

Título:

Introducción de técnicas del Personal Software Process desde los primeros años en la formación del ingeniero informático.

Autora:

Ing. Yeleny Zulueta Véliz.

Afiliación y direcciones:

Ingeniera Informática.

Profesora de la Universidad de las Ciencias Informáticas. Cuba.

yeleny@uci.cu

yeleny@gmail.com

Resumen:

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. Este trabajo propone un conjunto de consideraciones para introducir las prácticas de PSP, de forma paulatina e incremental en la formación del profesional en la universidad; con el objetivo de crear las bases para introducir métodos disciplinados de trabajo en equipo y cumplir con los requerimientos de la industria. Se exponen algunas experiencias de su implementación en el ámbito de la Universidad de las Ciencias Informáticas. Además se propone la herramienta Process Dashboard, para soportar las prácticas de disciplina personal.

Palabras Claves:

Personal Software Process, PSP, Proceso Software Personal, Process Dashboard.

Introducción

La informática está llamada a convertirse en un campo boyante internacionalmente, en una de las ramas más productivas para el país [CASTRO, 2004]. Para vencer esta meta es inevitablemente lograr el respaldo de un sólido sistema de educación superior donde es trascendental el lugar que ocupa la Universidad de las Ciencias Informáticas (UCI). Sobre el objetivo de este centro Fidel Castro Díaz-Balart planteó:

“El propósito fundamental es lograr un centro de excelencia para la formación masiva de profesionales de nivel superior. Ello debe alcanzarse con la ejecución de ambiciosos programas curriculares y de producción y con la aplicación de las más modernas tecnologías en la docencia.” [CASTRO, 2004]

Los estudiantes de la UCI generalmente se encuentran vinculados a proyectos productivos que ayudan a complementar sus prácticas profesionales a partir del segundo año de la carrera. Un papel de vital importancia en esta tarea lo juega la disciplina de estos futuros ingenieros para el cumplimiento de los plazos de entrega de resultados. Los estudiantes deben ser capaces de saber administrar su tiempo correctamente, priorizar sus actividades y, sobre todo, reajustar estas variables para no afectar el producto final. Ellos son quienes nutrirán la industria de software cubana y el desarrollo del país en esta rama depende en gran medida de que los ingenieros cumplan sus compromisos de trabajo.

Como ya ha sido señalado, es a partir del segundo año cuando los estudiantes se vinculan a la producción, sin embargo, no es hasta el cuarto año que el plan de estudio aborda con profundidad el tema de cómo aplicar métodos avanzados de ingeniería a sus tareas diarias para desarrollar las habilidades personales que le permitan disciplinar su trabajo como desarrolladores de software; aunque en Programación 1 se incluía el tema, la forma no fue la mejor y los resultados no fueron exitosos.

En la mayoría de las profesiones el trabajo competente requiere el uso de prácticas establecidas, planes y procedimientos que traen orden y eficiencia a cualquier trabajo y permite a los trabajadores concentrarse en producir productos de la más alta calidad.

Precisamente el Personal Software Process o Proceso Software Personal, puede ser usado por futuros ingenieros de software como guía para un enfoque disciplinado y estructurado en el desarrollo de software.

En nuestra universidad ya se han dado los primeros pasos para introducción paulatina de estas buenas prácticas para el desarrollo de software, pero uno de los aspectos que han restado vitalidad a la propuesta ha sido la resistencia al control de los datos necesarios, también en este trabajo se propone la utilización de la herramienta Process Dashboard, para apoyar esta tarea.

Desarrollo

Fundamentación del tema.

El 70% de los costos en el desarrollo de software se atribuye a costos personales, habilidades, experiencia y los hábitos de cada ingeniero determinan los resultados en el desarrollo de software. Esta relación del ingeniero con los resultados del proceso de desarrollo de software es la premisa en la que se basa el Proceso Software Personal.

Después de la segunda guerra mundial, la estrategia de calidad en la mayoría de las organizaciones industriales se basaba casi por completo en las pruebas. Las empresas establecieron departamentos especiales de la calidad para encontrar y arreglar problemas después de la producción de los productos. No fue sino hasta los años 70 y los años 80 que la industria estadounidense se centró en mejorar la forma en la que la gente hacía sus trabajos y desarrollaban sus procesos.

En los siguientes años este enfoque a los procesos de trabajo produjo mejoras importantes en la calidad de muchos productos y la estrategia tradicional de "prueba-y-arregla" fue reconocida como costosa por el desperdicio de tiempo y de materiales.

