizaciones.

El CMMI proporciona una oportunidad para evitar o eliminar estos canales y barreras, apoyándose en los modelos integrados que trascienden disciplinas. El CMMI para Desarrollo contempla las buenas prácticas relativas a las actividades de desarrollo y mantenimiento aplicadas a productos y servicios. Trata las prácticas que cubren el ciclo de vida del producto desde la concepción hasta la entrega y el mantenimiento. El énfasis lo pone en el trabajo necesario para construir y mantener el producto completo.

Un reporte creado y publicado en el año 2005 afirma que CMMI realmente funciona:

- Reducción de Costos en un 20% en promedio
- Reducción de Tiempos en un 37% en promedio
- Aumento de la productividad en un 62% en promedio
- Calidad en un 50% en promedio
- Satisfacción del Cliente en un 14% en promedio [REPORTE CMMI]

¿Qué cambios introduce la versión 1.2 de CMMI?

Esta última versión del modelo, integra los cuerpos del conocimiento que son esenciales para el desarrollo y el mantenimiento, pero que se han tratado por separado en el pasado, tales como la ingeniería del software, la ingeniería de sistemas, la ingeniería del hardware y de diseño, los aspectos no funcionales y la adquisición. Las denominaciones anteriores de CMMI para la ingeniería de sistemas y la ingeniería del software (CMMI-SE/SW) son reemplazadas por el título “CMMI para desarrollo”, reflejando así realmente la integración completa de estos cuerpos de conocimiento y la aplicación del modelo en el seno de una organización. CMMI para desarrollo (CMMI-DEV) propone una solución integrada y completa para las actividades de desarrollo y de mantenimiento aplicadas a los productos y a los servicios.

CMMI para desarrollo, versión 1.2, es una continuación y actualización de CMMI versión 1.1 y ha sido simplificada gracias al concepto de “constelaciones” de CMMI, donde un conjunto de componentes fundamentales puede ser ampliado mediante material adicional a fin de proponer unos modelos específicos de aplicación con elevado contenido común. CMMI-DEV es la primera de esas constelaciones y representa al dominio de interés de desarrollo.

Integración CMM

El proyecto de integración de CMM ha sido realizado para regular el problema de utilizar múltiples CMM. La misión inicial del equipo del producto CMMI (*CMMI Product Team*) fue combinar tres modelos fuente:

1. SW-CMM (*Capability Maturity Model for Software*), version v2.0 draft C
2. SECM (*Systems Engineering Capability Model*)
3. IPD-CMM (*Integrated Product Development Capability Maturity Model*), v0.98

La combinación de estos modelos en un único marco de mejora fue pensada para permitir a las organizaciones utilizar éste en su búsqueda de la mejora de procesos en toda la empresa.

Estos tres modelos fuente fueron seleccionados debido a su extensa adopción por las comunidades de desarrollo de sistemas y de software y también porque proponen diversos acercamientos a la mejora de procesos en el seno de una organización.

La integración de estos modelos pueden adoptarse tanto por aquellos que utilizan actualmente los modelos fuente como por aquellos nuevos al concepto de CMM. Así, el CMMI resulta directamente de la evolución de los modelos SW-CMM, SECM e IPD-CMM. [CMMI, 2da Edición]

El alcance del CMMI para el desarrollo

CMMI para desarrollo es un modelo de referencia que cubre las actividades del desarrollo y del mantenimiento aplicadas tanto a los productos como a los servicios.

Una constelación es una colección de componentes de CMMI que incluye un modelo, sus materiales de formación y los documentos de evaluación concernientes a un dominio de interés. Actualmente hay tres constelaciones planificadas que se sostienen en el marco del modelo de la v1.2: desarrollo, servicios y adquisición. Las “extensiones” se utilizan para extender las constelaciones mediante contenido específico adicional.

Los modelos de la constelación del CMMI para desarrollo contienen prácticas que cubren la gestión de proyectos, la gestión de procesos, la ingeniería de sistemas, la ingeniería del hardware, la ingeniería de software y otros procesos de soporte utilizados en el desarrollo y el mantenimiento. El modelo CMMI para desarrollo + IPPD (Desarrollo Integrado de Productos y Procesos) cubre también la utilización de equipos integrados que están implicados en las actividades de desarrollo y mantenimiento (IPPD).

CMMI vs. ISO 9000

A continuación, se presenta un resumen de las principales diferencias entre el modelo CMMI y el estándar ISO:

ISO 9000	CMMI
La norma ISO 9000 es muy general, ya que su concepción fue pensada para abarcar cualquier tipo de empresa.	CMMI nació enfocada hacia el desarrollo concreto de productos SW.
La norma ISO 9000, contempla un escenario más amplio en la producción y presenta una gran atención hacia el cliente.	CMMI se restringe al proceso de producción de forma más específica y también, por tanto más eficaz. CMMI y su espíritu incremental, motivan al personal.
La norma ISO 9000 tiene un carácter mucho más estático, proporcionando mecanismos de mejora mucho más débiles.	CMMI mueve al sistema de calidad de la empresa en una dirección de mejora continua.
La norma ISO 9000 provoca una realidad empresarial en la que el único reto es mantener la certificación.	CMMI es más flexible en su implantación.
La norma ISO es árida e inamovible con respecto a su definición.	CMMI aporta a la empresa capacidad de auto evaluación y cierta independencia.
La norma ISO 9000 requiere de auditorias externas periódicas, para comprobar su continuidad y adecuación.	CMMI se basa en la evaluación de la capacidad y madurez del proceso, mientras que la norma ISO 9000 chequea la organización en base a la comparación con unos estándares predefinidos.

Tabla 2: ISO vs. CMMI

3.2 Representaciones

CMMI soporta dos caminos de mejora. Un camino permite a las organizaciones mejorar de forma incremental los procesos que corresponden a un área de proceso individual seleccionada por la organización. El otro camino permite a las organizaciones mejorar un conjunto de procesos relacionados, tratando de forma incremental conjuntos sucesivos de áreas de proceso. Cada uno de estos caminos tiene ventajas sobre el otro.

Algunas organizaciones utilizan ambas representaciones para responder a unas necesidades particulares en diferentes momentos de sus programas de mejora.

La representación continua

La representación continua ofrece la máxima flexibilidad cuando se utiliza un modelo CMMI para la mejora de procesos. Una organización puede elegir mejorar el rendimiento de un punto problemático relacionado con un solo proceso, o puede trabajar en varios dominios que están fuertemente alineados con sus objetivos estratégicos. La representación continua también permite que una organización mejore diferentes procesos a diferentes niveles. Las dependencias que existen entre algunas áreas de proceso pueden, sin embargo, limitar un poco las elecciones. Si se sabe de antemano qué procesos necesitan ser mejorados en la organización y se conoce las dependencias existentes entre las áreas de proceso descritas en el CMMI, la representación continua constituye entonces la elección pertinente. Esta representación permite a la organización elegir el enfoque de sus esfuerzos de mejora de procesos mediante la elección de aquellas áreas de proceso, o conjuntos de áreas de proceso interrelacionadas, que benefician más a la organización y a sus objetivos de negocio.

La representación continua permite a la organización elegir el enfoque de sus esfuerzos de mejora de procesos mediante la elección de aquellas áreas de proceso, o conjuntos de áreas de proceso interrelacionadas, que benefician más a la organización y a sus objetivos de negocio. Aunque existen algunos límites sobre lo que una organización puede elegir debido a las dependencias entre las áreas de proceso, la organización tiene una libertad considerable en su selección.

La representación por etapas

La representación por etapas ofrece una manera sistemática y estructurada de aproximarse a la mejora de procesos basada en el modelo etapa a etapa. El logro de cada etapa asegura que una infraestructura de proceso adecuada se ha establecido como fundamento para el etapa siguiente.

Las áreas de proceso están organizadas por niveles de madurez, los cuales eliminan interpretaciones a la mejora de los procesos. La representación por etapas prescribe un orden para implementar las áreas del proceso según unos niveles de madurez, que determinan el camino seguido por una organización para pasar del nivel inicial al nivel “en optimización”.

Alcanzar cada nivel de madurez asegura que se ha establecido un fundamento adecuado para el siguiente nivel de madurez, lo que permite una mejora incremental y duradera.

Si no se sabe por dónde comenzar ni qué procesos elegir para mejorar, la representación por etapas es la opción designada. Esta ofrece un conjunto específico de procesos para mejorar en cada etapa.

Comparación de las representaciones continua y por etapas

La siguiente tabla compara las ventajas de cada representación y puede ayudar a determinar qué representación conviene a cada organización:

Representación continua	Representación por etapas
Concede la libertad explícita para seleccionar el orden de mejora que mejor satisface los objetivos de la organización y atenúa las áreas de riesgo de la organización.	Permite a las organizaciones tener una trayectoria predefinida y probada de mejora.
Permite visibilidad incrementada de la capacidad alcanzada en cada área de proceso individual.	Se centra en un conjunto de procesos que proveen a una organización con una capacidad específica que está caracterizada por cada nivel de madurez.
Permite que las mejoras de diversos procesos sean realizadas en diversos valores.	Resume resultados de la mejora de procesos en un simple número de nivel de madurez.
Refleja una aproximación nueva, que todavía no tiene los datos para demostrar sus relaciones con el retorno de la inversión.	Se construye sobre una historia relativamente larga del uso, que incluye casos de estudio y datos que demuestran el retorno de la inversión.

Tabla 3: Representación continua y por Etapas

¿Qué representación utilizar?

Usadas tanto para la mejora de procesos como para las evaluaciones, ambas representaciones están diseñadas para ofrecer esencialmente resultados equivalentes. Casi todo el contenido del modelo CMMI es común a ambas representaciones. Por lo tanto, una organización no necesita seleccionar una representación sobre otra. De hecho, una organización puede encontrar utilidad en ambas representaciones.

Es raro que una organización implemente una u otra representación exactamente según lo prescrito. Las organizaciones que tienen éxito en la mejora de procesos generalmente definen un plan de mejora que se centra en las necesidades propias de esa organización y por lo tanto utilizan los principios de ambas representaciones: por etapas y continua.

Por ejemplo, las organizaciones que seleccionan la representación por etapas y están en nivel de madurez 1 a menudo implementan las áreas de proceso del nivel de madurez 2, pero también el área de proceso “Enfoque en Procesos de la Organización”, que se encuentra en el nivel de madurez 3. Otro ejemplo es el de una organización que elige la representación continua para guiar su esfuerzo de mejora de procesos internos y luego elige la representación por etapas para realizar una evaluación. A continuación se describen los niveles para cada representación.

3.3 Modelo por Niveles

Comprendiendo los niveles

Los niveles se utilizan en CMMI para describir un camino evolutivo recomendado para una organización que quiera mejorar los procesos que utiliza para desarrollar y mantener sus productos y servicios.

Los niveles pueden también ser el resultado de la actividad de calificación de las evaluaciones. Las evaluaciones se pueden realizar para organizaciones, incluyendo

compañías (generalmente pequeñas) completas o grupos más pequeños, tal como un grupo de proyectos o una sección dentro de una compañía.

Los dos caminos de mejora están asociados con los dos tipos de niveles que corresponden a las dos representaciones presentadas anteriormente. Para la representación continua, se utiliza el término “nivel de capacidad”, para la representación por etapas, se utiliza el término “nivel de madurez”.

Independientemente de qué representación se seleccione, el concepto de niveles es el mismo. Los niveles caracterizan a la mejora desde un estado mal definido hasta un estado que utiliza información cuantitativa para determinar y gestionar las mejoras que se necesitan para satisfacer los objetivos de negocio de una organización. Para alcanzar un nivel particular, una organización debe satisfacer todas las metas apropiadas del área o conjunto de áreas de proceso que son objeto de la mejora, independientemente de si es un nivel de capacidad o de madurez.

Ambas representaciones también proporcionan caminos para implementar la mejora de procesos que permiten lograr los objetivos de negocio. Ambas representaciones proporcionan el mismo contenido esencial y utilizan los mismos componentes del modelo.

Estructuras de las representaciones continua y por etapas

Ambas representaciones (continua y por etapas) tienen muchos componentes iguales (p.ej. áreas de proceso, metas específicas y prácticas específicas) y estos componentes tienen la misma jerarquía y configuración.

Lo que no resulta tan aparente desde la visión de alto nivel de la Ilustración 1 es que la representación continua se enfoca en la capacidad del área de proceso cuando se mide por niveles de capacidad, y la representación por etapas se enfoca en la madurez de la organización cuando se mide por niveles de madurez. Estas dimensiones (las dimensiones de capacidad/madurez) de CMMI se utilizan para actividades de

benchmarking y evaluación, así como para guiar los esfuerzos de mejora de una organización.

Los niveles de capacidad, que pertenecen a la representación continua, se aplican al logro de mejora de procesos de una organización en áreas de proceso individuales. Estos niveles son un medio para mejorar de forma incremental los procesos que corresponden a un área de proceso dada. Existen seis niveles de capacidad, numerados de 0 a 5.

Los niveles de madurez, que pertenecen a la representación por etapas, se aplican al logro de mejora de procesos de una organización en múltiples áreas de proceso. Estos niveles son un medio de predecir los resultados generales del siguiente proyecto que se acometa. Existen cinco niveles de madurez, numerados de 1 a 5.

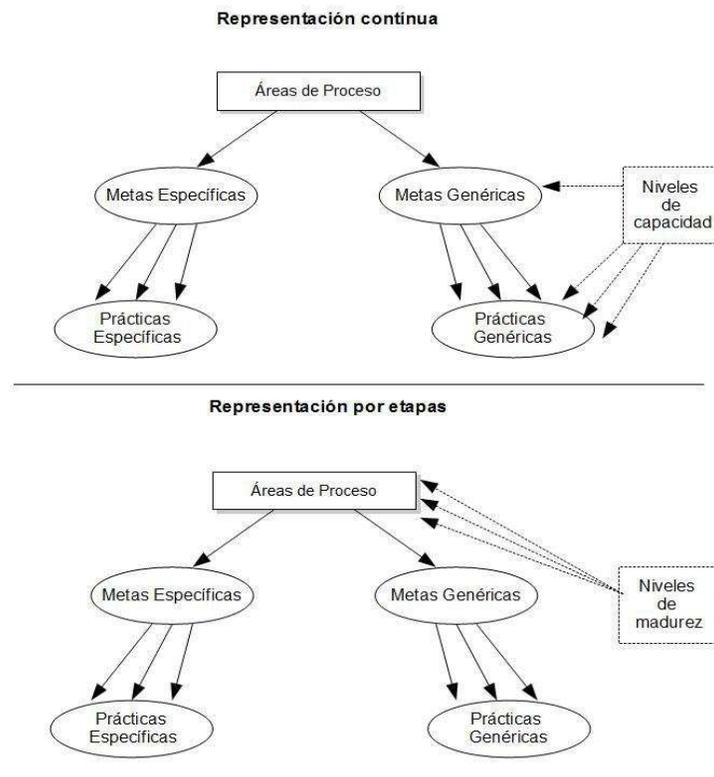


Ilustración 1: Representación continua y por Etapas

La Tabla 4 compara los seis niveles de capacidad con los cinco niveles de madurez. Los nombres de cuatro de los niveles son los mismos en ambas representaciones. Las diferencias son que no existe nivel de madurez 0 para la representación por etapas, y en el nivel 1, el nivel de capacidad es Realizado, mientras que el nivel de madurez es Inicial. Por lo tanto, el punto de partida es diferente para las dos representaciones.

La representación continua se interesa en seleccionar tanto un área de proceso particular a mejorar como en el nivel de capacidad deseado para ese área de proceso. En este contexto, es importante conocer si un proceso se ha realizado o está incompleto. Por lo tanto, al punto de partida de la representación continua se le da el nombre “incompleto”.

Debido a que la representación por etapas se interesa por la madurez global de la organización, no es interés primario si los procesos individuales son realizados o están incompletos. Por lo tanto, al punto de partida de la representación por etapas se le da el nombre “inicial”.

Tanto los niveles de capacidad como los niveles de madurez proporcionan una forma de medir cómo las organizaciones pueden y hacen mejorar sus procesos. Sin embargo, el enfoque asociado a la mejora de procesos es diferente.

Nivel	Niveles de capacidad	Niveles de madurez
Nivel 0	Incompleto	N/A
Nivel 1	Realizado	Inicial
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4	Gestionado cuantitativamente	Gestionado cuantitativamente
Nivel 5	En optimización	En optimización

Tabla 4: Correspondencia entre niveles de capacidad y de madurez

Comprendiendo los niveles de capacidad

Para dar soporte a aquellos que utilizan la representación continua, todos los modelos de CMMI reflejan niveles de capacidad en su diseño y contenido. Un nivel de capacidad

consiste en una meta genérica y sus prácticas genéricas relacionadas en la medida que se relacionan con un área de proceso, las cuales pueden mejorar los procesos de la organización asociados con ese área de proceso. En la medida que se satisface la meta genérica y sus prácticas genéricas en cada nivel de capacidad, se obtienen los beneficios de mejora de procesos para ese área de proceso. Los seis niveles de capacidad, especificados por los números 0 hasta 5, son los siguientes:

0. Incompleto.
1. Realizado.
2. Gestionado.
3. Definido.
4. Gestionado cuantitativamente.
5. En optimización.

El hecho de que los niveles de capacidad 2 a 5 utilicen los mismos términos que las metas genéricas 2 a 5 es intencionado, porque cada una de estas metas genéricas y prácticas genéricas reflejan el significado de los niveles de capacidad en términos de metas y prácticas que se pueden implementar. A continuación, se presenta una breve descripción de cada uno de los niveles de capacidad.

Nivel de capacidad 0: Incompleto

Un “proceso incompleto” es un proceso que, o bien no se ejecuta, o se ejecuta parcialmente. Al menos una de las metas específicas del área de proceso no se satisface y no existen metas genéricas para ese nivel, ya que no hay ninguna razón para institucionalizar un proceso ejecutado parcialmente.

Nivel de capacidad 1: Realizado

Un proceso de nivel de capacidad 1 se caracteriza como un “proceso realizado”. Un proceso realizado es un proceso que satisface las metas específicas del área de proceso. Soporta y permite el trabajo necesario para producir los productos del trabajo. Aunque el nivel de capacidad 1 da como resultado mejoras importantes, esas mejoras pueden perderse en el tiempo si no se institucionalizan. La aplicación de la institucionalización (las prácticas genéricas de CMMI en los niveles de capacidad 2 a 5) ayuda a asegurar que las mejoras se mantienen.

Nivel de capacidad 2: Gestionado

Un proceso de nivel de capacidad 2 se caracteriza como un “proceso gestionado”. Un proceso gestionado es un proceso realizado (nivel de capacidad 1) que tiene la infraestructura básica dispuesta para soportar el proceso. Se planifica y ejecuta de acuerdo a políticas; emplea personal con habilidades; tiene los recursos adecuados para producir resultados controlados; involucra a las partes interesadas relevantes; se monitoriza, controla y revisa; y se evalúa la adherencia a su descripción de proceso. La disciplina de proceso reflejada por el nivel de capacidad 2 ayuda a asegurar que las prácticas existentes se mantienen durante tiempos de estrés.

Nivel de capacidad 3: Definido

Un proceso de nivel de capacidad 3 se caracteriza como un “proceso definido”. Un proceso definido es un proceso gestionado (nivel de capacidad 2) que se adapta a partir del conjunto de procesos estándar de la organización, de acuerdo a las guías de adaptación de la organización, y contribuye a los activos de proceso de la organización con productos del trabajo, medidas e información adicional de mejora de procesos.

Una distinción crítica entre los niveles de capacidad 2 y 3 es el alcance de los estándares, descripciones de proceso y procedimientos. En el nivel de capacidad 2, los estándares, descripciones de proceso y procedimientos pueden ser bastante diferentes en cada instancia específica del proceso (p.ej., en cada proyecto particular). En el nivel de capacidad 3, los estándares, descripciones de proceso y procedimientos para un proyecto se adaptan a partir del conjunto de procesos estándar de la organización, para ajustarse a un proyecto o unidad organizativa particular, y son, por tanto, más consistentes, excepto para las diferencias permitidas por las guías de adaptación. Otra distinción crítica es que en el nivel de capacidad 3, los procesos se describen normalmente de forma más rigurosa que en el nivel de capacidad 2. Un proceso definido establece claramente el propósito, entradas, criterios de entrada, actividades, roles, medidas, etapas de verificación, salidas y criterios de salida. En el nivel de capacidad 3, los procesos se gestionan de forma más proactiva utilizando una comprensión de las interrelaciones de las actividades del proceso y de las medidas detalladas del proceso, de sus productos del trabajo y de sus servicios.

Nivel de capacidad 4: Gestionado cuantitativamente

Un proceso de nivel de capacidad 4 se caracteriza como un “proceso gestionado cuantitativamente”. Un proceso gestionado cuantitativamente es un proceso definido (nivel de capacidad 3) que se controla utilizando técnicas estadísticas y otras técnicas cuantitativas. Se establecen los objetivos cuantitativos de calidad y de ejecución del proceso, y se utilizan como criterios para gestionar el proceso. Se comprende la calidad y el rendimiento del proceso en términos estadísticos y se gestionan a lo largo de la vida del proceso.

Nivel de capacidad 5: En optimización

Un proceso de nivel de capacidad 5 se caracteriza como un “proceso en optimización”. Un proceso en optimización es un proceso gestionado cuantitativamente (nivel de capacidad 4) que se mejora en base a una comprensión de las causas comunes de variación inherentes al proceso. El enfoque de un proceso en optimización es mejorar continuamente el rango de la ejecución del proceso mediante mejoras, tanto incrementales como innovadoras.

Los niveles de capacidad de un área de proceso se logran mediante la aplicación de las prácticas genéricas o alternativas adecuadas a los procesos asociados con ese área de proceso.

Alcanzar el nivel de capacidad 1 para un área de proceso es equivalente a decir que los procesos asociados con esa área de proceso son “procesos realizados”.

Alcanzar el nivel de capacidad 2 para un área de proceso es equivalente a decir que existe una política que indica que se realizará el proceso. Existe un plan para realizarlo, se proporcionan los recursos, se asignan las responsabilidades, se proporciona formación para llevarlo a cabo, se controlan los productos del trabajo seleccionados relativos a la ejecución del proceso, y así sucesivamente. En otras palabras, un proceso de nivel de capacidad 2 puede planificarse y monitorizarse de la misma forma que cualquier proyecto o actividad de soporte.

Alcanzar el nivel de capacidad 3 para un área de proceso supone que existe un proceso estándar de la organización asociado con ese área de proceso, el cual puede adaptarse a las necesidades del proyecto. Los procesos en la organización se definen y aplican

ahora de forma más consistente porque se basan en los procesos estándar de la organización.

Alcanzar el nivel de capacidad 4 para un área de proceso supone que este área de proceso es un factor clave del negocio, que la organización quiere gestionar utilizando técnicas cuantitativas y estadísticas. Este análisis proporciona a la organización más visibilidad en la ejecución de los subprocesos seleccionados, que la harán más competitiva en el mercado.

Alcanzar el nivel de capacidad 5 para un área de proceso supone que se han estabilizado los subprocesos seleccionados y que se quieren reducir las causas comunes de variación dentro de ese proceso.

Comprendiendo los niveles de madurez

Para dar soporte a aquellos que utilizan la representación por etapas, todos los modelos CMMI reflejan niveles de madurez en su diseño y contexto. Un nivel de madurez consta de prácticas relacionadas específicas y genéricas para un conjunto predefinido de áreas de proceso que mejoran el rendimiento global de la organización. El nivel de madurez de una organización proporciona un camino para predecir el rendimiento en una disciplina dada o en un conjunto de disciplinas.

