

Implementación de metodología TSPi en centros universitarios de desarrollo de *software*

TSPi implementation in software development centers for academic purposes

José Alberto Vela Dávila^{1*}

¹ Instituto Tecnológico Superior de Fresnillo / Universidad Politécnica de Zacatecas
Fresnillo, Zacatecas, México
* Correo-e: veladavila@gmail.com

PALABRAS CLAVE:

procesos, planeación, estándares, TSPi, métricas, metodologías

RESUMEN

Actualmente los ingenieros en sistemas no sólo deben ser expertos en tecnologías de la información, además deben poseer formación como ingenieros de *software* capaces de atender las demandas actuales de las organizaciones en cuanto a modelos y estándares de calidad. En este artículo se describe la experiencia de la implementación de la metodología Team Software Process (TSP) para el desarrollo de *software* en la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico Superior de Fresnillo. Se instruyó a los estudiantes de la carrera en cuanto a procesos. Además se incluye el análisis del comportamiento de los estudiantes al ser integrados a equipos de trabajo dentro de centros de desarrollo de *software* con librerías de procesos establecidas. Por lo tanto, fueron disciplinados para trabajar siguiendo estándares, respetando tiempos, planificaciones y haciendo uso de algunas métricas para medir el desempeño de los proyectos. Con ello adquirieron bases para la gestión y seguimiento de un proyecto de *software*, con la intención de cubrir las necesidades actuales de la industria.

KEYWORDS:

process, schedules, standards, TSPi, metrics, methodologies

ABSTRACT

Nowadays, the computer systems engineers not only have to be experts in information technologies, but also must be formed as software engineers capable to attend the present demands from different organizations, referring to models and quality standards. In this paper it is presented the process implementation on software development done in the Computer Systems Major of a College or University, using the TSPi methodology. The paper shows the experience on preparing the students of Process Major, including the students behavior at the moment of being integrated to working teams within the Software development centers with libraries of processes established, then after this, they are disciplined to work following different standards, respecting times, schedules using metrics to measure the performance of the projects, giving the basis to the management and the tracking of a software project covering so the present needs of the software industry.

Recibido: 7 de febrero de 2015 • **Aceptado:** 25 de junio de 2015 • **Publicado en línea:** 30 de octubre de 2015

1 INTRODUCCIÓN

Hace algunos años, cuando los proyectos de *software* eran pequeños, el desarrollo del *software* se hacía de forma artesanal, es decir, parecía innecesario el uso de metodologías, estándares y menos procesos que incluyeran las mejores prácticas. A medida que los proyectos fueron cada vez más grandes en tamaño y complejidad, las compañías acumularon fracasos y las posibilidades de éxito de los proyectos disminuyeron drásticamente, tal como lo muestran las estadísticas del libro *A discipline for software engineering* [1].

Los proyectos de desarrollo de *software* generalmente comparten un problema: “la crisis del *software*”. Un término con el que muchos autores engloban los problemas comunes de los proyectos de *software*, como: costos de desarrollo que sobrepasan lo presupuestado; *software* que presenta una gran cantidad de errores, y tiempos de desarrollo o entrega fuera del calendario estipulado [2].

Debido a lo anterior, las organizaciones de desarrollo de *software* requieren alternativas para abordar estos problemas. Como una solución viable se propone la implementación de la metodología Team Software Process (TSP) o proceso de desarrollo de *software* en equipo. Ésta permite enfrentar el desafío de la crisis del *software* a través del incremento en la productividad, disminución de defectos de los equipos, disminución de los errores insertados y el cumplimiento de la calendarización en el tiempo estipulado. Se eligió TSP porque es una metodología que ha mostrado ser provechosa, como lo muestra el reporte de Noopur Davis y Julia Mullaney [3], en el cual se presentan los beneficios obtenidos de 20 proyectos TSP en 13 organizaciones alrededor del mundo.

