

Integración de marcos de trabajo para desarrollo de software: Scrum, PSP e ISO 25000*¹

[Integrating software development frameworks: Scrum, PSP and ISO25000]

JHON ALEXANDER HOLGUÍN BARRERA²

RECIBO: 14.08.2014 – APROBACIÓN: 08.04.2015

Resumen

Actualmente la industria del software tiene grandes retos para la construcción y mantenimiento del software en términos de calidad, costos y tiempo. Para esto se han generado diversidad de metodologías, marcos de trabajo, normas y modelos que permiten afrontar estos retos, pero es complejo que la aplicación de solo un referente logre cumplir con todo lo requerido, y por lo general se toman elementos de varios modelos o metodologías. El presente estudio contiene la definición de un proceso de desarrollo de software en una PYME a través de la integración de varios de estos marcos de trabajo (Scrum, PSP e ISO 25000) analizados bajo los lineamientos dados por SWEBOK y apoyados por técnicas como GQM, que permitan dar un aporte positivo en el cumplimiento de estos objetivos.

Palabras Claves: ISO25000, Procesos, PSP, Scrum, SWEBOK.

* Modelo para la citación de este artículo:

HOLGUÍN BARRERA, John Alexander (2015). Integración de marcos de trabajo para desarrollo de software: Scrum, PSP e ISO 25000. En: Ventana Informática No. 32 (ene-jun). Manizales (Colombia): Facultad de Ciencias e Ingeniería, Universidad de Manizales. p. 151-164. ISSN: 0123-9678

- 1 Reporte de caso proveniente del proyecto *Definición de un proceso de desarrollo de software con control de calidad del producto en una empresa PYME de la región*, ejecutado en el periodo 07.2013-03.2014, e inscrito en la Maestría en Gestión y Desarrollo de Proyectos de Software de la *Universidad Autónoma de Manizales*. [Proyecto para optar por el título de Magister en Gestión y Desarrollo de Proyectos de Software, bajo la tutoría de la MSc. Sandra Victoria Hurtado Gil].
- 2 Ingeniero de Sistemas y Telecomunicaciones, MSc(c) en Gestión y Desarrollo de Proyectos de Software. Líder de Departamento de I+D+i, Geminus Software de Colombia (Pereira, Risaralda, Colombia). Correos electrónicos: jhon.holguin@gmail.com, jholguin@geminus.com.co

Abstract

Currently, the software industry has big challenges for the construction and maintenance of software in terms of quality, costs and time. For this the industry generated diversity of methodologies, frameworks, standards and models that meet these challenges, but is complex to accomplish this with the single application of only one reference that meets the goal, usually elements from various models and methodologies are taken. This study contains the definition of a software development process inside of a SME (Small and Medium Enterprise), through the integration of several of this frameworks (Scrum, PSP, and ISO25000) analyzed under the guidelines given by SWEBOK and supported by techniques such as GQM, that allow to make a positive contribution to meet this goals.

Keywords: ISO25000, Process, PSP, Scrum, SWEBOK.

Introducción

Actualmente la industria del software posee grandes retos para el desarrollo de sus productos, tales como mejor calidad con bajos costos y en el menor tiempo posible. Aunque muchos factores pueden influir, existe uno en especial el cual podría decirse que es el más importante: *el proceso* según Mengjiao, Guoping & Dong (2013, 805), esto debido a la gran variedad de requerimientos, escenarios, restricciones o limitantes que se pueden presentar a la hora de iniciar un proyecto de desarrollo de software. Es difícil concebir que todos usarán un mismo proceso de desarrollo y que la aplicación de solo un proceso logre cumplir con los tres objetivos, es por esto que expertos han empezado a trabajar en la *integración de procesos de software*.

Un estudio realizado por The Standish Group (2013, 1) indica que los proyectos exitosos de software (considerando exitoso que se cumpla con los objetivos de calidad, costo y tiempo) se encuentran por debajo del 50%, corroborando la necesidad de mejorar o definir nuevas técnicas o modelos en los procesos de desarrollo de software, además de tener mecanismos que permitan evaluar la efectividad de dichos procesos en cuanto a calidad lo cual puede ser dado por normas como la ISO 25000 como lo indican Pardo et al. (2009, 14); por otro lado es necesario también tener en cuenta que otros factores además del proceso de desarrollo de software influyen en la calidad (organizacionales, humanos, de equipo, restricciones, etc.) como lo señalan Daughtrey & Carroll (2007, 47), lo cual es importante tener en cuenta para definir un modelo integral y efectivo.

