

TSP/PSP en Uruguay

Luego de este análisis uno se puede preguntar: ¿Si las inspecciones y revisiones son tan efectivas y tan poco costosas, por qué normalmente no se hacen?

Hay es 30 de octubre y estoy escribiendo este artículo que trata sobre dos procesos creados por Watts Humphrey. Lamentablemente, Watts falleció el día 28 de octubre de 2010; hace sólo dos días. Actualmente me encuentro en el SEI, el golpe de esta noticia fue enorme para la comunidad de ingeniería de software y, por supuesto, para todos sus compañeros del SEI. Sin lugar a dudas, Watts es el padre de la calidad de software y uno de los más grandes investigadores de la ingeniería de software de toda la historia. Tuve la oportunidad de conversar con él dos veces y de intercambiar varios correos electrónicos. Una persona estupenda que también será recordada por su calidad y calidez humana. ¡Gracias Watts!

Introducción

En el primer artículo "TSP/PSP en Uruguay", publicado en el número anterior de esta revista, comentamos que la principal prioridad en la gestión de negocios de software debe ser la calidad. También mencionamos que esta no sucede por casualidad. El Team Software Process es un proceso de desarrollo que al ser utilizado obtiene sistemáticamente productos de calidad, cumpliendo con el cronograma y los costos establecidos. Cada individuo que integra un equipo TSP utiliza el Personal Software Process (PSP) para realizar sus actividades. Aquí presentamos algunos resultados del uso del PSP y del TSP basándonos en dos reportes técnicos del Software Engineering Institute (1) (2).

¿Cómo se aprende el PSP?

El PSP se aprende mediante un curso teórico/práctico. El mismo no es un curso de programación, y debido a esto, tiene como requisito previo el manejo fluido de al menos un lenguaje de programación. Durante el desarrollo del curso se busca cambiar los hábitos del ingeniero e incrementar sus habilidades en distintos aspectos del desarrollo de software.

Dado que es complejo introducir el proceso completo desde el comienzo del curso, este se va introduciendo por niveles: 0, 1 y 2. El nivel 2 es el PSP completo.

Durante el curso los ingenieros desarrollan 10 programas de software. Los tres primeros se realizan con el PSP 0, los tres siguientes con el PSP 1 y los cuatro restantes con el PSP 2.

Resultados en cursos del PSP

Los datos que aquí se presentan corresponden al dictado de 23 cursos del PSP con 298 ingenieros en total; el estudio completo se presenta en el artículo de Hayes y Over (1). En esos cursos se escribieron más de 300.000 líneas de código (Loc) en 15.000 horas y los ingenieros descubrieron y corrigieron unos 22.000 defectos. Los cursos fueron impartidos tanto en ambientes académicos como en la industria.

La figura 1 presenta cómo mejoran los ingenieros en la estimación del tamaño (izquierda) y en la estimación del esfuerzo (derecha) a medida que van realizando los ejercicios. La línea negra es el error promedio que tuvieron los ingenieros al realizar las estimaciones de cada ejercicio. La línea azul presenta lo mismo pero por nivel del PSP.

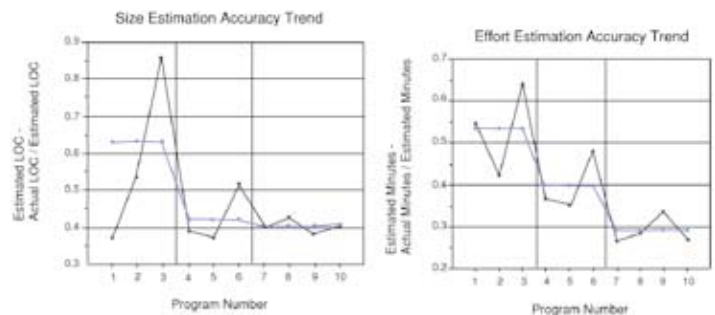


Figura 1 – Estimación de tamaño y esfuerzo

La figura 2 (izquierda) muestra cómo disminuye durante el curso la cantidad de defectos removidos por cada mil líneas de código (KLoc) durante las etapas de compilación y pruebas unitarias. La figura 2 (derecha) presenta la productividad. Al comenzar las prácticas del PSP la productividad disminuye, pero luego aumenta y se equipara a la productividad original.