El reto que se encuentra al implementar TSP es que se aplica en equipos de trabajo que pertenecen a empresas grandes e importantes; por lo tanto se asume que ya estaban entrenados en y dominaban la metodología. Con este artículo se contestará a la pregunta *¿qué pasa cuando se instruye a estudiantes de Ingeniería en Sistemas Computacionales para desarrollar software utilizando procesos por primera vez, específicamente TSP?*

De acuerdo con Humphrey [4], antes de adoptar TSP se debe entrenar a cada uno de los miembros de los equipos en proceso personal para el desarrollo de *software* (PSP, por sus siglas en inglés) [4]. Esto les

permitirá aprender una disciplina personal de trabajo, los introduce al uso de procesos, así como al empleo y seguimiento de métricas. A partir de lo anterior, en el Instituto Tecnológico Superior de Fresnillo se decidió dar formación a dos equipos de trabajo como prerrequisito. Cada equipo estuvo integrado por tres hombres y una mujer con edades entre 20 y 22 años. Se les capacitó primero en PSP y después en TSP con el desarrollo de un proyecto por equipo. Este artículo muestra el resultado e impacto de la formación de estos estudiantes de la carrera de Ingeniería en Sistemas en el uso y aplicación de esta metodología orientada a procesos. Es importante resaltar que para agilizar el registro de datos y su posterior análisis, algunas de las formas de TSP se sustituyeron con el uso de Process Dashboard. Éste es una herramienta de *software* libre creada con el propósito de facilitar el uso de TSP y PSP, y de esta manera disminuir la curva de adopción de estas metodologías [5]. Esta herramienta fue desarrollada originalmente por la defensa de los Estados Unidos en 1998 y ha continuado su desarrollo bajo la dirección de Tuman Solutions (empresa de desarrollo de *software*). En la tabla 1 se muestran las funcionalidades y fortalezas de esta herramienta.

Este documento está estructurado como sigue: la primera sección muestra las metodologías de procesos utilizadas; en la siguiente se describe la experiencia obtenida durante el desarrollo de este proyecto, que incluye la formación de los estudiantes miembros de los equipos, el escenario y el proceso de desarrollo del proyecto; en la penúltima sección se realiza el análisis de los datos, y finalmente se presentan las conclusiones.

2 METODOLOGÍAS DE PROCESOS

Las metodologías de procesos TSP y PSP que se describen a continuación fueron creadas por Watts Humphrey para proporcionar un marco de referencia en el desarrollo de *software*. Proveen lineamientos sobre los procedimientos, así como la adopción de estrategias para el uso de métodos de desarrollo que sirvan tanto para el programador como para el equipo [3].

- *Proceso personal para el desarrollo de software (PSP). Es una disciplina de trabajo. El PSP está compuesto de formas y estándares; provee un marco definido para medir, analizar y administrar*

el trabajo personal. El PSP divide el desarrollo de un programa en seis etapas: planeación, diseño, codificación, compilación, pruebas y *postmortem*. Durante el desarrollo de este programa se registran los datos o métricas, como el tiempo de desarrollo o el tamaño de los programas. Posteriormente los datos son analizados y utilizados como una forma de mejorar el desempeño personal [1]. PSP también refiere cómo estimar y planear proyectos además de administrar la calidad.

- *Proceso para el desarrollo de software en equipo (TSP)*. Es un marco de trabajo para desarrollo de software y un proceso estructurado para construir y guiar equipos de trabajo [6, 7]. Es una guía paso a paso para lograr un proyecto de software en equipo, define claramente los roles que cada miembro debe desempeñar, así como sus responsabilidades [8].
- *Introducción al Proceso para el desarrollo de software en equipo (TSPi)*. Es una versión académica de TSP. Es un proceso definido que proporciona un marco de desarrollo y para esto utiliza formas, procesos, estándares y métodos necesarios para desarrollar productos de software de alta calidad [9][8].