1. Antecedentes

De acuerdo con Mengjiao, Guoping & Dong (2013, 805), existen diversos casos de estudio que se han enfocado en la integración de diferentes marcos de trabajo de desarrollo de software, buscando tener como resultado un proceso («*serie de pasos parcialmente ordenados para lograr alcanzar una meta*» según Feiler & Humphrey (1992, 7)) que aumente las probabilidades de lograr un producto con calidad, a bajo costo y en el tiempo planeado, dentro de estos casos de estudio se ha rescatado algunos en los cuales han utilizado el marco de trabajo Scrum y el proceso PSP (Personal Software Process) integrado con otros modelos, arrojando resultados satisfactorios.

1.1 Integración de PSP con metodologías ágiles

Este es un artículo en el cual se realiza una revisión sistemática al estado del arte sobre el tema de integración de PSP con metodologías ágiles y donde se presenta una serie de conclusiones y resultados de la búsqueda de investigaciones realizadas sobre el tema. Se resalta cómo las metodologías ágiles (especialmente Scrum) proveen un marco de trabajo para la gestión del proceso aun en empresas grandes como *Yahoo*, *Google*, *Microsoft*, *SAP*, entre otras, según Deemer & Benefield (2006, 4), mientras que a través de PSP se puede obtener disciplinas para la estimación, planeación y gestión del trabajo individual. Algunas características que sobresalen de PSP en el artículo son:

- Ofrece mecanismos para las métricas y recolección de datos.
- Permite que se realicen estimaciones más razonables.
- Ayuda a optimizar los planes.
- Ofrece soporte para producir productos de software para cumplir las metas de calidad.
- Provee documentación adicional para los métodos ágiles.
- Ayuda de manera individual a los ingenieros a mejorar su trabajo personal gradualmente.

1.2 Scrum-PSP: Embracing Process Agility and Discipline

Guoping, Dong & He (2010, 317), muestran la integración de Scrum y PSP en un proyecto real, diseñado en dos capas, donde la primera (el Ciclo de Vida), refleja las iteraciones requeridas para realizar la entrega del producto final, se planea la estrategia de desarrollo, número de iteraciones requeridas, tiempo de cada ciclo y fases en cada iteración, mientras que la segunda capa (iteración Scrum-PSP), describe los pasos que van dentro de cada iteración, algunas conclusiones que se resaltan son:

- Lograr que cada desarrollador tenga la capacidad de predecir o estimar su trabajo, permitirá que el equipo pueda predecir mejor su trabajo.
- La inclusión del plan de calidad en cada iteración, permite que se logren establecer metas de calidad y maximizar la probabilidad de alcanzarlas.
- Permite realizar un seguimiento del proyecto dado por el valor ganado acumulado.
- Las reuniones establecidas en Scrum (Retrospectiva), permiten al equipo saber el estado del proyecto, posibles desviaciones, estado de la calidad, posibles riesgos.
- Scrum tiene definidos roles.

1.3 Scrum y CMMI

Este estudio posee datos contundentes sobre el uso de CMMI junto con Scrum resultado de la implementación de estos dos modelos en *Systematic*, empresa proveedora de software, que con productos complejos para sectores de la salud, defensa nacional y manufactura, hacen una mezcla contundente entre CMMI y Scrum la cual llaman la *Poción Mágica* según Sutherland, Jakobsen & Johnson (2008, 272), en la cual logran establecer que el re-trabajo de sus proyectos se redujo en un 38% a 42% con un nivel de madurez CMMI ML5 (*Maturity Level 5*) y más de 92% de entregas a tiempo a cliente; lo anterior permite corroborar que CMMI puede adaptarse como una ventaja para la mejora de procesos en entornos ágiles en los cuales se tengan ciclos cortos de liberaciones, como lo expresa Chrissis, Konrad & Shrum (2011, 102).

2. Metodología

Se realizó una investigación aplicada, relacionada con el proceso de desarrollo de software en una PYME, con componentes cuantitativos que incluyeron el análisis de datos históricos del proceso de desarrollo de software llevado hasta antes de la aplicación del nuevo proceso, junto a los datos recolectados en la prueba piloto aplicando el nuevo proceso que integra los modelos mencionados anteriormente.

2.1 Diagnóstico Inicial

La empresa seleccionada para el presente proyecto, es una PYME desarrolladora de productos de software ERP, CRM, educativos y otros productos especiales, con clientes en los ámbitos local y nacional.

