

## **2 Historia de las métricas de calidad de software**

### **2.1 Introducción**

#### **3.1.1 Un poco de Historia**

La industria de software, a diferencia de otras industrias, tiene muy poco tiempo de existir. Lo que ha llamado la atención del mercado hacia ella han sido dos factores esenciales: la velocidad con que ha crecido y su alcance. Desde su inicio existieron personas en distintos campos que vieron el avance que para ellos representaba hacer uso de software especializado que les permitiera automatizar procesos o acelerarlos.

Al haber tanta demanda en cuanto al campo se iniciaron muchas investigaciones en la rama de software y de hardware. Con el tiempo los costos se redujeron y el software se convirtió en un negocio rentable. Al haber tanto interés, muchas personas empezaron a desarrollar y ahí nacieron las primeras grandes empresas de software. Esto trajo consigo un problema natural en el proceso: al haber tantos desarrolladores en distintos países y para distintas aplicaciones empezó a haber diversidad de estilos así como la calidad del producto final variaba mucho entre producto y producto. En este marco se hizo necesario un estándar que permitiera a los consumidores de software decidir si el producto que estaban recibiendo era de calidad y si cumplía ciertos requisitos de funcionalidad.

#### **2.1.2 Definición de Estándar**

Según ISO, un estándar es “un conjunto de acuerdos documentados que contienen especificaciones técnicas u otros criterios precisos para ser usados constantemente, como reglas, lineamientos o definiciones de características. Todo esto con la finalidad de asegurar que los materiales, productos, procesos y servicios son óptimos para su propósito” [ISO 02a]. El primer organismo encargado de establecer un estándar sobre algún producto o proceso fue el International Electrotechnical Commission (IEC) creado en 1906 [IEC 03].

Uno de los propósitos principales de un estándar es promover un intercambio de productos en base a ciertos lineamientos comunes. Para que existiera una industria de software robusta primero hubo que establecer ciertos lineamientos.

### **2.1.3 Creación del Software Engineering Institute**

Uno de los primeros organismos que impulsaron el desarrollo de estándares en procesos de software fue el Departamento de Defensa de los Estados Unidos (DoD). A ellos les preocupaba el establecimiento de métricas para identificar a los contratistas potenciales referentes a ésta rama. De esta forma ellos crearon la Software Engineering Institute (SEI).

El SEI es un centro de investigación y desarrollo sostenido por el Departamento de Defensa y operado por la Universidad de Carnegie Mellon. El propósito principal de éste instituto es el ayudar a las empresas a buscar mejoras en cuanto a ingeniería de software basadas en métricas [SEI 03].

El primer mecanismo que sirvió para éste propósito fue un cuestionario para evaluar el estado actual sobre las prácticas de software en las empresas [Humphrey 87]. Gracias al éxito de éste, en 1988 la SEI creó un programa de entrenamiento a las organizaciones que querían realizar autodiagnósticos en su proceso de software. Éste programa de entrenamiento dio paso a la creación del Software Process Assessment (SPA) cuyo objetivo esencial era hacer llegar el programa de entrenamiento a más empresas.

El mercado creado por la SEI fue aprovechado por particulares que empezaron a dar asesorías individuales por lo que éstos decidieron llevar a cabo un proyecto de grandes magnitudes que con el tiempo sería llamado Capability Maturity Model (CMM) [Paulk 91, Weber 91].

#### **2.1.4 ISO 9001:1994**

Como respuesta a la estandarización de procesos de software por parte de la SEI, ISO creó su apartado de calidad en cuanto al software. La International Organization for Standardization (ISO) fue la primera empresa encargada de crear un estándar en cuanto a creación de software se refiere. ISO es creada en 1947 al final de la segunda guerra mundial con la participación de delegados de 25 países con la finalidad de facilitar la coordinación internacional y la unificación de estándares industriales [ISO 2002b].

ISO 9001:1994 fue la respuesta a CMM por parte de ISO. Ésta se encarga de los “sistemas de calidad. Modelo para asegurar la calidad en diseño o desarrollo, producción, instalación y servicio”. Tres años después salió al mercado el ISO 9000-3:1997 que es en realidad una expansión además de una guía para el ISO 9001:1994.

## **2.2 Capability Maturity Model (CMM)**

### **2.2.1 Introducción**

Como ya se dijo anteriormente, el SEI y posteriormente CMM nacieron de una necesidad en el mercado. En el desarrollo de software empezó a crecer la inconformidad en cuanto a la diferencia entre las fechas pactadas para entrega de proyectos y la fecha real de finalización. En un estudio hecho por el Departamento de Defensa a sus 17 contratistas mayores, encontraron que los proyectos pactados para 28 meses eran errados en un promedio de 20 meses y que un proyecto de 4 años no terminaba en menos de 7 [Paulk 95].

CMM tuvo varias influencias en su creación. La persona que más aportó a éste proyecto fue Watts S. Humphrey ya que CMM se apoya en sus trabajos realizados anteriormente. Él ideó un cuestionario [Humphrey 87] que fue usado como la primera métrica de calidad sobre desarrollo de software. Ésta métrica fue explicada más adelante en un libro escrito por él mismo, *Managing the Software Process* [Humphrey 89].

Otra cosa que observó la SEI sobre el proceso de software fue que se necesitaba un método de asesoramiento para facilitar la comparación de las compañías de software con las prácticas de Humphrey y proveer cierto tipo de evaluación sobre la madurez de su proceso. El primer método de asesoramiento público fue el Software Process Assessment (SPA). Éste sería reemplazado en 1995 por el CMM Based Appraisal for Internal Process Improvement (CBA IPI) [Dunaway 95] usado en la actualidad y el cual será descrito en el apartado 2.5 de este capítulo. Después de recopilar y procesar la información obtenida por éstos métodos, todo fue evolucionando hacia el Capability Maturity Model.