La experiencia ha mostrado que las organizaciones toman la mejor decisión cuando centran sus esfuerzos de mejora de procesos en un número controlable de áreas de proceso a la vez y que dichas áreas requieren aumentar su complejidad cuando la organización mejora.

Un nivel de madurez es una meseta evolutiva definida para la mejora de procesos de la organización. Cada nivel de madurez madura un subconjunto importante de procesos de la organización, preparándola para pasar al siguiente nivel de madurez. Los niveles de madurez se miden mediante el logro de metas específicas y genéricas asociadas a cada conjunto predefinido de áreas de proceso.

Existen cinco niveles de madurez, siendo cada uno de ellos una capa en la cimentación de la mejora de procesos en curso, denominados por los números 1 a 5.

1. Inicial.
2. Gestionado.
3. Definido.
4. Gestionado cuantitativamente.
5. En optimización.

Nivel de madurez 1: Inicial

En el nivel de madurez 1, los procesos son generalmente ad-hoc y caóticos. La organización generalmente no proporciona un entorno estable para dar soporte a los procesos. El éxito en estas organizaciones depende de la competencia y heroicidad del personal de la organización y no del uso de procesos probados. A pesar de este caos, las organizaciones de nivel de madurez 1 a menudo producen productos y servicios que funcionan; sin embargo, frecuentemente exceden sus presupuestos y no cumplen sus calendarios. Las organizaciones de nivel de madurez 1 se caracterizan por una tendencia a comprometerse en exceso, a abandonar los procesos en tiempos de crisis y a una incapacidad para repetir sus éxitos.

Nivel de madurez 2: Gestionado

En el nivel de madurez 2, los proyectos de la organización han asegurado que los procesos se planifican y realizan de acuerdo a políticas; los proyectos emplean personal con habilidad que dispone de recursos adecuados para producir resultados controlados; involucran a las partes interesadas relevantes; se monitorizan, controlan y revisan; y se evalúan en cuanto a su adherencia a sus descripciones de proceso. La disciplina de proceso reflejada por el nivel de madurez 2 ayuda a asegurar que las prácticas existentes se mantienen durante tiempos de estrés. Cuando estas prácticas están en su lugar, los proyectos se realizan y gestionan de acuerdo a sus planes documentados. En el nivel de madurez 2, el estado de los productos de trabajo y la entrega de los servicios son visibles a la dirección en puntos definidos (p.ej., en los hitos principales y al finalizar las tareas principales). Se establecen compromisos entre las partes interesadas relevantes y se revisan, según sea necesario. Los productos de trabajo se controlan de

forma apropiada. Los productos de trabajo y servicios satisfacen sus descripciones de proceso especificadas, estándares y procedimientos.

Nivel de madurez 3: Definido

En el nivel de madurez 3, los procesos son bien caracterizados y comprendidos, y se describen en estándares, procedimientos, herramientas y métodos. El conjunto de procesos estándar de la organización, que es la base del nivel de madurez 3, se establece y mejora a lo largo del tiempo. Estos procesos estándar se usan para establecer la consistencia en toda la organización. Los proyectos establecen sus procesos definidos adaptando el conjunto de procesos estándar de la organización de acuerdo a las guías de adaptación (consultar el glosario para una definición de “conjunto de procesos estándar de la organización”). Una distinción crítica entre los niveles de madurez 2 y 3 es el alcance de los estándares, descripciones de proceso y procedimientos. En el nivel de madurez 2, los estándares, descripciones de proceso y procedimientos pueden ser bastante diferentes en cada instancia específica de un proceso (p.ej., en un proyecto particular). En el nivel de madurez 3, los estándares, descripciones de proceso y procedimientos para un proyecto se adaptan para adecuarse a un proyecto particular o unidad organizativa a partir del conjunto de procesos estándar de la organización y, por tanto, son más consistentes, exceptuando las diferencias permitidas por las guías de adaptación. Otra distinción crítica es que en el nivel de madurez 3, los procesos normalmente se describen más rigurosamente que en el nivel de madurez 2. Un proceso definido establece claramente el propósito, entradas, criterios de entrada, actividades, roles, medidas, etapas de verificación, salidas y criterios de salida. En el nivel de madurez 3, los procesos se gestionan más pro activamente utilizando una comprensión de las interrelaciones de las actividades del proceso y las medidas detalladas del proceso, sus productos de trabajo y sus servicios. En el nivel de madurez 3, la organización debe madurar más las áreas de proceso de nivel de madurez 2. Para lograr el nivel de madurez 3, se aplican las prácticas genéricas asociadas con la meta genérica 3 que no fueron tratadas en el nivel de madurez 2.

Nivel de madurez 4: Gestionado cuantitativamente

En el nivel de madurez 4, la organización y los proyectos establecen objetivos cuantitativos en cuanto al rendimiento de calidad y del proceso, y los utilizan como criterios en la gestión de los procesos. Los objetivos cuantitativos se basan en las necesidades del cliente, usuarios finales, organización e implementadores del proceso. El rendimiento de calidad y del proceso se comprende en términos estadísticos y se gestiona durante la vida de los procesos ^[SEI 2001]. Para los subprocesos seleccionados, se recogen y analizan estadísticamente medidas detalladas de rendimiento del proceso. Las medidas de rendimiento de calidad y del proceso se incorporan en el repositorio de medición de la organización para dar soporte a la toma de decisiones basada en hechos ^[McGarry 2000]. Se identifican las causas especiales de variación y, donde sea apropiado, se corrigen las fuentes de las causas especiales para prevenir sus futuras ocurrencias. Una distinción crítica entre los niveles de madurez 3 y 4 es la predictibilidad del rendimiento del proceso. En el nivel de madurez 4, el rendimiento de los procesos se controla utilizando técnicas estadísticas y otras técnicas cuantitativas, y es predecible cuantitativamente. En el nivel de madurez 3, los procesos normalmente sólo son predecibles cualitativamente.

Nivel de madurez 5: En optimización

En el nivel de madurez 5, una organización mejora continuamente sus procesos basándose en una comprensión cuantitativa de las causas comunes de variación inherentes a los procesos. El nivel de madurez 5 se centra en mejorar continuamente el rendimiento de procesos mediante mejoras incrementales e innovadoras de proceso y tecnológicas. Los objetivos cuantitativos de mejora de procesos para una organización se establecen, se revisan continuamente para reflejar el cambio a los objetivos del negocio, y se utilizan como criterios para gestionar la mejora de procesos. Los efectos de las mejoras de procesos desplegadas se miden y evalúan frente a los objetivos cuantitativos de mejora de procesos. Tanto los procesos definidos como el conjunto de procesos estándar de la organización son objeto de las actividades de mejora cuantitativa. Una distinción crítica entre los niveles de madurez 4 y 5 es el tipo de variación del proceso tratado. En el nivel de madurez 4, la organización se preocupa por

tratar las causas especiales de variación del proceso y por proporcionan predictibilidad estadística de los resultados. Aunque los procesos pueden producir resultados predecibles, los resultados pueden ser insuficientes para alcanzar los objetivos establecidos. En el nivel de madurez 5, la organización se interesa en tratar las causas comunes de variación del proceso y en cambiar el proceso (para cambiar la media de rendimiento del proceso o reducir la variación inherente del proceso experimentada) para mejorar el rendimiento del proceso y para alcanzar sus objetivos cuantitativos de mejora de procesos establecidos.

3.4 Componentes del modelo

Componentes requeridos, esperados e informativos

Los componentes del modelo se agrupan en tres categorías —requerido, esperado e informativo— que indican cómo interpretarlos.

Los *componentes requeridos* describen lo que una organización debe realizar para satisfacer un área de proceso. Este logro se debe implementar de forma visible en los procesos de una organización. Los componentes requeridos en CMMI son las metas específicas y los objetivos genéricos. La satisfacción de objetivos se utiliza en las evaluaciones como base para determinar si un área de proceso ha sido realizada y satisfecha.

Los *componentes esperados* describen lo que una organización puede implementar para lograr un componente requerido. Los componentes esperados guían a los que implementan mejoras o realizan evaluaciones. Los componentes esperados incluyen las prácticas específicas y las prácticas genéricas.

Antes de que los objetivos puedan considerarse satisfechos, las prácticas tal como se describen o prácticas aceptables alternativas a ellas, deberán estar presentes en los procesos planificados e implementados de la organización.

Los *componentes informativos* proporcionan detalles que ayudan a las organizaciones a comenzar a pensar en cómo aproximarse a los componentes requeridos y esperados. Las sub-prácticas, los productos de trabajo típicos, las ampliaciones, las elaboraciones de las prácticas genéricas, los títulos de metas y prácticas, las notas de metas y prácticas, y las referencias son ejemplos de componentes informativos del modelo.

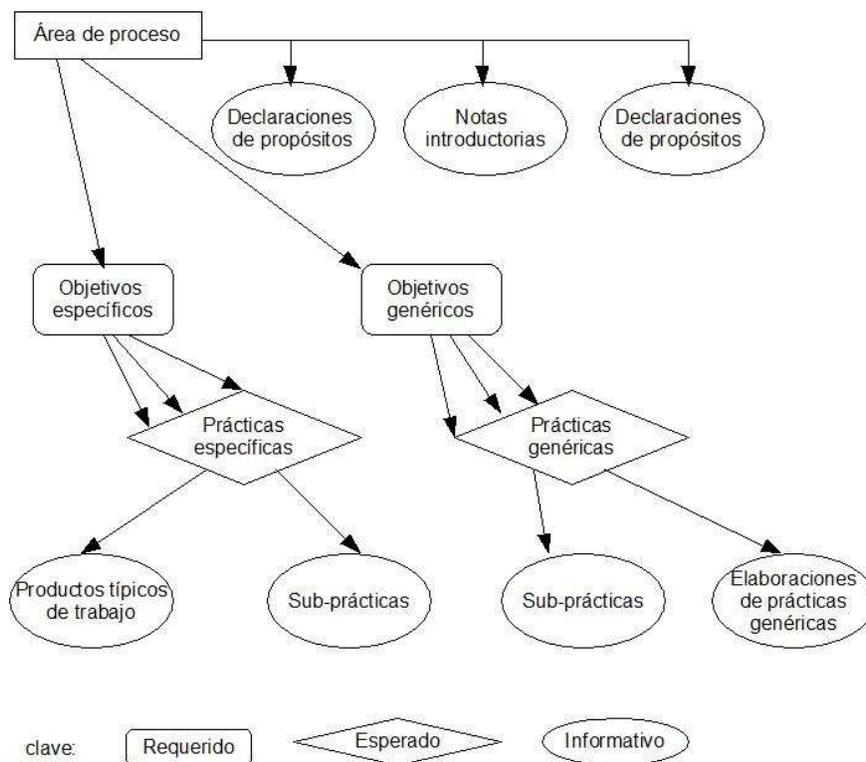


Ilustración 2: Componentes del modelo

Áreas clave de proceso (Key Process Area, KPA)

Este modelo establece un conjunto de prácticas o procesos clave agrupados en KPA's. Una KPA es un grupo de actividades relacionadas que cuando se llevan a cabo en conjunto alcanzan una serie de objetivos que se consideran importantes para aumentar la capacidad del proceso.

Para dar soporte a aquellos que utilizan la representación continua, las áreas de proceso se organizan en cuatro categorías:

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

Gestión de proyectos: las áreas de proceso de Gestión de proyectos cubren las actividades de gestión de proyectos relacionadas con la planificación, monitorización y control de proyectos.

Ingeniería: las áreas de proceso de Ingeniería cubren las actividades de desarrollo y de mantenimiento que se comparten entre las disciplinas de ingeniería.

Soporte: las áreas de proceso de Soporte cubren las actividades que dan soporte al desarrollo y al mantenimiento del producto. Las áreas de proceso de Soporte tratan los procesos que se usan en el contexto de la ejecución de otros procesos.

La tabla 3.2 proporciona una lista de todas las áreas de proceso, y sus categorías y niveles de madurez asociados.

Área de Proceso	Sigla en inglés	Categoría	Nivel de madurez
Análisis causal y resolución	CAR	Soporte	5
Gestión de configuración	CM	Soporte	2
Análisis de decisiones y resolución	DAR	Soporte	3
Gestión integrada del proyecto + IPPD	IPM + IPPD	Gestión de proyectos	3
Medición y análisis	MA	Soporte	2
Innovación y despliegue en la organización	OID	Gestión de procesos	5
Definición de procesos de la organización + IPPD	OPD + IPPD	Gestión de procesos	3
Enfoque en procesos de la organización	OPF	Gestión de procesos	3
Rendimiento del proceso de la organización	OPP	Gestión de procesos	4

Formación organizativa	OT	Gestión de procesos	3
Integración de producto	PI	Ingeniería	3
Monitorización y control del proyecto	PMC	Gestión de proyectos	2
Planificación de proyecto	PP	Gestión de proyectos	2
Aseguramiento de la calidad de proceso y de producto	PPQA	Soporte	2
Gestión cuantitativa de proyecto	QPM	Gestión de proyectos	4
Desarrollo de requerimientos	RD	Ingeniería	3
Gestión de requerimientos	REQM	Ingeniería	2
Gestión de riesgos	RSKM	Gestión de proyectos	3
Gestión de acuerdos con proveedores	SAM	Gestión de proyectos	2
Solución técnica	TS	Ingeniería	3
Validación	VAL	Ingeniería	3
Verificación	VER	Ingeniería	3

Tabla 5: Áreas Claves de Proceso

Declaraciones de propósitos

La “declaración de propósitos” describe la finalidad del área de proceso y es un componente informativo.

Notas introductorias

La sección de “notas introductorias” del área de proceso describe los conceptos principales cubiertos por el área de proceso y es un componente informativo.

Áreas de proceso relacionadas

La sección de “áreas de proceso relacionadas” lista las referencias a áreas de proceso que están en relación y refleja las relaciones de alto nivel entre las áreas de proceso. Es un componente informativo.

Metas específicas

Una meta específica describe las características únicas que deben estar presentes para satisfacer el área de proceso. Una meta específica es un componente requerido del modelo que se utiliza en las evaluaciones para ayudar a determinar si se satisface un área de proceso.

Metas genéricas

Las metas genéricas se denominan “genéricas” porque la misma declaración de la meta se aplica a múltiples áreas de proceso. Una meta genérica describe las características que deben estar presentes para institucionalizar los procesos que implementan un área de proceso.

Una meta genérica es un componente requerido del modelo y se utiliza en las evaluaciones para determinar si se satisface un área de proceso.

Las metas específicas y genéricas se numeran secuencialmente. Cada meta específica empieza con el prefijo SG (p. ej. SG 1). Cada meta genérica empieza con el prefijo GG (p. ej. GG 2).

Resúmenes de Metas específicas y prácticas específicas

El resumen de metas específicas y prácticas específicas proporciona un resumen de alto nivel de las metas específicas, que son componentes requeridos, y de las prácticas específicas, que son componentes esperados. El resumen de metas específicas y prácticas específicas es un componente informativo.

Prácticas específicas

Una práctica específica es la descripción de una actividad que se considera importante para alcanzar la meta específica asociada. Las prácticas específicas describen las actividades que se espera que produzcan la consecución de las metas específicas de un área de proceso. Una práctica específica es un componente esperado del modelo.

Productos de trabajo típicos

La sección de “productos de trabajo típicos” lista muestras de resultados de una práctica específica. Estos ejemplos se denominan productos de trabajo típicos porque a menudo hay otros productos de trabajo que son igual de eficaces pero no están en la lista. Un producto de trabajo típico es un componente informativo del modelo.

Subprácticas

Una subpráctica es una descripción detallada que proporciona una guía para interpretar e implantar una práctica específica o genérica. Las subprácticas pueden tomar un carácter prescriptivo, pero realmente son un componente informativo indicado sólo para proporcionar ideas que puedan ser útiles para la mejora de proceso.

Prácticas genéricas

Las prácticas genéricas se denominan “genéricas” porque la misma práctica se aplica a múltiples áreas de proceso. Una práctica genérica es la descripción de una actividad que se considera importante para el logro de la meta genérica asociada. Una práctica genérica es un componente esperado del modelo.

Cada práctica específica empieza con el prefijo SP, seguido por un número en la forma x.y (p. ej. SP 1.1). La x es el mismo número de la meta a la que pertenece la práctica específica. La y es el número de secuencia de la práctica específica para la meta específica correspondiente. Un ejemplo de numeración de práctica específica está en el área de proceso de Planificación de proyecto. La numeración de la primera práctica específica es SP 1.1 y la segunda SP 1.2. Cada práctica genérica empieza con el prefijo GP, seguido por un número en la forma x.y (p.ej. GP 1.1). La x corresponde al número de la meta genérica. La y es el número de secuencia de la práctica genérica para la meta genérica correspondiente. Por ejemplo, la primera práctica genérica asociada con GG 2 se numera GP 2.1 y la segunda GP 2.2.

Elaboraciones de las prácticas genéricas

Una elaboración de práctica genérica aparece después de una práctica genérica en un área de proceso, para proporcionar una guía sobre cómo la práctica genérica debería

aplicarse de forma exclusiva al área de proceso. Una elaboración de práctica genérica es un componente informativo del modelo.

Una extensión puede ser un material informativo, una práctica específica, una meta específica o un área de proceso que amplía el alcance de un modelo o hace hincapié en un aspecto particular de su utilización.

Componentes informativos de soporte

En muchas ocasiones, se necesita más información para describir un concepto. Este material informativo se presenta como uno de los siguientes componentes:

- *Nota*: es un texto que puede acompañar casi a cualquier otro componente del modelo. Puede proporcionar detalles, información previa o de base. Una nota es un componente informativo del modelo.
- *Ejemplo*: es un componente que comprende texto y, a menudo, una lista de elementos, por lo general en una caja, que puede acompañar a casi cualquier otro componente y proporciona uno o más ejemplos para clarificar un concepto o una actividad descrita. Un ejemplo es un componente informativo del modelo.
- *Ampliación*: es una nota o un ejemplo que es relevante para una disciplina particular. Las disciplinas cubiertas en este modelo son ingeniería del hardware, ingeniería de sistemas e ingeniería del software. Cada ampliación se etiqueta con una cabecera que indica la disciplina a la que se aplica. Una ampliación es un componente informativo del modelo.
- *Referencia*: es un enlace a información adicional o más detallada en las áreas de proceso relacionadas y puede acompañar a casi cualquier otro componente del modelo. Una referencia es un componente informativo del modelo.

Capítulo 4: Metodologías de Desarrollo y Ciclos de Vida

Todo producto que se desarrolla sigue un ciclo de vida, que va desde su concepción hasta su entrega final. Muchas veces no hay una definición formal, escrita y detallada del ciclo de vida empleado o de las etapas que se ejecutan, pero implícitamente se sigue un camino.

Si el producto no es complejo y requiere un mínimo de esfuerzo para desarrollarlo, esta falta de definición no afecta al resultado final. Se piensa un resultado, se describe algún camino para su construcción, se crea y se entrega. La cantidad de personas involucradas puede variar, y una vez finalizado se sigue con otro producto.

Pero, a medida que la complejidad aumenta, la cantidad de personas involucrada también, es imprescindible que se defina un ciclo de vida para que todos los productos tengan mismo resultado final, mismas actividades y fases de desarrollo.

Este capítulo trata sobre los ciclos de vida aplicados a los productos software y las metodologías que se utilizan para cumplir con las fases definidas en los ciclos de vida para alcanzar los resultados planteados.

Particularmente se analizará la metodología más conocida y adoptada por aquellas organizaciones cuyos proyectos software son de gran envergadura, el Proceso Unificado de Rational (RUP). Afortunadamente existe mucha información sobre el tema, y al ser una metodología con muchos años de antigüedad es posible hallar información sobre los pro y los contras de la metodología, su adaptación con otras metodologías, los resultados obtenidos por aquellas organizaciones que desean certificarse en CMMI cuando éstas aplican RUP en su proceso de desarrollo, etc.

Y también trata sobre MDD, que es un proceso relativamente nuevo dentro de las metodologías de desarrollo y desafortunadamente no existe tanta información como RUP.

Se describirán los aspectos más relevantes de RUP y de MDD para que pueda realizarse en el capítulo siguiente un mapeo con CMMI.

Al finalizar con la lectura de este capítulo se podrán responder las siguientes preguntas:

- ¿Cómo se define el ciclo de vida para un producto software?
- ¿Qué es una metodología de desarrollo?
- ¿Qué metodología se debe elegir en una empresa desarrolladora de software?
- ¿Qué es RUP?
- ¿Qué es MDD?

4.1- Metodologías de desarrollo

¿Cómo se organiza el desarrollo del software?

El ciclo de vida de un producto software se conoce como "modelo de ciclo de vida" dado que hay varios modelos para elegir y aplicar a un proyecto.

Un modelo de ciclo de vida representa todas las actividades y productos de trabajo necesarios para desarrollar un sistema. A menudo, se dice que los distintos componentes de software deben pasar por distintas fases o etapas durante su ciclo de vida.

Algunas de las características de tales modelos son:

- Describir las fases principales de desarrollo de software.
- Ayudar a administrar el progreso del desarrollo y
- Proveer un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

Cada una de las fases definidas puede ser abordada y resuelta de múltiples maneras, con distintas herramientas y utilizando distintas técnicas. Es necesario saber cuándo podemos dar por concluida una fase, quién debe realizarla, qué tareas preceden o anteceden a una dada, qué documentación se utilizará para llevar a cabo esa fase. Podrían pensarse todas estas tareas como detalles organizativos, un estilo de hacer las cosas. Pero yendo un poco más allá de un simple estilo, formalizándolo añadiendo algo de rigurosidad y norma obtenemos una *metodología*.

El estándar IEEE para los procesos del ciclo de vida del software describe el conjunto de actividades y procesos obligatorios para el desarrollo y mantenimiento del software [IEEE Std. 1074-1995]. Su objetivo es establecer un marco común para el desarrollo de modelos de ciclo de vida y proporciona ejemplos de situaciones típicas.

La selección del modelo de ciclo de vida también depende del modelo de sistema. La reingeniería de un sistema de software con casos de uso y modelos de objetos existentes requiere un conjunto de actividades diferentes que la construcción de un sistema a partir de cero.

El estándar no dicta un modelo de ciclo de vida específico, sino que proporciona una plantilla que puede personalizarse de muchas maneras diferentes.

¿Qué es una metodología de desarrollo de software?

Una metodología de desarrollo de software se refiere a un *framework* que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por sus fortalezas y debilidades.

El *framework* para metodología de desarrollo de software consiste en:

- Una filosofía de desarrollo de software con el enfoque del proceso de desarrollo de software.
- Herramientas, modelos y métodos para asistir al proceso de desarrollo de software.

Estos *frameworks* son a menudo vinculados a algún tipo de organización, que además desarrolla, apoya el uso y promueve la metodología. La metodología es a menudo documentada en algún tipo de documentación formal. ^[WIKIPEDIA 1]

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Un desarrollo de software por tanto, debe apoyarse en la metodología utilizada.

¿Qué metodología se debe utilizar?

El seleccionar una metodología en una organización, es una tarea fundamental y de respuesta no trivial, considerando además que pueden desarrollarse un número considerado de proyectos, cada uno con sus propias características. Tomando un proyecto cualquiera, se analizan ciertas características del proyecto (tamaño, complejidad, recursos asignados, tiempo, etc.) y se determina para el mismo qué metodología se utilizará.

¿Qué pasa cuando para un proyecto no sirve aplicar ninguna metodología de las ya conocidas?

En estos casos pueden suceder dos cosas: que no se aplique metodología alguna, o bien, se establece una nueva metodología consistente en una adaptación de una en particular, o de varias de ellas. Esto es totalmente válido siempre y cuando no se violen los objetivos y pautas de las metodologías bases utilizadas.

