

## Introducción

El desarrollo de software es cambiante, complejo y los equipos de trabajo se enfrentan a situaciones inesperadas, una de las causas es el tipo y tamaño de los sistemas que, en la actualidad, se utilizan. La construcción de un sistema de software fuertemente relacionada con la complejidad del problema y la complejidad de la solución, el fracaso de un proyecto es debido a que los defectos no son encontrados a tiempo y cerca del 70.5% de los defectos son encontrados entre la fase de integración y la fase de operación con el cliente. La administración del proceso de software es una labor pesada sino se tiene la dedicación y la calidad para abordarla en las primeras etapas. La combinación de métodos ágiles y modelos disciplinados es una solución aceptada por aquellas empresas que se dedican a la construcción de software. Este tipo de combinaciones proporciona mejoras al proceso de desarrollo entregando guías que sirven de apoyo a los desarrolladores para mejorar su desempeño y, como consecuencia, la calidad del producto final. Para esta investigación se eligió **Scrum** donde su filosofía resalta el trabajo en equipo y una estructura dinámica, y **Personal Software Process** (PSP) enfocado al trabajo a nivel personal, proporcionando una estructura de trabajo para mejorar las habilidades del ingeniero en software.

## Objetivo

Desarrollar un modelo para mejorar la administración del desarrollo de software, combinando el método ágil Scrum y el modelo disciplinado PSP.

## Metodología y Recursos

Como primer lugar se analizó la teoría que rodea la combinación entre métodos ágiles y modelos disciplinados, debido a que sus principios chocan entre si. Para encontrar un punto de equilibrio y realizar una adaptación exitosa de ambos enfoques se elaboró una comparación (Tabla 1) entre sus principales características y así, definir sus fortalezas y debilidades. Gracias a este análisis nos percatamos de sus similitudes, diferencias y carencias entre ambos enfoques, sirviendo como base para integrar PSP y Scrum dentro de un modelo para la administración del proceso de software centrado en MiPyMES. Además, se encontraron ciertas expectativas que ambos enfoques tienen que considerar para formar un esqueleto de integración sólido (Tabla 2).

Tabla 1. Análisis comparativo entre características de Scrum y PSP

Factores	Scrum	PSP
Tamaño de Equipo	Equipo de trabajo pequeño (7 ±2).	Se enfoca en el trabajo personal.
Enfoque	Agilidad y adaptación a cambios. Producción rápida.	Mejorar de habilidades personales del desarrollador. Estimación basada en datos históricos.
Conocimiento	Empírico y Tácito.	Teórico, basado en registros.
Prácticas en el Desarrollo	Gestión del proceso centrado en los requerimientos del cliente.	Centrado en el Proceso.
Mecanismo de Operación	Iterativo e incremental. Revisiones y Retrospectivas.	Iterativo y escalonado. Enfocado a la Introducción y Eliminación de Defectos.
Ambiente de Trabajo	Cambiante y rápido.	Estable, pocas modificaciones.
Tipo de Proyecto	Para proyectos cortos y medianos. Poca necesidad de documentación.	Para cualquier tipo de proyectos. Se enfoca en el trabajo personal.
Calidad	Implícita, a través de la definición de Terminado.	Explícita, la define y administra cuantitativamente.
Gestión de Riesgos	No definido.	Definido y administrado por el desarrollador.
Cultura	Permite una mejora continua. Equipos auto-dirigidos.	Disciplina y el respeto al proceso de trabajo. Proceso auto-dirigido.