Un paso de avance en este sentido lo constituyó la introducción, primero de las inspecciones del software en 1976 y posteriormente en 1987, del Modelo de Capacidad de Madurez (CMM). Después de que Watts S. Humphrey condujera el desarrollo inicial de este modelo para software, se decidió a aplicar los principios de CMM a los programas pequeños porque mucha gente preguntaba cómo aplicar CMM a las organizaciones pequeñas o al trabajo de los equipos reducidos de software. PSP nace como respuesta a esta interrogante, ampliando el proceso de mejora a quienes realizan el trabajo de desarrollo de software, concentrándose en las prácticas de trabajo de los ingenieros en una forma individual. [PELAEZ, 2003]

Los primeros cursos de PSP comenzaron a impartirse en 1994 en la universidad de Carnegie Mellon, por su descubridor y por otros profesores en la Universidad Aeronáutica de Embry. A finales de este año se publica el primer libro que describía el PSP. [SEI, 2002]

¿Por qué PSP?

El desarrollo de productos de software implica mucho más que escribir instrucciones de programación juntas y ejecutarlas en un ordenador. Requiere cumplir requisitos del cliente a un costo y planificación acordada [HUMPHREY, 2001]. El PSP muestra cómo producir de forma regular software de alta calidad. Utilizando el PSP se obtienen datos que muestran la efectividad del trabajo y se identifican los puntos fuertes y las debilidades, además se practican habilidades y métodos que ingenieros del software van a desarrollar durante muchos años de pruebas y errores.

El PSP enseña a ingenieros y futuros ingenieros, cómo administrar la calidad de sus productos y cómo hacer compromisos que ellos puedan cumplir. Puede ser empleado en muchas fases en el ciclo de desarrollo de programas pequeños, definición de requerimientos, documentación, pruebas y mantenimiento. [VELASCO, 2003]

El diseño de PSP se basa en los siguientes **principios** de planeación y de calidad [HUMPHREY, 1995]

- Cada ingeniero es esencialmente diferente; es decir, los ingenieros deben planear su trabajo y basar sus planes en sus propios datos personales.
- Para mejorar constantemente su funcionamiento, los ingenieros deben utilizar personalmente procesos bien definidos y medidos.
- Para desarrollar productos de calidad, los ingenieros deben sentirse personalmente comprometidos con la calidad de sus productos.
- Para hacer un trabajo de ingeniería de software de la manera correcta, los ingenieros deben planear de la mejor manera su trabajo antes de comenzarlo y deben utilizar un proceso bien definido para realizar de la mejor manera la planeación del trabajo.
- Para que los desarrolladores lleguen a entender su funcionamiento de manera personal, deben medir el tiempo que pasan en cada proceso, los defectos que inyectan y remueven de cada proyecto y finalmente medir los diferentes tamaños de los productos que llegan a producir.

Fundamentación del objetivo.

Si bien el trabajo disciplinado en cualquier esfera siempre ha demandado el uso de buenos estándares y un soporte sólido, la industria de software no tiene tradición de un desempeño disciplinado y tampoco incluye este aspecto en su enseñanza de manera profunda.

La industria cubana necesita profesionales con una visión total del proceso de desarrollo de software y con la disciplina para enfrentarlo. Conocimientos y disciplina deben adquirirse desde la universidad, pero en la carrera de Ingeniería Informática no se ha logrado que los estudiantes cubran de forma profunda y completa el proceso de desarrollo del software, ya sean sencillos o complicados, reales o académicos; como tampoco se ha logrado que los egresados apliquen las prácticas del trabajo en equipo siendo este un requisito indispensable en la mayoría de los proyectos de ingeniería.

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.

La disciplina, en el proceso de desarrollo de software, se define como una actividad o ejercicio que desarrolla o mejora habilidades. Los estudiantes tienden a verla como una limitación pero los profesores deben enseñarles a

asumirla como un marco de trabajo para aprender y mejorar personalmente: entre el ingeniero y los resultados del proceso de desarrollo de software hay una relación muy estrecha.

Aunque el PSP se introduce generalmente en programas de ingeniería de software para profesionales ya graduados, sus principios pueden ser aprendidos y practicados por estudiantes del curso inicial. Para ello las actividades son propuestas con un conjunto de elementos del PSP que un estudiante de primer año puede fácilmente asimilar junto con los temas más tradicionales del primer año de programación, pues están diseñadas para introducir de manera gradual los pasos del PSP que descubren cómo gestionar y controlar el tiempo y cómo planificar y controlar el trabajo de forma tal que los compromisos puedan ser cumplidos.