### **2.2.2 Descripción y características de CMM**

Según Paulk, CMM “describe los principios y prácticas relacionadas al proceso de madurez de software e intenta ayudar a las organizaciones de software a mejorar la madurez de sus procesos de software en términos de un camino evolutivo, desde procesos caóticos hasta procesos maduros y disciplinados”. CMM fue creado a partir del cuestionario inicial de la SEI y del SPA [Paulk 91].

Una de las características principales de CMM es que esta basado en la combinación de experiencias anteriores sobre mejoras en procesos tales como teorías académicas de tiempo sobre negocios y experiencias prácticas obtenidas de organismos de software de muchas compañías grandes como IBM o Xerox.

Otra característica importante es que CMM basa gran parte de su efectividad en la documentación. En todo el modelo frases tales como “de acuerdo a un proceso documentado” y siguiendo una “política organizacional escrita” son comunes entre los KPA de CMM. Siempre se hace énfasis en la necesidad de guardar información para su uso posterior en el proceso y el mejoramiento del mismo. La última característica y además muy importante es que CMM está al alcance de cualquier persona en forma gratuita.

Gracias a que se identificaron los proyectos exitosos en grandes compañías, se empezaron a localizar prácticas comunes entre éstos que además no se encontraban en proyectos menos exitosos. También se encontró que éstas prácticas podían ser agrupadas en distintas áreas. Esto más adelante se convirtió en los Key Process Areas (KPA) de CMM. Eventualmente se fueron definiendo distintos niveles que representaban etapas de madurez en los procesos. Las primeras etapas eran la base de las posteriores.

### **2.2.3 Los 5 niveles de CMM**

Un nivel de madurez es un sistema evolutivo y bien definido para alcanzar el proceso de madurez de software. Cada nivel de madurez tiene dentro de si mismo parámetros que permiten la mejora continua. Alcanzar un nivel dentro de la escala de CMM significa seguir en busca de mejores prácticas y a la vez mantener los logros alcanzados [Paulk 95].

Los niveles de madurez tienen varios objetivos:

- Definir un orden para medir la calidad de software gracias a la madurez de la compañía.

- Ayudar a la organización a ver que procesos debe de mejorar en forma gradual para alcanzar el nivel óptimo.
- Mantener un proceso bien documentado.
- Lograr un producto controlado, verificable, validado y medido.

Los niveles son: inicial, repetible, definido, administrado y óptimo. Estos se describirán a detalle a continuación.

### **Nivel 1: Inicial**

Éste nivel es el primer estado en la evolución de las organizaciones que desarrollan software. En éste nivel se encuentran todas las empresas que no han logrado implementar las prácticas básicas de administración de proyectos e ingeniería de software definidas a partir del nivel 2 o superiores.

En éste nivel suele ser frecuente el encontrar crisis dentro de la compañía. Esto se puede deber a diversos factores como comprometerse de más sobre los tiempos y alcances del producto, falta de comunicación en el equipo de desarrollo, etc. Cuando se presentan éstas crisis, generalmente el equipo deja de lado los procesos iniciales como el análisis y el diseño para dedicarse únicamente a codificar y si da tiempo realizar todas las pruebas necesarias.

En estas empresas, el software es producto del esfuerzo de artistas individuales como líderes de proyecto excepcionales o un efectivo equipo de desarrollo. Cada héroe crea su

proceso con un estilo propio y suele ser necesario que la misma persona este involucrada en el siguiente proyecto para que pueda ser repetido. Muchas veces los equipos realizan proyectos que cumplen los requerimientos necesarios sin entender claramente como lo lograron y otras lo logran pero exceden por mucho el presupuesto inicial o la calendarización.

La administración participa en gran manera del proceso en éste nivel, aunque generalmente en una forma ineficiente. Ellos ocupan una parte significativa de su tiempo en arreglar problemas y hablar con los clientes insatisfechos. Ante una situación de crisis permanente, se les hace difícil destinar recursos para definir o documentar procesos, lo que lleva a un ciclo infinito. Normalmente no existe buena relación entre administrativos y desarrolladores debido a que hay muchos roces debido a la presión de la entrega o de los errores existentes.

## **Nivel 2: Repetible**

En el nivel repetible se establecen prácticas básicas de administración de proyecto que permiten establecer control de requerimientos, calendarizaciones y costos. El equipo que desarrolló el proyecto puede aprovechar su experiencia e inversión en procesos para aplicarla en un nuevo proyecto.

Los proyectos repetibles llevan a cabo procesos definidos, medidos, entrenados y probados. Este nivel no garantiza que todos los proyectos dentro de la empresa tienen el mismo nivel de madurez. Algunos pueden estar todavía en el nivel inicial.



Es importante en éste nivel contar con procesos formales de documentación. Éste proceso debe ser comunicado a todos los niveles involucrados en el proyecto y continuamente mejorados por lo que debe de existir mucha precisión en la documentación de todos los procesos. Los análisis de proyectos anteriores proporcionan información importante acerca de todos los niveles de desarrollo. Ésta información permite hacer compromisos inteligentes en cuanto a tiempos o costos se refiere.

En este nivel los líderes de proyecto toman más importancia que en el nivel anterior. Dejan de encargarse de situaciones técnicas para dedicarse a administrar el proceso. Ya pueden hacer un compromiso y seguir planes realistas basados en los resultados de proyectos similares. Son responsables de la estimación y cambios en términos de calidad, costo y tiempo. Deben de identificar los problemas que surjan durante el proyecto en el momento y darles solución o tomarlos en cuenta para futuros procesos.

A partir de éste nivel es importante llevar una implementación controlada de métricas y medidas de software ya que si no se hace así desde los primeros niveles, en los próximos será muy tardado llevar acabo una transición.

### **Nivel 3: Definido**

En éste nivel la empresa ha definido un conjunto de procesos, metodologías y herramientas usados por todos los niveles involucrados en el proceso, tanto administrativos como desarrolladores. Existen pautas y criterios definidos para adaptar un proceso estándar a las

necesidades y características propias de cada proyecto. El nivel de definición es detallado y completo. En el proceso ya no existe dependencia de esfuerzos individuales, pues todos conocen el proceso.

