Aseguramiento de Calidad de Proceso de

software

En el siguiente informe daremos una visión general acerca del aseguramiento de calidad de proceso y los modelos creados para tal fin, haciendo énfasis en el modelo CMMI debido a su amplia adopción en la industria del software.

Antes de profundizar acerca del modelo CMMI, quisiéramos brindar una base de lo que se entiende por calidad de software, aseguramiento de calidad y los principales modelos existentes.

Citando la definición de Pressman, el concepto de calidad de software se asocia a la “concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo plenamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”.Teniendo en cuanto esto, que un sistema de software cumpla con estas características garantiza que el cliente cuente con un sistema confiable, lo cual aumenta la satisfacción del mismo frente a la funcionalidad y eficiencia del sistema construido.

La obtención de un software de calidad implica la utilización de metodologías y procedimientos estandarizados durante todas las fases del ciclo de vida del mismo; En este punto radica la importancia del aseguramiento de calidad,la cual puede definirse como un conjunto de actividades sistemáticas y planeadas para asegurar que los procesos y productos de software cumplen con los requerimientos, estándares y procedimientos.

En cuanto a los modelos de calidad diremos que son aquellos documentos que integran la mayor parte de las mejores prácticas, proponen temas de administración en los que cada organización debe hacer énfasis, integran diferentes prácticas dirigidas a los procesos clave y permiten medir los avances en calidad, dando de esta manera apoyo a la organización para tener una mejora continua y brindar productos o servicios de alto nivel.

En el caso del software los modelos deben ir enfocados a realizar el seguimiento y evaluación en cada etapa de construcción del producto software, permitiendo valorar el sistema, bien sea cualitativa o cuantitativamente, y de acuerdo con esta evaluación la organización podrá proponer e implementar estrategias que permitan la mejora del proceso dentro de las etapas de análisis, diseño, desarrollo y pruebas del software.

A continuación mencionaremos los tres modelos de calidad abarcados en este informe.

* **SPICE**

La ISO/IEC TR 15504, conocida como SPICE (Software Process Improvement and Capability dEtermination) es un modelo de evaluación y mejora de los procesos de desarrollo y mantenimiento de sistemas y productos de software. El estándar ISO 15504 es una herramienta que ayuda a reducir costos y mejorar la calidad evitando problemas.

La ISO/IEC TR 15504 es un marco de valoración de procesos, que puede ser empleado por las organizaciones involucradas en la planificación, gestión, monitorización, control y mejora de la adquisición, suministro, desarrollo, operación, evolución y soporte de software.

* **ISO 9000-3**

Es una norma derivada de la ISO 9001 dedicada al proceso de desarrollo con calidad del software. Cuando se publicó la ISO 9001 resultaba difícil aplicar esta norma genérica a el proceso de software. por esta razón se creo la ISO 9000-3 que define un conjunto de estándares utilizados para el desarrollo,suministro y soporte del software.

Dicha norma se basa en que el control de calidad debe ser aplicado a todas las fases de la producción de software , incluido el mantenimiento y tareas posterior a su implantación.

* **CMMI (Capability Maturity Model Integration)**

Es el modelo más utilizados en las empresas de construcción de software, con el propósito de verificar el cumplimiento de estándares de calidad a partir de la medición

con niveles de madurez.A continuación profundizaremos en el análisis de dicho modelos.

**CMMI (Capability Maturity Model Integration)**

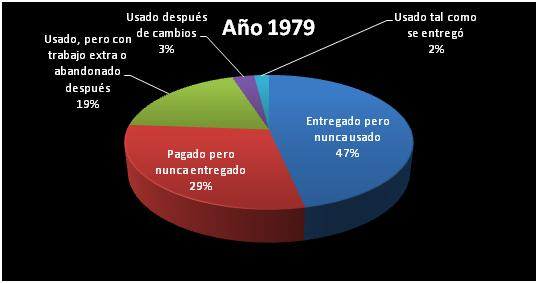
[CMMI](http://www.sei.cmu.edu/cmmi) nace a finales de los años 80 (exactamente, 1987) dentro de la industria militar de los Estados Unidos de América, y su primera finalidad fue la de evaluar los procesos de software más críticos. El principal desencadenante para su creación fue la llamada “Crisis del software”, un término que comenzó a usarse en los años 70, cuando la ingeniería de software era prácticamente inexistente. Éste se refería a la dificultad que presentaba la creación de software frente al rápido crecimiento de la demanda por software, de la complejidad de los problemas a ser resueltos y de la inexistencia de técnicas establecidas para el desarrollo de sistemas que funcionan adecuadamente o pudieran ser validados.