Es recomendado, y hasta se podría decir fundamental, que en los proyectos que sean de gran envergadura exista una metodología elegida o que la organización haya realizado una adaptación de metodologías que aplique al proyecto para que el trabajo no se torne desorganizado, caótico y conduzca a resultados no deseados.

Las metodologías más utilizadas varían entre ágiles tales como SCRUM, XP, hasta las más convencionales como RUP.

Dado que la madurez organizativa mejorada se asocia con la mejora en el rango de resultados esperados que pueden lograrse en una organización, es un modo de predecir los resultados generales del siguiente proyecto de la organización.

4.2 Proceso Unificado de Rational

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 proporcionan una guía específica en áreas tales como la de Modelado de Negocios, Arquitecturas Web, Pruebas y Calidad.

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.

El proceso RUP 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, aquellos fundamentales para un proyecto y los proyectos integrados) permiten obtener unos resultados más acordes con las previsiones gracias a la aplicación del proceso RUP.

Aspectos claves de RUP

RUP es un marco de desarrollo de software iterativo e incremental y junto con el UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Características:

Iterativo e Incremental

El Proceso Unificado es un marco de desarrollo iterativo e incremental compuesto de cuatro fases denominadas: Inicio, Elaboración, Construcción y Transición. Cada una de estas fases es a su vez dividida en una serie de iteraciones (la de inicio sólo consta de varias iteraciones en proyectos grandes). Estas iteraciones ofrecen como resultado un incremento del producto desarrollado que añade o mejora las funcionalidades del sistema en desarrollo.

Cada una de estas iteraciones se divide a su vez en una serie de disciplinas que recuerdan a las definidas en el ciclo de vida clásico o en cascada: Análisis de Requisitos, Diseño, Implementación y Prueba. Aunque todas las iteraciones suelen incluir trabajo en casi todas las disciplinas, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto.

Dirigido por los casos de uso

En el Proceso Unificado los Casos de Uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de Casos de Uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.

Centrado en la arquitectura

El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema. La analogía con la construcción es clara, cuando se construye un edificio existen diversos planos que incluyen los distintos servicios del mismo: electricidad, fontanería, etc.

Enfocado en los riesgos

El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.

RUP divide el proceso en 4 fases, dentro de las cuales se realizan varias iteraciones en número

variable según el proyecto y en las que se hace un mayor o menor hincapié en los distintas

actividades.

En las iteraciones de cada fase se hacen diferentes esfuerzos en diferentes actividades:

- **Inicio:** se hace un plan de fases, se identifican los principales Casos de Uso y se identifican los riesgos. Se define el alcance del proyecto.
- **Elaboración:** se hace un plan de proyecto, se completan los Casos de uso y se eliminan los riesgos.
- **Construcción:** se concentra en la elaboración de un producto totalmente operativo y eficiente. Además, se concentra en la elaboración del manual de usuario.
- **Transición:** se instala el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

La siguiente imagen, muestra el flujo de trabajo implementado en RUP:

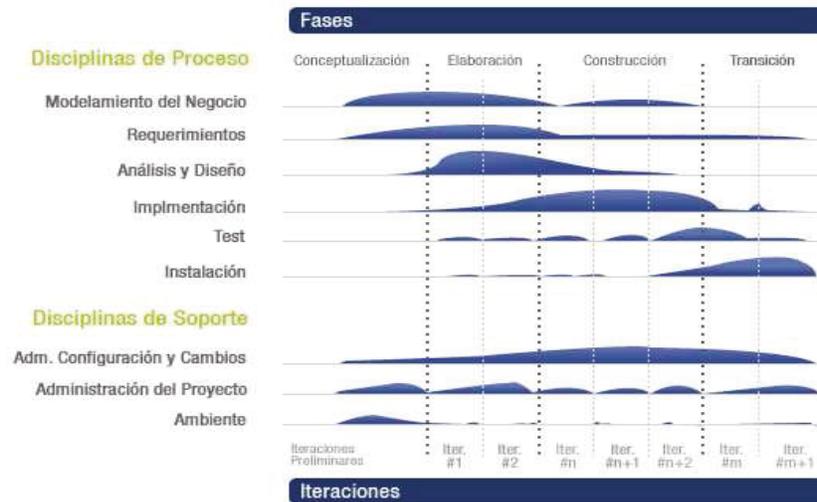


Ilustración 3: RUP

Actividades realizadas en cada fase

FASE DE INICIO

Durante esta fase las iteraciones ponen mayor énfasis en actividades de modelado del negocio y de requisitos.

Modelado del Negocio: el equipo se familiariza con el funcionamiento de la empresa.

- Entender la estructura y la dinámica de la organización para la cual se desarrollará el sistema.
- Entender el problema actual en la organización e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización.

Requerimientos: en esta línea los requerimientos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requerimientos que se especifiquen.

- Establecer y mantener un acuerdo entre clientes y otros *stakeholders* sobre lo que el sistema podría hacer.

- Proveer a los desarrolladores un mejor entendimiento de los requerimientos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

FASE DE ELABORACIÓN

En esta fase, las iteraciones se orientan al desarrollo de la línea base de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelado de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la línea base de arquitectura.

Análisis y Diseño: en esta actividad se especifican los requerimientos y se describen sobre cómo se van a implementar en el sistema.

- Transformar los requerimientos al diseño del sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación.

FASE DE CONSTRUCCIÓN

Implementación: se implementan las clases y objetos en ejecutables, binarios y demás. El resultado final es un sistema ejecutable.

- Planificar qué subsistemas deben ser implementados y en qué orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en qué orden implementa los elementos del subsistema.
- Si encuentra errores de diseño, los notifica.
- Se integra el sistema siguiendo el plan.

Test: este flujo de trabajo es el encargado de evaluar la calidad del producto que se está desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requerimientos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requerimientos tengan su apropiada implementación.

Despliegue: esta actividad tiene como objetivo producir con éxito distribuciones del producto hacia los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.
- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios y al cuerpo de ventas.
- Migrar el software existente o convertir bases de datos.

DURANTE TODO EL PROYECTO

Administración del proyecto: se vigila el cumplimiento de los objetivos, gestión de riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

- Proveer un marco de trabajo para la gestión de proyectos de software intensivos.
- Proveer guías prácticas, realizar planeación, contratar personal, ejecutar y monitorear el proyecto.
- Proveer un marco de trabajo para gestionar riesgos.

Adm. Configuración y Cambios: el control de cambios permite mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

Ambiente: la finalidad de esta actividad es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir. En concreto, las responsabilidades de este flujo de trabajo incluyen:

- Selección y adquisición de herramientas.
- Establecer y configurar las herramientas para que se ajusten a la organización.
- Configuración del proceso.
- Mejora del proceso.
- Servicios técnicos.

Comprendidos los conceptos básicos de RUP, se desarrolla a continuación una descripción del proceso MDD con el fin de tener claros los conceptos de estas dos formas de abordar el desarrollo de un producto software antes de poder realizar el mapeo con el modelo de madurez CMMI.

4.3 La organización en el desarrollo del software mediante MDD

Desarrollo de Software Dirigido por Modelos (MDD)

El Desarrollo de Software Dirigido por Modelos, MDD (por sus siglas en inglés: *Model Driven software Development*), se ha convertido en un nuevo paradigma de desarrollo software. MDD promete mejorar el proceso de construcción de software basándose en un proceso dirigido por modelos y soportado por potentes herramientas. El adjetivo “dirigido” (*driven*) en MDD, a diferencia de “basado” (*based*), enfatiza que este paradigma asigna a los modelos un rol central y activo: son al menos tan importantes como el código fuente. Los modelos se van generando desde los más abstractos a los más concretos a través de pasos de transformación y/o refinamientos, hasta llegar al código aplicando una última transformación. La transformación entre modelos constituye el motor del MDD.

Los principios para la generación de código se resumen en:

- Ingeniería hacia delante (*forward engineering*).
- El valor está en el modelo. El código es descartable: refactorizable, generable siempre que sea necesario.
- La tecnología cambia más rápido que los modelos.

Uno de los principios es el valor dado a los modelos, ya que el modelo es la pieza fundamental en MDD, por esto es importante dar una definición de este término. Si se busca en Internet hay muchas definiciones, muchos puntos de vista, y a veces es difícil quedarse con sólo una definición. Pero, para los fines de este trabajo alcanza con la siguiente simple definición:

Un modelo es una simplificación de la realidad

Esto significa, que de una dada realidad observada, se toman aquellos aspectos relevantes para el problema que se intenta solucionar y se omite el resto. ¿Por qué se

omite parte de la realidad observada? porque no aportaría información útil, relleno el análisis con información que nunca se utilizará. Si este análisis se plasma en un documento, éstos quedarían ilegibles. Por eso es que los modelos deben contener simplificaciones del *todo* que se observa.

Se suele llamar a tales modelos *planos de un sistema*. Esto es, porque al igual que los planos de un rascacielos sirven para realizar su construcción de principio a fin, integrando diferentes personas con diferentes necesidades de visualización del edificio (albañiles, electricistas, diseñadores, etc.) los planos del sistema sirven para el desarrollo del mismo integrando analistas, diseñadores, desarrolladores y demás personas involucradas. Se los utiliza en toda las etapas del ciclo de vida del software guiando su construcción.

¿Por qué modelamos? Hay una razón fundamental:

Construimos modelos para comprender mejor el sistema que estamos desarrollando.

A través del modelado, conseguimos cuatro objetivos:

1. Los modelos nos ayudan a visualizar cómo es o queremos que sea un sistema.
2. Los modelos nos permiten especificar la estructura o el comportamiento de un sistema.
3. Los modelos nos proporcionan plantillas que nos guían en la construcción de un sistema.
4. Los modelos documentan las decisiones que hemos adoptado.

Conocidas las razones por las cuales realizar modelado, lo que sigue es conocer en qué consiste la construcción de un software dirigido por modelos.

¿Qué propone MDD?

Los puntos claves de la iniciativa MDD fueron identificados en “*An MDA Manifesto*” (Booch 2004) de la siguiente forma:

1. El uso de un mayor nivel de abstracción en la especificación tanto del problema a resolver como de la solución correspondiente, en relación con los métodos tradicionales de desarrollo de software.
2. El aumento de confianza en la automatización asistida por computadora para soportar el análisis, el diseño y la ejecución.
3. El uso de estándares industriales como medio para facilitar las comunicaciones, la interacción entre diferentes aplicaciones y productos, y la especialización tecnológica.

Comparando el proceso MDD con el proceso tradicional se observa que muchas cosas permanecen iguales. Los requerimientos del sistema aun necesitan ser capturados y el sistema aun necesita ser probado e instalado. Lo que sí cambia de manera substancial son las actividades de análisis, diseño de bajo nivel y codificación.

El MDD tiene un efecto profundo sobre la forma en que construimos las aplicaciones de software. Las organizaciones y los proyectos frecuentemente dependen de expertos clave, quienes toman las decisiones respecto al sistema. MDD permite capturar su experiencia en los modelos y en las transformaciones permitiendo de esta forma que otros miembros del equipo pueden aprovecharla sin requerir de su presencia. Además, este conocimiento se mantiene aun cuando los expertos se alejen de la organización. Por otra parte, el costo de desarrollo y de pruebas se reduce significativamente al automatizar gran parte del trabajo de generación del código y otros artefactos a partir de los modelos. A través de la automatización MDD favorece la generación consistente de los artefactos reduciéndose el número de errores.^[7]

El ciclo de vida dirigido por modelos

El ciclo de vida de desarrollo de software usando MDD se muestra en la Ilustración 4. Una de las mayores diferencias frente a los ciclos de vida tradicionales está en el tipo de los artefactos que se crean durante el proceso de desarrollo. Los artefactos son modelos formales, es decir, modelos que pueden ser comprendidos por una computadora.

MDD identifica distintos tipos de modelos:

- Modelos con alto nivel de abstracción independientes de cualquier metodología computacional, llamados CIMs (*Computational Independent Model*).
- Modelos independientes de cualquier tecnología de implementación llamados PIMs (*Platform Independent Model*).
- Modelos que especifican el sistema en términos de construcciones de implementación disponibles en alguna tecnología específica, conocidos como PSMs (*Platform Specific Model*).
- Modelos que representan el código fuente en sí mismo, identificados como IMs (*Implementation Model*).



Ilustración 4: Ciclo de Vida Dirigido por Modelos

En el proceso de desarrollo de software tradicional, las transformaciones de modelo a modelo, o de modelo a código son hechas mayormente con intervención humana. Muchas herramientas pueden generar código a partir de modelos, pero generalmente no van más allá de la generación de algún esqueleto de código que luego se debe completar manualmente.

En contraste, las transformaciones MDD son siempre ejecutadas por herramientas, como se muestra en la Ilustración 5. Muchas herramientas pueden transformar un PSM a

código; no hay nada nuevo en eso. Dado que un PSM es un modelo muy cercano al código, esta transformación no es demasiado compleja. Lo novedoso que propone MDD es que las transformaciones entre modelos (por ejemplo de un PIM a PSMs) sean automatizadas.

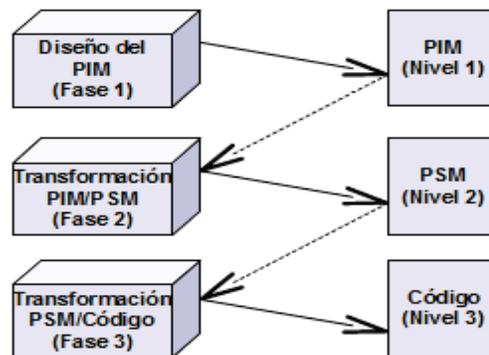


Ilustración 5: Los tres pasos principales en el proceso de desarrollo MDD

Los impactos en el ciclo de vida del software se resumen en los siguientes puntos:

- Más tiempo en análisis y diseño.
- Menos tiempo de codificación.
- Menos defectos, más calidad.
- Más productividad.
- Integración continua
- Ciclos de desarrollo más ágiles.
- Menos coste.

Visto hasta aquí como es el ciclo de vida de un proyecto aplicando MDD y comprendidas las transformaciones que se realizan durante el proceso, se verá a continuación cuáles son las tareas y los roles definidos por MDD.

Entendiendo las tareas y los roles en el proyecto de desarrollo

MDD tiene un efecto profundo sobre la forma en que construimos las aplicaciones de software. Las organizaciones y los proyectos frecuentemente dependen de expertos clave, quienes toman las decisiones respecto al sistema. MDD permite capturar su experiencia en los modelos y en las transformaciones permitiendo de esta forma que otros miembros del equipo pueden aprovecharla sin requerir de su presencia. Además, este conocimiento se mantiene aún cuando los expertos se alejen de la organización. Por otra parte, el costo de desarrollo y de pruebas se reduce significativamente al automatizar gran parte del trabajo de generación del código y otros artefactos a partir de los modelos. A través de la automatización MDD favorece la generación consistente de los artefactos reduciéndose el número de errores.

Para lograr los beneficios de MDD el proceso de desarrollo de software debe adaptarse.

Las tareas

El administrador de un proyecto MDD en realidad debe controlar un proyecto dentro de otro proyecto (Ilustración 6). El proyecto interno consiste en fabricar herramientas MDD para que sean usadas por el equipo que trabaja en el desarrollo del proyecto externo.

El hecho de tener dos proyectos acoplados nos obliga a organizar y planificar más cuidadosamente, especialmente al comienzo del proyecto. A las dificultades usuales asociadas con cualquier proyecto de desarrollo, ahora debemos sumarle un conjunto adicional de dependencias internas. Debemos identificar, desarrollar e instalar las herramientas MDD requeridas antes de que los desarrolladores las necesiten. Esto implica que los flujos de tareas de ambos proyectos están interrelacionados. La parte positiva es que podemos decidir como distribuir el esfuerzo entre ambos proyectos. Por ejemplo, al identificarse un nuevo requerimiento, podríamos temporariamente mover algunos desarrolladores del proyecto externo hacia el proyecto de desarrollo de herramientas, de tal forma que ellos puedan mejorar las herramientas disponibles.



Ilustración 6: MDD: Proyecto dentro de otro proyecto

Los Roles

MDD distingue dos roles diferentes en el proceso de desarrollo: aquellos que crean la aplicación y aquellos que desarrollan las herramientas. Esta separación no es algo nuevo sino que la encontramos aplicada ya en otras industrias. En el desarrollo basado en componentes, algunas personas fabrican los componentes, o bien la unidad organizacional completa podría actuar como una fábrica de componentes, y otras personas usan estos componentes para crear sus aplicaciones. Similarmente, en MDD algunas personas crean la plataforma común para varios proyectos y otras personas desarrollan los productos usando los elementos de esta plataforma común. El desarrollo de software está cambiando su enfoque desde la idea de *tener generalistas* a la idea de *tener especialistas*, en forma similar a otras empresas e industrias. Esta separación de dos roles en MDD no significa que las personas son necesariamente diferentes. A veces quienes desarrollan la plataforma MDD también la usan. Lo que si resulta crucial es que los desarrolladores más experimentados se encarguen de desarrollar la plataforma MDD. Estos desarrolladores experimentados pueden obviamente especificar la automatización en términos de lenguajes de modelado, modelos, transformaciones entre modelos, generadores de código y frameworks del dominio mejor que aquellos con menos experiencia. Además, ellos cuentan con el reconocimiento necesario entre sus colegas. Concretamente en MDD podemos identificar los siguientes roles:

- **Los expertos en el dominio** son personas que tienen conocimiento acerca del dominio del problema. Los desarrolladores de aplicaciones también califican aquí cuando han desarrollado múltiples aplicaciones de similar tipo en el pasado.
- **Los desarrolladores del lenguaje** seleccionan y/o crean el lenguaje de modelado si no se dispone de ninguno que se ajuste al dominio de nuestro problema. Ellos lo formalizan mediante un metamodelo o mediante un perfil de UML y proveen guías sobre como usarlo, tales como manuales y ejemplos de modelos.
- **Los modeladores** (o ingenieros del PIM) aplican los lenguajes de modelado. Este grupo es usualmente el más grande. Algunos modeladores crean los modelos a partir de los cuales se generará la aplicación.
- **El ergonomista** puede ayudar a los desarrolladores del lenguaje a mejorar la usabilidad del lenguaje. Si bien la usabilidad es siempre un factor relevante, este rol puede ser particularmente significativo en ciertos casos.
- **Los desarrolladores de las transformaciones** y de los generadores de código especifican las transformaciones del PIM al PSM y del PSM al código. Ellos son expertos en el uso de lenguajes de transformación como QVT, ATL y Mof2Text.
- **Los expertos en el framework del dominio** (o ingenieros del PSM) son usualmente arquitectos de software o desarrolladores con experiencia en el uso y desarrollo de componentes y frameworks.

Planificación y seguimiento del proyecto de desarrollo

Una vez que se han definido las tareas, las habilidades requeridas y las dependencias entre las tareas, el proceso de planificación y seguimiento de un proyecto MDD es similar al de cualquier proyecto de desarrollo de software. Se analiza la envergadura de las tareas y se le asignan los recursos apropiadamente. Se necesitan las habilidades usuales de un gerente de proyecto para detectar los lugares donde el cronograma no se cumple y realizar los ajustes apropiados a la asignación de recursos cuando resulte necesario.

Las siguientes secciones proveen algunos consejos adicionales acerca del esfuerzo de planificación y seguimiento del proyecto.

Una de las áreas de proceso de CMMI que debe satisfacerse para alcanzar el nivel 2, es la Planificación de Proyecto (PP). En el capítulo siguiente se verá entonces, cómo mapea la planificación y el seguimiento de un proyecto MDD para cubrir esa KPA.

Evaluación del proyecto

Al finalizar un proyecto MDD es útil generar las siguientes métricas:

- El costo de desarrollo de las herramientas.
- La productividad de los desarrolladores de la aplicación al usar las herramientas. Para realizar una comparación justa con respecto a los proyectos tradicionales, será necesario expresarla en términos del esfuerzo que hubiese requerido desarrollar todo el código manualmente.
- El nivel de calidad logrado por el equipo de desarrollo.
- El esfuerzo requerido para permitir que las herramientas MDD puedan ser reusadas en los siguientes proyectos y los beneficios potenciales que esto reporta.

Estas mediciones permitirán evaluar el valor real derivado de aplicar la propuesta MDD y también darán guías acerca de como explotar mejor sus potenciales en los proyectos futuros.

Administración y reuso de los artefactos

Modelos, transformaciones, y código son los principales artefactos que se generan y se usan en un proyecto MDD. Estos artefactos pueden ser tan simples como archivos de texto conteniendo propiedades de configuración y documentación, o tan complejos como archivos de proyecto conteniendo modelos, código ejecutable y descriptores de emplazamiento.

Típicamente, los ingenieros de software prefieren diseñar y construir sus propios artefactos, y los gerentes de los proyectos prefieren evitar las dependencias entre proyectos que se generan como consecuencia de compartir artefactos. Sin embargo, la

intención de reusar artefactos siempre está presente, debido a su gran potencial para ahorrar tiempo y costos en el desarrollo de software. MDD abre nuevas posibilidades de reuso que no deben ser desaprovechadas, por ejemplo:

- Cuando un proyecto define sus modelos conceptuales, es más fácil detectar puntos en común entre lo que se necesita construir y lo que ya fue construido, debido a que ambos modelos conceptuales tienen un nivel de abstracción más alto que el código, eliminando los detalles que oscurecen a los patrones comunes.
- Un pequeño cambio en un perfil de UML y en las transformaciones que lo usan, puede permitir a un proyecto reusar todo el ambiente de desarrollo de un proyecto anterior.
- Cuando las transformaciones hacen buen uso de archivos de datos de configuración y plantillas, un proyecto puede generar diferentes artefactos con sólo realizar pequeños cambios a dichos archivos de configuración y plantillas.

Reuso de los artefactos

El éxito de un proyecto MDD depende del reuso exitoso de los artefactos. El reuso en MDD incluye el reuso de patrones, modelos, transformaciones y en menor medida el del código. La administración de estos artefactos, sus descripciones relacionadas y el mantenimiento de los repositorios se torna cada vez más importante, incluyendo las siguientes cuestiones:

- Identificar y recuperar un artefacto para reusarlo.
- Asegurarse de haber recuperado el artefacto apropiado para la versión de la plataforma destino.
- Analizar la integridad de un artefacto y verificar que sea la última versión o la versión apropiada.
- Asegurarse de que el artefacto esté certificado cuando esto sea requerido.
- A la hora de decidir cuales métodos y herramientas se utilizarán para la administración de los artefactos, debemos considerar un conjunto de factores,

incluyendo servicios de administración de integridad y soporte para el emplazamiento de los artefactos.

Reuso de las herramientas

Las herramientas MDD que construimos en el subproyecto interno serán usadas numerosas veces por los desarrolladores para crear los artefactos de la aplicación en el proyecto externo. También es conveniente que estas herramientas se reusen en proyectos futuros, amortizando de esta manera el costo insumido en su construcción.

Las buenas herramientas MDD incrementan el porcentaje de tiempo que los ingenieros de software dedican al proceso creativo de diseñar el sistema, comparado con el proceso rutinario de codificar y depurar el software. Como resultado, una vez que ellos han visto los beneficios del proceso MDD, están motivados a explotar y extender las herramientas MDD en los siguientes proyectos. Por lo tanto, es valioso investigar en que casos las herramientas MDD que se crean para un proyecto tienen potencial para ser reusadas en otros proyectos futuros. Esta evaluación puede realizarse durante la fase de planificación, al comienzo del proyecto, o durante la fase de revisión hacia el final del proyecto.