El nivel 3 mantiene la misma línea que el nivel 2 pero lo expande a todo el proceso, hay un lenguaje común dentro de la organización. La administración del proyecto especifica en todo proceso que tareas realizar, quién las debe de realizar, el proceso y el tiempo necesario. La organización hace uso de las prácticas exitosas al estandarizar procesos de desarrollo.

También se recomienda contar con un grupo de personas dentro del equipo dedicadas al proceso de ingeniería de software. Ellos deben de encargarse de llevar a cabo actividades como la evaluación de líderes o revisiones de código con el fin de buscar mejoras en el proceso y saber como se desarrolla éste dentro de la organización.

#### **Nivel 4: Administrado**

El cambio del nivel definido al administrado es muy difícil ya que no está bien explicado en la estructura de CMM. En este nivel la organización mide la calidad del producto y del proceso de software. Ambas partes son seguidas en forma cuantitativa y se controlan mediante métricas detalladas. La capacidad de rendimiento del proceso es previsible.

Como su nombre lo dice, la administración de proyectos juega un papel esencial. En éste nivel el proyecto se entiende como un conjunto de procesos y la administración de los mismos y sus cambios. Todos los procedimientos y métricas necesarios en los niveles

anteriores se integran en éste nivel. El administrador del proyecto debe de dirigir todos los procesos paralelos y desarrollos hacia una meta definida. Debe de encargarse de procesos preventivos para corregir posibles errores antes de que se vuelvan incontrolables.

Los procesos de software son muy predecibles. Cuando sucede algo no contemplado es fácilmente corregible debido al control del proceso. La empresa es capaz de proponerse metas cuantitativas para la calidad de los productos y de los procesos de software. Es posible medir la productividad y calidad de los procesos de software a través de todo el proyecto. Las mediciones permiten detectar cuando las variaciones del rendimiento se salen de los rangos aceptables, de manera que se puedan tomar medidas correctivas para asegurar la calidad.

### **Nivel 5: Óptimo**

La característica principal aquí es que la organización entera lleva a cabo mejoras continuas en el proceso, en base a la retroalimentación cuantitativa y al ensayo de ideas y tecnologías innovadoras. Todos los cambios realizados en cualquier parte del proceso son vigilados y controlados con el sistema de métricas. Durante todo el desarrollo no suelen suceder errores.

La organización cuenta con los medios para identificar las debilidades y reforzar el proceso con el objeto de prevenir defectos. Los datos relativos a la eficiencia del proceso de software se usan para analizar el costo y el beneficio de usar nuevas tecnologías y de implementar cambios al proceso de software.

Los proyectos de software analizan los defectos para determinar sus causas. Los procesos de software se evalúan para prevenir que los defectos conocidos vuelvan a ocurrir, asimismo las lecciones aprendidas son difundidas a otros proyectos. Es importante llevar a cabo retroalimentación en esta fase. De ella depende el control efectivo del proceso. Ésta debe de estar dirigida hacia un objetivo para no caer en la retroalimentación de errores desde un principio y seguir alimentando en forma incorrecta el proceso.

#### **2.2.4 Key Process Areas (KPA)**

Para cada nivel (excepto nivel 1), el CMM especifica algunas Key Process Areas (KPA), que representan las áreas en las que una organización debe enfocarse si desea alcanzar algún nivel en particular. Cada una está asociada con metas que representan los requerimientos a satisfacer por el proceso para esa KPA.

Las KPA para los niveles de madurez existentes pueden ser usadas para verificar la capacidad del proceso actual e identificar las áreas que necesitan ser reforzadas para alcanzar el siguiente nivel. En forma breve, los KPA de los distintos niveles de CMM son:

##### **NIVEL 2:**

- **Requirements Management (RM):** Mantener coherencia entre el cliente y los requerimientos que éste busca en su sistema de software.
- **Software Project Planning (SPP):** Buscar diversos métodos para lograr un buen desempeño en la ingeniería de software y la administración del proyecto.

- **Software Project Tracking and Oversight (PTO):** Llevar un monitoreo del progreso actual del proyecto para que la administración puede tomar acciones acertadas cuando el desempeño del sistema se aleja de los planes iniciales.
- **Software Quality Assurance (SQA):** Tener una administración que tome en cuenta todo el proceso del proyecto y los productos desarrollados.
- **Software Configuration Management (SCM):** Alcanzar y mantener la integridad de los productos de software a través del ciclo de vida del proyecto.
- **Software Subcontract Management (SSM):** Elegir subcontratistas de software calificados y administrarlos en forma eficiente.

### **NIVEL 3:**

- **Organization Process Focus (OPF):** Fomentar un sentido de responsabilidad organizacional para generar actividades de proceso de software que pueden mejorar todo el proyecto.
- **Organization Process Definition (OPD):** Desarrollar y mantener activos utilizables en el proceso de software que mejoren el desempeño evolutivo de los proyectos y proporcionen una base para lograr beneficios a largo plazo en la organización.
- **Training Program (TP):** Establecer un programa de desarrollo de habilidades y conocimiento en las personas para que puedan desempeñar su trabajo en forma eficiente.
- **Integrated Software Management (ISM):** Integrar las actividades de ingeniería de software y administración en un proceso coherente, definido y adaptado del proceso existente.

- **Software Product Engineering (SPE):** Realizar procesos que integren todas actividades de ingeniería de software para producir productos de software consistentes y efectivos.
- **Intergroup Coordination (IC):** Buscar medios para que el equipo de ingeniería de software participe activamente con los demás equipos de desarrollo de modo que el proyecto esté facultado para satisfacer las necesidades del cliente.
- **Peer Reviews (PR):** Detectar los defectos en el producto de software con anticipación. Es importante desarrollar un vínculo entre los productos de software y sus posibles defectos para que puedan ser prevenidos.