3 EXPERIENCIA

A. Formación de los estudiantes

En esta sección se describe el proceso seguido para la formación de los alumnos en TSPi, que se muestra en la figura 1. Como se puede apreciar, primero se

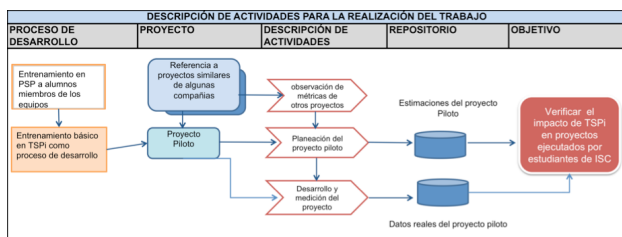


Figura 1. Resumen del trabajo realizado

entrenó a los miembros de los equipos en el uso del proceso personal de desarrollo de software (PSP), y se proporcionó una introducción básica al uso de procesos siguiendo TSPi.

Una vez terminada la formación, se ejecutó el proyecto piloto, durante el cual se registraron tanto los datos estimados como los reales a través de una herramienta de seguimiento y administración de proyectos llamada Process Dashboard. Finalmente se realizó el análisis del impacto en los proyectos.

Para realizar este análisis, al no contar con datos históricos, pues se está arrancando con un nuevo proyecto piloto, se decidió consultar algunos artículos, como [2, 9, 10], entre otros. En ellos se reportan resultados de la ejecución de proyectos utilizando TSPi, que sirven como referencia para poder medir el impacto del proyecto.

B. Escenario de aplicación

El Instituto Tecnológico Superior de Fresnillo, conocido por sus siglas ITSF, es un tecnológico descentralizado que se encuentra ubicado en Fresnillo, Zacatecas, México. En esta institución se imparte el programa educativo de Ingeniería en Sistemas Computacionales. Como una forma de que los estudiantes relacionen la teoría con la práctica y desarrollen habilidades y destrezas relacionadas con su carrera, se decidió crear una área de desarrollo cuyo objetivo inicial fue que los estudiantes desarrollaran proyectos de software utilizando estándares internacionales probados, de manera que cuando se integren a la industria la curva de adaptación sea menor.

Por lo mencionado en la introducción, se decidió implementar TSPi. En esta primera ocasión se integraron dos equipos, y cada uno desarrolló un proyecto real propuesto por ellos durante un semestre, como parte de la residencia profesional.

Como estrategia para agilizar la recolección de métricas y el seguimiento del proyecto se determinó utilizar la herramienta Process Dashboard [5], la cual sustituye algunas formas de TSPi. En la tabla 1 se puede observar la funcionalidad de la herramienta y en la tabla 2 se puede ver la equivalencia de formas con TSPi.

C. Desarrollo del proyecto

Una vez que los integrantes de los equipos contaban con un entrenamiento previo en PSP, se les proporcionó una introducción a TSPi y se les enseñó a seguir los scripts de TSPi. Por lo tanto, se siguió el proceso de desarrollo que establece Humphrey en su

Tabla 1. Funcionalidades de Process Dashboard

FUNCIONALIDADES DE LA HERRAMIENTA PROCESS DASHBOARD	
FUNCIONALIDAD	DESCRIPCIÓN
Recolección de datos	Tiempo, defectos, tamaño, datos actuales vs planeados
Planeación	Plantillas, formatos, valor ganado
Seguimiento	Reportes y estadísticas de valor ganado
Análisis de datos	Gráficas y reportes que ayudan a analizar tendencias históricas
Exportar datos	Exportar a otros formatos
Multiplataforma	Desarrollado en Java

libro *Introduction to the team software process* [8]. El primer paso fue ejecutar el *script* general llamado “*script* de desarrollo del proceso de TSPi”, que tiene como objetivo guiar a un equipo en el desarrollo de un proyecto de *software*. En la figura 2 se muestra el ciclo de desarrollo con las etapas que se siguieron para el desarrollo de los productos.