Con el avance del hardware y con la creación de los lenguajes de programación de alto nivel, las máquinas necesitaban programas mucho más complejos. Por este motivo se empezó a concebir el Software como producto, y se empezaron a desarrollar algunos proyectos para que funcionaran en las máquinas de la época.

A medida que aumentaba la complejidad aparecian importantes problemas: los productos excedían la estimación de costos, había retrasos en las entregas, no satisfacía las prestaciones las solicitadas, el mantenimiento era extremadamente complicado y costoso. El resultado era software de mala calidad.

Una de las principales causas de todo esto, si no la principal, era el enfoque dado al proceso de desarrollo de software, el cual era malo e incluso a veces era inexistente. En este proceso, solo ¼ del tiempo de desarrollo se dedicaba a las fases de análisis, diseño, codificación y pruebas, y más de ¾ del tiempo se dedicaba a correcciones y mantenimiento.Es evidente que dedicándole solo ¼ del tiempo a las primeras fases, se arrastran errores graves, sobre todo procedentes de las fases de análisis y diseño, lo que dificultaba la implementación, produciendo constantes paradas y retrocesos para revisar este análisis/diseño.

Los resultados acerca de proyectos de software, que tuvo lugar en EEUU arrojaron los siguientes resultados:



Con estos indicadores estaba claro que algo estaba fallando y que el proceso de desarrollo de software necesitaba un cambio radical.

Los resultados de este estudio llevan al departamento de defensa a tomar una decisión, y por lo tanto, a través del SEI (Software Engineering Institute), instaura, como estándar para la creación de software el CMM (Capability Maturity Model) y obliga a todos sus proveedores a certificarse en el mismo, para poder abastecer, de tal forma que prácticamente, todos aceptan a CMM como el método de trabajo.

El SEI ha tomado la premisa de la gestión de procesos para construir sus CMMs, la cual establece que la calidad del proceso de desarrollo es crítica para la calidad del producto. Esta última concepción ha influido tremendamente en el convencimiento de la Comunidad de Ingeniería del Software, para considerar seriamente la mejora de procesos, la cual implica entender los procesos existentes y cambiarlos para mejorar la calidad del producto y/o reducir los costos y el tiempo de desarrollo.

Los CMMs se centran en mejorar los procesos de una organización. Contienen los elementos esenciales de los procesos eficaces de una o más disciplinas y describen un camino evolutivo de mejora desde procesos ad hoc e inmaduros a procesos disciplinados y maduros con calidad y eficacia mejoradas.

En un intento de integrar los modelos que se habían desarrollado (incluyendo sus propios modelos), el SEI se embarcó en un nuevo programa para desarrollar un modelo integrado (CMMI).

Al igual que otros CMMs, los modelos CMMI orientan en el desarrollo de procesos.Los modelos CMMI no son procesos ni descripciones de proceso. Los procesos reales utilizados en una organización dependen de muchos factores, incluyendo dominios de aplicación, y estructura y tamaño de la organización. En particular, las áreas de proceso de un modelo CMMI normalmente no se corresponden uno a uno con los procesos utilizados en su organización.

El Modelo de Madurez de la capacidad Integrado(Capability Maturity Model for Integration) es un modelo de procesos que contiene las mejores prácticas de la industria para el desarrollo,mantenimiento, adquisición y operación de productos y servicios. No es una norma, no se certifica, sólo se evalúa a través de profesionales reconocidos por el SEI como Lead Appraisers. La versión actual de CMMI es la 1.3 y fue liberada el 1 de noviembre de 2010.

El proyecto CMMI se creó para resolver el problema de usar múltiples CMMs. La combinación de los modelos seleccionados en un marco de mejora único pretendía que fuera usado por organizaciones en su búsqueda de la mejora de procesos para toda la empresa. El desarrollo de un conjunto de modelos integrados implicó más que una simple combinación de los modelos existentes. Al usar procesos que fomentan el consenso, el Equipo del Producto CMMI creó un marco que da cabida a múltiples constelaciones.

Una “constelación” es una colección de componentes CMMI que se usan para construir modelos, materiales de formación y documentos relativos a la evaluación para un área de interés

CMMI propone 3 constelaciones:

CMMI para el Desarrollo (CMMI-DEV) : es un modelo desarrollado para que las empresas se dediquen al desarrollo y mantenimiento de productos y servicios. Esto es aplicable a todo tipo de industria donde el propósito es la entrega de los productos y servicios que serán utilizados por el cliente.