#### **NIVEL 4:**

- **Quantitative Process Management (QPM):** Controlar el desempeño del proceso de software en forma cuantitativa.
- **Software Quality Management (SQM):** Desarrollar un entendimiento cuantitativo de la calidad de los productos del proyecto de software y alcanzar metas de calidad específicas.

#### **NIVEL 5:**

- **Defect Prevention (DP):** Encontrar las causas de los defectos y prevenir que ocurran.
- **Technology Change Management (TCM):** Identificar nuevas tecnologías (herramientas, métodos y procesos) que sean de beneficio y transferirlas a la organización en forma ordenada.
- **Process Change Management (PCM):** Mejorar continuamente los procesos de software utilizados dentro de la organización con la idea de aumentar la calidad en el

software, su productividad y la disminución del tiempo del ciclo de vida de desarrollo del producto.

Un problema en la actualidad es que debido a que el modelo CMM se originó a principios de los años 90, las KPA enfocaron sus actividades y prácticas hacia un proceso tradicional de desarrollo en cascada. Estos requieren especificación de requisitos, planes de desarrollo documentados, revisiones e inspecciones de garantía de calidad, procedimientos documentados y documentación de los resultados de los proyectos.

Muy pocas KPA orientan sus actividades y prácticas a los resultados propios del desarrollo, es decir, el producto de software y a las herramientas asociadas a la ingeniería del software (CASE, reglas de codificación y compilación), que hacen del proceso de desarrollo del software su verdadero objetivo. Tampoco se profundiza en el proceso de la arquitectura del software o el proceso de evaluación que han demostrado ser factores limitantes para el éxito del proyecto.

## **2.3 International Organization for Standardization (ISO)**

### **2.3.1 Introducción**

“La International Organization for Standardization (ISO) es una organización a nivel mundial de estándares nacionales de más de 140 países, uno de cada país.

ISO es una organización no gubernamental creada en 1947. La misión de ISO es promover el desarrollo de la estandarización y actividades relacionadas a la técnica con la visión de facilitar el intercambio internacional de bienes y servicios y de desarrollar la cooperación en las áreas de actividades intelectuales, científicas, tecnológicas y económicas.

El trabajo de ISO provee resultados sobre acuerdos internacionales que son publicados como estándares internacionales” [ISO 02c].

ISO cuenta con distintas series dependiendo del enfoque que tengan. La que se refiere al desarrollo de software es la serie ISO 9000 (y más específico 9001:1994). Ésta puede ser usada con propósitos de manejo de calidad interna (ISO 9004) y de calidad externa (ISO 9001, 9002, 9003).

Los campos que comprende ISO 9001 son el diseño, el desarrollo, la producción, instalación y servicio. Una empresa que busca la certificación de ISO 9001 puede tardar de 3 meses a 3 años de dependiendo del nivel en que se encuentre.

El propósito de ISO 9001 es asegurar a los clientes que los proveedores pueden brindar productos y servicios de calidad. Está pensado para llenar las necesidades del cliente, las del proveedor son secundarias. Una organización debe de alcanzar y sostener la calidad de un producto o servicio producido para seguir en la búsqueda continua de las necesidades explícitas o implícitas del cliente.



Las ventajas de obtener ésta certificación son:

- El control de la calidad de los productos y servicios de la empresa.
- La reducción de costos asociados con la mala calidad.
- El reconocimiento por parte de los clientes de la certificación.
- Entrada a un mercado internacional.

### **2.3.2 ISO y el mercado de software**

ISO 9001:1994 es el primer acercamiento en cuanto a software se refiere. Ésta métrica contiene 20 procesos mínimos a ser cubiertos pero no está enfocada solamente a software. La norma ISO 9001 no fue bien aceptada por los desarrolladores ya que permite muchas interpretaciones en cuanto al tema por lo que en vista de la existencia de estos huecos se hizo un apartado especial 3 años después, el ISO 9000-3:1997.

Éste apartado está estructurado como una guía que facilita la aplicación de la norma ISO 9001 en organizaciones dedicadas a desarrollar, suministrar y mantener software. Tiene además extensiones en cuanto a ciertos temas como revisiones o requerimientos de compra.

Su intención es sugerir controles y procesos que garanticen al comprador que el software va a cumplir con los requerimientos establecidos. Esto se logra previniendo las inconformidades en los procedimientos de calidad a lo largo del ciclo de vida del producto de software.

La métrica de ISO sobre software tiene varios defectos ya que hay muchos campos que deja sin control como las prácticas y estilos de la administración en el proceso, el producto final y la interacción en la entrega con el cliente. Además no contiene un nivel detallado de los temas que trata, ya que el apartado de ISO 9000-3:1997 es de sólo 11 páginas.

### **2.3.3 ISO/IEC 15504**

En 1998 la IEC, organización encargada de los procesos de estandarización relacionados con la electrónica, decidieron participar en el proceso de software. Esta organización trabaja en conjunto con ISO por lo que sacaron una norma llamada ISO/IEC 15504.

Esta suite o conjunto de normas esta dividida en nueve partes que son las siguientes:

- Conceptos y guía introductoria.
- Modelo de referencia para procesos.
- Realización de asesorías.
- Guía para realizar asesorías.
- Modelo de asesorías y guía de indicaciones.
- Guía para la competencia de asesores.
- Guía para uso de mejora de procesos.
- Guía para determinar los procesos de proveedores.
- Vocabulario

En el apartado de modelo de referencia para procesos, se nombran los procesos universales de ingeniería de software para las actividades relacionadas con el desarrollo. Describe los procesos que ayudan a adquirir, desarrollar, operar y controlar software y las características de éstos. Estas características son cuantificables y necesarias para administrar un proceso y mejorar su desempeño. Son 9 características que son divididas en niveles (al estilo CMM) a veces agrupando dos por nivel. Los principios que manejan son los mismos que CMM solo que están enfocados a procesos más que a la organización. Los niveles son:

- **Nivel 0: Incompleto.** Existen grandes dificultades para alcanzar el propósito del proceso. Hay muy pocos productos o ninguno identificable y procesos.
- **Nivel 1: Realizado.** El propósito del proceso es generalmente alcanzado. Este no es necesariamente planeado o monitoreado. Hay resultados identificables de los procesos que respaldan el logro del propósito.
- **Nivel 2: Administrado.** El proceso produce resultados acorde a procedimientos específicos, planeados y monitoreados. Estos productos están apegados a estándares y requerimientos específicos.
- **Nivel 3: Establecido.** El proceso es realizado y administrado usando un proceso definido, basado en principios de ingeniería de software. Se implementan procesos aprobados, estandarizados y documentados para lograr resultados.
- **Nivel 4: Predecible.** El proceso definido es consistentemente llevado a la práctica con límites de control definidos para alcanzar las metas propuestas.
- **Nivel 5: Óptimo.** El desempeño del proceso es optimizado para conocer las necesidades actuales y futuras. El proceso es repetido constantemente para alcanzar las metas planteadas.

Los 9 procesos mencionados anteriormente, son muy similares a las funciones de los KPA. Sigue habiendo correcciones debido a que al ser muy pocos puntos, corren el riesgo de mantener tópicos inexplorados. Hubo una norma preeliminar en 1998 pero se sigue trabajando en la que se piensa lanzar definitivamente.

## **2.4 Team Software Process (TSP) y Personal Software Process (PSP)**

### **2.4.1 Introducción**

Aunque en éste título se nombren juntos, estos procesos son independientes y lo suficientemente grandes para ser explicados por separado. Se definirán en conjunto debido a la cercana relación que existe entre ellos. La mayoría de las empresas de software suelen tener un gran problema, que es llevar a cabo un proceso disciplinado de desarrollo. Esto se debe a varios factores como pueden ser distintos estilos de programación, formas erróneas de documentación u olvido de procesos necesarios como diseño o análisis.

Este problema se puede solucionar siguiendo los distintos lineamientos de CMM pero esto nos lleva a un nuevo problema, tener conciencia de qué debemos de hacer pero no saber cómo hacerlo. TSP y PSP fueron creados por Watts Humphrey como el siguiente paso en su modelo CMM. Fueron diseñados para proveer tanto lineamientos sobre los procedimientos como estrategias para el uso de métodos de desarrollo que sirvan al

programador y al equipo. También fueron una respuesta a la interrogante de cómo implantar CMM en organizaciones pequeñas que también buscaban maximizar resultados en sus proyectos.

PSP surgió en 1995 a partir de la necesidad de empresas pequeñas de implementar el modelo CMM. Fue probado en distintas organizaciones educativas con óptimos resultados aunque al llevarlos a la práctica seguía habiendo problemas. Estos surgían gracias a la falta de coordinación y comunicación entre los desarrolladores y los administrativos. Se hicieron estudios que mostraban que si la empresa no promovía procesos PSP, las personas que tenían esta costumbre la dejaban al poco tiempo. En consecuencia, se desarrolló TSP como una respuesta práctica al problema existente de implementación de PSP en equipos organizacionales [Humphrey 00a]. A continuación se mostrará brevemente el enfoque de cada proceso y sus ventajas.

## **2.4.2 PSP**

“El proceso de PSP consiste en un conjunto de métodos, formatos y reglas que muestran a los ingenieros como planear y administrar su trabajo. Es explicado con un libro de texto y un curso diseñados para uso industrial y académico.

PSP está diseñado para su uso con cualquier lenguaje de programación o método de diseño y puede ser usado para la mayor parte de aspectos de trabajo incluyendo requerimientos de trabajo, correr pruebas, definir procesos y reparar defectos” [Hayes 97].

PSP fue diseñado por la SEI como una respuesta a los ingenieros de software en la búsqueda de una aplicación de CMM. El enfoque que maneja favorece el trabajo realizado en empresas o equipos de proyectos pequeños. Tiene como principio el promover trabajo de calidad en cada persona involucrada en el proyecto para así producir software de calidad.

PSP aplica principios sobre procesos a los encargados del desarrollo a través del fomento de un proceso definido, incluyendo un conjunto de métricas de procesos que más adelante serán usados para evaluar el desempeño, la lucha continua por realizar procesos de calidad y la búsqueda de metas de mejoras progresivas. A grandes rasgos, PSP está conformado por siete procesos de desarrollo de software, divididos en 4 niveles:

- **Nivel 0 – Bases del proceso personal (0.0 y 0.1).** Esta es una introducción al proceso. Incluye ciertos conceptos básicos sobre métrica y planeación. En este nivel los desarrolladores hacen tres códigos que serán evaluados. Estos son libres de estilo y forma aunque deben de contener seis puntos que son: planeación, diseño, codificación, compilación, pruebas y resultados.
- **Nivel 1 – Administración del proceso personal (1.0 y 1.1).** En este nivel los dos procesos se enfocan a las técnicas personales de administración de proyectos, incluyendo estimación de tiempo y esfuerzos, formulación de una calendarización y métodos de rastreo de tiempo. Este nivel usa un método llamado PROBE (PROxy Based Estimating) que hace uso de proxys y datos históricos para solucionar los estimados de tiempo y esfuerzo en líneas de código.

PSP usa la técnica de valor ganado para planear y rastrear. Este método es una técnica estándar de administración que asigna un valor a cada tarea del proceso. Este valor está basado en el porcentaje del esfuerzo total planeado en el proyecto. Este valor sirve para medir el progreso y para saber la fecha de terminación del mismo.

- **Nivel 2 – Administración personal de calidad (2.0 y 2.1).** Este nivel añade procesos de calidad al proyecto: diseño personal y revisión de códigos, diseños de notaciones, verificación de técnicas y medidas para administrar un proceso y la calidad del producto. La meta de éste nivel es erradicar cualquier error en el código antes de compilarlo. Esto se logra mediante la revisión del diseño y la del código.