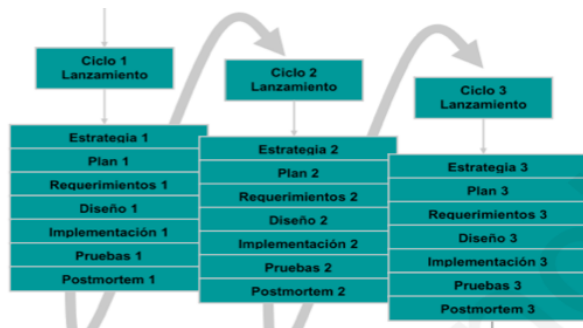


Figura 2. Ciclo de desarrollo TSPi [1]

Tal como lo muestran el *script* y la figura 2, el primer paso es ejecutar el lanzamiento 1; es decir, el que corresponde al ciclo uno. Para esto se consultó el *script* específico para el lanzamiento 1, el cual, entre otras cosas, indica que se debe: crear la estrategia, definir los equipos y establecer los papeles para cada uno de los integrantes. Para ello se utilizó la forma “Info”, que permite recabar datos de la experiencia, disponibilidad y habilidades de los miembros de los equipos, con la finalidad de asignar las funciones en el equipo. Por lo tanto, se asignó a cada uno de los integrantes de los equipos el papel para desempeñar durante el desarrollo del proyecto. Las funciones adoptadas fueron: líder de equipo, administrador de

planeación, administrador de calidad y procesos (se fusionaron los roles), administrador de desarrollo y administrador de configuraciones o soporte. Además del rol de TSPi, todos los miembros del equipo fueron ingenieros de desarrollo. TSPi sugiere el uso de tres ciclos de desarrollo; en este caso sólo se utilizó un ciclo por tratarse de un proyecto piloto.

Tal como lo muestra la figura 2, la siguiente etapa es la planeación. Para ello, en lugar de utilizar la forma *sump*, los equipos utilizaron la opción *WBS* de *dashboard*, que permite crear una lista de tareas con tiempo y tamaño estimado. En la etapa de requerimientos se utilizó el estándar de documentación SRS IEEE830; en la etapa de diseño, un documento llamado “Diseño de alto nivel”, que concentra el diseño de la base de datos y la arquitectura descrita con el modelo 4+1 vistas. El *software* se codificó en PHP. En la etapa de pruebas se crearon el plan y los casos de prueba, y finalmente se llevó a cabo una junta de *postmortem* en la cual se analizó si se cumplieron los objetivos definidos en el plan de calidad. Posteriormente se analizó el desempeño individual de los miembros del equipo, así como del conjunto.

Se utilizó también una versión inicial de la biblioteca de procesos llamada “Tritón”, que se desarrolló en el Centro de Investigación en Matemáticas (CIMAT) por el grupo de ingeniería de *software* y que puso a disposición de las universidades en Zacatecas.

Nuevamente, para agilizar el registro de métricas y el seguimiento a los proyectos, se utilizó la herramienta *Process Dashboard*. La tabla 2 muestra las formas que se utilizaron como fuente de datos y para control del proyecto.

Con el objetivo de lograr que los estudiantes apliquen de forma adecuada el TSPi, se aplicaron y monitorearon de manera especial los principios básicos de TSPi que se muestran en la tabla 3.

4 ANÁLISIS DE DATOS

Objetivo 1. Que los estudiantes den seguimiento al proyecto a través del valor ganado y que cumplan con el valor planeado

D. Seguimiento del proyecto mediante el uso del valor ganado.

Una de las métricas más importantes para dar seguimiento a proyectos es el valor ganado (EV).

