

REFERENCIAS BIBLIOGRÁFICAS

- [1] Sommerville, I. Ingeniería del software. Madrid: Pearson Educación, 2005.
- [2] Léxico en programación orientada a objetos. Consultado en agosto 10 de 2010. Disponible: <http://ojarami.com/riosur.net/index.php>
- [3] Proyecto Cupi2, Universidad de los Andes. Facultad de Ingeniería, Departamento de Ingeniería de Sistemas y Computación. Consultado en junio 23 de 2010. Disponible: <http://cupi2.uniandes.edu.co/sitio/>
- [4] Gayo, D. et al. Reflexiones y experiencias sobre la enseñanza de la programación orientada a objetos como único paradigma. Actas de las IX Jornadas de Enseñanza Universitaria de Informática. Cádiz. Julio de 2003. Consultado en abril de 2007. Disponible: <http://www.di.uniovi.es/~dani/publications/jenui03.pdf>
- [5] Villalobos, J. y Casallas, R. Fundamentos de programación: aprendizaje activo basado en casos. México: Pearson Educación, 2006.
- [6] Booch, G., Rumbaugh, J. y Jacobson, I. El Proceso unificado de desarrollo de software. Madrid: Pearson Educación, 2000.
- [7] Botero, R., Castro, C., Taborda, G., Valencia, M. y Maya, J. Lógica y programación orientada a objetos: un enfoque basado en problemas. Grupo GIISTA. Medellín: Divegráficas, 2009.
- [8] XXVII Reunión Nacional y VI Encuentro Iberoamericano de Instituciones de Enseñanza de la Ingeniería. El profesor de ingeniería, profesional de la formación de ingenieros. Cartagena de Indias: ACOFI, 2007.
- [9] Sexto Simposio Iberoamericano en Educación, Cibernética e Informática- SIECI 2009, Orlando, Florida - EE.UU. Disponible: <http://www.iiis.org/CDs2008/CD2009CSC/CISC12009/>
- [10] Castro, C., Taborda, G., Botero, R. (2009). Método y entorno integrado de desarrollo para el aprendizaje en lógica de programación orientada por objetos. Revista Iberoamericana de Sistemas, Cibernética e Informática. [En línea]. 6, (2). Disponible: <http://www.iiisci.org/journal/risci/>

ARTÍCULO II

INTRODUCCIÓN A LOS MODELOS DE CALIDAD DEL SOFTWARE BASADOS EN PROCESOS

Darío Enrique Soto Durán

*Profesor de Tiempo Completo, Facultad de Informática
Tecnológico de Antioquia - Institución Universitaria
dsoto@tdea.edu.co*

Adriana Xiomara Reyes Gamboa

*Profesora de Tiempo Completo, Facultad de Ingeniería
Politécnico Colombiano Jaime Isaza Cadavid*

RESUMEN

En un mundo globalizado, en donde las organizaciones se ven enfrentadas a competencias de orden mundial, la calidad, además de aumentar la satisfacción general del cliente, disminuir costos y optimizar los recursos, se convierte en un importante punto diferenciador. Los productos o servicios que ostentan certificados de calidad son preferidos por los compradores porque transmiten seguridad y confianza. Esto también constituye un atributo de valor para las estrategias de comercialización en el exterior. Si bien la industria del software es nueva, ha tenido que madurar rápidamente, tal como lo exigen los avances tecnológicos y su alta participación al interior de las empresas. Esta industria comparte con las demás, el interés por la calidad y la competitividad. Lo anterior no debe ser ajeno a la academia, por el contrario, esta debe incorporar estas temáticas dentro de sus procesos de formación en los programas académicos del área tecnológica, con el fin de ofrecer profesionales competitivos en el país. Para esto, debe retomar y evaluar experiencias exitosas empleadas en otros países y aplicar las más viables en nuestro contexto.

Palabras clave: Calidad, Procesos, Software, Organización.

