

CREANDO UN PROFESIONAL CON DISCIPLINA EN EL PROCESO DE DESARROLLO DE SOFTWARE

Resumen / Abstract

Para desarrollar la industria cubana de software es preciso contar con procesos bien definidos y un personal competente, entrenado en una disciplina personal y de trabajo en equipo. El proceso de software personal (PSP) contribuye a alcanzar disciplina en el trabajo individual. Aquí se propone un conjunto de consideraciones para introducir las prácticas de PSP y el desarrollo de roles de forma paulatina e incremental con el objetivo de crear las bases para introducir métodos disciplinados de trabajo en equipo y cumplir con los requerimientos de la industria.

To develop the Cuban industry of software it is necessary to have very defined processes and a competent personnel, trained in a personal discipline and of work in team. The Process of Personal Software (PSP) contributes to reach discipline in the individual work. This paper proposes a group of considerations to introduce the practices of PSP and the development of roles in a gradual and incremental way with the objective of creating the bases to introduce disciplined methods of work in team and to fulfill the requirements of the industry.

Palabras clave / Key words

Disciplina personal, proceso de software personal, PSP, roles

Personal discipline, personal software process, PSP, roles

Margarita André Ampuero,
Ingeniera en Sistemas Automatizados
de Dirección, Profesora Auxiliar,
Centro de Estudios de Ingeniería de
Sistemas (CEIS), Instituto Superior
Politécnico José Antonio Echeverría
Cujae, Ciudad de La Habana, Cuba
e-mail: mayi@ceis.cujae.edu.cu

Yucely López Trujillo, Ingeniera
Informática, Instructora, CEIS,
Instituto Superior Politécnico José
Antonio Echeverría, Cujae, Ciudad de
La Habana, Cuba
e-mail: ylopez@ceis.cujae.edu.cu

Recibido: octubre del 2005

Aprobado: diciembre del 2005

INTRODUCCIÓN

La formación del ingeniero informático debe corresponderse con las exigencias de la Industria Nacional de Software, con el objetivo de fomentar su desarrollo. Este aspecto cobra mayor importancia en Cuba teniendo en cuenta que dentro de las proyecciones del gobierno se encuentra el desarrollo de la Industria Cubana del Software, de manera tal, que garantice el diseño y desarrollo de sistemas informáticos que beneficien a la sociedad y se cree un rubro de exportación que aporte a todos los programas del país.¹

Para desarrollar software de calidad no basta con tecnología, es preciso contar con un proceso de desarrollo de software bien definido y un personal capacitado y disciplinado. Dentro de las normas y modelos líderes en el tópico de la mejora de procesos se pueden citar el modelo de madurez de las capacidades (CMM), el modelo de madurez de las capacidades integrado (CMMi), el proceso de software personal (PSP), el proceso de software en equipo (TSP) y las normas ISO 9000, entre otros.

El CMM, desarrollado en el Software Engineering Institute (SEI)*, tiene como punto de partida la mejora del proceso de software y como meta, alcanzar un proceso maduro que permita el mejoramiento continuo. Pretende incrementar la capacidad administrativa de las organizaciones para predecir y controlar la calidad, la planificación, los costos, los tiempos y la productividad de los sistemas de software adquiridos o construidos. En el SEI, Watts S. Humphrey**, considerado el visionario detrás de CMMi, ha desarrollado el PSP con el objetivo de proveer un punto de partida a los individuos para seguir el camino de alto desempeño. El PSP es visto como una aplicación del modelo CMMi que provee directamente a los ingenieros de software de una metodología para la planeación y seguimiento de su trabajo con el objeto de producir consistente y eficientemente desarrollos de alta calidad.

Por su parte, el propósito fundamental del proceso de software en equipo instructivo (TSPi) es adiestrar a estudiantes e ingenieros en el uso de TSP. Este refina los métodos de CMM y PSP para guiar a los ingenieros en el desarrollo y mantenimiento de los equipos. Así, para las empresas que no tienen aplicado ningún método de mejora de proceso, aplicar PSP y TSP les permitirá arribar en menos tiempo a niveles altos de CMMi.

Este trabajo pretende demostrar la necesidad que tiene la industria de software nacional de contar con un personal disciplinado y competente y para ello se propone ofrecer una serie de consideraciones sobre cómo introducir gradualmente las prácticas de PSP y el desarrollo de roles durante el proceso de formación del profesional informático con vistas a entrenarlo para ser miembros efectivos de equipos de desarrollo y poder cumplir con las exigencias de la industria.