Al inicio del estudio sus problemas de producto y proceso se concentraban en la calidad del producto, estimación y cumplimiento de entregas,

esta información se obtuvo a través de la recolección de datos históricos y encuestas desarrolladas a los clientes (Tabla 1). Tales registros permiten visualizar la necesidad de la implementación de un modelo que permita contrarrestar las falencias presentadas actualmente en la organización.

Tabla 1. Indicadores diagnóstico inicial PYME

Descripción Indicador	Valor
Promedio diferencia de tareas Estimado vs. Real por proyecto	20.02 Hrs
Promedio diferencia de tareas Estimado vs. Real por Sprint	26.10 Hrs
Promedio de cantidad de tareas no completadas x Sprint	34.32%
Promedio de cantidad errores liberados por versión	67.18

2.2 Selección de Norma ISO 25000 y División

La selección del uso de la norma ISO 25000 para la evaluación de calidad del producto en Scrum-P³, se realizó a través de la comparación de las normas ISO 9126, ISO 14598 y ISO25000 donde se valoraron características de las normas tales como requerimientos de cumplimiento para atributos funcionales, no funcionales, vigencia de la norma, métricas establecidas y si permite certificación, de esta evaluación se seleccionó la norma mencionada, en donde la principal justificación es su vigencia frente a las otras, al ser la generación siguiente de estándares de calidad de producto de software que reemplazara la ISO 9126 e ISO 14598, además de tener como objetivo la coordinación y armonización con la ISO/IEC 15939⁴ según Abran et al. (2005, 104-105); esta norma tiene definidas cinco divisiones establecidas por ISO (2005, vi) de la cual se selecciona la división 25023 que comprende de manera explícita las métricas para la evaluación de la calidad de un producto de software.

2.2.1 Definición de características y métricas a evaluar. La ISO/IEC 25023 (ISO, 2011, 14) provee un conjunto básico de métricas para la calidad de software o sistema a nivel interno y externo, las cuales permiten definir cómo se va a medir la calidad de las características seleccionadas de la división 25010, la selección de estas métricas está a consideración y de acuerdo con las necesidades del usuario de la norma e incluso puede incluir métricas que no se encuentren establecidas en ISO/IEC 25023 (ISO, 2011, 16).

Teniendo en cuenta lo anterior, se seleccionaron las características más relevantes debido al tamaño y extensión de la norma, para esta selec-

³ Scrum-P: Nombre que se le dio al modelo donde se integran Scrum, PSP e ISO.

⁴ ISO 15939: Estándar de definición de procesos de medición para sistemas e ingeniería de software

ción se consultó con el propietario de producto (*Product Owner*), quien definió las más relevantes para la calidad del producto: Funcionalidad, Usabilidad, Rendimiento/Eficiencia y Mantenibilidad.

De acuerdo con las características seleccionadas se utilizó el método GQM (*Goal Question Metric*) con el fin de establecer las métricas a evaluar, alineadas a los indicadores de calidad establecidos en la norma para la medición como lo establecen Van Soligen & Berghout (1999, 23), estas permitirán realizar la medición para verificar la calidad del producto. En la Tabla 2 se muestra un ejemplo del resultado de la definición de las métricas de calidad de producto para la categoría de funcionalidad, de esta misma manera se definieron las otras categorías seleccionadas

Tabla 2. Medidas de funcionalidad seleccionadas norma ISO 25023

Funcionalidad		
Meta	Preguntas	Métricas
Evaluar el cumplimiento de los requerimientos definidos en un Sprint	¿Se implementaron todos los requerimientos en el Sprint?	Total de requerimientos se entregaron / Total planeados en el Sprint
	¿Los requerimientos desarrollados cumplen su función?	Porcentaje de desarrollos que cumplen con todas las expectativas descritas en el requerimiento
	¿Los requerimientos son exactos en la información presentada al usuario?	Total de desarrollos entregados con errores durante la fase de pruebas / total de requerimientos desarrollados en el Sprint

2.3 Caracterización Scrum y PSP bajo lineamientos del SWEBOK

Para la caracterización del marco de trabajo Scrum y el modelo PSP, se realizó un desglose de las actividades y artefactos encontrados en cada modelo con el fin de establecer el mapeo del mismo con el área de conocimiento (*KA*), tópico y sub-tópico correspondiente, definido en el SWEBOK, según Bourque & Fairley (2014,29); de esta manera cada elemento analizado de los modelos se confrontó con la definición dada en el cuerpo de conocimientos para establecer si se cumple parcial o completamente con lo definido allí; las áreas de conocimiento que se mapearon fueron RS (Requerimientos de Software), DS (Diseño de Software), CO (Construcción de Software), PR (Pruebas de Software), CA (Calidad de Software) e IS (Gestión de la Ingeniería de Software).