1. EVOLUCIÓN DE LA CALIDAD DEL SOFTWARE

A finales de la década de los sesenta se acuñó el término *Crisis del software* para referirse a la dificultad de construir programas libres de defectos, fácilmente comprensibles y que fueran verificables. La *crisis del software* resumió una serie de problemas que se venían observando en los procesos de desarrollo de software, tales como: los proyectos no terminaban en el plazo establecido, no se ajustaban al presupuesto inicial, baja calidad del software generado, software que no cumplía las especificaciones y código difícil de mantener que obstaculizaba la gestión y evolución del proyecto, entre otros.

A partir de este momento crucial en la evolución del desarrollo de software, nace formalmente la Ingeniería de Software y con ella la gestión de la calidad del software.

Al hablar de gestión de la calidad del software[1] es necesario incluir los principales factores que intervienen en el desarrollo del mismo: las personas, los procesos, el producto y el proyecto. Dentro de ellos, un factor

crítico son las personas, pues son ellas quienes finalmente logran el producto software, como resultado tangible del proceso de desarrollo.

En el ámbito internacional del software, a partir de la segunda mitad de los años ochenta, surgieron metodologías, modelos y estándares que abordan cada uno de los factores que intervienen en el proceso de desarrollo, cuya adopción muestra inicialmente un énfasis en aquellos orientados a los procesos, el producto y el proyecto. Un poco más tarde, surgen metodologías orientadas a medir los niveles de calidad de los procesos a los productos de software.

Al dar una mirada a la historia formal del aseguramiento de la calidad del software en Colombia, encontramos hitos importantes como la adopción de modelos para sistemas de gestión de la calidad y de mejoramiento del proceso de desarrollo de software, a partir de iniciativas del Estado y de la empresa privada; asimismo, de modelos de calidad de producto y de evaluación del mismo, con logros significativos a nivel organizacional, de proyecto y de producto, incluyendo certificaciones en diferentes niveles.

Con relación al factor persona, específicamente al desarrollador de software, a pesar de ser crítico para lograr el éxito de los proyectos de desarrollo, no se conocen iniciativas (tanto en el sector productivo como en el ámbito educativo) en cuanto a la adopción de modelos o metodologías orientados a mejorar sus competencias en la estimación y planeación de su trabajo, definir compromisos que se puedan cumplir, administrar la calidad de su trabajo y reducir el número de defectos en sus productos. Modelos como estos se han adoptado en otros países, como es el caso de la metodología Proceso de Software Personal y grupal denominada *Personal Software Process - PSP* y *Team Software Process - TSP* respectivamente.

2. MODELOS DE CALIDAD ORIENTADOS AL PROCESO

Es bien sabido que para desarrollar software de calidad de manera consistente se requiere contar con una alta madurez de procesos. A nivel internacional el modelo de madurez de procesos más popular es denominado **Capability Maturity Model Integration- CMMI**. Sin embargo, este modelo es complejo para implementar en empresas pequeñas. La estrategia para incrementar la madurez en la industria de software, no contempla solamente los procesos de las empresas, sino que incluye el mejoramiento del elemento básico que da sustento a la industria: las personas. Precisa-

mente en las personas se enfoca el Personal Software Process - PSP y a los equipos de desarrollo el Team Software Process - TSP, creados por el Dr. Watts Humphrey del Software Engineering Institute - SEI.

2.1 Proceso de Software Personal - PSP

El PSP es una línea de trabajo de medición y análisis para ayudarnos a caracterizar nuestro proceso. También es un procedimiento definido que nos ayuda a mejorar nuestro desempeño. Algunos de sus principios son [2]:

- La calidad de un sistema de software está dada por la calidad del proceso utilizado para desarrollarlo y mantenerlo.
- La calidad de un sistema de software está determinada por la calidad de sus componentes más deficientes.
- La calidad de un componente de software está dada por el individuo que lo desarrolla.
- El desempeño individual está dado por el conocimiento, la disciplina y el compromiso del individuo.
- Se pretende lograr que como profesional del software, conozcamos nuestro desempeño personal.
- Debemos medir, darle seguimiento y analizar el trabajo.
- Debemos aprender a partir de las variaciones del desempeño personal.
- Debemos incorporar las lecciones aprendidas en las prácticas personales.

La implementación del marco PSP, se realiza por fases como se muestra en la figura 7:



Figura 7. Niveles del proceso personal de software.