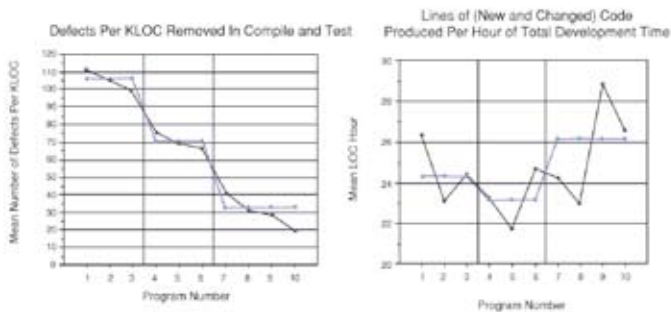


Figura 1 – Etapas del desarrollo de software

Estos resultados se dan en todos los cursos del PSP. Luego de incorporar las prácticas del proceso los ingenieros estiman mucho mejor tanto el tamaño de los productos como el tiempo que lleva construirlos.

Además de mejorar en la estimación, los ingenieros disminuyen los defectos que “llegan” a la fase de compilación y de prueba. Esto quiere decir que logran remover los defectos de forma temprana: fases de revisión del PSP. Remover los defectos tempranamente mejora notablemente la calidad del producto final, el cronograma y los costos de un proyecto. Como presentamos en el artículo anterior las pruebas tienen un costo varias veces mayor que las revisiones. Disminuir la cantidad de defectos que se tienen que remover durante las pruebas hace que el costo del proyecto disminuya sustancialmente. Por último, el incorporar las prácticas del PSP no afecta la productividad del ingeniero.

Los productos desarrollados con PSP son los componentes de un sistema desarrollado con TSP. El tiempo de re-trabajo disminuye significativamente, debido a la calidad de cada componente, y por lo tanto aumentará la productividad del equipo TSP.

En definitiva: mejores estimaciones, cumplimiento del cronograma, productos de calidad y a menor costo.

Resultados del TSP en la industria de software

Los resultados que obtienen equipos que usan el TSP son sustancialmente mejores a los encontrados normalmente en la industria de software. Estos resultados se repiten a medida que distintas compañías de software adoptan el TSP. La siguiente situación se describe en el artículo de Noopur Davis (2):

El producto se liberó en la semana 62, esto fue 4 semanas después del compromiso establecido con el cliente un año antes (6,9% de desviación del cronograma). Además, el producto se entregó con más funcionalidades de las acordadas. El tamaño del mismo fue de 28.900 líneas de código. La cantidad de defectos durante las pruebas de sistema fue de 0,44 defectos por KLoc. Comparado con el desarrollo de la versión anterior, la cual no usó TSP, se redujo 10 veces la cantidad de defectos detectados por el grupo de pruebas y se redujo 8 veces la duración de las pruebas de sistema. Esta era la primera vez que el equipo de desarrollo usaba el TSP.

En ese mismo artículo se analizan datos de varios proyectos realizados en 13 compañías de software que usan el TSP. El tamaño de los productos en esos proyectos varía desde 600 Loc hasta 110.000 Loc, el tamaño de los equipos varía desde 4 hasta 47 ingenieros y la duración de los proyectos desde unos pocos meses a unos 2-3 años. Los tipos de aplicaciones incluyen software de tiempo real, aplicaciones cliente-servidor y software financiero, entre otros (2).

El análisis de estos proyectos indica que el promedio de desviación en el cronograma fue de un 6% y el rango de desviación fue desde un -20% (liberación temprana) hasta un 27% (liberación tardía). Según el CHAOS Report de 2009 de The Standish Group International, los proyectos se desvían en promedio un 79% del cronograma. La diferencia con TSP es de un orden de magnitud y el proyecto TSP con mayor desvío es aún mejor que el promedio de la industria.

Además, según el CHAOS Report 2009, el 24% de los proyectos fracasan (se cancelan o el producto no es utilizado). Todos los proyectos que usaron el TSP en el estudio de Noopur Davis finalizaron y se consideraron exitosos dentro de sus compañías. En lo que refiere a calidad, se tuvo como promedio 0,4 defectos por KLoc en las pruebas de sistema (con varios proyectos sin detectar defectos). Los productos liberados contienen alrededor de 0,06 defectos por KLoc, dos órdenes de magnitud menos que la media de la industria que es de 7,5 defectos por KLoc. El cuadro 1 compara datos del uso de TSP con el promedio de la industria.

Medida	Proyectos TSP Promedio y rango	Proyectos típicos Promedio
Defectos en pruebas de sistema (defectos/KLoc)	0,4 De 0 a 0,9	15
Defectos en producción (defectos/KLoc)	0,06 De 0 a 0,2	7,5
Esfuerzo en pruebas de sistema	4%	40%
(% del esfuerzo total del desarrollo)	De 2% a 7%	
Cronograma de las pruebas de sistema (% de la duración total del desarrollo)	18% De 8% a 25%	40%

Cuadro 1 – Calidad: comparación entre el TSP y la media de la industria

Los integrantes de un equipo TSP

También existen varios reportes acerca de cómo se sienten los ingenieros que trabajan en equipos TSP. En este artículo comentamos sólo un caso (para muestra basta un botón).