DISCIPLINA PERSONAL

Humphrey define el trabajo de la Ingeniería del Software como "la entrega de productos de alta calidad de acuerdo con un costo y un cronograma fijado". Precisa que hay tres aspectos claves para que se realice de manera efectiva: planificar el trabajo, trabajar acorde con un plan y esforzarse para producir productos de alta calidad.²

Sin embargo, esto exige disciplina y resulta muy difícil para los ingenieros de software desarrollar de manera regular un trabajo personal disciplinado debido principalmente a tres razones:³

- La ingeniería de software no tiene tradición de ejecución personal disciplinada.
- El proceso de software no le impone una disciplina natural a los ingenieros ya que el hecho de que el diseño de software no implique producción a gran escala no ha exigido una revisión minuciosa de este.

* Centro de Investigación y Desarrollo, creado por el Departamento de Defensa de los EE.UU., asignado en 1984 a la Universidad de Carnegie Mellon.

**Watts S. Humphrey, científico e investigador del Software Engineering Institute (SEI). De 1959 a 1986 estuvo asociado con IBM Corporation donde fue director del Departamento de Proceso y Calidad.

- A pesar de que el trabajo disciplinado en cualquier campo siempre ha exigido contar con buenos estándares y soporte competente, la industria de software carece de una adecuada formación para el desempeño de los diferentes roles.

Con vistas a lograr cambiar esta actitud es preciso que los ingenieros comprendan la necesidad de utilizar métodos disciplinados, conozcan cómo aplicarlos y constaten que el empleo de los métodos realmente mejora su trabajo. Por lo tanto, para alcanzar disciplina en el trabajo individual es preciso: usar un proceso personal definido, planificar cada tarea, registrar tiempos, tamaños y defectos, seguir la ejecución del proceso y medir la calidad del producto.

PROCESO DE SOFTWARE PERSONAL

El diseño del PSP se basó en los siguientes principios de planificación y mejora de la calidad:⁴

- Cada ingeniero es diferente y para ser más eficiente debe planificar el trabajo basándose en datos tomados de su propia trayectoria profesional.
- Los ingenieros deben usar procesos personales bien definidos y cuantificados para mejorar realmente su trabajo.
- Cada ingeniero debe responsabilizarse con la calidad de los productos que desarrolla.
- Cuanto antes se detecten y corrijan los defectos menos esfuerzo será necesario.
- Resulta más efectivo evitar los defectos que detectarlos y corregirlos.
- Trabajar bien es siempre la forma más rápida y económica de trabajar.

El PSP está estructurado en niveles y cada uno incorpora buenas prácticas incrementalmente, exigiendo del personal una mayor disciplina en el desempeño del trabajo individual. La figura 1 muestra la estructura incremental de PSP.

Asimilar conscientemente PSP contribuye a mejorar el control y la administración del tiempo, facilitando la elaboración de planes de trabajo más reales. Además, posibilita la realización de mejores estimaciones de tamaño y permite medir y prevenir los defectos en las diferentes fases del desarrollo del software.

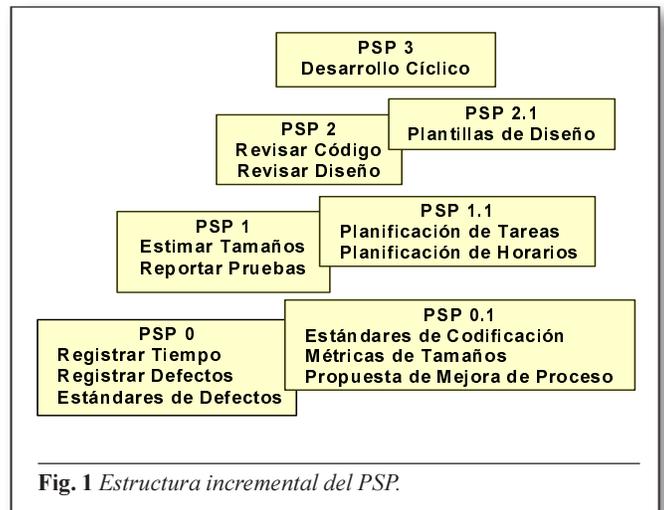


Fig. 1 Estructura incremental del PSP.

En fin, ayuda a adquirir una disciplina en el proceso de desarrollo de software creando paulatinamente un compromiso estrecho con la calidad.

Análisis del proceso de formación del ingeniero informático

La industria cubana de software ha tomando conciencia de la importancia de contar con procesos de desarrollo de software bien definidos que garanticen un buen uso de los talentos y recursos con que cuenta y que estos procesos mejoren de forma continua. Pero para lograr las mejoras deseadas resulta vital entrenar al personal en una disciplina individual y de trabajo en equipo en el desarrollo de software, y a su vez, entrenar a los directivos con el objetivo de cambiar la cultura de las organizaciones, de modo tal, que promuevan y soporten el trabajo disciplinado. Sin embargo, no se cuenta con cursos que garanticen el entrenamiento requerido y en ocasiones se aplican los modelos y metodologías internacionales tal y como están concebidas, sin tomar en cuenta la idiosincrasia y las particularidades de la realidad cubana.