2.2 Proceso de Software en Equipo - TSP

El modelo TSP proporciona directrices para ayudar a un equipo a establecer sus objetivos, a planificar sus procesos y a revisar su trabajo con el fin de que la organización pueda establecer prácticas de ingeniería avanzadas y así obtener productos eficientes, fiables y de calidad.

El TSP provee un esquema de trabajo, donde cada desarrollador tiene perfectamente definidos sus roles, actividades y responsabilidades. Asimismo, el TSP incluye procedimientos para la mejora continua del proceso de desarrollo, para mejorar la calidad del software producido, para mejorar la estimación del tiempo de desarrollo, para la disminución de defectos en el producto y para promover la integración del equipo de desarrollo. Es decir, el TSP apoya tanto al equipo de desarrollo como a los administradores del proyecto para la culminación de proyectos de desarrollo de software, dentro del tiempo y presupuesto establecido (ver figura 8).

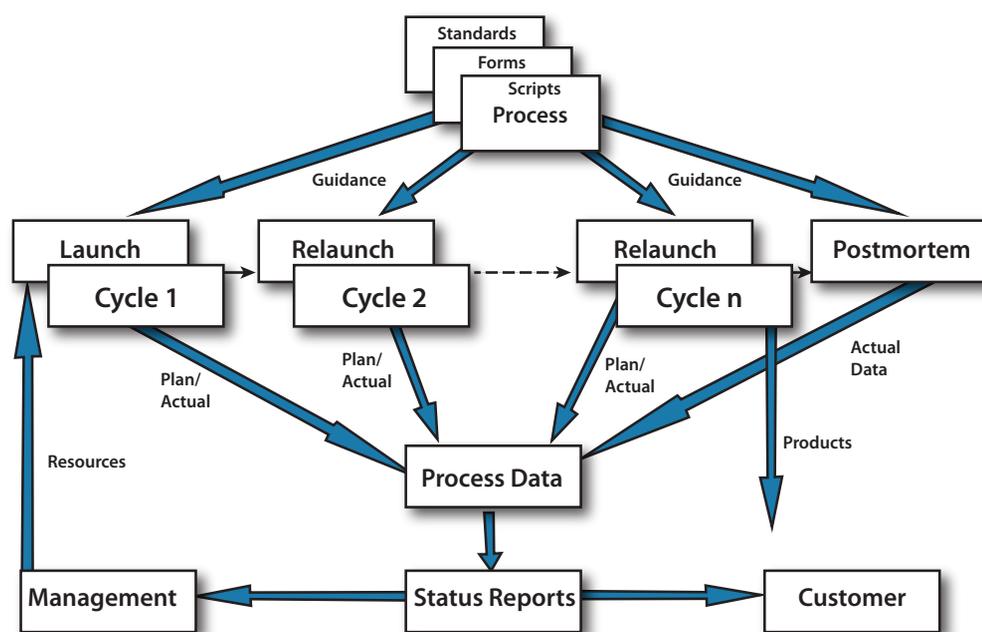


Figura 8. Flujo de proceso de TSP.

TSP se basa en los siguientes principios[3]:

- Un seguimiento preciso de un proyecto requiere planes bien detallados y datos precisos. Únicamente el personal que realiza el trabajo es capaz de recoger con precisión dichos datos.
- Para minimizar el tiempo del proyecto, los ingenieros deben equilibrar su carga de trabajo para maximizar la productividad, el primer foco de atención debe ser la calidad.
- TSP está formado por dos componentes primarios bien diferenciados que abarcan distintos aspectos del trabajo en equipo y que pueden observarse de manera resumida en la figura 9:

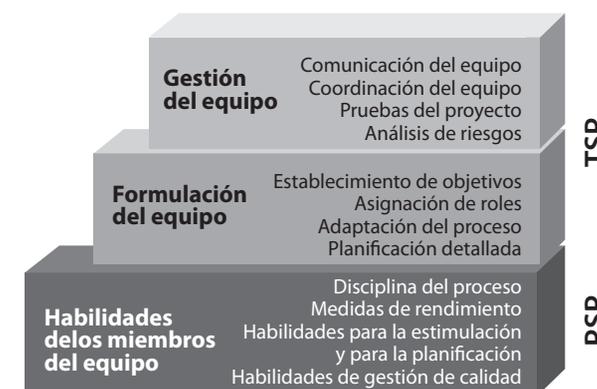


Figura 9. Aspectos de trabajo PSP y TSP.