En una entrevista que Humphrey tuvo con el presidente de Kaiser Electronics, este último le comentó acerca de la rotación de personal: "En Silicon Valley normalmente hay un 25% de rotación, el año pasado nosotros tuvimos un 23%. En software, nuestra rotación fue de 7%, sin embargo, en los equipos que ya tenemos el TSP adoptado fue de 0%" (3).

¡A los ingenieros les gusta mucho trabajar con TSP, son "dueños" de los compromisos del equipo, se mantienen motivados, contentos y altamente productivos! En el artículo de Noopur Davis se presentan comentarios de integrantes de los equipos TSP de las 13 compañías relevadas. Vale la pena leerlos.

Otros resultados del uso del TSP

Además de los mencionados aquí, existen diversos reportes que indican las mejoras logradas con el TSP. Uno interesante es el escrito por McAndrews, donde compara los resultados de algunas compañías antes de adoptar el TSP y luego de adoptarlo (4).

Compartimos, con aproximadamente 125 colegas provenientes de unas 30 empresas y de unas 5 universidades, el placer de asistir al "TSP Symposium 2010". Año a año en este simposio se presentan resultados del uso del TSP en la industria. Las presentaciones realizadas están disponibles en la web: <http://www.sei.cmu.edu/tspsymposium/2010/proceedings.cfm>.

Diversas empresas de distintas partes del mundo están usando TSP. Algunas de las más conocidas son: Adobe Systems, Bursatec, Fuji Xerox, IBM, Intuit, Kernel Technologies, Microsoft, Mitsubishi, Nedbank y Oracle, Toshiba.

Vale la pena resaltar que Capers Jones, reconocido experto en ingeniería de software, indica en su último libro que el TSP es el mejor proceso de desarrollo para proyectos de mediano y

gran tamaño, y el segundo mejor proceso para proyectos de pequeño tamaño (5).

Hay que considerar que cuando un producto pequeño es un éxito se espera una segunda versión del producto, luego una tercera, etc. Probablemente, alguna versión pase a ser de tamaño medio. Utilizar el TSP desde el inicio es una opción a tener en cuenta.

Resumiendo

Hemos mostrado resultados del PSP durante su aprendizaje y también resultados del uso del TSP en la industria. Los resultados son contundentes: se mejora la calidad del producto final en 2 órdenes de magnitud y se cumple tanto con el cronograma como con los costos establecidos. Además, los ingenieros de un equipo TSP están muy motivados y son altamente productivos. Estos magníficos resultados se obtienen porque el PSP y el TSP se basan en mejorar la forma de trabajo del individuo y del equipo que desarrolla software. Los equipos TSP logran una sinergia particular y son estos equipos los verdaderos responsables del éxito. En próximos artículos presentaremos cómo TSP logra crear equipos con individuos altamente calificados, motivados y comprometidos a alcanzar el objetivo definido, en un ambiente de total cooperación entre los integrantes.

Las exigencias de los clientes son cada vez mayores. Sobre todo en lo que refiere a calidad de software, entrega en plazo y dentro de los costos establecidos. El TSP tiene resultados excelentes en la industria de software, Capers Jones lo considera el mejor proceso existente y está siendo adoptado por numerosas compañías. ¿Será el 2011 el comienzo del TSP en la industria de software del Uruguay?

Bibliografía

1. Hayes, Will y Over, James W. The Personal Software Process: An Empirical Study of the Impact of PSP on Individual Engineers. Software Engineering Institute, 1997. CMU/SEI-97-TR-001.
2. Davis, Noopur y Mullaney, Julia. The Team Software Process in Practice: A Summary of Recent Results. Software Engineering Institute, 2003. CMU/SEI-2003-TR-014.
3. Humphrey, Watts S. Winning with Software: An Executive Strategy. Addison-Wesley Professional, 2001.
4. McAndrews, Donald R. The Team Software Process: An Overview and Preliminary Results of Using Disciplined Practices. Software Engineering Institute, 2000. CMU/SEI-2000-TR-015.
5. Jones, Capers. Software Engineering Best Practices: Lessons from successful projects in the top companies. McGraw Hill, 2010.

*Diego Vallespir
Profesor de Facultad de Ingeniería, Udelar
TSP Uruguay - Grupo de Ingeniería de Software
dvallesp@fing.edu.uy*