La gestión del tiempo en la UCI reviste mayor importancia que en otras instituciones de Educación Superior, teniendo en cuenta que este es un centro involucrado directamente y con alta responsabilidad en la vida política del país y en la Batalla de Ideas. Nuestros estudiantes deben ser capaces de saber planificar su tiempo y sus proyectos, ya sean pequeños softwares como los que deberán enfrentar en 1er año o los que deberán asumir durante la carrera o después de egresados; esto les ayudará a adaptarse a los cambios que implica el dinamismo que caracteriza la escuela.

Tanto PSP como TSO (Team Software Process: Proceso de Software del Equipo) requieren la recopilación y análisis de métricas con un elevado nivel de detalle. Una vez que los datos necesarios han sido recopilados al nivel preciso, el análisis estadístico de los mismos, permite efectuar una planificación realista, rastreo, predicción, y control de los proyectos y productos de software. Esta recopilación y análisis de métricas, sin embargo, no es algo trivial. En cualquier proyecto real, el empleo de herramientas de apoyo a PSP y TSP se convierte en un elemento a considerar muy importante. Aunque estudios han demostrado que las personas pueden mantener su productividad, al utilizar PSP sin el empleo de herramientas de apoyo, el “factor de frustración” inherente a tal hecho, pone a prueba la paciencia de incluso los ingenieros más disciplinados, haciendo la aplicación de PSP difícil de sostener.

Herramientas de apoyo a PSP

Las investigaciones en este tema hasta el momento, se dividen en tres generaciones. En la tabla se muestran algunas de sus características [HUMPHREY, 1994].

La primera generación hacía uso del PSP como se describe originalmente por su creador en *A Discipline for Software Engineering*: los usuarios creaban e imprimían tablas o formularios que llenaban de forma manual con gran esfuerzo y cansancio por lo trabajoso del llenado de datos.

En la segunda generación se usaban herramientas automatizadas como Leap, PSP Studio o PSP Dashboard. Estas básicamente tienen las mismas prestaciones: cajas de diálogo donde el usuario registra los datos de esfuerzo, tamaño

y defectos, aunque también se pueden mostrar análisis si el usuario lo desea. Esta segunda generación, sin dudas minimiza el trabajo del usuario en la recolección de datos así como en su análisis para la toma de decisiones.

Con el desarrollo de Hackystat en Mayo del 2001, el tema evoluciona hacia la 3ra generación con una herramienta que colecciona automáticamente las métricas mediante sensores adjuntos a las herramientas de desarrollo, los datos son enviados por los sensores al servidor y independencia de los análisis pueden ser enviados mensajes de alerta, por ejemplo, a los desarrolladores.

Características	1ra Generación (PSP manual)	2da Generación (Leap, PSP Studio y PSP Dashboard)	3ra generación (Hackystat)
Dificultad en la recopilación de datos	Alta	Media	Nula
Dificultad para el análisis	Alta	Baja	Nula
Cambios en las métricas	Simple	Software	Dependiente de la herramienta

Como se planteó anteriormente, la herramienta recomendada por sus características es el Process Dashboard (tablero de instrumentos de proceso). Fue desarrollado originalmente en 1998 por la fuerza aérea de Estados Unidos, y ha continuado desarrollándose bajo modelo open-source. Está libremente disponible para la descarga bajo condiciones de la GNU Public License (GPL: Licencia Pública de GNU), siendo la última versión la 1.6.5 de diciembre de 2005.

El Process Dashboard es un software libre de apoyo a PSP con módulos que incluyen los guiones y formularios que aparecen en el libro de Watts Humphrey “A discipline for Software Engineering”, donde se describe el Proceso Personal de Software. El PD fue desarrollado en Java, y necesita para correr en cualquier sistema de una Máquina Virtual Java. Corre en sistemas Windows, Linux, Unix, y Macintosh. Para acceder a los formularios y guiones se requiere disponer de algún navegador web.

Después de tener instalado el ambiente para correr programas Java, se ejecuta el fichero de instalación nombrado ***pdash-install-offline-1-6-5.jar*** y se siguen las indicaciones pertinentes. Lo más importante es ubicar los ficheros de instalación generalmente en *C:\Program Files\Process Dashboard* y señalar el lugar donde serán guardados los datos históricos.