CMMI para servicios (CMMI-SVC) : este modelo está diseñado para cubrir todas las actividades que requieren gestionar, establecer y entregar Servicios.En este caso las organizaciones se caracterizan por el desarrollo de servicios que proporcionan y operan para sus clientes o áreas internas.

CMMI para la adquisición (CMMI- ACQ) : este modelo trata la gestión de la cadena de suministro, adquisición y contratación externa en los procesos del gobierno y la industria. Aquí entran las grandes organizaciones que manejan diversos proveedores y contratos que se requieren gestionar.

El modelo identifica 16 Áreas de Procesos fundacionales comunes a todas las constelaciones.

Nos enfocaremos en el CMMI-DEV ya que nos interesa entender cómo se relaciona con el desarrollo de un producto Software. Antes de comenzar a hablar de esta constelación, es necesario entender el concepto de nivel. Los niveles se utilizan en CMMI-DEV para describir un camino evolutivo recomendado para una organización que quiera mejorar los procesos que utiliza para desarrollar productos o servicios. Los niveles pueden ser el resultado de la actividad de calificación en las evaluaciones. Las evaluaciones se pueden aplicar a organizaciones enteras o a grupos más pequeños, tales como un grupo de proyectos o una división.

CMMI da soporte a dos caminos de mejora usando niveles. Un camino permite a las organizaciones mejorar de forma incremental los procesos que corresponden a un área de proceso individual (o áreas de procesos) seleccionada por la organización. El otro camino permite a las organizaciones mejorar un conjunto sucesivos de procesos relacionados tratando, de forma incremental, conjuntos sucesivos de áreas de proceso.

Estos dos caminos de mejora están asociados con los dos tipos de niveles: niveles de capacidad y niveles de madurez. Estos niveles corresponden a las dos aproximaciones de mejora de procesos denominadas “representaciones”. Las dos representaciones se denominan “continua” y “por etapas.” El uso de la representación continua permite alcanzar “niveles de capacidad” mientras que el uso de la representación por etapas permite alcanzar “niveles de madurez”. Para alcanzar un nivel particular, una organización debe satisfacer todas las metas del área de proceso o del conjunto de áreas de proceso que son objeto de la mejora independientemente de si es un nivel de capacidad o de madurez. Ambas representaciones proporcionan caminos para mejorar sus procesos con el fin de lograr los objetivos de negocio, proporcionan el mismo contenido esencial y utilizan los mismos componentes del modelo ( por ejemplo, áreas de proceso, metas específicas, prácticas específicas) y estos componentes tienen la misma jerarquía y configuración.

Las diferencias entre las estructuras son sutiles pero significativas. La representación por etapas utiliza los niveles de madurez para caracterizar el estado global de los procesos de la organización con respecto al modelo como un todo, mientras que la representación continua utiliza los niveles de capacidad para caracterizar el estado de los procesos de la organización con respecto a un área de proceso individual.

En el modelo CMMI por etapas, cada nivel de madurez ( desde el 1 al 5) tiene asociado un conjunto de áreas de proceso definidas como podemos ver en el siguiente cuadro:

|  |  |  |
| --- | --- | --- |
| Nivel | Nombre del Nivel | Áreas de Proceso |
| 1 | Inicial | Procesos improvisados e impredecibles, pobremente controlados, reaccionarios. No es necesario acreditarlo. |
| 2 | Administrado | * Administración de requerimientos * Planeamiento de Proyectos * Monitoreo y control de Proyectos * Administración de Acuerdo con el Proveedor * Medición y Análisis * Aseguramiento de Calidad del Proceso y del Producto: * Administración de Configuración |
| 3 | Definido | * Desarrollo de Requerimientos * Solución Técnica * Integración de Producto * Verificación * Validación * Foco en el proceso Organizacional * Definición del Proceso Organizacional * Capacitación Organizacional. * Administración Integrada de Proyectos * Administración de Riesgos * Análisis y Resolución de Decisiones |
| 4 | Gestionado  Cuantitativamente | * Performance del Proceso Organizacional * Gestión Cuantitativa del Proyecto |
| 5 | Optimizado | * Administración de Performance Organizacional. * Análisis Causal y Resolución |

Los niveles 2 y 3 hacen foco a la ingeniería de software, mientras que los niveles 4 y 5 hacen foco en el control estadístico de procesos.A medida que se avanzan en los niveles se administra mejor el riesgo y se aumenta la productividad y calidad.