El rol de UML en MDD

Generalmente UML es el lenguaje elegido para definir los modelos en MDD. UML es un estándar abierto y es el estándar de facto para el modelado de software. UML es un lenguaje de modelado de software de propósito general que podemos aplicar de varias formas. Las técnicas de modelado apropiadas para diseñar una aplicación de telefonía, embebida y de tiempo real y las técnicas de modelado para desarrollar una aplicación de comercio electrónico son bastante diferentes, sin embargo podemos usar UML en ambos casos.

Los perfiles de UML son ortogonales entre sí, por lo tanto varios perfiles pueden aplicarse simultáneamente.

El uso de UML como lenguaje en MDD tiene los siguientes beneficios:

- El uso de perfiles de UML para MDD nos permite aprovechar la experiencia que se ha ganado en el desarrollo de este lenguaje. Esto significa que podemos proveer un ambiente de modelado personalizado sin afrontar el costo de diseñarlo e implementarlo desde cero;
- UML es un estándar abierto y el estándar de facto para el modelado de software en la industria;
- UML ha demostrado ser durable, su primera versión apareció en 1995;
- El éxito de UML ha propiciado la disponibilidad de muchos libros y cursos de muy buena calidad;
- La mayoría de los estudiantes de las carreras relacionadas con la ingeniería de software han aprendido algo de UML en la universidad;
- Existen muchas herramientas maduras que soportan UML;
- Muchas organizaciones necesitan desarrollar varios tipos diferentes de software. Si podemos usar una solución de modelado en común, configurada de forma apropiada para describir cada uno de esos dominios de software, entonces será más fácil lidiar con su posterior integración.

El primer paso en el proceso MDD consiste en la construcción de un modelo independiente de la plataforma (PIM) que describa (represente o modele) al sistema que se quiere implementar. Mediante el uso del lenguaje UML se puede realizar la representación del PIM. Al ser UML un lenguaje extensible, permite la creación de perfiles. Dependiendo el sistema que se esté desarrollando, es posible que sea necesario la creación de perfiles (UML profiles). De esta manera, contamos con elementos para modelar definidos por nosotros mismos y estandarizados para todo aquel que participe en la construcción del sistema.

Conclusión

Es posible afirmar que MDD cuenta con ciertas bases sólidas que hacen más simple su aplicación en el desarrollo de un sistema software. Se pueden resumir los principales avances de MDD en los siguientes aspectos:

- Modelos y Lenguajes de modelado: el lenguaje más conocido y utilizado para la especificación de **PIMs** y **PSMs** es el estándar UML definido por la OMG, acompañado por el lenguaje de consulta OCL.
- Transformación de modelos: el OMG esta actualmente trabajando en la definición de un lenguaje estándar para describir transformaciones. Este estándar es conocido como **QVT**, que quiere decir Query, View and Transformations y se encuentra en proceso de estandarización.
- Herramientas automáticas: actualmente existen herramientas que combinan algunas de las funciones requeridas. Estas funciones pueden ser específicas para un lenguaje o pueden ser genéricas.
- Aplicación de métodos formales al **MDD**: acompañando a las actividades de estandarización de lenguajes y procesos coordinadas por el **OMG**, las cuales tienen un enfoque básicamente “pragmático”, existe una bien organizada comunidad científica, de la cual los integrantes de este proyecto forman parte, abocada al análisis de los elementos del **MDD** desde la perspectiva científica.

El administrador de un proyecto MDD en realidad debe controlar un proyecto dentro de otro proyecto. El proyecto interno consiste en fabricar herramientas MDD para que sean usadas por el equipo que trabaja en el desarrollo del proyecto externo. Por lo tanto, MDD distingue dos roles diferentes en el proceso de desarrollo: aquellos que crean la aplicación y aquellos que desarrollan las herramientas. Concretamente se pueden identificar los siguientes roles: los expertos en el dominio, los desarrolladores del lenguaje, los modeladores, el ergonomista, los desarrolladores de las transformaciones y de los generadores de código y los expertos en el framework del dominio y la plataforma de implementación.

Una vez que ha quedado claro el orden de las tareas en un proyecto MDD, el proceso de planificación y seguimiento es similar al de cualquier proyecto de software. Se definen las tareas, se analiza su envergadura y se le asignan recursos apropiadamente. En gran medida el éxito de un proyecto MDD depende del reuso exitoso de los artefactos. El reuso en MDD incluye el reuso de patrones, modelos, transformaciones y en menor medida el del código. La administración de estos artefactos, sus descripciones relacionadas y el mantenimiento de los repositorios se torna entonces un tema relevante.

Capítulo 5: *Una perspectiva común entre RUP, MDD y CMMI*

Habiendo desarrollado en los capítulos anteriores el concepto de calidad, modelo de madurez CMMI y las metodologías de desarrollo que una organización puede adoptar, este capítulo se centra en destacar los puntos de conexión entre metodologías para el desarrollo de software y CMMI.

Para conocer los puntos de conexión entre CMMI nivel 3 y las tareas que implementa una organización desarrolladora de software que utiliza MDD, es importante y útil hacer el mismo trabajo con el Nivel 2, ya que para este nivel la organización deja sus proceso ad-hoc para alinearse con los estándares y es donde las metodologías de desarrollo ya pueden empezar a cubrir ciertas KPA. Se realiza un mapeo primero con Nivel 2 y luego con Nivel 3. Como se ha mencionado anteriormente, una organización debe trabajar paulatinamente para alcanzar la madurez, los niveles no pueden pasarse de uno a otro en forma abrupta, el camino debe ser progresivo e incremental sino sería contraproducente.

Si una organización opta por tomar solamente algunas KPA omitiendo el resto, esto no la convierte en una organización madura, y las falencias por tales omisiones no tardarán en salir a la luz.

Independientemente del trabajo global que realice una organización para poder certificarse, en este capítulo se centralizan los puntos de conexión entre alguna práctica específica de cada KPA con CMMI.

Mapeo de cada Nivel de CMMI con RUP y MDD

Las metodologías no cubren la totalidad de las prácticas específicas definidas por cada nivel en CMMI, sino que existen ciertas conexiones que se explicarán en este capítulo. Se recorrerán todas las KPA de los niveles 2 y 3 de CMMI; luego se toma algún aspecto de la KPA que se esté tratando y se hace una conexión con RUP o MDD. Para aquellas KPA que no tengan ninguna conexión concreta con RUP o MDD se describen los aspectos, actividades, roles, etc. que debe cubrir la organización para poder establecer algún tipo de mapeo.

La organización debe analizar las KPA que ya está cubriendo o bien aquellas KPA a las cuales se está encaminando. Una vez detectadas las KPA, como primer paso se las debe reforzar para completar su definición y luego recién ir sumando las restantes hasta completar los requisitos de cada nivel.

En este capítulo se presentan las prácticas específicas de las KPA cubiertas por la metodología adoptada y se sugieren los caminos a seguir por la organización para completar las restantes.

La descripción de los niveles CMMI en este capítulo es completa, es decir, se nombran todas y cada una de las KPA de nivel 2 y 3. Luego se nombran las metas y prácticas específicas de cada KPA. Esto nos ayudará a comprender el contexto de cada KPA, cual es su objetivo y en qué parte de las tareas organizacionales se centra.

Al finalizar este capítulo se podrán responder las siguientes preguntas:

- ¿Qué aspectos de las KPA de nivel 2 y 3 se cumplen por estar utilizando RUP
- ¿Qué aspectos de las KPA de nivel 2 y 3 se cumplen por estar utilizando MDD?

Nivel 2 – Mapeo con RUP y CMMI

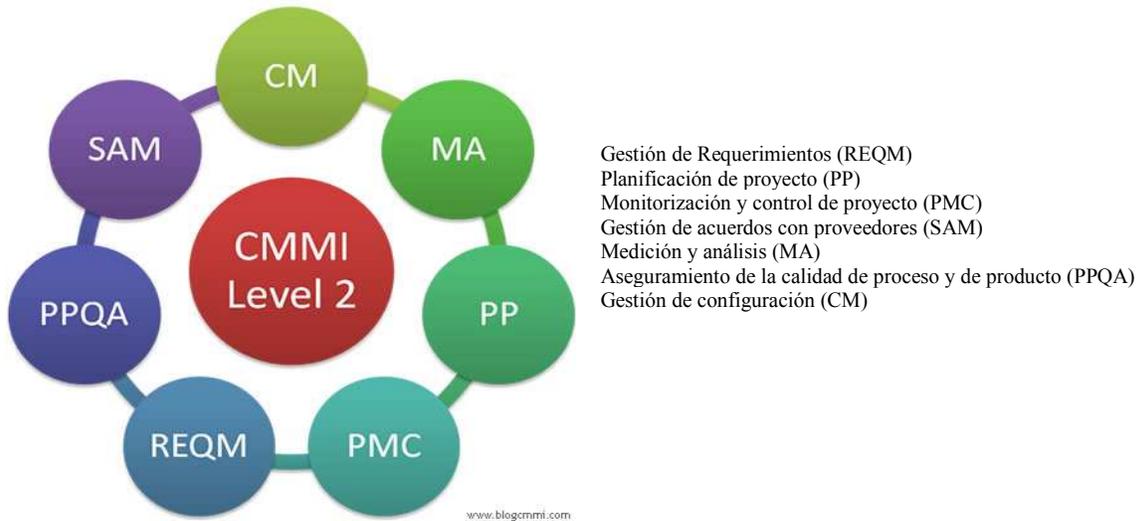


Ilustración 7: KPAs del Nivel 2

En el Nivel de madurez 2, los proyectos de la organización han asegurado que los procesos se planifican y realizan de acuerdo a políticas: los proyectos emplean personal con habilidad que dispone de recursos adecuados para producir resultados controlados; involucran a las partes interesadas relevantes; se monitorizan, controlan y revisan; y se evalúan en cuando a su adherencia a sus descripciones de proceso. La disciplina de proceso reflejada por el nivel de madurez 2 ayuda a asegurar que las prácticas existentes se mantienen durante tiempos de estrés. Cuando estas prácticas están en su lugar, los proyectos se realizan y gestionan de acuerdo a sus planes documentados.

En nivel 2, el estado de los productos de trabajo y la entrega de los servicios son visibles a la dirección en puntos definidos (p.ej., en los hitos principales y al finalizar las tareas principales). Se establecen compromisos entre las partes interesadas relevantes y se revisan, según sea necesario. Los productos de trabajo se controlan de forma apropiada. Los productos de trabajo y servicios satisfacen sus descripciones de proceso especificadas, estándares y procedimientos. La ilustración 7 muestra todas las KPA que deben cumplirse para alcanzar el nivel 2 de CMMI. A continuación se recorren las KPA del nivel 2, y se realiza su correspondencia con RUP y MDD:

KPA REQM (Gestión de Requerimientos)

Propósito: gestionar los requerimientos de los productos y de los componentes del producto del proyecto e identificar inconsistencias entre esos requerimientos y los planes y productos de trabajo del proyecto.

Notas Introdutorias: los procesos de gestión de requerimientos gestionan todos los requerimientos recibidos o generados por el proyecto, incluyendo tanto requerimientos técnicos como no técnicos, así como aquellos requerimientos impuestos al proyecto por la organización. En particular, si se implementa el área de proceso de Desarrollo de requerimientos, sus procesos generarán requerimientos de producto y de componentes del producto que también serán gestionados por los procesos de gestión de requerimientos. Durante las áreas de proceso, donde se usan los términos producto y componentes del producto, sus significados previstos también comprenden los servicios y sus componentes. Cuando las áreas de proceso de Gestión de requerimientos, de Desarrollo de requerimientos y de Solución técnica se implementan todas, sus procesos asociados pueden estar estrechamente ligados y ser realizados de forma concurrente.

El proyecto toma las medidas apropiadas para asegurar que el conjunto de requerimientos acordados se gestiona para dar soporte a las necesidades de planificación y de realización del proyecto. Cuando un proyecto recibe requerimientos de un proveedor de requerimientos aprobado, se revisan éstos con el proveedor de requerimientos para resolver los problemas y para prevenir malentendidos antes de que los requerimientos se incorporen en los planes del proyecto. Una vez que el proveedor y el receptor de los requerimientos alcanzan un acuerdo, se obtiene un compromiso sobre los requerimientos por parte de los participantes del proyecto. El proyecto gestiona los cambios a los requerimientos a medida que evolucionan e identifica cualquier inconsistencia que ocurra entre los planes, los productos de trabajo y los requerimientos. RUP provee mecanismos de soporte adecuados, integra el seguimiento (con control de cambios) de los requerimientos mediante la captura y el análisis de requerimientos.

Resumen de metas y prácticas específicas

SG 1 Gestionar los requerimientos

- SP 1.1 Obtener una comprensión de los requerimientos.
- SP 1.2 Obtener el compromiso sobre los requerimientos.
- SP 1.3 Gestionar los cambios de los requerimientos.
- SP 1.4 Mantener la trazabilidad bidireccional de los requerimientos.
- SP 1.5 Identificar las inconsistencias entre el trabajo del proyecto y los requerimientos.

La obtención de requerimientos se enfoca en la descripción del propósito del sistema. El cliente, los desarrolladores y los usuarios identifican un área problema y definen un sistema que ataca al problema. A tal definición se le llama especificación del sistema y sirve como contrato entre el cliente y los desarrolladores.

Práctica específica a mapear:

SP 1.2 Obtener el compromiso sobre los requerimientos

Obtener el compromiso de los participantes de proyecto sobre los requerimientos. Esta práctica específica se ocupa de los acuerdos y compromisos entre aquellos que tienen que llevar a cabo las actividades necesarias para implementar los requerimientos. Los requerimientos evolucionan a través del proyecto, especialmente según lo descrito por las prácticas específicas del área de proceso de Desarrollo de requerimientos y del área de proceso de Solución técnica. A medida que los requerimientos evolucionan, esta práctica específica asegura que los participantes del proyecto se comprometen con los requerimientos actuales y aprobados, y con los cambios resultantes en los planes, actividades y productos de trabajo del proyecto.

Mapeo CMMI-RUP

RUP emplea los Casos de Uso para planificar y empaquetar el trabajo que debe realizarse. Se obtiene en una primera iteración, una lista priorizada de Casos de Uso.

Todo el equipo de trabajo está involucrado en la revisión de esta lista de prioridades basándose en los riesgos del proyecto, la disponibilidad de los recursos y las necesidades de los usuarios.

Mapeo CMMI-MDD

El primer paso que se realiza en un proceso MDD es el relevamiento de los requerimientos y su posterior documentación. Tal como se mencionó anteriormente, MDD utiliza el lenguaje UML para documentar los modelos. En particular, para documentar los requerimientos de un sistema, utiliza los ya nombrados por RUP: Casos de Uso.

Documentando con UML

Luego de representar mediante diagramas de Casos de Uso los requerimientos, cada uno puede ser extendido mediante la conversación (escenarios) de cada Caso de Uso. Los escenarios sirven para documentar las interacciones entre los usuarios y el sistema para satisfacer el objetivo del Caso de Uso.

UML no define la forma de escribir los escenarios. Esta decisión debe ser tomada por cada persona o empresa. Los formatos más utilizados son de una columna (donde se representan mediante pasos numerados las interacciones entre los usuarios y el sistema), dos columnas (una para las interacciones del usuario y otra para las interacciones del sistema), y así, se puede continuar con más columnas dependiendo el nivel de detalle y organización que se desee.

Documentar los escenarios es de suma importancia para no dejar librada a la imaginación de quien lee un diagrama de Casos de Uso como es la interacción con el sistema. Además, ayuda a las etapas siguientes de construcción del software porque se detallan todos los pasos de interacción.

La simplicidad de los Casos de Uso permite que diferentes personas interesadas en el desarrollo de un sistema puedan interactuar para definir, refinar, completar cada funcionalidad que debe proveer el sistema. Ese es el objetivo principal de un Caso de Uso: poder documentar los requerimientos del sistema utilizando un lenguaje que pueda ser entendido por todos sin necesidad de conocimientos técnicos.

La documentación puede ser tan completa como se requiera, así, se puede añadir a los escenarios documentación para casos de error, precondiciones, postcondiciones y documentación adjunta, entre otros datos.

Queda claro que en MDD tenemos que controlar dos proyectos, uno de los cuales es la generación de herramientas que realizarán la transformación de los modelos. Este proyecto también maneja sus propios requerimientos. Pero al fomentar el reuso de

herramientas, la cantidad de nuevos requerimientos a incorporar en las herramientas, irá disminuyendo con el tiempo.

Los Casos de Uso sirven para documentar las nuevas funcionalidades de ambos proyectos, y más generalmente, los Casos de Uso nos permiten documentar las funcionalidades que debe proveer un sistema (proyecto) independientemente de su naturaleza.

Es importante destacar, que existe un punto de conexión entre los Casos de Uso del proyecto externo y del interno. Esta conexión se da porque no se puede crear una aplicación sin que las herramientas estén listas. Es decir, los requerimientos del proyecto externo no se van a poder satisfacer si los requerimientos del proyecto interno no fueron satisfechos.

KPA PP (Planificación de Proyecto)

Propósito: establecer y mantener planes que definan las actividades del proyecto.

Estos planes son necesarios para gestionar el proyecto software. Sin planes realistas, no es posible implementar una gestión efectiva del proyecto.

Notas introductorias: el área de proceso de Planificación de Proyecto implica:

- Desarrollar el plan del proyecto.
- Interactuar con las partes interesadas de forma apropiada.
- Obtener el compromiso con el plan.
- Mantener el plan.

La planificación comienza con los requerimientos que definen el producto y el proyecto.

La planificación incluye la estimación de los atributos de los productos de trabajo y de las tareas, a determinación de los recursos necesarios, la negociación de los compromisos, la elaboración de un calendario, y la identificación y el análisis de los riesgos del proyecto. Para establecer el plan del proyecto, puede ser necesario realizar iteraciones sobre estas actividades. El plan del proyecto proporciona la base para realizar y controlar las actividades del proyecto dirigidas a los compromisos con el cliente del proyecto.

El plan del proyecto usualmente necesitará corregirse a medida que el proyecto progresa, para dar tratamiento a los cambios en los requerimientos y en los compromisos, las estimaciones inexactas, las acciones correctivas y los cambios del proceso. Las prácticas específicas que describen tanto la planificación como la replanificación están contenidas en esta área de proceso.

El término "plan de proyecto" se usa durante todas las prácticas genéricas y específicas en esta área de proceso para referirse al plan global para controlar el proyecto.

Resumen de metas y prácticas específicas

SG1 Establecer estimaciones

SP 1.1 Estimar el alcance del proyecto.

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

SP 1.3 Definir el ciclo de vida del proyecto.

SP 1.4 Determinar las estimaciones de esfuerzo y coste.

SG2 Desarrollar un plan de proyecto

SP 2.1 Establecer el presupuesto y el calendario.

SP 2.2 Identificar los riesgos del proyecto.

SP 2.3 Planificar la gestión de los datos.

SP 2.4 Planificar los recursos del proyecto.

SP 2.5 Planificar el conocimiento y las habilidades necesarias.

SP 2.6 Planificar la involucración de las partes interesadas.

SP 2.7 Establecer el plan de proyecto.

SG3 Obtener el compromiso con el plan

SP 3.1 Revisar los planes que afectan al proyecto.

SP 3.2 Reconciliar los niveles de trabajo y de recursos.

SP 3.3 Obtener el compromiso con el plan.

Los grupos e individuos afectados están de acuerdo con los compromisos relacionados con el proyecto software.

La planificación se deriva del contrato establecido entre la empresa y el cliente realizando un desglose de cada uno de los Casos de Uso. Cada Caso de Uso tiene un objetivo, un hito a alcanzar.

La planificación se realiza en varias etapas: en una primera etapa se listan los hitos que se deben cumplir. Por cada hito se define una fecha estimativa de entrega. Una vez definidos los hitos y las fechas, es posible armar un cronograma más detallado realizando un desglose de tareas que se requieren para alcanzar un hito.

Esta información se plasma en algún documento que permita hacer seguimiento de tareas (un buen ejemplo, son los documentos que permiten visualizar diagramas de Gantt). Este documento es el que sirve de guía para conocer el estado del proyecto y proponer estrategias de soluciones para los casos en que se produzcan desvíos de las fechas iniciales estimadas.

El realizar un seguimiento organizado de los proyectos que desarrolla una empresa es una tarea sumamente importante, para conocer el avance de cada proyecto, los desvíos si los hubiere y las asignaciones de las personas a cada proyecto para tener un mapa completo del estado actual de todos y poder hacer mejores estimaciones en cuanto a tiempo y personas para próximos proyectos a realizar. Por lo tanto, todas las organizaciones deben adoptar una forma de poder realizar la planificación del proyecto y su consecuente seguimiento.

Práctica específica a mapear:

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

Establecer y mantener las estimaciones de los atributos de los productos de trabajo y de las tareas.

El tamaño es la principal entrada para muchos modelos usados para estimar el esfuerzo, el coste y el calendario. Los modelos también pueden estar basados en datos de entrada tales como la conectividad, la complejidad y la estructura.

Las estimaciones deberían ser consistentes con los requerimientos del proyecto para determinar el esfuerzo, el coste y el calendario del proyecto. Para cada atributo de tamaño, debería asignarse un nivel relativo de dificultad o complejidad.

Mapeo CMMI-RUP

En RUP el *Plan de Desarrollo del Software* define el plan general para el proyecto y el *Plan de Iteración* define en detalle qué es lo que se logra en cada iteración.

La *Revisión del Plan de Iteración* expone el Plan de Iteración a todos los intervinientes (*stakeholders*), permitiendo un consenso previo a que la iteración comience.

A partir del Plan de Iteración aceptado, el Project Manager realiza una serie de órdenes de trabajo. Una vez que el Project Manager obtiene las aceptaciones de las órdenes de trabajo con el equipo afectado la iteración puede comenzar.

RUP proporciona orientación para medir el tamaño del esfuerzo para un proyecto software. Se determina el tamaño por analogía y por análisis. RUP no provee asistencia para determinar el esfuerzo de atributos no-software (por ejemplo: labor, maquinaria, materiales y métodos requeridos por el proyecto).

Las estimaciones del Software se documentan para utilizarlas en la planificación y seguimiento del proyecto software.

Uno de los objetivos de RUP es asegurar que las expectativas de todas las partes se sincronicen y sean consistentes.

Esto se asegura mediante valoraciones periódicas a través del ciclo de vida del software.

- RUP realiza los siguientes tipos de métricas:
 - Progreso (líneas de código, número de clases, puntos de función por iteración, cambio)

- Estabilidad (tipo de cambio, requisitos o volatilidad de la implementación)
- Adaptabilidad (coste del cambio)
- Modularidad (extender el impacto del cambio)
- Calidad (ratio de descubrimiento de defectos, densidad, profundidad, indicador de cambio)
- Madurez (horas de pruebas por fallo)
- Perfil de cambios de los recursos (planificados vs. actuales)
- Actividades del proyecto software y compromisos se planifican y documentan.
- Los documentos de RUP que capturan los planes y compromisos del proyecto son:
 - Business Case
 - Software Development Plan
 - Measurement Plan
 - Risk List
 - Project Plan
 - Iteration Plan(s)
 - Iteration Assessment(s)
 - Status Assessment(s)

Mapeo CMMI-MDD

El proceso de planificación y seguimiento de un proyecto MDD es similar al de cualquier proyecto de desarrollo de software. Se analiza la envergadura de las tareas y se le asignan los recursos apropiadamente. Se necesitan las habilidades usuales de un gerente de proyecto para detectar los lugares donde el cronograma no se cumple y realizar los ajustes apropiados a la asignación de recursos cuando resulte necesario.