El PDB en un espacio muy pequeño el tablero de instrumentos da el acceso a una especie de cronómetro como el contador de tiempo para las actividades de la sincronización, un diálogo de la entrada del defecto para la información del defecto que captura, scripts y formas para seguir procesos definidos, un “checkbox práctico de la terminación” que permite una manera rápida y fácil de decir a la herramienta que una fase del desarrollo sea completa, menús que permiten la navegación con la jerarquía de la estructura de la interrupción del trabajo de proyecto, y un menú de configuración que permite el acceso a otras partes de la herramienta.



De manera general el PD es un software que resulta muy conveniente usar para fines docentes en la formación de profesionales con hábitos de disciplina personal y por el personal encargado de enfrentar el proceso de desarrollo de software en las empresas cubanas. Uno de los aspectos fundamentales es que es un software libre, y su uso estaría al margen de las políticas dictadas al respecto en Cuba. En este trabajo solo se describieron las funcionalidades que brinda el software fundamentalmente en el manejo y control, de tiempo y defectos, pero el software brinda otras funcionalidades muy importantes, tal es el caso de posibilitar realizar estimaciones de tiempo y tamaño basadas en datos históricos.

Algunas experiencias.

La asignatura Práctica Profesional 1, corresponde al primer año de la carrera Ingeniería Informática en la Universidad de las Ciencias informáticas. Su misión fundamental radica en la ejercitación y desarrollo, por parte de los estudiantes, de los roles, habilidades y funciones que deberán desplegar en su vida como profesional de la informática. Es por ello que el acercamiento a las técnicas de PSP se presenta como parte de esta asignatura y comienza a ponerse en práctica a partir del curso 2005/2006.

Durante este período se produjo una introducción al PSP pues no se trató por completo, no se abordan, por ejemplo, las técnicas estadísticas para hacer estimaciones con exactitud ni análisis de datos, defectos y calidad del software. Solo se pretende que los estudiantes se familiaricen con el control del tiempo, no solo de trabajos de programación sino de la vida en general, que interioricen la necesidad de gestionar el tiempo, los compromisos y las programaciones como una vía para mejorar la calidad del software.

Con la introducción se pretende que los estudiantes reconozcan el valor de registrar los datos de esfuerzo y tamaño y que puedan usar estos datos en la planificación de sus proyectos y analizar su efectividad personal. La estimación de trabajos futuros estaría basada precisamente en la referencia de trabajos anteriores. Retrasar la introducción de PSP, podría influir en que los estudiantes adquieran prácticas de programación indisciplinadas y en cuarto año sería mucho más difícil que entendieran y, sobre todo, aplicaran las bases de una correcta disciplina personal para el desarrollo de un software de calidad en los plazos y costes comprometidos; por otra parte, el cambio podría ser traumático si el acercamiento no es gradual.

En el cuarto año de la carrera, en la asignatura Gestión de Software, se tratan estos temas, cuando 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 y además ha adquirido conocimientos y habilidades importantes desde asignaturas como, por ejemplo, Probabilidades y Estadística e Ingeniería de Software. 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 Process Dashboard.

Es irrefutable el hecho de que la capacidad de los equipos de ingeniería para desarrollar productos de calidad de forma eficiente y eficaz, depende en gran medida de la capacidad individual de los ingenieros [HUMPHREY, 2001]. La implementación de la propuesta también proporciona las bases para la posterior introducción de cuestiones más avanzadas de tipo individual y de equipo.

Conclusiones

La disciplina en el proceso de desarrollo de software es, sin lugar a dudas, uno de los elementos fundamentales para tal propósito por lo que nuestros estudiantes deben comenzar a entenderla y aplicarla desde el primer año de la carrera.

Ante la tarea de convertir la informática en una de las más importantes ramas productivas del país, la UCI tiene el compromiso de formar ingenieros capaces de asumir y cumplir sus compromisos de trabajo con la más alta calidad.

Con la introducción de PSP desde los primeros años de la carrera Ingeniería Informática y de forma gradual, los futuros ingenieros informáticos del país inferirán la necesidad de saber gestionar correctamente sus tiempos y compromisos, no solo para el trabajo que desempeñarán sino para otras facetas de su vida.

El uso del Process Dashboard puede ayudar a facilitar el proceso de recopilación y obtención de datos y métricas para PSP.