Del mapeo de los elementos de Scrum y PSP con SWEBOK se generó la Figura 1, en la cual se estableció el porcentaje de cubrimiento

donde estos elementos se mapean de manera simultánea en un área del conocimiento del SWEBOK y de manera individual, permitiendo establecer que PSP es un modelo que aporta bastante en el proceso de construcción, y Scrum favorece al proceso de Ingeniería de Software (El ítem no mapeado corresponde a los sub-tópicos a los cuales no se les definió correspondencia en los modelos dados).

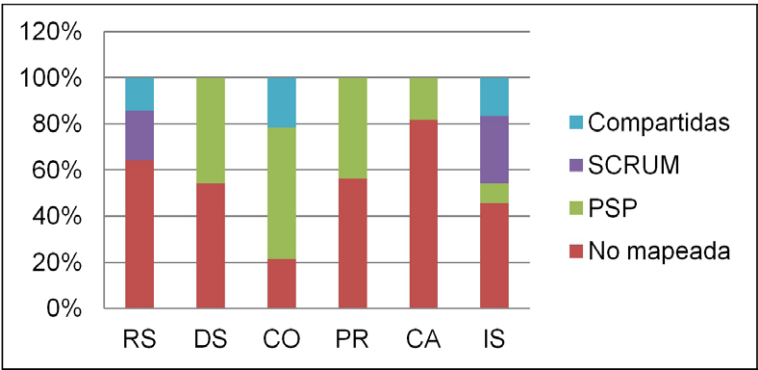


Figura 1. Porcentajes de cubrimiento de SCRUM y PSP versus SWEBOK

2.4 Integración Scrum - PSP y ISO 25000

El proceso de integración se realizó a través de la valoración de las diferentes actividades que se desarrollaban en Scrum y PSP, a través de un cruce de las mismas que permitió establecer aquellas que se complementaban y las que eran redundantes, en este caso se seleccionaron aquellas que tuvieran un criterio más completo y acertado de acuerdo a lo evaluado a través del SWEBOK, en la Tabla 3 se muestra un ejemplo de la valoración de una actividad compartida por los dos modelos y el método de selección.

Tabla 3. Método de selección para integración Scrum y PSP (Ejemplo)

KA SWEBOK	Scrum	PSP	SEL	OBS
Gestión en la ingeniería del software/ Planeación del proyecto de software	Sprint Planning (Estimar tiempo)	Estimación de tiempo requerido (PROBE)	PSP	La estimación de tiempo requerido a través del método PROBE de PSP es una herramienta que ha demostrado ser muy efectiva y acercada a la realidad, de acuerdo a los resultados presentados por el SEI

De acuerdo con la selección realizada, se definieron los procesos con el modelo integrado en el cual las actividades de Scrum y PSP se integran en diferentes momentos, de tal manera que exista un complemento de

las diferentes ventajas que ofrece cada uno; posteriormente las actividades de revisión de calidad del producto a través de la norma ISO 25000 se adicionan como actividades complementarias de calidad en la salida de cada *Sprint*. En la Figura 2 se puede encontrar un mapa del proceso integrado donde se resalta que Scrum mantiene la estructura definida y que PSP y el modelo ISO se integran como mecanismos complementarios para el desarrollo de las actividades ya definidas o sin definir por Scrum.

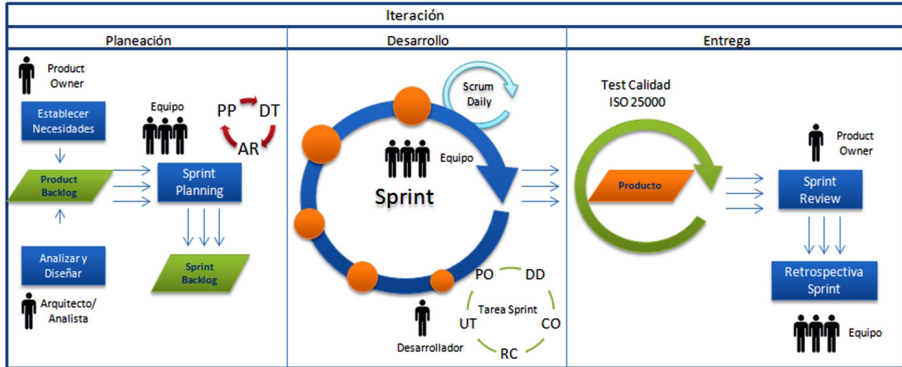


Figura 2. Proceso Integrado Scrum, PSP y Norma ISO 25000

Durante el desarrollo de un proyecto se pueden tener varias iteraciones con el fin de que al final de cada una exista un incremento del producto de valor para el usuario final; internamente estas iteraciones están divididas en tres fases (Planeación, Desarrollo, Entrega), en donde cada fase posee diferentes actividades que corresponden a los modelos integrados.