Tabla 2. Equivalencias de formas TSPi en Dashboard

FORMA TSPi	DESCRIPCIÓN	EQUIVALENCIA EN PROCESS DASHBOARD
Task	Muestra la lista de tareas el tiempo real de ejecución así como el estimado	Work break Down Structure (WBS)
Schedule	Se registran horas semanales	Calendario individual
Week	Reporte semanal de trabajo realizado	Reporte semanal de valor ganado
Strat	Se define la estrategia de desarrollo	
Sump	Muestra datos de tamaño, tiempo, incluye productividad, cpi, porcentaje Reuse y porcentaje new reuso	Work break Down Structure (WBS)
Sums	Resume los datos para el tamaño del producto	
Logt	Registro de tiempo	Registro de tiempo por tarea

Tabla 3. Principios de TSPi aplicados

PRINCIPIO TSPi APLICADO	COMO SE HACÍA ANTES
Integración de equipos y roles bien definidos	No existían roles
Trabajo en equipo, integrantes comprometidos	No se conocía un proceso de integración de equipos
Estimación y delimitación del proyecto	No se estimaban los proyectos
Seguimiento a los proyectos a través de valor ganado	Sólo se establecían fechas de entrega, se confía en la palabra
Reuniones semanales de seguimiento	Reuniones informales sin agenda
Establecimiento de objetivos	Se definen objetivos que no son medibles
Seguimiento a procesos a través de scripts	No existían procesos documentados

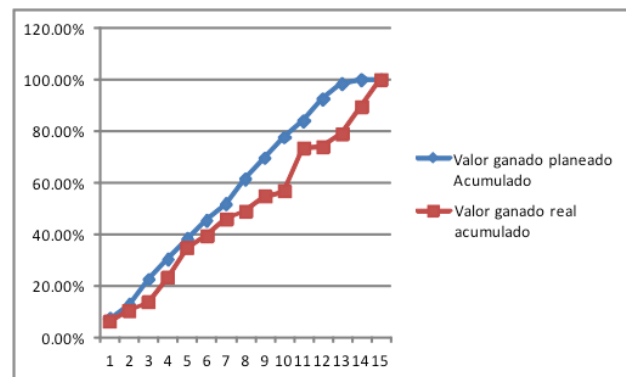
Se calcula determinando el porcentaje que le corresponde a una tarea del total de horas planeadas para el proyecto. Por ejemplo: si el proyecto se calcula en 1,000 horas, una tarea que se estime en 16 horas, representa 1.6% del valor ganado [4]. En la tabla tabla 4 se muestra un porcentaje de cumplimiento del valor ganado planeado para el equipo 1, y en la tabla 5, para el equipo 2. Éste se calculó al dividir el valor actual entre el valor real. Porcentaje de cumplimiento = valor ganado real acumulado (VG) / valor ganado planeado acumulado (CPV).

Tabla 4. Valor ganado por semana equipo 1

SEMANAS	CPV	VG	CUMPLIMIENTO
Semana 1	7.30%	6.20%	84.93%
Semana 2	12.80%	10.40%	81.25%
Semana 3	22.60%	13.70%	60.62%
Semana 4	30.40%	23.20%	76.32%
Semana 5	38.20%	34.80%	91.10%
Semana 6	45.40%	39.50%	87.00%
Semana 7	51.70%	45.90%	88.78%
Semana 8	61.50%	49%	79.67%
Semana 9	69.60%	54.80%	78.74%
Semana 10	77.70%	56.70%	72.97%
Semana 11	84%	73.40%	87.38%
Semana 12	92.50%	73.90%	79.89%
Semana 13	98.40%	78.90%	80.18%
Semana 14	100%	89.60%	89.60%
Semana 15	100%	100%	100%