La industria espera por un profesional que tenga una visión total del proceso de desarrollo de software y que posea una disciplina para enfrentar este proceso. Sin embargo, la universidad carece de cursos que cubran los conocimientos acerca del proceso entero, ya que los semestres académicos resultan muy cortos para garantizar que los estudiantes desarrollen todo el ciclo de vida de un proyecto, con independencia de si los proyectos son reales o académicos.

Por otra parte, la carrera está enfocada en la contribución individual de los profesionales que forma, y carece de cursos para la construcción de equipos y la coordinación intergrupal. En el desarrollo de los proyectos no se ejercitan los roles. Temas como planificación, asignación de recursos, estimación de costos, tamaños y tiempos, definición y trabajo con estándares, calidad, entre otros no son abordados de forma profunda. Por lo tanto, es deber tanto de los profesionales de la industria como de las universidades afrontar este reto.

Así, al analizar el proceso de formación del ingeniero informático se puede concluir lo siguiente:

- Las implementaciones del plan de estudio de la carrera están concebidas, de forma tal, que en los primeros años se imparte un ciclo de programación donde apenas se introducen algunos aspectos relacionados con la ingeniería de software y solo se ejercita con fuerza el rol de programador. Además, no se desarrolla una disciplina en el desarrollo personal de software. Es a partir del segundo semestre de 3er. año que se comienzan a enseñar las prácticas de la ingeniería de software, por lo que resulta muy difícil tratar de cambiar los hábitos adquiridos por los estudiantes en los años previos.

- Se imparte un curso de PSP en el último año, donde ya resulta difícil para los estudiantes convertir las prácticas de PSP en hábitos de trabajo. A pesar que ellos comprenden la importancia y utilidad de las prácticas de PSP les resulta engorroso aplicarlas. En primer lugar porque aprendieron a enfrentar las diferentes

fases del proceso sin planificar ni registrar tiempos, tamaños y defectos por lo que el esfuerzo y el tiempo que dedican a estas tareas lo consideran una pérdida en lugar de una inversión en aras de mejorar su propio proceso. En segundo lugar, la falta de una herramienta de apoyo a la enseñanza que facilite el registro de las métricas hace tedioso la aplicación de PSP.

ESTRATEGIA PARA LOGRAR DISCIPLINA PERSONAL EN LOS EGRESADOS

A partir del análisis del contenido de las asignaturas que se encuentran dentro del plan de estudio de la carrera de ingeniería informática en Cuba, y con el objetivo de lograr que los nuevos profesionales adquieran una disciplina personal y de trabajo en equipo se tienen las siguientes consideraciones iniciales:

- La industria no necesita que los nuevos graduados conozcan PSP y TSP en detalle, sino que tengan hábitos de organización y que estén preparados para planificar, medir y documentar sus actividades en aras de mejorar el proceso de desarrollo.

- Las prácticas de PSP deben impartirse y aplicarse tan pronto como sea posible pero paulatina e incrementalmente, de forma tal, que se dosifique su introducción y que la disciplina adquirida se torne un hábito debido a su empleo durante el resto de la carrera.

Consideraciones sobre las prácticas a introducir en el primer año de la carrera

- Resulta muy difícil para los estudiantes de primer año asimilar la temática de administración de tiempo, ya que se encuentran ante el desafío de aprender a programar en un ambiente de desarrollo y con un lenguaje determinado, y además, no tienen idea de lo que significa un proyecto ni de su magnitud en cuanto a tamaño, tiempo de desarrollo y esfuerzo.

- En el primer año de la carrera, específicamente desde la primera asignatura de programación, se debe introducir la temática de prevención de defectos. Sin embargo, al analizar el estándar de tipos de defectos propuesto por PSP, y tomando en cuenta los conocimientos que poseen los estudiantes en ese nivel, se concluye que solo podrán identificar defectos sintácticos y lógicos. El resto de los tipos de defectos incluidos en el estándar deberán incorporarse paulatinamente en las siguientes asignaturas de programación. Lo importante es darle la noción al estudiante de lo que es un defecto, los tipos de defectos y la importancia de registrarlos y prevenirlos. Los estudiantes desde su primera tarea deben demostrar que son capaces de depurar su código y de elaborar un reporte de defectos. A partir de este momento los estudiantes deben registrar defectos de forma rutinaria.