2.3 CMMI

Este es un modelo para la mejora de procesos que proporciona a las organizaciones los elementos esenciales para procesos de desarrollo y mantenimiento de software. Durante los años 90, SEI desarrolló modelos para la mejora y medición de la madurez específicos para varias áreas[4]:

- CMM-SW: CMM for software
- P-CMM: People CMM.
- SA-CMM: Software Acquisition CMM.
- SSE-CMM: Security Systems Engineering CMM.
- T-CMM: Trusted CMM
- SE-CMM: Systems Engineering CMM.
- IPD-CMM: Integrated Product Development CMM.

Luego del uso y aplicación individual de estos modelos de madurez, el SEI desarrolló CMMI para facilitar y simplificar la adopción de forma simultánea de CMM-SW(CMM for Software), SE-CMM(SystemsEngineering Capability Maturity Model) e IPD-CMM(Integrated Product Development) y de ahí la palabra Integración en la sigla. Antes de CMMI el modelo más común era CMM-SW y se puede ver CMMI como la evolución de este último [5] (Figura 10).

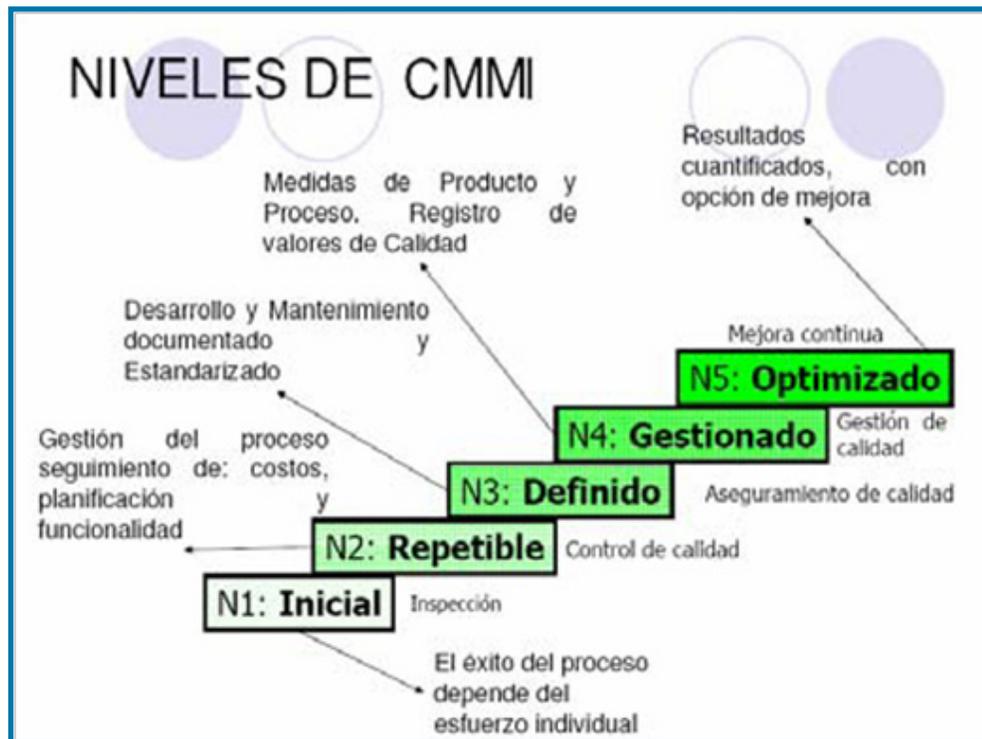


Figura 10. Niveles de madurez del modelo CMMI.

Este modelo presenta una estructura de cinco niveles de madurez, en los cuales una organización puede determinar su madurez en la producción de software en función de la consecución de los objetivos establecidos en cada nivel.

Según el nivel de madurez en que se encuentre la empresa, las medidas se enfocarán más al grupo de objetivos del nivel correspondiente, para que mejore la capacidad de producir software y pueda avanzar hacia el siguiente nivel.