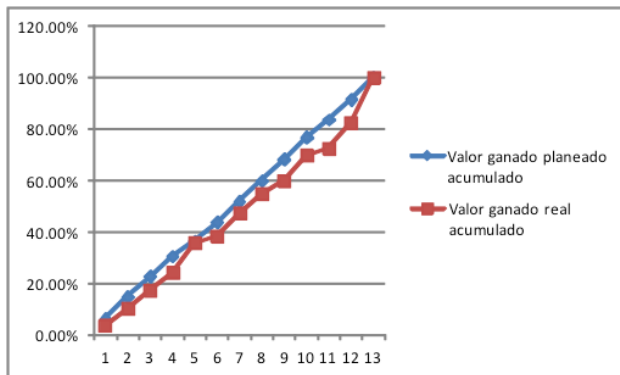
El análisis de estos datos se llevó a cabo a través de la implementación de juntas de seguimiento semanales o también llamadas juntas de estatus, en las cuales se revisó el valor ganado acumulado por semana planeado y real.

En la gráfica 1 y 2 se puede observar que se cumple uno de los objetivos primordiales que es enseñar al estudiante a registrar el valor ganado y a dar seguimiento al proyecto a través de éste.



Gráfica 1. Valor ganado por semana equipo 1

Si bien se puede observar que no se cumple al 100% con el valor ganado planeado por semana, se ve que el valor que se va ganando por semana se va



Gráfica 2. Valor ganado por semana equipo 2

manteniendo constante (sobre todo en el equipo 2) a medida que avanzan las semanas y se va asimilando el uso de TSPi, lo cual permitió el retraso de sólo una semana en la agenda para el equipo 1 y ninguna semana de retraso para el equipo 2.

En el comportamiento del equipo 1 se puede observar que la semana 11 y 12 fueron críticas, debido a que no se obtuvo valor ganado, se invirtió el tiempo en revisión y corrección de errores.

Objetivo 2. Terminar el proyecto dentro de la agenda establecida

La estimación del proyecto se hizo a través del método de estimación de tareas que los alumnos aprendieron en PSP llamado estimación basada en proxys o Proxy based estimating (PROBE, por sus siglas en inglés) [1]. Los resultados obtenidos se pueden ver en la tabla 5.

La fórmula que se utilizó para obtener la desviación, tanto de esfuerzo como de número de semanas de duración del proyecto, fue la siguiente: $100 * ((Actual\ x - Estimado\ x) / Estimado\ x)$, en donde x se sustituye por esfuerzo, semanas o tamaño.

Debido a que no se cuenta con datos históricos, se decidió medir el impacto del trabajo realizado comparándolo con otro proyecto similar. En este caso se trata del mencionado en el artículo “A small settings case study using TSPi in a software project” que aborda los beneficios de usar TSPi en un proyecto de software. En las tablas 6 y 7 se muestran algunos de los datos obtenidos en el proyecto ya mencionado así como las metas que planearon alcanzar con respecto a desviaciones en la agenda, el esfuerzo y el tamaño de los programas. En la tabla 8 se muestra el porcentaje de desviación de la agenda del proyecto del artículo [2].

Tabla 5. Valor ganado por semana equipo 2

SEMANAS	CPV	VG	CUMPLIMIENTO
Semana 1	6.40%	3.80%	59.38%
Semana 2	14.90%	10.10%	67.79%
Semana 3	22.80%	17.50%	76.75%
Semana 4	30.60%	24.30%	79.41%
Semana 5	36.50%	35.80%	98.08%
Semana 6	43.80%	38.60%	88.13%
Semana 7	51.90%	47.30%	91.14%
Semana 8	59.90%	54.70%	91.32%
Semana 9	68.30%	59.90%	87.70%
Semana 10	76.80%	69.80%	90.89%
Semana 11	83.70%	72.60%	86.74%
Semana 12	91.50%	82.50%	90.16%
Semana 13	100%	100%	100%

Tabla 6. Estimación y datos reales de los proyectos piloto

MEDIDA	EQUIPO 1			EQUIPO 2		
	ESTIMACIÓN	ACTUAL	DESVIACIÓN	ESTIMACIÓN	ACTUAL	DESVIACIÓN
Agenda (Semanas)	14	15	7.1	13	13	0.0
Esfuerzo (horas)	2080.75	2144.8	3.1	1267.5	1236.2	-2.5