No se debe entender que una planificación de un proyecto MDD consta de:

- Hito 1: Construcción de Casos de Uso.
- Hito 2: Construcción de Modelos PIM.

- Hito 3: Construcción de Modelos PSM.
- Hito 4: Generación de código.

Es un error pensar que los hitos de un proyecto se corresponden con las fases de MDD. Los hitos deben corresponderse con metas a alcanzar de un proyecto, los productos entregables a los cuales se les puede asignar fecha de finalización. Por ejemplo, podría ser un portal para logueo de usuarios. Los pasos que lleven a obtener ese objetivo es lo propio de MDD.

Es importante destacar que un “cliente” no necesariamente es una entidad externa que requiere de un servicio. En MDD podemos asumir que cliente es el grupo de trabajo que desarrolla el proyecto externo y que consume el producto realizado por el equipo que desarrolla el proyecto interno.

En este caso, se debe especificar esta dependencia de tareas en el plan de proyecto. En caso de que se retrase el desarrollo de las herramientas, impacta directamente en el proyecto externo ya que se proyecta como un desvío de las fechas. Una dependencia que se genera con el último hito entregable del proyecto interno del cual depende el hito de comienzo del proyecto externo.

¿Cómo se determina el esfuerzo de planificación?

Usando un proceso iterativo. Es aconsejable separar el desarrollo en varias iteraciones. En la primera iteración se desarrolla un ejemplo de cada tipo de modelo y transformación. Esto tiene los siguientes beneficios:

- El equipo de desarrollo gana experiencia y puede estimar cuanto tiempo y esfuerzo implicará la construcción del resto del proyecto.
- Todo el equipo de desarrollo está activo, es decir, los implementadores de transformaciones no tienen que esperar a que todo el modelo PIM este listo para comenzar a trabajar en sus transformaciones.
- Permite mostrar algunos resultados anticipadamente lo cual ayudará a disminuir el escepticismo y aumentar la confianza en el éxito del proyecto.

Las siguientes iteraciones trabajarán sobre la experiencia ya ganada para soportar escenarios de la aplicación cada vez más amplios.

¿Cómo se realiza el seguimiento del proyecto?

En MDD se sugieren realizar las siguientes tareas durante la etapa de desarrollo:

- Realizar una inversión explícita en las herramientas de soporte.
- Usar a la gente más calificada para desarrollar las herramientas MDD con el objetivo de capturar y automatizar su experiencia.
- Considerar que además del código el proyecto generará documentos, configuraciones, reportes y casos de prueba.
- Asegurarse de que el proceso de desarrollo soporta ambientes de prueba además de ambientes de producción.
- Recordar definir las estrategias de manejo de configuraciones para sus herramientas MDD.
- Asigne tiempo al entrenamiento del equipo sobre el uso de las herramientas MDD.
- Tomarse tiempo para considerar si sus herramientas MDD serán re-usables en proyectos futuros.

KPA PMC (Monitorización y Control de Proyecto)

Propósito: proporcionar una comprensión del progreso del proyecto para que se puedan tomar las acciones correctivas apropiadas, cuando el rendimiento del proyecto se desvíe significativamente del plan.

Notas introductorias: un plan documentado de proyecto es la base para la monitorización de las actividades, la comunicación del estado y la toma de acciones correctivas. El progreso se determina principalmente comparando los atributos de los productos de trabajo y de las tareas, el esfuerzo, el coste y el calendario reales, con el plan en los hitos o niveles de control prescritos dentro del calendario del proyecto o de la estructura de descomposición del trabajo (WBS, *Work Breakdown Structure*). La visibilidad apropiada permite que las acciones correctivas sean llevadas a cabo de manera

oportuna cuando el rendimiento se desvía significativamente del plan. Una desviación es significativa si, cuando se deja sin resolver, impide al proyecto cumplir sus objetivos.

El término "plan de proyecto" se usa en todas estas prácticas para hacer referencia al plan global para controlar el proyecto.

Cuando el estado real se desvía significativamente de los valores esperados, se llevan a cabo acciones correctivas según sea apropiado. Estas acciones pueden requerir replanificación, la cual puede incluir la corrección del plan original, el establecimiento de nuevos acuerdos o la inclusión de actividades adicionales de mitigación dentro del plan actual.

Resumen de metas y prácticas específicas

SG1 Monitorizar el proyecto frente al plan

SP 1.1 Monitorizar los parámetros de planificación del proyecto.

SP 1.2 Monitorizar los compromisos.

SP 1.3 Monitorizar los riesgos del proyecto.

SP 1.4 Monitorizar la gestión de datos.

SP 1.5 Monitorizar la involucración de las partes interesadas.

SP 1.6 Llevar a cabo revisiones de progreso.

SP 1.7 Llevar a cabo revisiones de hitos.

SG2 Gestionar las acciones correctivas hasta su cierre

SP 2.1 Analizar problemas.

SP 2.2 Llevar a cabo las acciones correctivas.

SP 2.3 Gestionar las acciones correctivas.

Todas las KPA de este nivel tienen relación entre sí, todas apuntan a las mismas prácticas genéricas, pero vamos a tomar algunas de ellas para hacer el análisis. En principio, nos enfocamos en las prácticas SP 1.6 y SP 1.7 y éstas tienen que ver con el seguimiento que se hace respecto a una planificación previa que se realiza de los proyectos (punto también tratado en la KPA PP). Ya se ha planteado la importancia de

tener una planificación previa al comienzo del desarrollo del proyecto. Ahora, para asegurarnos que el desarrollo siga por el camino que fue planeado, es fundamental hacer un seguimiento del mismo. Cada empresa define su metodología de hacer el seguimiento. Es común que el seguimiento se realice entre el coordinador del proyecto y las personas involucradas, pero está surgiendo con mucho peso el tener un área central en la empresa que se dedique a gestionar la cartera de todos los proyectos, la PMO (*Project Management Office*). Una de las tareas que realiza esta área es realizar el seguimiento de todos los proyectos que se desarrollan y tomar decisiones frente a problemas y desvíos que puedan surgir.

Por un lado se tiene la planificación que consta básicamente de: tareas, fechas y asignaciones. Toda esta información fue planteada y consensuada antes de comenzar el proyecto. En las reuniones de seguimiento se realiza una revisión si estas tareas se cumplen según lo planificado o si existen desviaciones de fechas. Es muy importante conocer los motivos de los desvíos para poder tomar acciones.

Práctica específica a mapear:

SP 1.4 Monitorizar la gestión de datos

Monitorizar la gestión de los datos del proyecto frente al plan de proyecto.

Una vez que los planes para la gestión de los datos del proyecto se realizan, la gestión de esos datos debe monitorizarse para asegurar que se cumplen esos planes.

Mapeo RUP-CMMI

Los resultados actuales y de rendimiento son comparados con los planes de software. Como se indica en la sección PP, el RUP tiene varios niveles de planes de proyecto y un *Status Assessment Report* que se genera para valorar el rendimiento planificado comparado con el actual. Este informe, generado para hitos específicos, es responsabilidad del Project Manager.

Se toman y gestionan acciones correctivas cuando el rendimiento de los resultados actuales se desvía de forma significativa de los planes de software.

RUP requiere el seguimiento del rendimiento a dos niveles: como parte de la rutina de monitorización y control del proyecto, el Project Manager recopilará métricas para producir un *Status Assessment*.

Los cambios de los compromisos del software se acuerdan por los grupos e individuos afectados.

El control iterativo del proceso, como se indica en RUP, asegura que los intervinientes (stakeholders) obtengan una visión del avance del proyecto y cualquier cambio que pueda ser necesario para que el proyecto permanezca supervisado.

Los cambios propuestos se revisan en un *Change Control Board* (CCB) para asegurar que son realistas y pueden ser adecuados al calendario general del proyecto.

Mapeo MDD-CMMI

El seguimiento de los proyectos que utilicen MDD no varía de otras metodologías. Se realiza un seguimiento según la planificación inicial y se verifica si todo se cumple según lo acordado o si hay algún tipo de desvío.

Por lo tanto, en estas reuniones de seguimiento se cuenta principalmente con dos fuentes de información: la primera es la planificación inicial que se creó al comienzo del proyecto, y segundo, es el estado actual del proyecto en particular.

Cabe destacar que, para una empresa que desarrolle simultáneamente muchos proyectos, cada responsable de proyecto puede mantener una planificación interna además de contar con una planificación general para todos los proyectos. Esta última planificación la mantiene un gerente, un coordinador, PMO, etc.

Cuando se realiza el seguimiento de los proyectos entonces se realiza una revisión de la planificación inicial y se revisa junto a la planificación propia del proyecto.

Una planificación general contiene todos los proyectos a un nivel alto de abstracción. Por cada uno, va a interesar conocer el avance para alcanzar los hitos que fueron planificados.

En la primera fase del proyecto, se obtuvo la lista priorizada de los Casos de Uso. Supongamos que un coordinador es el responsable de coordinar un Caso de Uso específico. Este coordinador, crea una planificación interna donde puede figurar el desglose de tarea y acá sí es válido incluir las fechas de finalización de los PIM y PSM por ejemplo.

Al momento de realizar el seguimiento, tiene sentido que el coordinador de tal proyecto indique un porcentaje de avance general sin entrar en detalle en qué etapa definida por MDD se está ejecutando en ese momento.

No olvidar que quien lleva a cabo la revisión y seguimiento de los proyectos le interesa conocer fechas de hitos generales, no a corto plazo ni conocer el desglose de tareas realizado ya que son temas ajenos a tal persona.

Una buena práctica a incluir en los planes de proyectos, es la elaboración documentos donde figuren los posibles riesgos que pueden ocurrir durante el desarrollo del proyecto, especificando las consecuencias.

Dentro de ese documento, podrían incluirse registros relacionados a la entrega de sus proyectos. Por ejemplo: “la demora en la entrega de la herramienta puede generar desvíos en el comienzo del proyecto externo”, se especifica quién reporta este registro y los artefactos que afecta.

KPA SAM (Gestión de Acuerdos con Proveedores)

Propósito: gestionar la compra de productos

Notas introductorias: el área de proceso de Gestión de acuerdos con proveedores involucra:

- Determinar el mecanismo de compra.
- Seleccionar los proveedores.

- Establecer y mantener los acuerdos con los proveedores.
- Realizar el acuerdo del proveedor.
- Evaluar los productos suministrados por el proveedor.
- Aceptar la entrega de los productos adquiridos.
- Entregar los productos al proyecto.

Esta área de proceso trata principalmente la compra de los productos y de los componentes del mismo que se entregan al cliente del proyecto. En todas las áreas de proceso donde usamos los términos producto y componente de producto, sus significados previstos engloban también a los servicios y a sus componentes.

Resumen de metas y prácticas específicas

SG1 Establecer los acuerdos con proveedores

- SP 1.1 Determinar el tipo de compra.
- SP 1.2 Seleccionar los proveedores.
- SP 1.3 Establecer los acuerdos con el proveedor.

SG2 Satisfacer los acuerdos del proveedor

- SP 2.1 Realizar el acuerdo del proveedor.
- SP 2.2 Monitorizar los procesos seleccionados del proveedor.
- SP 2.3 Evaluar los productos de trabajo seleccionados del proveedor.
- SP 2.4 Aceptar los productos adquiridos.
- SP 2.5 Transferir los productos.

Mapeo RUP-CMMI

Los objetivos de esta KPA caen fuera del ámbito actual de RUP y dependen de la organización.

Mientras que la subcontratación no se especifica especialmente en RUP, se supone que si los subcontratistas utilizan herramientas, técnicas y mecanismos de RUP, podemos suponer que el proceso permanece homogéneo.

Mapeo MDD-CMMI

Supongamos que una empresa aplica MDD en el desarrollo de sus proyectos y trabaja con una tecnología particular. Si esta empresa es proveedora de servicios y ocurre que un cliente solicita un sistema en otra tecnología (distinta), la empresa puede optar por la compra de un servicio a una tercera empresa.

Este servicio podría ser la transformación a PSM. La empresa podría realizar las primeras etapas del desarrollo (requerimientos, modelado de Casos de Uso y generación del PIM) y luego con este PIM finalizado y revisado se cede a otra empresa para que genere el PSM y el consecuente código.

El PIM generado puede ser reutilizado. De esta manera, si el día de mañana se necesita cambiar de tecnología, la única modificación sería en el PSM.

Ahora bien, al igual que sucede con RUP, la selección del proveedor depende de cada organización. Uno de los parámetros que se puede sugerir podría ser que la empresa tercerizada trabaje con MDD y que además, trabaje con la tecnología requerida.

KPA MA (Medición y Análisis)

Propósito: desarrollar y sustentar una capacidad de medición que se utiliza para poder dar soporte a las necesidades de información de la gerencia.

Notas introductorias: el área de Medición y análisis involucra:

- Especificar los objetivos de medición y análisis de modo que estos estén alineados con las necesidades de información y los objetivos identificados.
- Especificar las medidas, las técnicas de análisis y los mecanismos para la recogida de datos, almacenamiento de datos, informes y realimentación.
- Implementar la recogida, almacenamiento, análisis e informes de los datos.
- Proporcionar resultados objetivos que puedan utilizarse en la toma de decisiones informadas y en la toma de acciones correctivas apropiadas.

La integración de las actividades de medición y análisis en los procesos del proyecto da soporte a:

- La planificación y estimación objetivas.
- El seguimiento del rendimiento real frente a los planes y objetivos establecidos.
- La identificación y resolución de problemas relativos al proceso.
- El suministro de una base para incorporar la medición en procesos adicionales en el futuro.

El personal requerido para implementar una capacidad de medición puede o no estar integrado dentro de un programa constituido a nivel organizativo por separado. La capacidad de medición puede integrarse dentro de proyectos individuales o dentro de otras funciones de la organización (p.ej. en aseguramiento de la calidad).

El enfoque inicial para las actividades de medición se encuentra a nivel proyecto. Sin embargo, una capacidad de medición puede resultar útil para tratar las necesidades de información de la organización y/o de toda la empresa. Para poder dar soporte a esta capacidad, las actividades de medición deberían dar soporte a las necesidades de información a varios niveles, incluyendo el negocio, la unidad organizativa y el proyecto, para minimizar el re-trabajo a medida que madura la organización.

Los proyectos pueden decidir almacenar datos y resultados específicos del proyecto en un repositorio específico del proyecto. Cuando los datos se comparten más ampliamente entre los proyectos, éstos pueden residir en el repositorio de medición de la organización.

La medición y el análisis de los componentes del producto proporcionados por los proveedores son fundamentales para la gestión eficaz de la calidad y de los costes del proyecto. Es posible, con una gestión minuciosa de los acuerdos con el proveedor, proporcionar la visión de los datos que dan soporte al análisis del rendimiento del proveedor.

Resumen de metas y prácticas específicas

SG1 Alinear las actividades de medición y análisis.

SP 1.1 Establecer los objetivos de medición.

SP 1.2 Especificar las medidas.

SP 1.3 Especificar los procedimientos de recogida y de almacenamiento de datos.

SP 1.4 Especificar los procedimientos de análisis.

SG2 Proporcionar los resultados de la medición.

SP 2.1 Recoger los datos de la medición.

SP 2.2 Analizar los datos de la medición.

SP 2.3 Almacenar los datos y resultados.

SP 2.4 Comunicar los resultados.

Práctica específica a mapear:

SP 2.4 Comunicar los resultados

Informar de los resultados de las actividades de medición y análisis a todas las partes interesadas relevantes.

Los resultados del proceso de medición y análisis se comunican a las partes interesadas relevantes de forma utilizable y oportunamente para soportar la toma de decisiones y ayudar en la toma de acciones correctivas.

Las partes interesadas relevantes incluyen a los usuarios previstos, a los patrocinadores, a los analistas de datos y a los proveedores de datos.

Mapeo RUP-CMMI

RUP define el *Status Assessment* que es derivado del *Project Measurements* en la actividad "Monitoreo del Estado del Proyecto". El *Status Assessment* es utilizado para asegurar que las expectativas (del *Project Manager*, líderes de distintas áreas funcionales y la autoridad que revisa el proyecto) sean sincronizadas y consistentes.

Mapeo MDD-CMMI

Cuando se quieren realizar mediciones en una empresa, es fundamental que los criterios de medición se encuentren establecidos y conocidos por todos. No se pueden hacer mediciones y obtener resultados de las mediciones sin parámetros definidos con los cuales comparar.

Las mediciones se pueden realizar sobre muchos factores: sobre los modelos, sobre el código generado, sobre la documentación, sobre los informes que se generan, sobre costos, esfuerzo, calidad, etc.

Las mediciones se realizan al comenzar los proyectos para determinar costos de las herramientas que se requieren para la cuando los proyectos se están ejecutando para ir encaminando el desarrollo en caso de desvíos en cuanto a las pautas establecidas.

La forma de documentar las pautas varía de empresa a empresa. Algunas utilizan documentos y los comunican al resto del equipo, otros utilizan un repositorio de información donde se publican las novedades, y así, pueden existir otras formas de establecer las normas. Lo importante, es que cada vez que se define una nueva norma, ésta sea publicada.

Toda norma publicada se asume conocida por todos. La forma de hacer las publicaciones debe ser estandarizada también para que todo el equipo se acostumbre a recibir las notificaciones y la lectura de las novedades no varíe.

Muchas de las normas a estandarizar surgen de las buenas prácticas obtenidas con la ejecución de los proyectos.

Una vez definidas las normas se pueden hacer revisiones.

KPA PPQA (Aseguramiento de la Calidad de Proceso y de Producto)

Propósito: proporcionar al personal y a la gerencia una visión objetiva de los procesos y de los productos de trabajo asociados.

Notas introductorias: el área de proceso de Aseguramiento de la calidad de proceso y de producto involucra:

- Evaluar objetivamente los procesos, los productos de trabajo y los servicios ejecutados frente a las descripciones de proceso, los estándares y los procedimientos aplicables.
- Identificar y documentar las no conformidades.
- Proporcionar realimentación al equipo del proyecto y a los gerentes sobre los resultados de las actividades de aseguramiento de calidad.
- Asegurar que sean tratadas las no conformidades.

Esta KPA da soporte a la entrega de productos y servicios de alta calidad, proporcionando al equipo del proyecto a los gerentes, en todos los niveles, la apropiada visibilidad y realimentación sobre los procesos y los productos de trabajo asociados durante toda la vida del proyecto.

Las prácticas en esta KPA aseguran que los procesos planificados se implementan, mientras que las prácticas en el área de proceso de Verificación aseguran que se satisfacen los requerimientos especificados. Estas dos áreas de proceso pueden en ocasiones tratar los mismos productos de trabajo, pero desde diferentes perspectivas. Los proyectos deberían aprovechar este solapamiento con el fin de minimizar la duplicación de esfuerzos, aunque deben mantener las perspectivas separadas.

Resumen de metas y prácticas específicas

SG1 Evaluar objetivamente los procesos y los productos de trabajo.

SP 1.1 Evaluar objetivamente los procesos.

SP 1.2 Evaluar objetivamente los productos de trabajo y los servicios.

SG2 Proporcionar una visión objetiva

SP 2.1 Comunicar y asegurar la resolución de las no conformidades.

SP 2.2 Establecer registros.

Práctica específica a mapear:

SP 2.1 Comunicar y asegurar la resolución de las no conformidades

Comunicar problemas de calidad y asegurar la resolución de las no conformidades con el personal y con los gerentes.

Las no conformidades son problemas identificados en las evaluaciones que reflejan la falta de adherencia a los estándares, descripciones de proceso o procedimientos aplicables. El estado de las no conformidades proporciona una indicación de las tendencias de calidad. Los problemas de calidad incluyen no conformidades y resultados del análisis de tendencia.

Cuando las no conformidades no pueden resolverse localmente, hay que usar los mecanismos de escalado establecidos para asegurar que el nivel apropiado de gerencia pueda resolver el problema. Seguir las no conformidades hasta su resolución.

Mapeo RUP-CMMI

RUP tiene varios atributos que permiten por sí solos realizar un programa efectivo de la garantía de la calidad del proyecto.

Cada hito de RUP tiene un criterio específico para completarlo que puede servir como una base para la auditoría.

La observancia de los productos de software y actividades a los estándares aplicables, procedimientos, y requisitos es verificada objetivamente.

Este objetivo podría ser cumplido adoptando personal cualificado en la organización.

RUP proporciona las listas de verificación necesarias de revisión y plantillas de documentos que pueden ser aplicables como estándares de proyectos.

Los grupos e individuos afectados son informados de las actividades y resultados de la Garantía de la Calidad del Software (SQA).

Como se describe en la Planificación del Proyecto de Software (*Software Project Planning*), uno de los objetivos de RUP es asegurar que las expectativas de todas las partes se sincronicen y sean consistentes.

Las no conformidades que no puedan ser resueltas dentro del proyecto de software se derivan a la gestión senior.

Esto cae fuera del ámbito de RUP y es una responsabilidad de la organización.

Sin embargo, el *Change Control Process* descrito en RUP podría establecer un mecanismo por medio del cual las no conformidades podrían ser documentadas y promocionadas para su resolución.

Mapeo MDD-CMMI

En MDD se pueden definir los criterios de calidad por cada paso en el desarrollo del software, así tendríamos: calidad de los PIM (¿Qué define la calidad para los modelos PIM?), calidad de los PSM (¿Qué define la calidad para los modelos PSM?), calidad del código (¿Qué define la calidad para el código?) y por supuesto la calidad relacionada con las herramientas que se utilizan para las transformaciones (complejidad, resultados, etc.).

En sí, la calidad del producto final dependerá fuertemente de la calidad de los modelos que se usaron para su desarrollo, pero los criterios deben ser únicos en toda la organización y adoptados por todos. Por un lado, deben definirse los estándares en cada aspecto. Estos estándares deben ser conocidos por todo el equipo en la organización. Una vez que esto es conocido pueden hacerse seguimientos para validar que las normas impuestas sean cumplidas.

La organización debe conocer que se han establecido estándares de calidad. De otro modo, no se podría hacer un seguimiento y eventual alta de una no conformidad.

Las no conformidades pueden ser creadas solamente cuando las normas son conocidas y no cumplidas.

Al igual que se realiza un seguimiento de que las normas sean cumplidas, también debe realizarse un seguimiento de las no conformidades que se creen, para saber si las mismas fueron resueltas por el responsable.

Es imprescindible contar con algún mecanismo en la organización para gestionar las no conformidades (tal como se reportan los bugs de una aplicación). De este modo, es

simple el manejo de las no conformidades tanto por quien las crea como para quien las recibe y debe resolverlas.

Tanto RUP como MDD consideran que la calidad es responsabilidad de todos los miembros del equipo y no se concentra en ninguna parte de la organización.

La planificación de las tareas de la Garantía de la Calidad del Software es una responsabilidad de la organización.

KPA CM (Gestión de Configuración)

Propósito: establecer y mantener la integridad de los productos de trabajo utilizando la identificación de configuración, el control de configuración, el registro del estado de configuración y las auditorias de configuración.

Notas introductorias: el área de proceso de Gestión de configuración implica:

- Identificar la configuración de los productos de trabajo seleccionados que componen las líneas base en puntos determinados en el tiempo.
- Controlar los cambios a los elementos de configuración.
- Construir o proporcionar especificaciones para construir los productos de trabajo a partir del sistema de gestión de configuración.
- Mantener la integridad de las líneas base.
- Proporcionar a los desarrolladores, usuarios finales y clientes datos del estado exacto y de la configuración actual.

Los productos de trabajo situados bajo gestión de configuración incluyen los productos que se entregan al cliente, los productos de trabajo internos designados, los productos adquiridos, las herramientas y otros elementos que se usan para crear y describir estos productos de trabajo.