La idea de éstas técnicas no es crear un nuevo método de diseño, sino que el programador examine y documente el diseño desde otra perspectiva. PSP muestra 4 enfoques para el diseño: operacional, funcional, estado y lógico.

- **Nivel 3 – Proceso personal cíclico (3.0).** Este es el último de los niveles de procesos y busca enfocar a las personas a proyectos más grandes sin que esto fomente la pérdida de calidad y productividad en su trabajo. Para esto usan un modelo de ciclo de vida de ciclos, para descomponer el proyecto en partes para ser desarrolladas y después integrarlas.

En el nivel 3, los ingenieros descomponen su proyecto en varios ciclos PSP 2.1 y luego integran y prueban la salida de cada ciclo. Debido a que ya se domina el proceso 2.1, los costos de integración y pruebas son minimizados.

Estos procesos se van desarrollando a través de pequeños proyectos que permitan entenderlos en forma gradual. Es recomendable llevar un proceso paralelo al estudio de la

teoría de PSP a través de la práctica de desarrollo de software. De esta forma es mejor entendido y aprovechado el material.

En estos procesos se detallan varias cosas como la arquitectura. En las guías se lleva un proceso estructurado en cuanto al tema y pasos bien definidos. PSP toma como modelo base para el ciclo de vida del proyecto al modelo de cascada ya que en cada fase se vuelven a trabajar fases anteriores. Estos procesos de modificación a fases anteriores sólo se dan cuando se encuentran defectos. En PSP 3, que es el último proceso en la secuencia, es el único en que se permite implementar el proceso a través de ciclos. Los ciclos de vida en los proyectos de PSP suelen ser muy pequeños aunque van incrementando conforme se avanza en los procesos (el de PSP 3 suele ser el más largo).

En el modelo de PSP se incluyen muchos estándares que son requeridos para las prácticas de proceso. Cubre prácticas avanzadas como pueden ser requerimientos de los niveles 4 y 5 de CMM. Los requerimientos básicos de nivel 2 generalmente son dejados a un lado como puede ser el asegurar la calidad y el proceso de administración de la configuración.

Para llevar a la práctica el proceso PSP no es necesario contar con herramientas especiales, todo lo que necesita son formas y hojas de trabajo, dependiendo del número de proceso, que son parte de la enseñanza del modelo.



### 2.4.3 TSP

La conformación de equipos es uno de los requisitos de la mayor parte de los proyectos de ingeniería. Aunque ciertos proyectos pequeños de software pueden ser realizados en forma individual, la complejidad de los sistemas actuales y la demanda de cortos tiempos de entrega es tal, que ya no es práctico para una sola persona encargarse de proyectos de software.

En los equipos de desarrollo de software, aunque las personas que los conforman suelen tener distintas especialidades, es necesario que todos trabajen en forma cooperativa en busca de un objetivo en común. Para realizar trabajo en equipo se necesita desarrollar habilidades específicas que permitan una correcta cohesión entre sus integrantes [Humphrey 00b].

Entre estas habilidades se encuentran la existencia de una meta definida y realista, la obtención de recursos adecuados para el trabajo, integrantes expertos y comprometidos con el logro del objetivo, trabajo disciplinado y constante. El desarrollo de estas habilidades es entre otras cosas lo que TSP ofrece al mercado de software.

TSP al igual que PSP es un proceso que consiste en un conjunto de métodos, formatos y reglas que ayudan al desarrollo de equipos de proyecto de gran calidad y madurez. TSP fue creado en 1996 y tiene como prerrequisito el conocer PSP ya que los conceptos individuales de éste son retomados e integrados al trabajo en equipo [Davis 03].

TSP es un proceso muy importante ya que junto con PSP y CMM conforman las herramientas ideales para ayudar a las organizaciones en la formación de sistemas de software de alta calidad. TSP es un proceso intermedio entre los dos anteriores ya que CMM mejora el proceso de toda la organización y se enfoca en la forma de administrar los proyectos mientras que PSP mejora las habilidades personales, crea un hábito de desarrollo y se enfoca a las personas. TSP conecta las dos anteriores ya que mejora el desempeño de los equipos y se enfoca en los productos y los equipos.

Una característica de gran importancia es que mientras PSP y CMM proveen una lista de las habilidades necesarias para llevar a cabo un proyecto de software efectivo, TSP es una guía para su realización. TSP está basado en un modelo “secuencial incremental” que divide el proceso de software en un conjunto de ciclos de desarrollo, donde cada ciclo incluye la producción de software que cumple algunas características de los requerimientos de software. El ciclo final reúne la integración y las pruebas del sistema completo.

Otro elemento importante de TSP es que para crear un equipo de trabajo efectivo se necesita la determinación y definición clara de roles para cada uno de los integrantes del mismo. Estos roles a grandes rasgos son los siguientes:

- **Líder de proyecto:** El líder de proyecto en general guía al equipo y se asegura que los ingenieros reporten las estadísticas de avance y que se complete el trabajo en la forma en que fue planeado.

- **Encargado de Desarrollo:** El encargado de desarrollo dirige al equipo en los asuntos de diseño y desarrollo del producto.
- **Encargado de Planeación:** El encargado de planeación guía al equipo en la planeación y seguimiento del producto.
- **Encargado de Procesos y Calidad:** El encargado de procesos y calidad ayuda al equipo en la definición de los procesos necesarios para la realización del sistema y en el establecimiento y administración de planes de calidad que den eficiencia al proyecto.
- **Encargado de Soporte:** El encargado de soporte ayuda al equipo a determinar, obtener y administrar las herramientas necesarias para cubrir las necesidades de tecnología y soporte administrativo.