Referencias Bibliográficas

- [CASTRO, 2004] CASTRO DÍAZ-BALART, Fidel. *Ciencia, Tecnología y Sociedad. Hacia un desarrollo sostenible en la Era de la Globalización*. Editorial Científico-Técnica, La Habana, 2004. (segunda edición)
- [HUMPHREY, 1994] HUMPHREY W. *A Personal Commitment to Software Quality*. 1994. En <http://www.sei.cmu.edu/publications/documents/95.reports/95.ar.psp.qual.html> [29 Junio 2006]
- [HUMPHREY, 1995] HUMPHREY, Watts S. *A Discipline for Software Engineering*. Addison-Wesley, 1995.
- [HUMPHREY, 2001] HUMPHREY, Watts S. *Introducción al Proceso Software Personal*. Addison Wesley, 2001. ISBN 84 7829 052 4
- [PELAEZ, 2003] PELAEZ RAMÍREZ, Juan José. El modelo de capacidad de madurez y su enfoque al proceso personal de software (PSP). Capítulo 2. Proceso Personal de Software. http://www.pue.udlap.mx/~tesis/lis/pelaez_r_jj/capitulo2.pdf [13 Julio 2005]
- [SEI; 2002] Software Engineering Institute. *Publications*. [en línea]. <http://www.sei.cmu.edu/publications/publications.html> [09 Julio 2006]
- [VELASCO, 2003] VELASCO, Perla Inés. *PSP: Una alternativa para mejorar los procesos del software*. LANIA. Año 12, Vol. 39 y 40. 2003. <http://www.lania.mx/biblioteca/newsletters/2003-primavera-verano/psp.html> [02 Junio 2006]
- [ZABALZA, 1987] ZABALZA, M. *Diseño y desarrollo curricular*, Madrid, Nancea, 1987.

Bibliografía

- BENEDITO, J.: *La formación universitaria a debate*, Barcelona, Servicio de Publicaciones de la Universidad, 1995.
- CABERO ALMENARA, Julio. *Nuevas tecnologías, comunicación y educación*. EDUTEC, *Revista Electrónica de tecnología educativa*. Núm. 1. Febrero, 1996. <http://www.uib.es/depart/gte/edutec-e/revelec/revelec1.html> [24 Junio 2006]
- CARNEGIE MELLON UNIVERSITY. *What is PSP?* <http://www.sei.cmu.edu/tsp/psp.html> [24 Junio 2006]
- CASTELLS, M. y otros. *El desafío tecnológico. España y las nuevas tecnologías*, Madrid, Alianza Editorial, 1996.
- CEDILLO, Karina. *Personal Software Process (PSP)*. <http://correo.cimat.mx:1666/Investigacion/Computacion/infoIS.php> [24 Junio 2006]
- CEPES Colectivo de autores. *Tendencias pedagógicas contemporáneas*. Ediciones Mercadu. Habana. 2004.
- CHAPELA, Leonardo. *Calidad y competitividad en la industria del software*. Newsletters Primavera Verano, Año 12, Vol. 39 y 40. 2003. <http://www.lania.mx/biblioteca/newsletters/2003-primavera-verano/psp.html> [24 Junio 2006]
- DANILOV M, A, y M. N. Skatkin. *Didáctica de la escuela Media*. Editorial Pueblo y Educación, 1981.
- FUENTES GONZÁLEZ, Homero Calixto. *El método del proceso de formación de los profesionales*. 2004.
- FUENTES GONZÁLEZ, Homero Calixto. *El contenido de enseñanza*. 2004.
- FUENTES H., ÁLVAREZ S. *Dinámica del Proceso Docente Educativo en la Educación Superior*, CeeS "M. F. Gran". U. O. 1998
- GARCÍA BATISTA, Dr. Gilberto. *Compendio de Pedagogía*. Pueblo y Educación, La Habana, 2002.
- HAYES, W. OVER, J. *The Personal Software Process: An empirical study of the impact of PSP on individual engineers*. (CMU/SEI-97-TR-001), Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1997. www.sei.cmu.edu/publications/documents/97.reports/97tr001/97tr001abstract.html [24 Junio 2006]
- JACOBSON, Ivar; BOOCH, Gardy y RUMBAUGH, James. *El Proceso Unificado de Desarrollo de Software* Addison Wesley Madrid, 2000. ISBN 84-7829-036-2
- ROSELL PUIG, Washington y MÁS GARCÍA, Martha. *El enfoque sistémico en el contenido de la enseñanza*, 2004.
- ZILBERSTEIN TORUNCHA, José, et. all. *Preparación Pedagógica Integral para profesores universitarios*. Editorial Félix Varela, La Habana, 2003.