Los productos adquiridos pueden necesitar ser puestos bajo gestión de configuración tanto por el proveedor como por el proyecto. Las provisiones para llevar a cabo la

gestión de configuración deberían establecerse en los acuerdos con los proveedores. Se deberían establecer y mantener métodos para asegurar que los datos están completos y son consistentes.

Resumen de metas y prácticas específicas

SG1 Establecer líneas base

- SP 1.1 Identificar elementos de configuración.
- SP 1.2 Establecer un sistema de gestión de configuración.
- SP 1.3 Crear o liberar líneas base.

SG2 Seguir y controlar los cambios

- SP 2.1 Seguir las peticiones de cambio.
- SP 2.2 Controlar los elementos de configuración.

SG3 Establecer la integridad

- SP 3.1 Establecer registros de gestión de configuración.
- SP 3.2 Realizar auditorías de configuración.

Práctica específica a mapear:

SP 3.1 Establecer registros de gestión de configuración

Establecer y mantener los registros que describen los elementos de configuración.

Mapeo RUP-CMMI

RUP tiene dos instrumentos principales para definir cómo se mantiene la valoración del desarrollo de software de un proyecto y cómo se integran:

- El Plan de la Gestión de la Configuración (*Configuration Management Plan*).
- El Plan de Construcción de la Integración (*Integration Build Plan*).

Los productos de trabajo del software son identificados, controlados y están disponibles. El Plan de Gestión de la Configuración de RUP solicita una descripción del control de la configuración y gestión del proceso para asegurar que los productos de trabajo son identificados, controlados y están disponibles.

Los cambios para identificar productos de software están controlados.

RUP apoya que un proyecto mantiene un CCB (*Change Control Board*) y tiene un Sistema de Gestión del Cambio (*Change Management System*) para gestionar adecuadamente el coste, seguimiento e implementación del cambio.

Los grupos e individuos son informados del estado y contenido de las líneas base del software.

RUP establece que las líneas base de los requisitos, diseño e implementación, y trazabilidad entre ellas, se mantienen en formato electrónico.

Mapeo MDD-CMMI

Haciendo foco en la SP 1.1, se define como “elemento de configuración” a una entidad designada para la gestión de configuración, la cual puede consistir en múltiples productos de trabajo relacionados que forman una línea base. Esta agrupación lógica proporciona facilidad de identificación y acceso controlado. La selección de los productos de trabajo para la gestión de configuración debería basarse en criterios establecidos durante la planificación.

Resulta simple identificar elementos de configuración en un modelo MDD, dada la clara cadena de elementos que conforman la creación de un software, que van desde la concepción del producto hasta su implementación. Los elementos sobre los cuales queremos mantener una configuración, podemos tomarlos de la siguiente lista:

- Documentos y modelos de análisis de requerimientos.
- Documentos y modelos de diseño del sistema.
- PIM.
- PSM.

- Herramientas utilizadas para las transformaciones de los modelos y generación de código.
- Código generado a partir de los modelos previos.
- Resultados de las pruebas.

Una vez definidos los elementos de configuración, el paso siguiente es asignarle a cada elemento un identificador único para diferenciarlo de los elementos pertenecientes a otra configuración o línea base. Definidos los identificadores, se especifican las características más importantes de cada elemento de configuración (autor, documento o tipo de archivo y lenguaje de programación para archivos de código fuente).

Por último, se especifica cuándo cada elemento de configuración se pone bajo gestión de configuración y quién es el propietario responsable de cada elemento de configuración.

Nivel 3 – Mapeo con RUP y CMMI

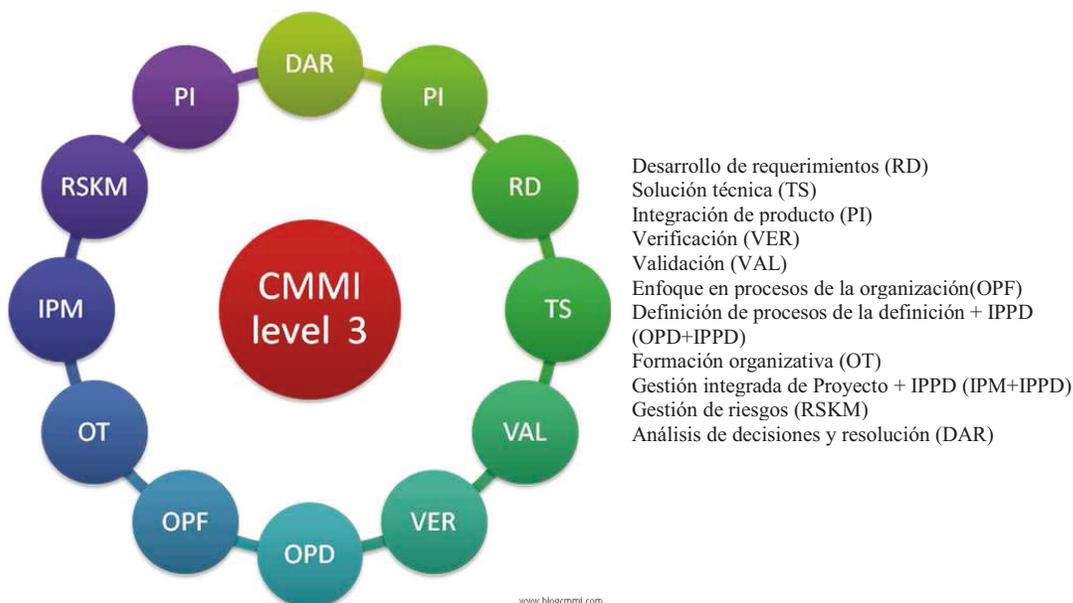


Ilustración 8: KPAs del Nivel 3

En el nivel de madurez 3, los procesos son bien caracterizados y comprendidos, y se describen en estándares, procedimientos, herramientas y métodos. El conjunto de procesos estándar de la organización, que es la base del nivel de madurez 3, se establece y mejora a lo largo del tiempo. Estos procesos estándar se usan para establecer la consistencia en toda la organización. Los proyectos establecen sus procesos definidos adaptando el conjunto de procesos estándar de la organización de acuerdo a las guías de adaptación.

Una distinción crítica entre los niveles de madurez 2 y 3 es el alcance de los estándares, descripciones de proceso y procedimientos. En el nivel de madurez 2, los estándares, descripciones de proceso y procedimientos pueden ser bastante diferentes en cada instancia específica de un proceso (p.ej., en un proyecto particular). En el nivel de madurez 3, los estándares, descripciones de proceso y procedimientos para un proyecto se adaptan para adecuarse a un proyecto particular o unidad organizativa a partir del conjunto de procesos estándar de la organización y, por tanto, son más consistentes, exceptuando las diferencias permitidas por las guías de adaptación.

Otra distinción crítica es que en el nivel de madurez 3, los procesos normalmente se describen más rigurosamente que en el nivel de madurez 2. Un proceso definido establece claramente el propósito, entradas, criterios de entrada, actividades, roles, medidas, etapas de verificación, salidas y criterios de salida. En el nivel de madurez 3, los procesos se gestionan más proactivamente utilizando una comprensión de las interrelaciones de las actividades del proceso y las medidas detalladas del proceso, sus productos de trabajo y sus servicios.

En el nivel de madurez 3, la organización debe madurar más las áreas de proceso de nivel de madurez 2. Para lograr el nivel de madurez 3, se aplican las prácticas genéricas asociadas con la meta genérica 3 que no fueron tratadas en el nivel de madurez 2. A continuación se recorren las KPA del nivel 2, y se realiza su correspondencia con RUP y MDD:

KPA RD (Desarrollo de Requerimientos)

Propósito: producir y analizar los requerimientos de cliente, de producto y de componente del producto.

Notas introductorias: este área de proceso describe tres tipos de requerimientos: de cliente, de producto y de componente de producto. Tomados en conjunto, estos requerimientos tratan las necesidades de las partes interesadas relevante, incluyendo aquellas pertinentes a las distintas fases del ciclo de vida del producto (p.ej.: criterios de pruebas de aceptación) y a los atributos del producto (p.ej.: seguridad, fiabilidad y facilidad de mantenimiento). Los requerimientos también tratan las restricciones causadas por la selección de soluciones de diseño (p.ej.: integración de productos disponibles comercialmente).

Todos los proyectos de desarrollo tienen requerimientos. En el caso de que un proyecto se enfoque en actividades de mantenimiento, los cambios al producto o a los componentes del producto se basan en los cambios a los requerimientos, al diseño o a la implementación existente. Los cambios a los requerimientos, si existen, podrían documentarse en peticiones de cambio del cliente o de los usuarios, o podrían tomar la forma de nuevos requerimientos recibidos desde el proceso de desarrollo de requerimientos. Sin importar su fuente o forma, las actividades de requerimientos, que son conducidas por los cambios a los requerimientos, se gestionan consecuentemente.

Metas y prácticas específicas

SG1 Desarrollar los requerimientos del cliente.

SP 1.1 Obtener las necesidades.

SP 1.2 Desarrollar los requerimientos del cliente.

SG2 Desarrollar los requerimientos de producto.

SP 2.1 Establecer los requerimientos de producto y de componentes del producto.

SP 2.2 Asignar los requerimientos de componentes del producto.

SP 2.3 Identificar los requerimientos de interfaz.

SG3 Analizar y validar los requerimientos.

- SP 3.1 Establecer los conceptos operativos y los escenarios.
- SP 3.2 Establecer una definición de la funcionalidad requerida.
- SP 3.3 Analizar los requerimientos.
- SP 3.4 Analizar los requerimientos para alcanzar el equilibrio.
- SP 3.5 Validar los requerimientos.

Práctica específica a mapear:

SP 2.2 Asignar los requerimientos de componentes del producto

Asignar los requerimientos para cada componente del producto.

Los requerimientos para los componentes del producto de la solución definida incluyen la asignación del rendimiento del producto; las restricciones de diseño; y el encaje, la forma y la función para cumplir los requerimientos y facilitar la producción. En los casos donde un requerimiento de nivel más alto especifique el rendimiento que será responsabilidad de dos o más componentes del producto, el rendimiento debe dividirse para la asignación única a cada componente del producto como un requerimiento derivado.

Mapeo RUP-CMMI

Para asignar los requerimientos de componentes del producto, RUP transforma las descripciones de comportamiento de los Casos de Uso en un conjunto de elementos de diseño del producto orientada a objetivos usando una aproximación para análisis y diseño.

Mapeo MDD-CMMI

Poder adaptarse a los cambios es un requerimiento clave para los negocios, y los sistemas informáticos deben ser capaces de soportarlos. Cuando usamos un proceso MDD, agregar o modificar una funcionalidad de negocios es una tarea bastante sencilla, ya que el trabajo de automatización ya está hecho. Cuando agregamos una nueva función, sólo necesitamos desarrollar el modelo específico para esa nueva función. El

resto de la información necesaria para generar los artefactos de implementación ya ha sido capturada en las transformaciones y puede ser re usada.

El mayor impacto se ve asociado si hay que rediseñar los Casos de Uso iniciales porque la funcionalidad inicial descrita cambia en su totalidad, pues esto requiere tener que crear el PIM nuevamente y de éste se derivarán los PSM y el código.

Cualquier nuevo requerimiento cambios un nuevo diseño de PIM. Este punto debe ser tratado con sumo cuidado y depende de la propia empresa. Generalmente hay un período estipulado para que se puedan definir los requerimientos nuevos. Una vez finalizada esta fase, la empresa que desarrolla el sistema puede continuar con su trabajo. Por lo tanto, si surge un nuevo requerimiento luego de finalizada esta fase, debe tener su propia planificación y tiempo de desarrollo, extra al planificado inicialmente.

En cambio, si los requerimientos surgen en las últimas etapas, significa que los primeros productos obtenidos quedan inalterados. Un requerimiento podría ser el cambio de tecnología para implementar el código (un sistema que en principio se solicitó implementar en PHP y luego se solicitó su implementación en Java). En este caso, las funcionalidades solicitadas por el cliente son las mismas pero un requerimiento sobre tecnología solamente se ve afectado en la etapa final, transparente para el cliente.

KPA TS (Solución Técnica)

Propósito: diseñar, desarrollar e implementar soluciones para los requerimientos. Las soluciones, los diseños y las implementaciones engloban productos, componentes de producto y procesos del ciclo de vida asociados al producto, individualmente o en combinación, según sea apropiado.

Notas introductorias: esta área es aplicable en cualquier nivel de la arquitectura de producto y a cada producto, componente de producto y proceso del ciclo de vida asociado al producto. En todas las áreas de proceso, donde usamos los términos de producto y componentes de producto, su significado previsto engloba también a los servicios y a sus componentes.

El área de proceso se enfoca en:

- Evaluar y seleccionar soluciones (referidas a veces como "planteamiento de diseño", "conceptos de diseño" o "diseño preliminares") que potencialmente satisfacen un conjunto apropiado de requerimientos asignados.
- Desarrollar diseños detallados para las soluciones seleccionadas (detallados en el contexto de contener toda la información necesaria para fabricar, codificar o, de otra manera, implementar el diseño como un producto o componente de producto).
- Implementar los diseños como un producto o componente de producto.

Resumen de metas y prácticas específicas

SG1 Seleccionar las soluciones de componentes de producto.

SP 1.1 Desarrollar las soluciones alternativas y establecer los criterios de selección.

SP 1.2 Seleccionar las soluciones de componentes de producto.

SG2 Desarrollar el diseño.

SP 2.1 Diseñar el producto o el componente de producto.

SP 2.2 Establecer un paquete de datos técnicos.

SP 2.3 Diseñar las interfaces usando criterios.

SP 2.4 Realizar el análisis sobre si hacer, comprar o reutilizar.

SG3 Implementar el diseño de producto.

SP 3.1 Implementar el diseño.

SP 3.2 Desarrollar la documentación de soporte de producto.

Práctica específica a mapear:

SP 1.1 Desarrollar las soluciones alternativas y establecer los criterios de selección

Desarrollar las soluciones alternativas y los criterios de selección.

Necesitan identificarse y analizarse soluciones alternativas, para permitir la selección de una solución balanceada a través de la vida de producto en términos de coste, de calendario y de rendimiento. Estas soluciones se basan en las arquitecturas propuestas de producto que tratan las cualidades críticas de producto y se extienden en un espacio de diseño de soluciones factibles.

Las soluciones alternativas engloban frecuentemente asignaciones alternativas de requerimientos para diferentes componentes de producto.

Los criterios de selección normalmente tratarían costes (p. ej., tiempo, personal y dinero), beneficios, (p. ej., rendimiento, capacidad y eficacia) y riesgos (p. ej., técnicos, de costes y de calendario).

Mapeo RUP-CMMI

Disciplinas de proceso con las que mapea RUP: Análisis y Diseño, Implementación, Gestión de Proyectos.

- RUP direcciona los objetivos centrales de formación, diseño e implementación.
- RUP proporciona mayor dirección sobre arquitectura de desarrollo y validación que CMMI.
- RUP no cubre explícitamente consideración de alternativas de diseño excepto a nivel de arquitectura.
- RUP no cubre explícitamente el uso de criterios de selección para soluciones de producto o componentes.

Los temas de negocio describen al menos dos aproximaciones para realizar la Visión del sistema, y analizarlos en términos de capacidad, riesgo de impacto, calendario y resultados económicos. Las opciones del proyecto podrían incluir diferentes bases contractuales, diferentes ciclos de vida, diferentes combinaciones entre "hacer" y "comprar" y así sucesivamente durante la Revisión de Aprobación de Proyecto, una de las opciones es seleccionada. RUP provee referencias a fuentes de riesgos y técnicas de

análisis de decisión pero no explícitamente establece los criterios a aplicar. Aunque RUP se focaliza en la arquitectura, no provee directrices sobre la selección de las componentes del producto.

Mapeo MDD-CMMI

El área de proceso de desarrollo de requerimientos suministra los requerimientos al área de proceso de Solución técnica, donde los requerimientos se convierten en la arquitectura del producto, el diseño de los componentes del producto, y los propios componentes del producto (p.ej., codificación y fabricación). Los requerimientos se suministran también al área de proceso de integración de producto, donde se combinan los componentes del producto y se verifican las interfaces para asegurar que cumplen con los requerimientos de interfaz suministrados por el área de proceso de Desarrollo de requerimientos.

Una vez definidos los requerimientos, sabemos lo que hay que hacer, está completa la etapa de análisis y los requisitos están claramente identificados. El paso siguiente, es llevar el desarrollo a una etapa más, que es su implementación, para comenzar a tener resultados tangibles del producto. Supongamos que hemos obtenidos los PIM, que son la primera etapa de transformación a partir de los modelos obtenidos de la etapa de análisis y diseño. Como paso siguiente, debemos responder los siguientes interrogantes: ¿se implementa una herramienta para realizar la transformación? O ¿se compra una herramienta disponible en el mercado que permita hacer la transformación? Por definición, MDD debe ser implementado mediante una serie de estándares industriales abiertos. Estos estándares permiten a los fabricantes de herramientas centrar su atención en su principal área de experiencia, sin tener que recrear y competir con funcionalidades implementadas por otros proveedores. Por ejemplo, una herramienta que transforma modelos no necesita incluir una funcionalidad de edición de modelos. En lugar de ello, puede usar otra herramienta de edición de modelos de otro fabricante que se ajuste a un estándar común.

KPA PI (Integración de Producto)

Propósito: ensamblar el producto a partir de sus componentes, asegurar que el producto, una vez integrado, funciona correctamente, y entregar el producto.

Notas introductorias: esta área de proceso trata la integración de los componentes de producto dentro de componente de producto más complejos o de productos más completos.

El alcance de esta área de proceso es lograr la integración del producto completo a través de un ensamblaje progresivo de los componentes, en una etapa o en etapas incrementales, de acuerdo a una secuencia y procedimientos de integración definidos. En todas las áreas de proceso, donde usamos los términos de producto y componente de producto, su significado previsto engloba también a los servicios y a sus componentes.

Resumen de metas y prácticas específicas

SG1 Preparar para la integración de producto.

SP 1.1 Determinar la secuencia de integración.

SP 1.2 Establecer el entorno de integración de producto.

SP 1.3 Establecer los procedimientos y los criterios de integración de producto.

SG2 Asegurar la compatibilidad de la interfaz.

SP 2.1 Revisar la completitud de las descripciones de la interfaz.

SP 2.2 Gestionar las interfaces.

SG3 Ensamblar los componentes de producto y entregar el producto.

SP 3.1 Confirmar la disponibilidad de los componentes de producto para su integración.

SP 3.2 Ensamblar los componentes de producto.

SP 3.3 Evaluar los componentes de producto ensamblados.

SP 3.4 Empaquetar y entregar el producto o el componente de producto.

Práctica específica a mapear:

SP 2.1 Revisar la completitud de las descripciones de la interfaz

Revisar las descripciones de la interfaz en cuanto a cobertura y completitud. Las interfaces deberían incluir, además de las interfaces de los componentes de producto, todas las interfaces con el entorno de integración de producto.

Mapeo RUP-CMMI

RUP apoya la integración general del producto.

Las interfaces son una parte crítica de la arquitectura y diseño en RUP. El primer mecanismo de revisión es la construcción arquitectónica de prototipos y la integración y el testeado de ejecutables en cada iteración de modo que las interfaces son ejercidas en un contexto más realista.

Mapeo MDD-CMMI

Un sistema es construido a partir de los modelos. Las interfaces, conexiones entre subsistemas y servicios pueden ser definidos desde los primeros modelos e ir enriqueciéndolos con el avance del proyecto, tal como define MDD la evolución del sistema. Conociendo los subsistemas, servicios requeridos y ofrecidos por cada uno se puede hacer un diseño de alto nivel y luego ir completando las definiciones e implementaciones de los mismos. De esta manera la integración es considerada desde las primeras instancias y el ensamblaje de todas las partes es probado en cada entregable.

KPA VER (Verificación)

Propósito: asegurar que los productos de trabajo seleccionados cumplen sus requerimientos especificados.

Notas introductorias: este área de proceso implica: preparación de la verificación, realización de la verificación e identificación de acciones correctivas. La verificación incluye la verificación del producto y de los productos de trabajo intermedios frente a todos los requerimientos seleccionados, incluyendo requerimientos del cliente, del producto y del componente de producto.

La verificación es inherentemente un proceso incremental, debido a que ocurre durante todo el desarrollo del producto y de los productos de trabajo, comenzando con la verificación de los requerimientos, progresando a través de la verificación de los productos de trabajo según van evolucionando y culminando en la verificación del producto finalizado.

Las áreas de proceso de Verificación y Validación son similares, pero tratan aspectos diferentes. La validación demuestra que el producto, tal como se proporciona (o tal como se proporcionará), se ajustará a su uso previsto, mientras que la verificación trata sobre si el producto de trabajo refleja apropiadamente los requerimientos especificados. En otras palabras, la verificación asegura que "se construye correctamente"; mientras que la validación asegura que "se construye la cosa correcta".

Resumen de metas y prácticas específicas

SG1 Preparar la verificación.

SP 1.1 Seleccionar los productos de trabajo a verificar.

SP 1.2 Establecer el entorno de verificación.

SP 1.3 Establecer los procedimientos y los criterios de verificación.

SG 2 Realizar revisiones entre pares.

SP 2.1 Preparar las revisiones entre pares.

SP 2.2 Llevar a cabo las revisiones entre pares.

SP 2.3 Analizar los datos de la revisión entre pares.

SG 3 Verificar los productos de trabajo seleccionados.

SP 3.1 Realizar la verificación.

SP 3.2 Analizar los resultados de la verificación.

Práctica específica a mapear:

SP 3.2 Analizar los resultados de la verificación

Analizar los resultados de todas las actividades de verificación.

Los resultados reales deben compararse con los criterios de verificación establecidos para determinar su aceptabilidad.

Los resultados del análisis se registran como evidencia de que se ha llevado a cabo la verificación.

Para cada producto de trabajo, se analizan incrementalmente todos los resultados disponibles de verificación, para asegurar que se han cumplido los requerimientos. Ya que una revisión entre pares es uno de los diferentes métodos de verificación, los datos de la revisión entre pares deberían incluirse en esta actividad de análisis para asegurar que se analizan suficientemente los resultados de la verificación.

Los informes de análisis o la documentación del método de “tal como se ejecutó” también puede indicar que los resultados malos de verificación se deben a problemas del método, problemas de criterios, o a un problema del entorno de verificación.

Mapeo RUP-CMMI

RUP provee buen soporte para las prácticas de verificación.

Las verificaciones se realizan en cada iteración.

Mapeo MDD-CMMI

Los términos verificación y validación (V&V) se refieren a los procesos de comprobación y análisis que aseguran que el software esté acorde con su especificación y cumpla las necesidades de los clientes.

Tal como se mencionó en las notas introductorias, la verificación y validación son dos conceptos distintos. La verificación determina si un producto de software de una fase cumple los requerimientos de la fase anterior. La validación determina si el software satisface los requerimientos del usuario. Sobre estas definiciones podemos afirmar que:

- La validación sólo se puede hacer con la activa participación del usuario.
- La validación establece si estamos haciendo el software correcto.
- La verificación establece si estamos haciendo el software correctamente.

Esta definición también nos sirve para la KPA siguiente VAL.

Hay distintas técnicas de V&V, entre ellas la verificación formal de programas y modelos, chequeo de modelos (model checking) y testing.