Los niveles de madurez de una organización en CMMI son:

- Inicial.
- Gestionado.
- Definido.
- Gestionado cuantitativamente.
- Optimizando o en optimización continua

3. RELACIÓN ENTRE PSP/TSP Y CMMI

PSP forma técnicos de equipos establecidos con TSP en la mayoría de las prácticas genéricas de CMM/CMMI. TSP por sí solo, aunque sea aplicado a todos los equipos de desarrollo, no cubre todas las prácticas de cada área de proceso de CMM/CMMI, razón de más para que sea utilizado de forma complementaria a este modelo, no de forma aislada (ver figura 11).

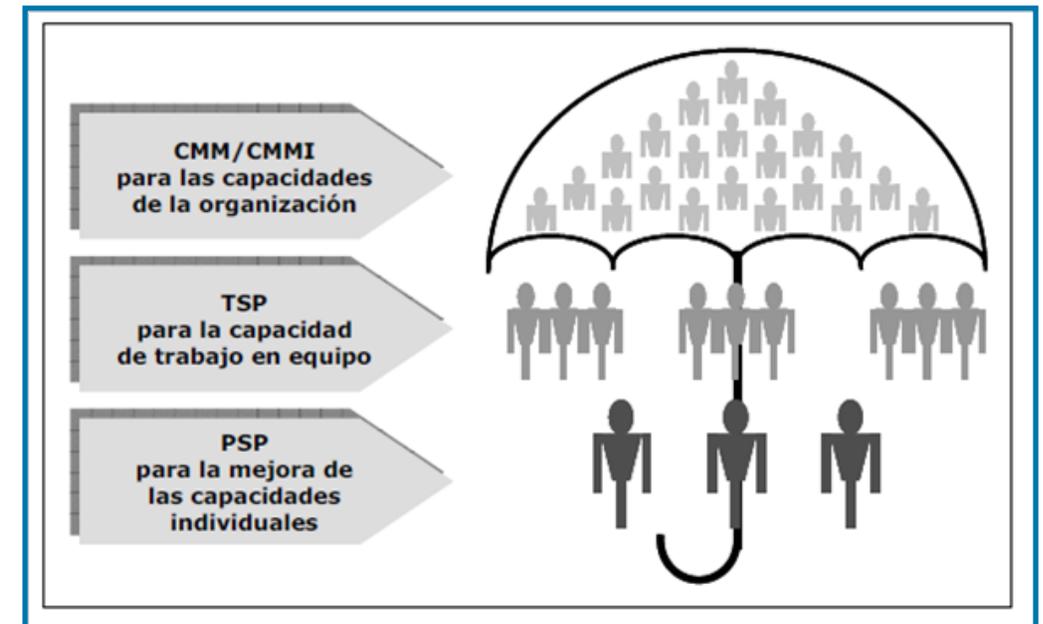


Figura 11. Relación de los modelos de calidad PSP, TSP y CMMI.

Debido a que las actividades de medición y análisis de los resultados son fundamentales en PSP y en TSP, su utilización durante la aplicación de CMM/CMMI en una organización, permite acelerar el progreso y aumentar el nivel de capacidad de la empresa en un tiempo menor que sin su uso.

El objetivo de cualquier esfuerzo de mejora de procesos de software es aumentar el rendimiento de la organización, y ello requiere cambios en el funcionamiento de los procesos de ingeniería, cualquier esfuerzo de mejora debe estar acompañado de pasos que puedan demostrar cambios en los comportamientos de ingeniería. TSP y PSP se utilizan para este fin. TSP genera una mejora substancial en el rendimiento de los grupos de software en una organización.

3.1. ¿Por qué incorporar PSP y TSP en una organización desarrolladora de software?

Las razones fundamentales para seleccionar los modelos de calidad denominados: Personal Software Process - PSP y Team Software Process - TSP son:

- La gran mayoría de las empresas que desarrollan software en nuestro contexto, son menores a 50 empleados.
- El modelo que utilizan nuestros competidores (CMMI) es complejo y apropiado para organizaciones grandes.
- El TSP/PSP, cuando se implementa correctamente, ha probado ser más eficaz que el CMMI Nivel 5.