Tabla 7. Muestra tabla 3 estimación vs actual del artículo con referencia 2

MEASURE	ESTIMATION	ACTUAL	DEVIATION
Schedule [SEM]	13	14.0	7.7%
Effort [HRA]	950.0	1121.0	18.0%
Size [KLOC]	6.9	8.5	22.5%

Tabla 8. Muestra tabla 4 resultados del objetivo 1 del artículo con referencia 2

MEASURE	GOAL	ACTUAL	DEVIATION
Schedule deviation	> 8% (1 week)	7.7%	-3.8%
Effort deviation	< 15%	18.0%	20.2%
Cost deviation	< 15%	18.0%	20.2%

Como se puede ver en la tabla 5, la desviación del proyecto piloto del equipo 1 es de 7.1% con respecto a las semanas planeadas de desarrollo contra las reales. Si se compara con la desviación mostrada en

la tabla 6 del artículo [2], se puede observar que la desviación del equipo del ITSF es un poco menor y cumple con la meta que allí se establece; es decir, una desviación de agenda menor a 8% equivalente a una semana de retraso. La desviación del equipo 2 es cero. Se considera que esto sucedió debido a que es más preciso y fácil estimar un proyecto de pocas líneas.

En cuanto a la desviación del esfuerzo, se puede observar que las métricas de ambos equipos están por debajo del objetivo utilizado en el artículo comparado; es decir, equipo 1, 3.1; equipo 2, -2.5. En los dos casos menor a la meta establecida <15%. De esta manera se puede afirmar, a partir de las mediciones, que el impacto de utilizar TSPi en estudiantes de ingeniería es positivo.

Objetivo 3. *Productividad aceptable de acuerdo a estándares de la industria*

El esfuerzo, la cantidad de líneas desarrolladas en el proyecto y la productividad de los equipos son datos relevantes en todo trabajo de desarrollo de *software*. Por este motivo se muestran esos indicadores en la tabla 10. Como se puede observar, la productividad es de 9.4 y 9 líneas de código por hora, para los equipos 1 y 2 respectivamente. Estos indicadores están por encima de la productividad que muestra la tabla 9, los cuales son resultados del trabajo tomado como referencia. La productividad está incluso por encima del estándar de la industria, entre 6 y 8 líneas de código por hora, según algunos artículos [1, 7, 10]. Estos resultados se explican porque son proyectos controlados y asistidos por asesores para lograr un correcto entrenamiento en el uso de TSPi por parte de los estudiantes.

5 CONCLUSIONES

El objetivo final es entrenar a los alumnos en el uso de TSPi para que cuando se integren a la industria puedan ser agentes de cambio al aplicar estas metodologías con una curva de aprendizaje lo más corta posible, de manera que puedan ser productivos en un tiempo menor al que necesitarían de no contar con esta capacitación.

Uno de los retos importantes fue observar y medir el rendimiento de los equipos de trabajo, una vez que se les exigió utilizar una metodología estándar que los obligó a establecer disciplina de trabajo. De acuerdo

Tabla 9. Muestra tabla 6 resultados del objetivo 3 del artículo

MEASURE	GOAL	ACTUAL	DEVIATION
Project productivity [LOC/Hour]	> 7.3%	7.6%	3.9%
% Released defects	< 5.0	3.8	-24.8%

Tabla 10. Productividad de los equipos

	NÚMERO DE LÍNEAS DESARROLLADAS PARA EL PROYECTO	HORAS TOTALES ESFUERZO	PRODUCTIVIDAD LOC x HORA
Equipo 1	20055	2144.8	9.4
Equipo 2	17850	1984	9.0

con los resultados mostrados en la sección “Análisis de datos”, se puede determinar que los estudiantes adoptaron el uso de procesos. Esto les permitió dar seguimiento a su trabajo, semana tras semana, a través de métricas como el valor ganado, así como tomar decisiones para realizar ajustes en los casos en que fueron necesarios y monitorear el proyecto. De esa manera el proyecto pudo ser entregado en tiempo y con base en una agenda establecida desde la planeación inicial, basada en estimaciones realizadas por los mismos alumnos, que emplearon el método PROBE de PSP [1].