Tabla 2. Expectativas para combinar Scrum y PSP

Enfoque	Principio	Descripción
Scrum	Transparencia	Los procesos significativos deben ser visibles para aquellos que son responsables del resultado.
	Inspección	Los involucrados tienen que inspeccionar con frecuencia los artefactos y el progreso hacia el objetivo del <i>Sprint</i> para detectar variaciones indeseables.
	Adaptación	Si la inspección determina que varios aspectos se alejan de los límites aceptables, el proceso o el material que se está siendo trabajado tiene que ser ajustado.
Personal Software Process	Definir las métricas calidad	Definidas de manera personal y aplicadas a los productos creados de manera individual.
	Recolección de datos de todas las actividades	Para generar una base de conocimientos que permita hacer estimas más precisas en el futuro.
	Todo el equipo tiene que disponer de recursos necesarios	Es obligatorio que cada integrante tenga acceso con libertad y facilidad a los recursos que apoyen sus actividades.
	Seguir el flujo del proceso con disciplina	Una vez definido su proceso personal se tiene que apegar a este sin cambios.
	Realizar la medición del producto final	Verificar la calidad del resultado del proceso y que cumpla con la meta establecida.

Una vez encontrado el punto de equilibrio se procedió a desarrollar el modelo de combinación entre Scrum y PSP, tomando prestadas algunas actividades de la norma mexicana **MoProSoft** de dentro de la categoría de **Operación** (OP) en los procesos de **Administración de Proyectos Específicos** (APE) y **Desarrollo y Mantenimiento de Software** (DMS), nivel de capacidad **Realizado** y **Gestionado**; con el objetivo de proporcionar lista de verificación y obtener un proceso controlado y medible.

## Resultados

Como resultado se obtuvo la definición de un modelo para el desarrollo de software llamado **Mixing Scrum-PSP** (MSP), estructurado por dos capas. La primera, el **Ciclo de Vida** (Figura 1) define el flujo de actividades y la interacción con los roles y sus actividades. La segunda capa es la **Iteración de Desarrollo** (Figura 2), donde se define el flujo de actividades a nivel personal por medio de las guías proporcionadas por PSP. El modelo generado fue presentado a la Gerencia de una empresa dedicada al desarrollo de software, el Jefe del Área de Desarrollo proporcionó críticas constructivas y mostró su aprobación e interés por implementarlo a sus equipos de desarrollo y con ello mejorar el proceso de software actual.