Partiendo de estos modelos se puede escalar de forma acelerada a los diferentes niveles del modelo corporativo denominado CMMI. A este método de mejora acelerado en la implementación de las prácticas CMMI utilizando el modelo TSP como plataforma se denomina TCAIM.

El método TCAIM [6], permite responder los interrogantes propios de una mejora de proceso en una organización, como son: "¿qué cambiar?" y "¿cómo hacer esto?". Así: el primer interrogante, desde la visión holística derivada del modelo CMMI y el segundo interrogante, abordado por el control detallado que permite realizar los modelos TSP y PSP, desde las métricas definidas en el proceso de desarrollo.

4. ¿CÓMO INCORPORAR DESDE LA ACADEMIA ESTOS MODELOS DE CALIDAD EN LOS PROCESOS DE ENSEÑANZA?

Los modelos de calidad dentro de la oferta académica no deben ser islas, sino por el contrario estructurar el conocimiento técnico en el estudiante, para que le permita al profesional entrar en una cultura de automejoramiento proactivo y realizable.

Por esta razón se deben buscar estrategias en el aula que surtan efecto en las competencias asociadas a las áreas técnicas; articulando desde los primeros semestres, prácticas de programación y calidad propicias para su nivel de formación, y así vivenciar las prácticas de calidad.

En el ámbito académico vale la pena resaltar experiencias de orden internacional, como el caso de México, que ha implementado las siguientes estrategias:

- La universidad y la empresa deben fortalecer la relación existente generando espacios compartidos para la discusión y la búsqueda de oportunidades creando un lenguaje común alrededor de la calidad de software.
- La enseñanza de los conceptos de calidad debe iniciar con el modelo PSP, para que permitan al estudiante conocer los beneficios de utilizar un proceso definido y crear una cultura de autocontrol.
- Los métodos de calidad requieren de una madurez en el estudiante a nivel técnico y conceptual, que solo se alcanzan en los semestres finales, porque en este momento el estudiante ha consolidado los conocimientos de diseño y programación necesarios para avanzar en una mejora constante en su proceso de desarrollo, evidenciable y controlable desde las métricas de calidad reflejadas en el marco PSP a nivel de indicadores de productividad.
- A su vez, el modelo TSP integra las condiciones de mejora identificadas en el proceso de desarrollo a nivel individual, que solo serán apropiadas por el estudiante en un equipo de desarrollo siguiendo la dinámica que los proyectos reales imponen en este sector. Teniendo como premisa que el modelo TSP, es un marco complementario al modelo PSP, el cual le da visibilidad a este último en el ambiente empresarial. El estudiante solo se apropia de estas competencias realizando prácticas empresariales simulando un ambiente que demanda el dinamismo productivo.

REFERENCIAS BIBLIOGRÁFICAS

[1] Sistemas de gestión de calidad: conceptos y vocabulario (traducción certificada), ISO 9000:2000, ISO 2000.

[2] Watts S. Humphrey. PSP(sm): A Self-Improvement Process for Software Engineers. Addison Wesley, 2005.

[3] Watts S. Humphrey. Winning with Software: An Executive Strategy. Addison Wesley, 2001.

[4] CMMI. Calidad. Ingeniería del Software. Consultado en Septiembre de 2010. Disponible: <http://www.ingenierosoftware.com/calidad/cmmcmmi.php>

[5] CMMI Transition Plan. Consultado en Septiembre de 2010. Disponible: <http://www.sei.cmu.edu/cmmi/background/transplan.html>

[6] Miluk, Gene; McHale, Jim; y Chick, Tim. Accelerating CMMI Adoption with PSP/TSP-TCAIM. Proceedings of the TSP Symposium (September 2008). Consultado en Septiembre de 2010. Disponible: http://www.sei.cmu.edu/tsp_symposium, visitada en Septiembre de 2010.

ARTÍCULO III

ESTADÍSTICA BÁSICA EN EXCEL

Leonardo Ceballos Urrego

*Profesor de Tiempo Completo, Facultad de Informática
Tecnológico de Antioquia - Institución Universitaria
lceballos@tdea.edu.co*