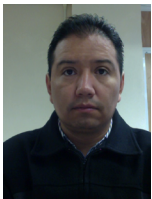
También se llevó a cabo una administración de riesgos, con intención de medir, controlar o disminuir el impacto de alguna situación no planeada que pudiera afectar el desarrollo del proyecto. Como resultado se observó que los riesgos se pudieron identificar fácilmente; sin embargo, fue más difícil para los equipos definir la estrategia adecuada para evitar o disminuir el impacto del riesgo.

Cabe mencionar que es importante la implementación de herramientas computacionales robustas que apoyen y faciliten la asimilación de metodologías de desarrollo de procesos, por ejemplo: TSPi. Un trabajo futuro será implementar metodologías ágiles dentro del centro de desarrollo de *software* de la institución para poder medir el nivel de adopción de éstas y compararlo con los resultados obtenidos con la herramienta TSPi. Además se pretende realizar una investigación que dé seguimiento a los alumnos que participaron en estos proyectos y medir su desempeño dentro de la industria.

REFERENCIAS

1. Humphrey, W. S., *A discipline for software engineering*. Nueva York: Addison-Wesley, 1995. p. 816.
2. Calvo Manzano, J., Cuevas, G., San Feliu, T., Caballero, E., I. Journal and I. Technologies, A Small Settings Case Study using TSPi in a Software Project " International Journal, vol. 2, pp. 245-250, 2008.
3. Mullaney, J., Davis, N. *The Team Software Process (TSP) in practice: a summary of recent results*, Software Engineering Institute, 2003. En línea.
4. Humphrey, W. S., *TSP: Leading a Development Team*. Nueva York: Addison-Wesley, 2006.
5. Tuman D., Functionality. *The Software Process Dashboard Initiative* (consultado: marzo de 2013). Disponible en: <http://www.processdash.com/functionality>
6. Humphrey, W. S., *TSP: Coaching Development Teams*. Nueva York: Addison-Wesley, 2006.
7. Nichols, W. R., *Deploying TSP on a National Scale: An Experience Report from Pilot Projects in Mexico*. *Softw. Eng. Process Manag*, 2009, 1(March), pp. 53-57.
8. Humphrey, W. S., *Introduction to the team software process*. Nueva York: Addison-Wesley.
9. Calvo-Manzano, J., Cuevas, G., San Feliu, T., Impact of TSPi on Software Projects. *Electronics, Robotics and Automotive Mechanics Conference*, 2007. CERMA/IEEE, 2007, pp. 706-711.
10. Calvo-Manzano, J., San Feliu, T., Análisis de la calidad y productividad en el desarrollo de un proyecto software en una microempresa con TSPi, *EICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*, 2009, 5(2), pp. 28-37.

Acerca de los autores



José Alberto Vela Dávila es Maestro en Ingeniería de Software por el Centro de Investigación en Matemáticas (CIMAT), campus Zacatecas en 2008. Su trabajo de investigación fue "Ambientes de Desarrollo de Software basado en componentes". Actualmente es candidato al Doctorado en Ciencias Computacionales del CIMAT, en la línea de investigación de Human Computer Interaction (HCI). Se desempeña como docente del programa educativo de Ingeniería en Sistemas Computacionales en el Instituto Tecnológico Superior de Fresnillo y la Universidad Politécnica de Zacatecas. Sus áreas de interés son: *brain computer interface*, ingeniería de *software*, redes y cómputo paralelo.