La **verificación formal** es un método de verificación estática (se analiza a través del propio código del programa, a partir de una abstracción o de una representación simbólica), en el que partiendo de un conjunto axiomático, reglas de inferencia y algún lenguaje lógico (como la lógica de primer orden), se puede encontrar una demostración o prueba de corrección de un programa. Entre las herramientas para la verificación formal de programas podemos citar a ESC/Java (Extended Static Checker for Java), desarrollada por el Compaq Systems Research Center para encontrar errores en tiempo de ejecución de programas Java por análisis estático del texto del programa. Las recientes versiones de ESC/Java están basadas alrededor de Java Modeling Language (JML).

El **model checking** es un método automático de verificación de un sistema formal. Este es descrito mediante un modelo, que debe satisfacer una especificación formal definida mediante una fórmula, a menudo escrita en alguna variedad de lógica temporal. Entre las herramientas de model checking se encuentra Java PathFinder, empleada para el bytecode ejecutable de Java.

Las técnicas de V&V comentadas anteriormente son muy complicadas y requieren tener experiencias en métodos formales, en cambio el **testing** es una alternativa práctica. Esta técnica consiste en el proceso de ejercitar un producto para verificar que satisface los requerimientos e identificar diferencias entre el comportamiento real y el comportamiento esperado (IEEE Standard for Software Test Documentation, 1983). De acuerdo a la definición anterior el testing juega un rol importante en el aseguramiento de calidad del software. Aunque cabe señalar que esta técnica no puede garantizar la ausencia de errores, sólo puede asegurar la existencia de errores.

El éxito del testing depende fuertemente de nuestra capacidad para crear pruebas útiles para descubrir y eliminar problemas en todas las etapas del desarrollo; esto requiere diseñarlas sistemáticamente desde la fase de análisis. Los casos de prueba determinan el tipo y alcance de la prueba. Dada la complejidad del proceso, su automatización (aunque sea parcial) es indispensable.

En el contexto de MDD, el Model-Driven Testing (MDT) es una de las propuestas más recientes que enfrentan este problema. El MDT es una forma de prueba de caja negra que utiliza modelos estructurales y de comportamiento descritos por ejemplo, con el lenguaje estándar UML, para automatizar el proceso de generación de casos de prueba. MDT reduce el costo de mantenimiento de los tests, ya que los desarrolladores sólo tienen que regenerarlos para reflejar los cambios del modelo que afectan a todas las pruebas. Las herramientas de MDT refuerzan la comunicación del equipo porque los modelos proporcionan una vista clara y unificada del sistema y de las pruebas.

KPA VAL (Validación)

Propósito: demostrar que un producto o componente de producto se ajusta a su uso previsto cuando se sitúa en su entorno previsto.

Notas introductorias: las actividades de validación pueden aplicarse a todos los aspectos del producto en cualquiera de sus entornos previstos, tales como los servicios de operación, de formación, de fabricación, de mantenimiento y de soporte. Los métodos

empleados para lograr la validación pueden aplicarse a los productos de trabajo, así como al producto y a los componentes de producto. Los productos de trabajo (p.ej.: requerimientos, diseños y prototipos) deberían seleccionarse en base a cuáles son los mejores predictores acerca de cómo el producto y el componente de producto satisfarán las necesidades del usuario, y así la validación se realiza temprana e incrementalmente en todo el ciclo de vida del producto.

Resumen de metas y prácticas específicas

SG1 Preparar la validación.

SP 1.1 Seleccionar los productos a validar.

SP 1.2 Establecer el entorno de validación.

SP 1.3 Establecer los procedimientos y los criterios de validación.

SG 2 Validar el producto o los componentes del producto.

SP 2.1 Realizar la validación.

SP 2.2 Analizar los resultados de la validación.

Práctica específica a mapear:

SP 1.1 Seleccionar los productos a validar

Seleccionar los productos y los componentes de producto a validar y los métodos de validación que serán usados para cada uno.

Los productos y los componentes de producto se seleccionan para ser validados en base a su relación con las necesidades del usuario.

Para cada componente de producto, debería determinarse el alcance de la validación (p. ej., comportamiento operacional, mantenimiento, formación e interfaz de usuario).

Mapeo RUP-CMMI

RUP comienza muy temprano con las tareas de validación mediante la revisión de los Casos de Uso. Esta revisión la realiza con los usuarios y se repite en cada iteración.

La aceptación del producto es definida como parte del Plan de Proyecto.

La validación se realiza con el fin de identificar posibles defectos lo antes posible.

Mapeo MDD-CMMI

Los modelos además de ser entidades descriptivas, son entidades productivas que van sufriendo sucesivas transformaciones a lo largo del proceso de desarrollo. Por ello, además de analizar los modelos en sí mismos debemos también analizar sus transformaciones.

Actualmente, los lenguajes de transformación se enfocan principalmente en cómo expresar las transformaciones pero les prestan menos atención a cómo validarlas. En general, la validación de la transformación puede incluir propiedades como correctitud sintáctica de la transformación con respecto a su lenguaje de especificación y correctitud sintáctica de los modelos producidos por la transformación. Pero pocas propuestas tratan la consistencia semántica de la transformación, es decir la preservación de la correctitud del modelo destino con respecto al modelo origen correspondiente.

Sin embargo, el problema de la validación semántica de transformación de modelos no es un nuevo desafío originado por la filosofía de MDD. La idea de desarrollo de software basada en pasos incrementales, se remonta a los días de la teoría de refinamiento de Dijkstra.

Esta teoría cuenta con gran popularidad y desarrollo en la comunidad de los métodos formales, donde la mayoría de los esquemas de refinamiento se basan en el principio de 'sustitución'. Normalmente el refinamiento se verifica demostrando que el sistema concreto simula al sistema abstracto.

Adaptar aquellas definiciones de refinamiento bien fundamentadas hacia la validación de la transformación de modelos se torna un gran desafío. En se describe una propuesta formal ágil para la validación semántica del refinamiento de modelos. Los autores definen las estructuras de refinamiento en términos de UML/OCL, resultando equivalentes a las estructuras empleadas en los lenguajes formales. La ventaja de esta

propuesta reside en que se reemplaza la aplicación de lenguajes matemáticos ajenos al área de modelado, que normalmente no son aceptados por los modeladores, por lenguajes estándar que han sido ampliamente adoptados.

KPA OPF (Enfoque de Procesos de la Organización)

Propósito: planificar, implementar y desplegar las mejoras de procesos de la organización, basadas en una comprensión completa de las fortalezas y debilidades actuales de los procesos y de los activos de proceso de la organización.

Notas introductorias: los procesos de la organización incluyen todos los procesos usados por la organización y sus proyectos. Las mejoras candidatas a los procesos y a los activos de proceso de la organización se obtienen de diferentes fuentes, incluyendo la medición de procesos, las lecciones aprendidas en la implementación de procesos, los resultados de las actividades de evaluación de productos, los resultados de evaluación comparativa (*benchmarking*) frente a los procesos de otras organizaciones, y recomendaciones desde otras iniciativas de mejora en la organización.

La mejora de procesos ocurre en el contexto de las necesidades de la organización y se usa para tratar los objetivos de la organización. La organización promueve la participación en actividades de mejora de procesos de aquellos que realizarán el proceso. La responsabilidad de facilitar y gestionar las actividades de mejora de procesos de la organización, incluyendo la coordinación de la participación de otros, se asigna normalmente a un grupo de procesos. La organización proporciona el compromiso a largo plazo y los recursos requeridos para patrocinar a este grupo y para asegurar el despliegue eficaz y oportuno de las mejoras.

Resumen de metas y prácticas específicas

SG1 Determinar las oportunidades de mejora de procesos.

SP 1.1 Establecer las necesidades de procesos de la organización.

SP 1.2 Evaluar los procesos de la organización.

SP 1.3 Identificar las mejoras de procesos de la organización.

SG 2 Planificar e implementar las mejoras de procesos.

SP 2.1 Establecer planes de acción de procesos.

SP 2.2 Implementar los planes de acción de procesos.

SG 3 Desplegar los activos de proceso de la organización e incorporar las lecciones aprendidas.

SP 3.1 Desplegar los activos de proceso de la organización.

SP 3.2 Desplegar los procesos estándar.

SP 3.3 Monitorizar la implementación.

SP 3.4 Incorporar las experiencias relativas al proceso en los activos de proceso de la organización.

Práctica específica a mapear:

SP 1.2 Evaluar los procesos de la organización

Evaluar los procesos de la organización periódicamente y según sea necesario para mantener una comprensión de sus fortalezas y debilidades.

Las evaluaciones de proceso pueden realizarse por las siguientes razones:

Para identificar los procesos que deberían mejorarse.

- Para confirmar el progreso y hacer visibles los beneficios de la mejora de proceso.
- Para satisfacer las necesidades de una relación cliente-proveedor.
- Para motivar y facilitar la aceptación

La aceptación obtenida durante una evaluación de procesos puede deteriorarse significativamente si no es seguida por un plan de acción basado en la evaluación.

Mapeo RUP-CMMI

RUP principalmente es una herramienta a nivel proyecto. Las organizaciones que deseen utilizar RUP para la administración de sus proyectos, deberán añadir workflows para direccionar sus procesos al enfoque de procesos de organización.

Mapeo MDD-CMMI

Esta KPA cubre aspectos que van más allá de la metodología implementada para el desarrollo de software. Toda organización debe cumplir con procesos estandarizados donde enmarcar el desarrollo de sus proyectos. Generalmente, para estandarizar los desarrollos se determinan workflows generales que consideran las actividades que van desde la concepción de un nuevo proyecto hasta su entrega. Pero, este workflow involucra otras áreas de la empresa, y no solamente aquellas involucradas con el desarrollo en sí del sistema.

Una de las propuestas sería tener revisiones en cada una de las fases definidas por MDD. Si se incluyen revisiones antes de comenzar con el desarrollo, se podrían detectar si en la organización existen soluciones similares o bien, si para el cliente ya se han entregado soluciones relacionadas y poder enmarcar el nuevo desarrollo dentro de algún contexto ya existente. Si se incluyen revisiones luego de la entrega del código, se pueden realizar reuniones de retrospectiva con el equipo e informar las soluciones encontradas para este sistema de modo que las mismas sean adoptadas para futuros sistemas.

KPA OPD+IPPD (Definición de Procesos de la Definición + IPPD)

Propósito: el propósito de la Definición de procesos de la organización (OPD) es establecer y mantener un conjunto usable de activos de proceso de la organización y de estándares del entorno de trabajo.

Para IPPD, el área de proceso de Definición de procesos de la organización + IPPD cubre también el establecimiento de reglas y guías de la organización que permiten llevar a cabo el trabajo usando equipos integrados.

Notas introductorias: los activos de proceso de la organización permiten conseguir un rendimiento de procesos consistente en toda la organización y proporcionan una base para beneficios acumulativos a largo plazo para la organización.

La biblioteca de activos de proceso de la organización es una colección de elementos que mantiene la organización para ser usados por el personal y los proyectos de la misma. Esta colección de elementos incluye descripciones de procesos y de elementos de proceso, descripciones de modelos del ciclo de vida, guías de adaptación de procesos, documentación relativa a procesos y datos. La biblioteca de activos de proceso de la organización da soporte al aprendizaje y la mejora de procesos, al permitir compartir las mejores prácticas y las lecciones aprendidas en la organización.

Resumen de metas y prácticas específicas

SG1 Establecer los activos de proceso de la organización.

SP 1.1 Establecer los procesos estándar.

SP 1.2 Establecer las descripciones de los modelos de ciclo de vida.

SP 1.3 Establecer los criterios y las guías de adaptación.

SP 1.4 Establecer el repositorio de medición de la organización.

SP 1.5 Establecer la biblioteca de activos de proceso de la organización.

SP 1.6 Establecer los estándares del entorno de trabajo.

SG 2 Facilitar la gestión IPPD.

SP 2.1 Establecer los mecanismos para otorgar la autoridad.

SP 2.2 Establecer las reglas y las guías para los equipos integrados.

SP 2.3 Equilibrar entre las responsabilidades en el equipo y en la organización de origen.

Práctica específica a mapear:

SP 1.2 Establecer las descripciones de los modelos de ciclo de vida

Establecer y mantener las descripciones de los modelos de ciclo de vida aprobados para su uso en la organización.

Los modelos de ciclo de vida se pueden desarrollar para una variedad de clientes o de situaciones, ya que un modelo de ciclo de vida puede no ser apropiado para todas las situaciones. Los modelos de ciclo de vida se usan frecuentemente para definir las fases del proyecto. La organización también puede definir modelos de ciclo de vida diferentes para cada tipo de producto y servicio que entrega.

Mapeo RUP-CMMI

La información complementaria sugiere que RUP podría ser el proceso de normalización de una organización.

Las organizaciones deberían poder describir muchos tipos de ciclos de vida. RUP sugiere que puede haber más de uno en toda la organización, uno por cada tipo diferente de desarrollo.

Mapeo MDD-CMMI

Una organización puede realizar una combinación de metodologías e incluso, adoptar buenas prácticas de MDD a medida que adquiere experiencia en su uso. Es habitual que el proceso se vaya adaptando con el tiempo, porque a medida que transcurren los proyectos, se van adquiriendo buenas prácticas que luego son adoptadas como parte del flujo de trabajo. Puede ser el caso de herramientas para transformación o modelos base que se utilizaron en varios proyectos y se detectó que son lo suficientemente independientes de cada proyecto como para considerarlo dentro del flujo de trabajo como una actividad estándar a incluir. Cada proyecto particular ampliará esos modelos o documentos base con información propia del dominio.

Parte del proceso, tal como se definió anteriormente es la planificación de tareas y su seguimiento. También se pueden incluir las reuniones de retrospectiva al finalizar un proyecto (de las cuales se obtienen buenas prácticas), reuniones semanales sobre avance de todos los proyectos que se estén desarrollando, documentación a generar

durante el proyecto y la posterior publicación de los mismos con previa revisión del área de gestión de calidad de la empresa, entre otras actividades.

Toda ampliación de un proceso **debe** ser documentado. ¿Por qué debe ser documentado? Porque si este proceso no es conocido por todo los participantes y si no existe un registro del proceso, no hay forma de hacer un seguimiento y de dar de alta no conformidades cuando algo se desvía de lo estandarizado.

Lo más común es que una empresa ya haya adoptado alguna metodología (like RUP o ágil) e incorpore el proceso MDD. Esta adaptación es la que debería documentarse.

KPA OT (Formación Organizativa)

Propósito: desarrollar las habilidades y el conocimiento de las personas para que puedan realizar sus roles eficaz y eficientemente.

Notas introductorias: formación organizativa incluye la formación para dar soporte a los objetivos del negocio de la organización y para cumplir las necesidades tácticas de formación identificadas por proyectos y grupos de soporte. Las necesidades específicas de formación identificadas por proyectos y grupos de soporte individuales se tratan a nivel de proyecto y de grupo de soporte, y están fuera del alcance de Formación organizativa. Los proyectos y grupos de soporte son responsables de identificar y de tratar sus necesidades específicas de formación.

Un programa de formación organizativa implica:

- Identificar las necesidades de formación de la organización.
- Obtener y proporcionar formación para tratar esas necesidades.
- Establecer y mantener la capacidad de formación.
- Establecer y mantener los registros de formación.
- Evaluar la eficacia de la formación.

Resumen de metas y prácticas específicas

SG1 Establecer una capacidad de formación organizativa.

SP 1.1 Establecer las necesidades de formación estratégicas.

SP 1.2 Determinar qué necesidades de formación son responsabilidad de la organización.

SP 1.3 Establecer un plan táctico de formación organizativa.

SP 1.4 Establecer la capacidad de formación.

SG 2 Proporcionar la formación necesaria.

SP 2.1 Impartir la formación.

SP 2.2 Establecer los registros de formación.

SP 2.3 Evaluar la eficacia de la formación.

Esta KPA cae fuera de cualquier metodología que adopte una organización.

Por tanto, independientemente de la metodología, una organización puede organizar planes de formación interna para su equipo. La formación puede ser variada: una organización puede contratar organismos externos que se dediquen y sean especialistas en un tema particular y también puede generar la formación desde dentro de la organización, en base a la experiencia de trabajo que perciben los altos cargos. Las formas de detectar el conocimiento actual de su equipo puede ir desde las evaluaciones personales pero, el valor más importante se obtiene del trabajo diario donde se observan los resultados a corto plazo.

Este es un punto a tener en cuenta por las organizaciones ya que generalmente los equipos son predispuestos a recibir formación de cualquier tipo, no solamente la formación de temas relacionados al trabajo que realice.

La forma de documentar las formaciones que se tomen varía de organización en organización según cómo se maneje la misma a la hora de guardar sus activos.

Toda documentación, incluyendo el material de formación, debe ser guardada de tal manera que su acceso y búsqueda resulte simple. Por cada material de formación que se almacene es importante registrar el tema, temario y la fecha. De este modo se puede mantener un ciclo de versiones del mismo material e ir mejorando con cada iteración.

KPA IPM+IPPD (Gestión Integrada de Proyecto + IPPD)

Propósito: el propósito de la Gestión integrada de proyecto (IPM) es establecer y gestionar el proyecto y la involucración de las partes interesadas relevantes de acuerdo a un proceso integrado y definido que se adapta a partir del conjunto de procesos estándar de la organización.

Para IPPD, la Gestión integrada de proyecto + IPPD también cubre el establecimiento de una visión compartida para el proyecto y el establecimiento de equipos integrados que llevarán a cabo los objetivos del proyecto.

Notas introductorias: la gestión integrada de proyecto implica:

- Establecer el proceso definido del proyecto al inicio del mismo, mediante la adaptación del conjunto de procesos estándar de la organización.
- Gestionar el proyecto utilizando el proceso definido del proyecto.
- Establecer el entorno de trabajo para el proyecto, basándose en los estándares del entorno de trabajo de la organización.
- Utilizar y contribuir a los activos de proceso de la organización.
- Permitir que las inquietudes de las partes interesadas relevantes sean identificadas, consideradas, y, cuando sea apropiado, tratadas durante el desarrollo del producto.
- Asegurar que las partes interesadas relevantes realizan sus tareas de una forma coordinada y oportuna para tratar los requerimientos del producto y de los componentes del producto, los planes, los objetivos, los problemas y los riesgos, para satisfacer sus compromisos y para identificar, seguir y resolver los problemas de coordinación.

La Gestión integrada de proyecto + IPPD también implica:

- Establecer una visión compartida del proyecto.
- Establecer equipos integrados que se encarguen de cumplir los objetivos del proyecto.

Resumen de metas y prácticas específicas

SG1 Utilizar el proceso definido del proyecto.

SP 1.1 Establecer el proceso definido del proyecto.

SP 1.2 Utilizar los activos de proceso de la organización para planificar las actividades del proyecto.

SP 1.3 Establecer el entorno de trabajo del proyecto.

SP 1.4 Integrar los planes.

SP 1.5 Gestionar el proyecto utilizando los planes integrados.

SP 1.6 Contribuir a los activos de proceso de la organización.

SG2 Coordinar y colaborar con las partes interesadas relevantes.

SP 2.1 Gestionar la involucración de las partes interesadas.

SP 2.2 Gestionar las dependencias.

SP 2.3 Resolver los problemas de coordinación.

SG3 Aplicar los principios de IPPD.

SP 3.1 Establecer la visión compartida del proyecto.

SP 3.2 Establecer la estructura del equipo integrado.

SP 3.3 Asignar los requerimientos a los equipos integrados.

SP 3.4 Establecer los equipos integrados.

SP 3.5 Asegurar la colaboración entre los equipos interconectados.

Práctica específica a mapear:

SP 1.1 Establecer el proceso definido del proyecto

Establecer y mantener el proceso definido del proyecto a lo largo de la vida del proyecto desde su inicio.

El proceso definido del proyecto consiste en procesos definidos que forman un ciclo de vida integrado y coherente para el proyecto.

Mapeo RUP-CMMI

Para el desarrollo de un proyecto, se evaluará el desarrollo de software actual de la organización y se tomarán los procesos y herramientas que soporten al proyecto en cuestión. Esto es menos formal que tener un “proceso organizacional estándar” compuesto por un conjunto de ciclos de vida y procesos para elegir una completa adaptación de los guidelines.

Mapeo MDD-CMMI

En MDD el proceso de desarrollo se encuentra bien definido (tal como lo muestra la Ilustración 5) y para cada nuevo proyecto en la organización, se seguirán los mismos pasos aunque siempre pueden surgir mejoras en algún paso particular. Estas mejoras surgirán de las “buenas prácticas” obtenidas de las retrospectivas de proyectos anteriores finalizados.

Independientemente de la metodología que se implemente, es fundamental no desviarse del plan definido al comienzo, de modo que todo lo que se decida, documente y planifique sirva de guía durante el desarrollo del proyecto. Si esto no se cumpliera seguramente el proyecto terminaría en el fracaso. Es por esto, que la fase inicial debe ser planificada y lleva un determinado tiempo realizarla (dependiendo de la complejidad del proyecto), pero una planificación bien realizada hará que el proyecto termine en el tiempo estimado y cumpliendo con los requisitos analizados. Esta etapa es tal vez la más importante

KPA RSKM (Gestión de Riesgos)

Propósito: identificar los problemas potenciales antes de que ocurran para que las actividades de tratamiento de riesgos puedan planificar.

Notas introductorias: la gestión de riesgos es un proceso continuo, orientado a evaluar el futuro, y una parte importante de la gestión. La gestión de riesgos debería tratar los aspectos que podrían poner en peligro el logro de los objetivos críticos. Una aproximación de gestión de riesgos continua se aplica para anticipar y mitigar eficazmente los riesgos que puedan tener un impacto crítico sobre el proyecto.

La gestión de riesgos eficaz incluye la identificación temprana y agresiva de cada riesgo a través de la colaboración y la involucración de las partes interesadas relevantes, tal y como se describió en el plan para la involucración de las partes interesadas tratado en el área de proceso de Planificación de proyecto. Es necesario un fuerte liderazgo entre las partes interesadas relevantes para establecer un entorno para la libre y abierta divulgación y discusión de los riesgos.

La gestión de riesgos debe considerar fuentes tanto internas como externas para riesgos de coste, de calendario y de rendimiento, así como de otros tipos. La detección temprana y agresiva del riesgo es importante porque normalmente es más fácil, menos costoso y menos perjudicial hacer los cambios y corregir los esfuerzos de trabajo durante las fases más tempranas del proyecto, en lugar de en fases posteriores.

Resumen de metas y prácticas específicas

SG1 Preparar la gestión de riesgos.

SP 1.1 Determinar las fuentes y las categorías de los riesgos.

SP 1.2 Definir los parámetros de los riesgos.

SP 1.3 Establecer una estrategia de gestión de riesgos.

SG2 Identificar y analizar los riesgos.

SP 2.1 Identificar riesgos.

SP 2.2 Evaluar, categorizar y priorizar los riesgos.

SG3 Mitigar los riesgos.

SP 3.1 Desarrollar los planes de mitigación de riesgo.

SP 3.2 Implementar los planes de mitigación de riesgo.

Práctica específica a mapear:

SP 2.1 Identificar riesgos

Identificar y documentar los riesgos.

Deberían considerarse los riesgos particulares asociados con la realización del proyecto usando equipos integrados, tales como los riesgos asociados con la pérdida de coordinación entre equipos o dentro del equipo.

La identificación de problemas potenciales, peligros, amenazas y vulnerabilidades que podrían afectar negativamente a los esfuerzos de trabajo o a los planes es la base para una gestión de riesgos sólida y con éxito. Los riesgos deben identificarse y describirse de manera comprensible antes de que puedan analizarse y gestionarse apropiadamente.

Los riesgos se documentan en una declaración concisa que incluye el contexto, las condiciones y las consecuencias de la ocurrencia de riesgo.