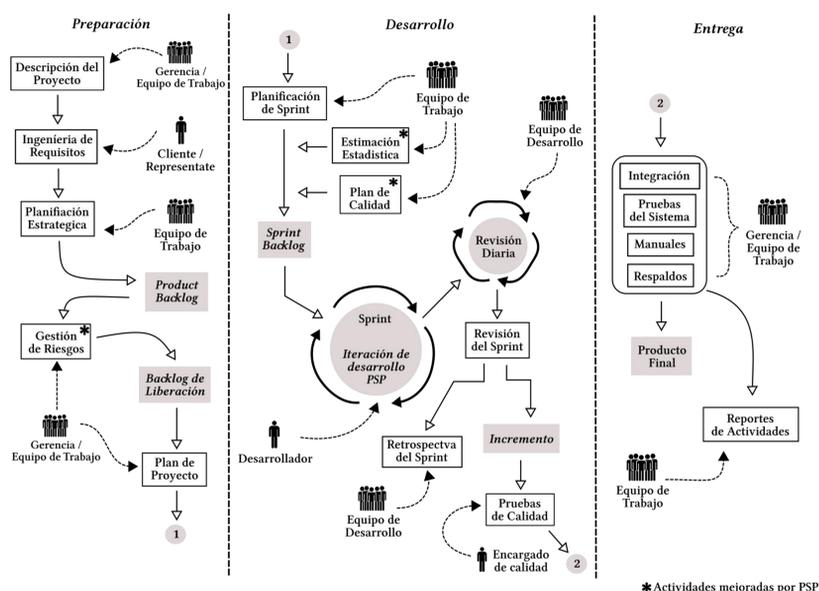


Figura 1. Modelo Mixing Scrum-PSP

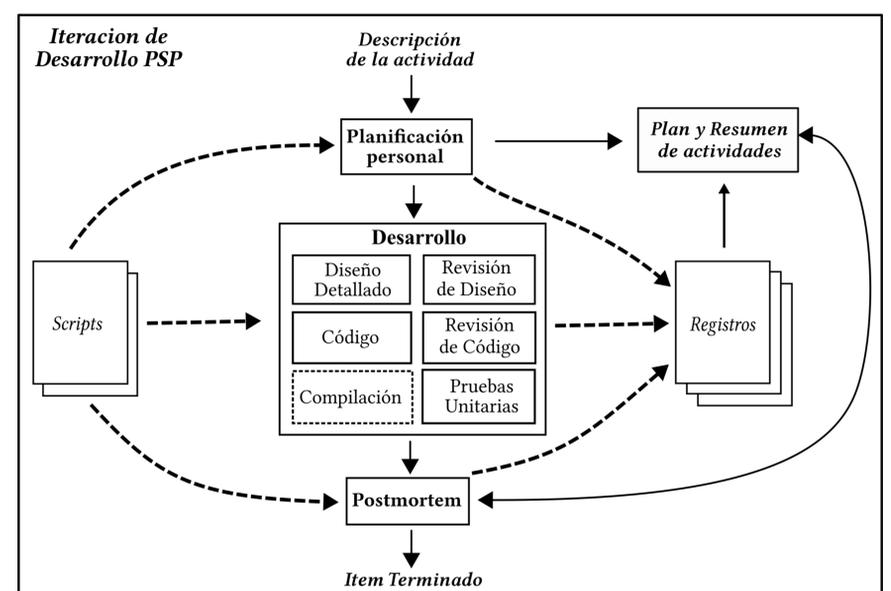


Figura 2. Iteración de Desarrollo PSP

## Conclusiones

Con este proyecto se ofrece una propuesta de un modelo de desarrollo para software que apoya a un equipo de desarrollo en un ambiente empresarial en mejorar la calidad final de su producto como consecuencia de la institucionalización de un proceso. Se considera como principal obstáculo la mentalidad de cambio de cada integrante del equipo para conseguir una implementación exitosa de cualquier mejora de procesos. Esta guía podrá ser utilizada por cualquier empresa dedicada al desarrollo de software, que deseen mejorar la calidad de su proceso, sin importar el tipo de proyecto, el tipo de aplicación y el lenguaje utilizado para construirlo.

## Referencias

- [1] Secretaría de Economía, "Modelo de Procesos para la Industria de Software, MoProSoft, Por Niveles de Capacidad de Procesos", Hanna Oktaba, México, 2005.
- [2] Scrum MEXICO. *Scrum & Agile*. (2014). Disponible en: <http://Scrum.org.mx/Scrum-agile/>.
- [3] W. Humphrey, *PSP*. Upper Saddle River, NJ: Addison-Wesley, 2005.
- [4] Boehm, B. 2002. "Get Ready For Agile Methods, With Care". *Computer* 35 (1): 64-69. doi:10.1109/2.976920.
- [5] Sutherland, Jeff, Carsten Ruseng Jakobsen, and Kent Johnson. 2007. "Scrum And CMMI Level 5: The Magic Potion For Code Warriors". *AGILE* 2007. doi:10.1109/agile.2007.52.
- [6] Dzhurov, Y. et al.: Personal Extreme Programming – An Agile Process for Autonomous Developers. Proceedings of the International Conference on Software, Services & Semantic Technologies. (2009).
- [7] Brown, D... 2014. "PSP Implementations for agile methods: a SEMAT-based approach". *Software Engineering: Methods, Modeling, and Teaching*. 3, 41-45 (2014).
- [8] Arauz Ortiz, G., Morales Trujillo, M., Oktaba, H., Ramirez Hernandez, E.: Integrating Agile Methods into a Level 5 CMMI-DEV Organization: a Case Study. *IEEE Latin America Transactions*. 14, 1440-1446 (2016).