CMMI por etapas provee una única clasificación que facilita comparaciones entre organizaciones y provee una secuencia probada de mejoras.

Cada área de proceso posee metas genéricas y metas específicas que definen prácticas sugeridas. Las metas genéricas son comunes a todas las áreas de proceso.

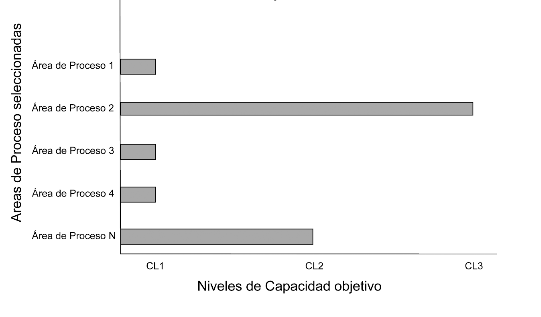
Las áreas de proceso no son procesos sino puede estar integrada en varios procesos o actividades como conjuntos de buenas prácticas.

En cuanto al modelo CMMI continuo evalúa cada área de proceso y le asigna un nivel de 0 a 5 a cada una, permitiendo comparaciones sobre la base de cada Área de Proceso. Además le permite a la empresa elegir el orden de las mejoras. Normalmente, las organizaciones operan a diferentes niveles de capacidad en distintas áreas de proceso. Si un proceso no existe en una organización se le asigna el nivel 0.

Modelo CMMI por etapas



Modelo CMMI Continuo



Luego de dar un vistazo general sobre CMMI podemos ver que se basa en la evaluación de procesos definidos, pero nos interesa saber cómo es posible aplicar este modelo en ambientes ágiles, ya que allí los procesos son definidos por el equipo al momento de llevar a cabo el proyecto. Para ello CMMI (desde la versión 1.3) viene hablando de temas ágiles y propone una forma de llevar las implementaciones de modelos ágiles en conjunto con CMMI.

Algunas prácticas que marca CMMI pueden ser complementadas con elementos que marca SCRUM o viceversa.Los modelos se complementan no se contraponen. A continuación, podemos ver qué elementos de SCRUM podemos utilizar para cubrir las Áreas de Proceso del Nivel 2:

|  |  |
| --- | --- |
| **Áreas de proceso** | **Elementos SCRUM** |
| Planificación de proyectos | * Sprint planning * Poker estimation * Daily meeting |
| Monitoreo y control | * Daily meeting * Review * Retrospective |
| Aseguramiento de la calidad | * Daily meeting * Retrospectivas * Inspecciones |
| Configuration management | * Uso de herramientas de versionado |
| Medición y análisis | * Daily meeting * Burndown charts |

**Planificación de proyectos**

Para planificar, SCRUM propone dos ceremonias. Una de ellas es el sprint planning, en donde el equipo scrum se compromete a realizar una cierta cantidad de trabajo durante la duración del sprint, valiéndose de estimaciones para luego definir un objetivo. La otra ceremonia es la Daily meeting, la cual permite planificar a un nivel de granularidad menor (diaria) hablando de lo que se hizo el dia anterior, ese mismo día y los inconvenientes existentes.

**Monitoreo y control**

Las daily, además, permiten detectar problemas que enfrenta el equipo de desarrollo. Por otro lado, gracias a las review (donde el product owner acepta o rechaza los ítem implementados) se puede calcular la métrica de velocidad, comparando el trabajo aceptado con el que se comprometió el equipo al principio del sprint. Si se detecta algún desvío de esta métrica, se deberá tomar alguna acción correctiva. Estas decisiones se toman en la retrospectivas,donde se busca mejorar el proceso.

**Aseguramiento de calidad**

Como se mencionó anteriormente, las daily permiten detectar inconvenientes que se presentan a diario, lo que luego permite tomar decisiones en consecuencia; así como también en las retrospectivas se inspecciona lo realizado en el sprint anterior, tanto lo bueno como lo malo, para adaptar el proceso buscando la mejora del mismo.

**Configuration Management**

Con la complejidad actual del software, la utilización de una herramienta de gestión de configuraciones no es opcional, ya que es necesario para gestionar las distintas versiones de los ítems de configuración.

**Medición y Análisis**

Los burndown charts nos permiten realizar mediciones diarias del trabajo terminado, lo cual nos permite detectar retrasos en el sprint.