Mapeo RUP-CMMI

RUP es un proceso enfocado en los riesgos. Las actividades iniciales de RUP son realizadas para mitigar los altos riesgos y hacer frente al trabajo más difícil en las primeras etapas.

Se documentan los riesgos identificados.

Mapeo MDD-CMMI

El reuso de artefactos ya probados incrementa la confianza en el desarrollo de nuevas funcionalidades y reduce los riesgos ya que los temas técnicos han sido previamente resueltos.

Como parte del plan de trabajo, se especifican los riesgos que pueden presentarse durante el desarrollo del proyecto y se definen las consecuencias de tales riesgos. Esta definición se revisa en cada seguimiento que se realice del proyecto. Algunos riesgos serán eliminados de la lista y otros nuevos podrían ser incluidos.

La idea es estar preparado ante eventuales riesgos que puedan aparecer.

Independientemente de la metodología que se utilice y del tipo de proyecto que se esté abordando, constituir un plan de riesgos ayuda a definir los casos en los cuales pueden producirse desvíos en los caminos principales de ejecución. Estar preparado ante estos eventuales hechos ayuda a la toma de decisiones rápidamente, ya que éstas han sido planteadas con tiempo.

KPA DAR (Análisis de Decisiones y Resolución)

Propósito: analizar las decisiones posibles utilizando un proceso de evaluación formal que evalúa alternativas identificadas frente a criterios establecidos.

Notas introductorias: esta área implica establecer guías para determinar qué problemas deberían estar sujetos a un proceso de evaluación formal y después aplicar los procesos de evaluación formal a estos problemas.

Un proceso de evaluación formal es un enfoque estructurado para evaluar soluciones alternativas frente a criterios establecidos con el fin de determinar una solución recomendada para tratar un problema. Un proceso de evaluación formal implica las siguientes acciones:

- Establecer los criterios para la evaluación de alternativas.
- Identificar soluciones alternativas.
- Seleccionar los métodos para la evaluación de las alternativas.
- Evaluar las soluciones alternativas utilizando los criterios y métodos establecidos.
- Seleccionar las soluciones recomendadas a partir de las alternativas en base a los criterios de evaluación.

Resumen de metas y prácticas específicas

SG1 Evaluar las alternativas

SP 1.1 Establecer guías para el análisis de decisiones.

SP 1.2 Establecer criterios de evaluación.

SP 1.3 Identificar soluciones alternativas.

SP 1.4 Seleccionar métodos de evaluación.

SP 1.5 Evaluar alternativas.

SP 1.6 Seleccionar soluciones.

Mapeo RUP-CMMI

Fuera del alcance de RUP

Mapeo MDD-CMMI

También cae fuera de MDD ya que son cuestiones a tratar por áreas de la empresa que no están directamente involucradas con una metodología o proceso de desarrollo sino con la toma de decisiones y seguimiento a nivel gerencial. Son áreas across a todas las áreas de una empresa (puede ser responsabilidad de una PMO).

Los problemas con los que se puede enfrentar una empresa varían, es decir no es posible definir a priori una lista de problemas con resoluciones alternativas. Para algunos casos si, dado que las alternativas de solución se adquieren con la misma práctica. Por ejemplo, soluciones a aplicar ante los desvíos de un proyecto, ante la falta de recursos, etc.

Los problemas que tienen que ver directamente con el proceso de desarrollo de software se pueden minimizar con un plan definido de antemano para realizar una planificación y posterior seguimiento ya que los posibles problemas podrán ser detectados con tiempo.

Capítulo 6: *Trabajos Relacionados y Futuros*

¿Cuáles trabajos se relacionan con la presente tesina?

Para la elaboración de la presente tesina se ha investigado sobre los dos grandes conceptos: MDD y CMMI, referencias que se encontrarán en la sección de bibliografía. Rescato de la investigación realizada los siguientes títulos de trabajos que tienen relación en cuanto a la integración con CMMI:

- “*The Rational Unified Process and the Capability Maturity Model – Integrated System/Software Development*”: es un trabajo publicado en el sitio oficial de CMMI y trata sobre el mapeo de la metodología RUP para realizar la certificación en CMMI nivel 3. En esta tesina se ha utilizado este trabajo para realizar el mapeo entre RUP y CMMI como introducción al mapeo entre MDD y CMMI para cada KPA. La versión online puede encontrarse en: <http://www.sei.cmu.edu/library/assets/rup.pdf>
- “*Introduction to the MDD Maturity Model - Buenos Aires – INTI – June 2007*”: este trabajo trata sobre la evolución del modelo MDD a través de los 5 niveles de CMMI, describiendo los pasos en la adopción de MDD, donde cada paso describe un conjunto coherente de prácticas de ingeniería y gestión en el enfoque MDD que sea compatible con CMMI.
- “Proceso de desarrollo con UML y el modelo CMM”: el objetivo de este trabajo es presentar la **adecuación** de las técnicas y métodos de **UML** (Unified Modeling Language) **al proceso de mejora** de una organización, tomando como modelo el **CMM** (Capability Maturity Model).

¿Qué otras investigaciones podrían realizarse tomando como base esta tesina?

Siguiendo con la línea de investigación pienso en dos posibles futuros trabajos a realizar, los cuales entitularía como:

- “Evolución de una empresa que implementa MDD a través de los niveles de CMMI”
- “Adaptación de una empresa que implementa RUP y ha certificado en CMMI en la inclusión de MDD”

La primera propuesta consistiría en realizar una investigación y descripción de una empresa que ha incluido MDD en la ejecución de los proyectos y desea certificarse en CMMI. El trabajo consistiría en realizar un seguimiento de la empresa a través de los niveles de CMMI tomando un caso base de ejemplo. Es decir, podría definirse un ejemplo de una empresa que desarrolla cierto tipo de sistema, considerando una determinada cantidad de gente, con cierta organización estructural y una descripción de su trabajo ad-hoc antes de la inclusión de MDD. Luego, ir describiendo paso a paso cómo sería la evolución a través de las KPA de nivel 2 y 3. Además de los puntos de conexión entre MDD y CMMI realizado en la presente tesina, habría que cumplimentar con el resto de las prácticas específicas de cada meta definida por las KPA de nivel 2 y 3.

La segunda propuesta sería dar un ejemplo evolutivo de una empresa de gran envergadura que ya ha certificado en CMMI e implementa RUP y desea incorporar MDD a su proceso. Este caso implica que la empresa ya satisface cada práctica específica de todas las KPA de ambos niveles, entonces el trabajo consistiría en la adaptación del proceso involucrando las tareas de MDD sin perder el mapeo con CMMI. Es claro que esto sería una propuesta a implementar a largo plazo ya que la inclusión de MDD en una empresa que se encuentra organizada mediante RUP puede ser caótica y sobre todo cuando un proyecto se encuentra en pleno desarrollo.

Capítulo 7: *Conclusión*

Sobre la calidad en las empresas

Cada día más las empresas se preocupan y dedican gran parte de su trabajo en mejorar la productividad y escalar posiciones entre las organizaciones desarrolladoras de software para ubicarse entre las primeras que ofrezcan un servicio con la calidad esperada por los clientes.

En el mercado, cuando se deben contratar o subcontratar los servicios de una empresa, la calidad es uno de los factores que se evalúa y tiene gran peso sobre empresas que no cuentan con certificaciones internacionales.

Esta tendencia de los últimos tiempos hace que las nuevas organizaciones tomen la calidad como uno de los puntos a incluir en la definición de su misión.

Las organizaciones más antiguas con cientos de empleados que quieran incorporar estándares de calidad, el tiempo de adopción es más largo y seguramente los interesados puedan desanimarse al ver lo lejano que se encuentran del camino ideal, esto es porque están acostumbrados a una forma de trabajo distinta a la que propone un estándar. Hay que pensar en estos casos que solamente llevará más tiempo, pero el objetivo se alcanzará si hay predisposición de todos los involucrados.

Sucede también que se piensa en los estándares como procesos complejos a implementar, por la gran cantidad de puntos que definen. Pero, esto no debe ser motivo para desanimarse ya que cuando uno empieza a leer ese detalle cae en cuenta que muchas de las cosas que el estándar define ya se están realizando en la organización, solamente que no se les estaba dando un nombre formal. O bien, para muchos puntos se puede concluir que la organización va encaminada a cumplirlos pero necesita trabajar un poco más para cumplir 100% con esa definición.

El mantenimiento de la calidad puede llevarse a cabo si la organización se concientiza en que debe cambiar su manera de trabajo, un trabajo que durante años se realizó, se adoptó y se asumió como el mejor.

Todo el personal involucrado debe realizar su parte del trabajo con la inclusión de calidad y tiene la responsabilidad de tener influencia positiva sobre el resto del trabajo, de manera que unos y otros se apoyen y maduren juntos en el trabajo diario. Esto es muy importante ya que genera la motivación de realizar un trabajo de calidad a diario y para los nuevos integrantes de la empresa esto es un ejemplo a seguir. Todo se aprende y se influye sobre el resto, los buenos y malos ejemplos. Por esto es muy fundamental concientizar a toda la empresa de la importancia que tiene la calidad sobre el trabajo para que no sea tomado como algo extra a realizar sino como parte del objetivo a alcanzar.

Una empresa desorganizada que quiera ser una que cumpla con un estándar de calidad, debe comprender que no es un trabajo que se realiza de un día a otro, ni de una semana a otra ni de un mes a otro. Lleva mucho tiempo (meses y hasta años) conducir el trabajo organizacional para que ésta pueda ser evaluada y consecuentemente certificada.

Para las empresas que buscan certificarse, se puede hacer una división entre aquellas que buscan una certificación por algún requisito ocasional y aquellas donde lograr una certificación implica seguir buscando mejoras y seguir incorporando estándares en el trabajo.

En el primer caso, las empresas buscan una certificación por exigencia del mercado pero no representa una motivación propia. En estos casos el mantenimiento es sumamente costoso porque es una tarea que se debe hacer bajo obligación para poder recertificarse y seguir manteniendo el sello de “empresa con calidad”.

El segundo caso es el ideal. Donde la motivación y el trabajo en equipo hace que la empresa siempre quiera buscar mejoras y sobresalir por las demás. Luego de alcanzar una certificación de calidad pueden focalizarse en ciertas tareas y aplicar estándares y

buscar certificaciones de tareas concretas (por ejemplo eSCM-SP para el trabajo de outsourcing).

Para las empresas que incluyan calidad, es fundamental crear un área que monitoree periódicamente el trabajo y que además, con su trabajo diario pueda estar en contacto con el resto de las áreas de la empresa y con las personas que la conforman detectando posibles mejoras a incluir, y que la calidad sea un trabajo cotidiano e incremental.

Cada área en una empresa se focaliza en una parte específica del desarrollo de un sistema y muchas veces incluir tareas de calidad lo perciben como algo tedioso y no encuentran justificaciones para dedicarle valioso tiempo a estas tareas. Es aquí cuando la alta gerencia debe plantear a todo el equipo la importancia que tiene la calidad y cómo afecta en las metas gerenciales. Cuando tales tareas deben incluirse sin saber en qué afecta y de qué manera, entonces, el mantener calidad resulta una tarea pesada para la mayoría, y generalmente el resultado de las mismas poco aporta al trabajo gerencial.

Obtener una certificación internacional además de estampar a la empresa con un sello de calidad ayuda enormemente al trabajo diario, ya que todas las tareas son ordenadas y seguidas de manera que los desvíos y errores se aminoren con el tiempo.

Apuesto a la inclusión de la calidad en la empresa, incluso si la misma no está próxima a certificarse puede incluir las definiciones de las normas para aplicar mejores prácticas y con el tiempo ir encaminando a la empresa para llegar a la meta de la certificación.

La elección de CMMI en este trabajo se debe a que la mayoría de las empresas buscan una certificación ISO, por costumbre, por práctica, por cantidad de empresas certificadoras, y otras cuestiones. CMMI es relativamente nuevo y no es tan común que las empresas certifiquen en este modelo de madurez. Seguramente con el correr de los años será tan común certificar en ISO como en CMMI.

Sobre los modelos

En mis años de experiencia trabajando con modelos y diseños conocí la importancia que tiene la documentación de los mismos para la comunicación entre personas de distintas áreas y con distintos intereses sobre el mismo sistema. Además, resulta de gran utilidad el contar con una buena documentación para cambios y mantenimiento futuro. El utilizar para esta documentación el estándar UML es una gran ventaja ya que se utiliza un idioma comprendido por todos los involucrados, y permite que todo elemento sea definido una sola vez y sin ambigüedades.

La realidad es que no todos son amigos de los documentos, la mayoría prefiere poner manos a la obra directamente, pero si esto se realiza así hay muchas decisiones que se van a estar omitiendo y detalles que no serán tenidos en cuenta. El resultado será un sistema que inicialmente cumpla con lo esperado, pero con el paso del tiempo los puntos no contemplados saldrán a la luz y serán problemas mayores a solucionar. Todo esto se podría haber evitado siguiendo los documentos que registran todas las tomas de decisiones y las validaciones con los usuarios.

La mayoría de las veces el desarrollo de un sistema consiste en generar un modelo mental, poco documentado y volcar directamente todo el conocimiento a líneas de código. Por más que el diseño se esté realizando, no se realiza una documentación fuerte que sustente todo lo que se está codificando.

En estos casos resulta difícil y muchas veces imposible su mantenimiento si es que el sistema es grande, el cambio de personas es poco recomendable ya que no contarán con el conocimiento y experiencia de aquellos que desarrollaron el sistema anteriormente.

Existe el erróneo pensamiento de que realizar una documentación del diseño es tarea inútil. Esto puede suceder simplemente porque la empresa no cuenta con personal especializado en documentación y no se quiere invertir en capacitaciones, o bien, porque el personal rechaza la idea de documentar lo que ellos mismos están construyendo (muchos codificadores no quieren realizar otra tarea que no sea codificar). Siendo estas las causas más comunes cada empresa puede tener sus propias justificaciones.

Resulta necesario remarcar que diseñar no es documentar. Se puede o no documentar un diseño. En este trabajo se le da importancia a documentar los diseños y a partir de esa documentación generar el código, todo esto mediante MDD.

La incorporación de MDD hace que el equipo se involucre con los modelos en forma más comprometida ya que todo código que se generará será a partir de los modelos. Contamos con la ventaja de mantener nuestro sistema completamente documentado y cualquier cambio que haya que realizar se hará sobre los modelos y nuevamente es el mismo equipo el que participará de las modificaciones.

Toda nueva metodología o proceso que se incorpore a una empresa toma tiempo de adaptación y sobre todo aceptación por parte del equipo. Se debe tomar como un cambio en la tarea de trabajar y hacer un fuerte hincapié en las ventajas que trae MDD sobre el trabajo sin contemplación de documentación de modelos.

La comunicación es uno de los factores que más influye en un cambio organizacional. Ya se mencionó anteriormente que cuando algo se comunica y es entendido por todos, entonces la incorporación de una tarea en el trabajo cotidiano no resulta tediosa.

Se ha visto en los capítulos anteriores cada tema por separado y el mapeo de ambos temas. Queda claro que muchas de las definiciones de CMMI caen fuera de cualquier metodología o proceso. Considero un punto positivo esto último ya que es propio de cada empresa el ver de qué manera se cumplen todas las KPA.

No se limita a definir una lista de KPA y decir de qué manera se deben cumplir. CMMI lista por cada nivel las KPA que se deben satisfacer y el trabajo de cada empresa consiste justamente en definir el “cómo” se realiza.

Si bien en esta tesina se han tocado algunos puntos de conexión entre una metodología de desarrollo y CMMI, sirve como un primer paso hacia una certificación, sirviendo de base para un trabajo que puede llevar un tiempo prolongado. Al encontrar esta conexión nos anima para seguir investigando sobre otros puntos de conexión. Esto se ha propuesto como trabajo futuro en el capítulo anterior.

En lo personal, he mencionado anteriormente que me interesan ambos temas por la cercanía que tengo en mi trabajo cotidiano. Conozco las ventajas que trae el seguir un proceso estandarizado y sé lo complejo que es el tratar de incluir a todo el personal a seguir un estándar. Pero no es una tarea imposible, solamente requiere un trabajo continuo y constante.

Lograr seguir un proceso documentado y estandarizado ayuda a mantener el trabajo diario en una organización, haciendo que todos los proyectos se ejecuten de manera ordenada y siguiendo un proceso que ha sido mejorado por la organización adaptándolo según las necesidades de la organización.

Es difícil encontrar dos organizaciones que realicen su proceso de desarrollo exactamente igual, ya que cada uno lo adapta y eso es lo que diferencia unas de otras.

Considero también, que una empresa que quiera incluir MDD a su proceso, primero debe familiarizarse con los modelos, sacarle provecho y ver los beneficios que tiene la inclusión de los modelos para lograr los proyectos propuestos. Una vez comprendidos y adoptados los modelos, la organización está preparada para dar un paso más.

La concientización de todo el equipo al uso de los modelos y a la inclusión de CMMI es fundamental porque el trabajo a realizar tiene que ser en equipo y no solamente de una persona o de un área que está tratando de lograr una certificación. Esto va a resultar en el fracaso, por eso hay que incluir desde el principio a todos y de las ventajas que trae la nueva inclusión.

Para finalizar, concluyo en que esta tesina sirve como base para realizar muchos trabajos futuros que ayuden a entender que alcanzar un estándar de calidad no está tan lejano como pensamos y que la inclusión de MDD a la empresa nos va a ayudar a generar proyectos fáciles de crear, mantener, entender y comunicar.

Apéndice I: *Glosario*

Abstracción: características esenciales de una entidad que la distinguen de otros tipos de entidades. Una abstracción define una frontera desde la perspectiva del observador.

Actividad: es un conjunto de tareas que se realiza con un propósito específico.

Constelación: colección de componentes de CMMI que incluye un modelo, sus materiales de formación y los documentos de evaluación concernientes a un dominio de interés.

Desarrollo: consiste en los procesos que se dirigen a la construcción del sistema.

CMMI: es un modelo de madurez de mejora de los procesos para el desarrollo de productos y de servicios.

Framework: estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado.

Madurez de una organización: el grado en el cual una organización tiene procesos explícita y constantemente desplegados que están documentados, administrados, medidos, controlados, y continuamente mejorados. La madurez organizacional se puede medir por medio de evaluaciones.

MDD: concibe al desarrollo de software bajo la idea central de que los artefactos fundamentales del desarrollo son los modelos (y no los programas). Implica la generación automática de programas a partir de modelos. En MDD “El modelo es la implementación”.

Mejora del proceso de software: metodología deliberada y planeada que sigue prácticas de documentación estandarizadas para capturar en papel (y en la práctica) cada elemento del proceso de software y determinar su valor agregado.

Método: es una técnica repetible para la resolución de un problema específico.

Metodología: es una colección de métodos para la resolución de una clase de problemas.

Metodología de Desarrollo: se refiere a un framework que es usado para estructurar, planear y controlar el proceso de desarrollo en sistemas de información.

Modelo: simplificación de la realidad, creada para comprender mejor el sistema que se está creando; abstracción semánticamente cerrada de un sistema.

Objetivo: principio de alto nivel que se usa para guiar al proyecto. Los objetivos definen los atributos del sistema que son importantes.

Proceso: secuencia de pasos ejecutados con un propósito específico.

Proceso de software: conjunto estructurado de actividades requeridas para desarrollar un sistema de software de alta calidad y proporciona el marco de trabajo desde el cual se puede establecer un plan detallado para el desarrollo del software.

UML: un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software.

Apéndice II: *Abreviaturas*

Abreviatura	Significado
CMMI	Capability Maturity Model Integration
MDD	Model Driven Development
RUP	Rational Unified Process
MDA	Model Driven Architecture
OMG	Object Management Group
IEEE	The Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
SEI	Software Engineering Institute
IPPD	Integrated Product and Process Development
SP	Specific Practice
UML	Unified Modeling Language
KPA	Key Process Area
PMO	Project Management Office

Tabla 6: Abreviaturas

Bibliografía

[BLOG] <http://CMMI-en-castellano.blogspot.com/2009/05/CMMI.html>
 Actividades y artefactos que constituyen el ciclo de vida del software (nombrar IEEE)
 [RAE] <http://www.rae.es/rae.html>
 [CMMI, 2da Edición] <http://www.sei.cmu.edu/library/assets/cmmi-dev-v12-spanish.pdf>
 [UML, 2da Edición] "El Lenguaje Unificado del Modelado" - Segunda Edición - Booch, Jacobson, Rumbaugh
 - Pearson / Addison Wesley, 2006
 [OMG] <http://www.omg.org/>
 [MDA] <http://www.omg.org/mda/>
 [UML] <http://www.uml.org/>
 [SEI]: <http://www.sei.cmu.edu/>
 [CMMI]: <http://www.sei.cmu.edu/CMMI/>
 [RUP] <http://www-01.ibm.com/software/ar/rational/rup.shtml>
 [WIKIPEDIA 1] http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software
 [Saltiveri 2005] "Diseño de sistemas interactivos centrados en el usuario" - Saltiveri, Vidal, Delgado - Editorial UOC, Primera edición, Octubre 2005. Vista previa: <http://books.google.com.ar/books?id=Bk5Uv0Ais0C&printsec=frontcover&client=firefox-a#v=onepage&q=&f=false>
 [MET-SOF1] <http://www.willydev.net/Descargas/cualmetodologia.pdf>
 [Weitzenfeld] "Ingeniería de software orientada a objetos con UML, Java e Internet" - Alfredo Weitzenfeld, Editorial Thomson. Vista previa: <http://books.google.com.ar/books?id=MOViEp0ApQcC&printsec=frontcover&client=firefox-a#v=onepage&q=&f=false>
 "Ingeniería de Software Orientado a Objetos" - Bernd Bruegge, Allen H. Dutoit - Prentice Hall, 2002
 [REPORTE CMMI] <http://seir.sei.cmu.edu/CMMIresearch/results/2005results.asp>
 CMM y RUP: Una perspectiva común
http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Presentacion_CMM_RUP_02.pdf
 Influencia de Paradigmas de Gestión de Proyectos en CMMI
 Leonel Caviedes, Samuel Sepúlveda, Carlos Cares - Universidad de La Frontera, Temuco, Chile Sitio Web:
[http://sistemas.dis.ufro.cl/eig2008/files/papers/Caviedes-Sepulveda-Cares\(pp.218-233\).pdf](http://sistemas.dis.ufro.cl/eig2008/files/papers/Caviedes-Sepulveda-Cares(pp.218-233).pdf)
 "Realizing CMMI using Enterprise Architect and UML for Process Improvement" -
<http://www.sparxsystems.com/downloads/whitepapers/RealizingCMMIusingEnterpriseArchitect.pdf>
<http://html.rincondelvago.com/el-ciclo-de-vida-del-software.html>
<http://latecladeescape.com/w0/ingenieria-del-software/metodologias-de-desarrollo-del-software.html>
http://www.acis.org.co/fileadmin/Conferencias/IntroduccionCMMI_CarlosMendez.pdf
 Adopción del Capability Maturity Model Integration (CMMI) -->
<http://www.rational.com.ar/CMMI/CMMIesp.html>
<http://www.springerlink.com/content/07h22335507h4366/fulltext.pdf>
<http://www.springerlink.com/content/p817381456316408/fulltext.pdf>
<http://www.ingenierosoftware.com/calidad/cmm-cmmi-nivel-2.php>
 "Identifying Extensions Requeri by RUP (Rational Unifed Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3" - http://www.inf.ufrgs.br/amadeus/download/manzoni_jeetosen2003.pdf

Para la elaboración de la presente tesina se ha utilizado el procesador de texto
 OpenOffice.org (Writer) versión 3.1.0 y PDFCreator 0.9.6