- También desde el primer curso de programación los estudiantes deben ser capaces de escribir código siguiendo un estándar definido en el lenguaje utilizado. Esta disciplina debe tornarse en un hábito por el resto de su carrera. Además, deben ser capaces de elaborar su propio estándar de estilo de

código, de forma tal, que lo definan y apliquen en los proyectos de cursos, las prácticas profesionales y en el trabajo de diploma. Un buen estándar de codificación facilita la lectura y entendimiento del código y también garantiza que las revisiones al código, el mantenimiento, el reuso y el proceso de depuración sea más fácil.

- Partiendo del estándar de codificación definido, los estudiantes deben aprender a revisar su código, tratando de encontrar defectos antes de la primera compilación. El concepto de lista de chequeo debe ser introducido y los defectos encontrados aplicando las listas de chequeo en las revisiones deben ser registrados con vistas a demostrar el valor que tienen las revisiones y la importancia de trabajar por la calidad desde las primeras fases de desarrollo del software.

- Se propone introducir el registro del tiempo a partir del segundo semestre del 1er. año. El objetivo no es que los alumnos sepan planificar un proyecto sino que dominen en qué emplean su tiempo y que comprendan lo importante que resulta registrarlo para poder administrarlo adecuadamente. Por el momento solo será necesario que registren el tiempo que dedican a las clases, al estudio y preparación de trabajos y tareas, así como al desarrollo del proyecto de curso.

- Dado que uno de los roles principales a desarrollar en el 1er. año de la carrera es el de probador se debe introducir el reporte de prueba. Los estudiantes deben elaborar los casos de prueba asociados al proyecto de programación, ejecutar las pruebas y reportar los resultados obtenidos.

Por lo tanto, un estudiante de 1er. año debe conocer y ejercitar las habilidades de los roles de programador y probador debiendo ser capaz de definir y aplicar un estándar para escribir código, registrar el tiempo planificado para el desarrollo de proyecto de curso y los tiempos reales dedicados a cada fase del desarrollo, así como elaborar los casos de prueba, ejecutar las pruebas, registrar los defectos y escribir un reporte de defectos.

Consideraciones sobre las prácticas a desarrollar a partir de segundo año de la carrera

- Un rol que comienza a desarrollarse a partir de 2do. año es el de diseñador. Los estudiantes deben aprender a diseñar y revisar el diseño tan pronto como sea posible. Aquí los estudiantes implementan algoritmos más complejos donde el diseño preliminar les ahorrará tiempo. El diseño puede ser documentado utilizando artefactos como el diagrama de clase aunque sin un grado de formalización elevado.

- Las métricas de tamaño y los métodos de estimación deben ser introducidos en las asignaturas de Ingeniería de Software a partir del 3er. año de la carrera. Un rol importante a desarrollar a partir de este año es el de analista.

- En 4to. año el estudiante ya tiene una visión completa del ciclo de vida del proyecto por lo que ya está listo para recibir un curso formal de PSP. Para este curso se debe contar

con el apoyo de una herramienta que automatice dichas prácticas, con el objetivo de no cargar al estudiante con el registro de métricas y el posterior procesamiento requerido para su análisis. Se sugiere el uso del Personal Process Dashboard.^{5*} A partir de este año los estudiantes están listos para desarrollar el rol de arquitecto de software.

- En el primer semestre de 5to. año se debe impartir un curso de TSP, los estudiantes cuentan con disciplina personal y están capacitados para asumir distintos roles. En este curso un mismo estudiante debe ejercitar más de un rol y se propone utilizar proyectos académicos por lo complejidad que introduce el trabajo con proyectos reales.

CONCLUSIONES

1. La industria de software exige de un personal competente y disciplinado y entrenar con PSP a los estudiantes de la carrera ingeniería de ingeniería informática sin dudas puede contribuir con la disciplina personal que se requiere.

2. Introducir paulatina e incrementalmente a partir del 1er. año las buenas prácticas propuestas en PSP, y exigir su uso consistente durante la carrera constituye la base para formar un miembro eficiente de un equipo de trabajo.

3. La introducción con éxito de PSP requiere una buena herramienta de apoyo y el Personal Process Dashboard constituye una excelente propuesta. ☐

REFERENCIAS

1. **FEBLES, A.:** "Modelo de referencia para la gestión de configuración", Tesis de doctorado, CEIS, Cujae, Ciudad de La Habana, 2004.
2. **HUMPHREY, W.:** *Introducción al proceso de software personal*, Addison Wesley, Madrid, 2001.
3. ———: *Introduction to the Team Software Process. TSPi*, Addison Wesley Longman, USA, 1999.
4. ———: *The Personal Software Process. Technical Report CMU/SEI-2000-TR-022*, November, 2000.
5. sourceforge.net Software Process Dashboard Initiative. Disponible en: <<http://processdash.sourceforge.net>>

* Software libre de apoyo al PSP con módulos que incluyen los guiones y formularios que aparecen en el libro de Watts Humphrey: *A Discipline for Software Engineering*.