## **2.5 CMM Based Appraisal for Internal Process Improvement (CBA IPI)**

### **2.5.1 Introducción**

La historia de CBA IPI es en realidad la historia de la SEI en el campo de las asesorías a externos. Como se mencionó anteriormente, la creación de métodos de asesoría por parte de la SEI nació de la necesidad del departamento de la defensa de Estados Unidos para mejorar los proyectos de desarrollo de software. Ellos pidieron un método que sirviera a sus contratistas para reducir los tiempos de entrega. Así nació el Cuestionario de Madurez de Humphrey [Humphrey 87] que sería el primer modelo en base al cual serían realizadas las primeras asesorías de la SEI.

Después de la creación del cuestionario, surgió un proceso hecho especialmente para llevar a cabo asesorías llamado Software Process Assessment (SPA). El objetivo de este era hacer llegar la tecnología a más personas. En 1991 sale al mercado el modelo CMM [Paulk 91, Weber 91] y con esto varias organizaciones buscan hacer modificaciones al SPA para que sirva como una herramienta de CMM. Es en 1995 cuando nace CBA IPI como el primer método de asesoría basado en el modelo CMM. Cabe mencionar que el SPA dejó de funcionar en 1996 después de comprobar que CBA IPI proporcionaba información más útil y confiable mientras que contiene la mayor parte de los aspectos benéficos de SPA [Dunaway 99].

### **2.5.2 Definición y características de CBA IPI**

CBA IPI es una herramienta de diagnóstico que permite a una organización llevar a cabo una introspección en su proceso de desarrollo de software identificando fortalezas y debilidades de sus procesos actuales, para relacionarlas a CMM. Esto con el objeto de dar prioridad a ciertas mejoras necesarias y a enfocarse en las mejoras de software que son de mayor beneficio, dado el nivel de madurez actual de la compañía y sus metas de negocios.

La forma en que CBA IPI realiza las asesorías a las empresas es a través de un Lead Assessor o asesor guía. Esta persona es la encargada de conformar equipos para generar resultados relacionados a los KPA de CMM dentro del marco de la asesoría. Todos los

procesos realizados por los equipos deben de seguir los lineamientos de CBA IPI en orden ya que si no se siguen así el proceso no será certificado como CBA IPI.

El asesor guía debe de entregar un reporte de los resultados obtenidos en la asesoría a la SEI. Estos resultados se guardan en un archivo de resultados cuya información es confidencial. Cabe resaltar que el hecho de que se entreguen resultados a la SEI no significa que ésta apruebe el proceso ya que la SEI no aprueba procesos de CBA IPI. Los resultados que reciben son usados para evaluar la efectividad del proceso y para buscar mejoras en campos en donde se hayan presentado contratiempos por parte del modelo.

Otro detalle importante es que para que una asesoría sea certificada CBA IPI se debe de contar con material adquirido en la SEI. El material debe de estar relacionado con el equipo asesor, el plan de asesoría, la recolección de datos y su validación, el rango donde se encuentra la empresa y los reportes necesarios.

### **2.5.3 SCE y CBA IPI**

Existe un proceso muy similar a CBA IPI llamado SCE (Software Capability Evaluation) en donde se obtienen resultados muy similares aunque su enfoque es muy distinto. La decisión del estudio a realizar depende de a quien están destinados los resultados y como los va a usar. Si los resultados del estudio son para uso interno y para encontrar mejoras en los procesos, siendo solicitados por la misma empresa entonces se utiliza CBA IPI. Este método proporciona un ambiente de colaboración y autoconocimiento.

Por el otro lado si el estudio se realiza a una compañía externa y los resultados del mismo son para propósitos personales (ya sea que se planee dar a conocer los resultados o no), entonces SCE es la herramienta recomendada. Este método es muy similar a una auditoría y generalmente no incluye a personas de la misma organización dentro de los equipos. Generalmente son realizadas por grandes organizaciones a sus proveedores potenciales para ver el nivel de desarrollo y desempeño de los mismos [Dunaway 95].

#### **2.5.4 Planeación de la asesoría**

La planeación es el paso inicial dentro de una asesoría CBA IPI. Lo primero que debe de hacer un líder asesor es resolver un cuestionario que lleva los detalles principales del proceso junto al coordinador de la asesoría dentro de la empresa. En este cuestionario se debe de determinar el alcance del estudio, los participantes, tiempos de finalización y las necesidades de la asesoría. El cuestionario va a evaluar dependiendo del conjunto de KPA que cumpla, es decir, en el nivel de CMM que cumpla todos los requisitos.

Por otro lado, debe de juntar distintos documentos como un organigrama, una lista de empleados, sus puestos y una lista de proyectos y sus características. Todo esto es necesario para determinar el número de proyectos que son representativos de la organización, qué proyectos deben de elegirse y que empleados deben de ser entrevistados para la elaboración de equipos. Los proyectos que se elijan deben de ser representativos de las áreas a evaluar

en la empresa ya que si no, no se podrán hacer las evaluaciones correspondientes para la asesoría.

Generalmente los equipos de trabajo están conformados entre 4 y 8 personas. En estos debe haber al menos una persona que forme parte de la compañía. Los integrantes del equipo deben de ser personas imparciales, entrenadas y con cultura de software. Además de esto, no deben de tener intereses representados en la asesoría como ser jefes de alguien que este siendo evaluado, obtener alguna recompensa si se lleva acabo la asesoría, etc.

El siguiente paso es hacer las citas correspondientes para aplicar un cuestionario a los administradores de proyecto y a otras personas cuya participación es necesaria dentro de la evaluación. Esto incluye un lugar para entrevistar, material necesario y tiempo suficiente.

Antes de iniciar la asesoría se debe de hacer un archivo con todos los documentos necesarios de cada proyecto. Este archivo sirve para acelerar el proceso ya que permite tener todo a la mano. Aunque no es necesario, ayuda para ver que tan organizado es un equipo de trabajo además que evita la omisión de documentos ya que si el equipo asesor no encuentra alguno, asumirá que no existe.

Este es el último paso antes del inicio de la asesoría. Es necesario realizar toda la planeación antes de la fecha de inicio.

### **2.5.5 Proceso de la Asesoría**

Para iniciar la asesoría con apoyo de las personas de la compañía, tanto integrantes del equipo como administrativos y empleados en general, el asesor da unas palabras con el objeto de infundir confianza sin importar los resultados del estudio. En esta generalmente se explica la importancia de una asesoría, algo de historia de CMM, la meta de una administración frente una asesoría, los KPA a evaluar, el equipo asesor y la gente que será entrevistada, lugar de la cita y horario.

Una vez iniciada la asesoría, el equipo asesor se dividirá en pequeños equipos (generalmente de dos personas) encargados de KPA específicos de modo que puedan trabajar mucha información en paralelo. Después del análisis se procesa la información recolectada y se presenta al equipo asesor completo.

Durante el proceso se debe de llevar a cabo el cuestionario de madurez a las personas seleccionadas. Esta información debe de ser evaluada por el equipo asesor y discutida en conjunto. Muchas veces estos cuestionarios se grafican para dar una mejor idea de los resultados. Otra forma de lograr mejores resultados es que cada integrante del equipo evalúe los cuestionarios y después se discutan en conjunto.

El equipo durante todo el proceso estará recopilando información acerca de las key practices de CMM para sustentar que han sido institucionalizadas e implementadas en forma correcta en la compañía. Si la documentación buscada se encuentra, simplemente



será validada por el equipo. Si no se encuentra se pedirá una explicación o alternativa durante las entrevistas. Para ésta práctica es bueno hacer una matriz de relaciones para llevar un control de la documentación.

Antes de realizar las entrevistas se debe de establecer un objetivo para estas, de modo que se recopile la información necesaria. El objetivo debe de buscar obtener información distinta a la recopilada anteriormente en busca de complementar el producto final. Los subequipos se encargan de redactar las preguntas que harán a sus entrevistados y una vez hechas las entrevistas, se evaluarán las respuestas en consenso por el equipo asesor en conjunto.

Al final de cada proceso se debe de hacer un vaciado de información a una base de datos en común de donde se obtendrán copias para todos los integrantes del equipo para que tengan todos la misma información.

### **2.5.6 Entrevistas**

Existen dos tipos de entrevistas conducidas durante el proceso de CBA IPI:

- **Entrevista Individual:** Sirve para obtener información de alguien que tiene conocimientos específicos acerca de la compañía, sus proyectos o funciones incluidas dentro del alcance de la asesoría. Este tipo de entrevista se lleva a cabo cuando existen temas difíciles que pueden ser discutidos durante la entrevista y no es conveniente que

sean discutidos en grupo. Al menos los administradores de proyectos serán entrevistados de esta forma.

- **Entrevista de representativos de un área funcional:** Esta se usa para obtener información acerca de grupos funcionales de personas como asesores de calidad, desarrolladores, encargados de pruebas, etc. Este grupo discute como su trabajo es desarrollado y los posibles problemas que se presentan.

Todo el equipo asesor participa de esta actividad en ambos casos y todos deben de tomar notas sobre lo que se dijo. Generalmente el proceso de entrevistas inicia con las individuales. En las entrevistas normalmente el asesor líder es quien dirige el proceso y algún integrante del equipo puede intervenir si se menciona algún tema relacionado al KPA que este investiga.

Después de la entrevista, cada integrante del equipo asesor deberá de revisar sus notas y poner en orden éstas de acuerdo al key practice al que pertenece. Esto debe de ser realizado mientras la entrevista esta fresca en la mente de los entrevistadores.

### **2.5.7 Proceso Final**

Cuando todas las entrevistas se han llevado a cabo, cada subequipo evaluará sus notas e integrará la información a su matriz de key practices. Ellos decidirán si recopilaron suficiente información para hacer alguna observación acerca de cada key practice que incluya toda la información obtenida.

Las observaciones se discuten en consenso con todo el equipo. El líder debe de guiar la discusión de modo que se analice cada punto y se verifique si es objetiva y válida. La aceptación de cada punto debe de ser votada y en caso de existir integrantes en contra se les da la oportunidad de convencer a los demás de su razón. Esta actividad es generalmente la que lleva más tiempo en la asesoría y por la cual deben de ser escogidos los integrantes del equipo en forma especial.

Una vez que se obtienen los resultados de las fortalezas y debilidades, éstas se ubican en cada KPA: Los resultados son mostrados a las personas entrevistadas para que proporcionen retroalimentación y tengan la última oportunidad de agregar información adicional al proceso de asesoría.

Ya que se tiene la información final se vacía en un documento de la SEI y es presentado en forma oficial a toda la organización. Cabe recalcar que toda información obtenida durante la asesoría es confidencial y que el líder asesor debe de promover esto dentro de los integrantes del equipo.

### **2.5.8 Prácticas Posteriores**

Ya que se obtienen los resultados finales de la asesoría se incluyen dentro de los reportes las fortalezas y debilidades de la organización. Las fortalezas representan las mejores

prácticas actuales de la organización y son tomadas en cuenta para ser usadas en forma generalizada en todos los proyectos.

Por otro lado, aunque las debilidades no necesariamente sean grandes problemas, si son un obstáculo a la hora de buscar procesos de calidad. Después de la asesoría una forma de mostrar el progreso es evaluando la eliminación de estas debilidades dentro del proceso.

Cabe mencionar que en el reporte se ponen las prácticas comunes, es decir que si se encuentran procesos que muestren características favorables o desfavorables pero solo se han realizado en uno o dos proyectos, no son tomados en cuenta.

Lo importante de esto es que aunque no sean parte de los resultados, las personas que estuvieron involucradas en el proceso, como integrantes de algún equipo de evaluación, si conocen esta información y puede ser de mucha utilidad. Las fortalezas representan una ruta potencial a las metas de calidad del programa actual mientras que las debilidades deben de ser polarizadas y asociadas a acciones preventivas para evitar su crecimiento.