A continuación se realiza una breve explicación del modelo integrado:

- **Planeación:** En esta fase se realiza la construcción, documentación, asignación y estimación de recursos del *product backlog*. En la estimación se utiliza la plantilla de tamaño (LOC) y tiempos de PSP según lo establecido en el modelo de PSP por Humphrey (2005, 96), por lo cual se realiza una documentación más detallada que facilite el método de estimación.
- **Desarrollo:** corresponde al desarrollo de cada una de las tareas planeadas en el *Sprint Planning*, a través de la ejecución de las actividades integradas de PSP, en donde se cumplen los siguientes procesos: (DD) Diseño Detallado, (CO) Codificación, (RC) Revisiones de Código, (UT) Pruebas Unitarias y (PO) Post-mortem. El *scrum daily* continúa como actividad proveniente de Scrum, de tal

manera que permita hacer una monitorización de avance del *Sprint* y quitar posibles obstáculos que estén impidiendo el avance normal del mismo.

- Entrega: La finalización del Sprint, es la bandera que da inicio al inicio de la aplicación de las pruebas de calidad de producto definidas en la norma ISO 25000 para el proceso integrado de Scrum-P; una vez se ejecuten estas pruebas, se deberá proceder a realizar las respectivas correcciones.

2.5 Prueba Piloto

La implementación del proceso Scrum-P se realizó a través del desarrollo de un Sprint en la organización, donde se aplicaron y usaron las actividades definidas previamente en el proceso; a continuación se mencionan algunas particularidades para la implementación del proceso:

- Introducción del proceso al equipo: La inducción del equipo a la metodología se realizó a través de la explicación individual de Scrum y PSP por separado estableciendo las características de cada uno, ventajas y actividades realizadas, posteriormente se explica cómo se integraron las actividades que se seleccionaron de cada uno, y cómo se va a realizar la implementación del proceso integrado a través del desarrollo del próximo Sprint en la organización

- Herramientas utilizadas: Para la implementación del proceso de Scrum-P en la organización, se hizo uso de las herramientas con las cuales ya contaban, pero de las cuales no todas son indispensables, para esto se resalta con un asterisco (*) aquellas que se consideran como imprescindibles pero que pueden ser reemplazadas por otras que cumplan el mismo fin:

- JIRA: Herramienta utilizada para la gestión del ciclo de vida del proyecto, gestión de los *Sprint*, *product backlog* y liberaciones de producto.
- FishEye/Crucible: Herramienta utilizada para realizar las revisiones de código.
- Git (*): Herramienta de versionamiento.
- PSP Geminus: Herramienta propietaria elaborada para el registro de tiempos a través de cronometro.
- WorkBook PSP (*): Herramienta provista por el SEI (*Software Engineering Institute*) para la gestión de PSP.
- UCC tool version 2011.10 (*): Herramienta provista por la universidad del sur de california para realizar el conteo de líneas de código.
- Indigo (Infragistics): Herramienta de prototipado.

- McCabe IQ (McCabe): Herramienta utilizada para realizar pruebas de calidad sobre el software especialmente en el área de la mantenibilidad.
- Enterprise Architect (Sparx Systems): Herramienta utilizada para la elaboración de documentación específica para tareas (Casos de uso, Diagramas de Bases de datos, Casos de prueba, etc.).

3. Resultados y discusión

3.1 Resultados

3.1.1 Implementación de Scrum-P (Prueba Piloto). En el *sprint* desarrollado en la empresa como prueba piloto de la implementación del modelo integrado entre Scrum-P y el modelo de calidad ISO, se logró mejorar el porcentaje de tareas completadas con respecto a Sprint anteriores en donde el promedio de porcentaje completado era del 65.78% contra un 88% en el *Sprint* de la prueba piloto, aunque no es una cifra contundente y se deben esperar los resultados de futuros *Sprint*, se ve una mejora en el promedio de tareas completadas. Otro síntoma de mejora se puede evidenciar en la gráfica *burndown* (Figura 3), donde se ve como la línea real de trabajo restante en el *Sprint* (en rojo), es cercana a la línea guía ideal de trabajo restante del *Sprint* (en gris), la cual mejoró considerablemente comparándola con *sprints* anteriores, dando un indicador de en las estimaciones de tiempo.

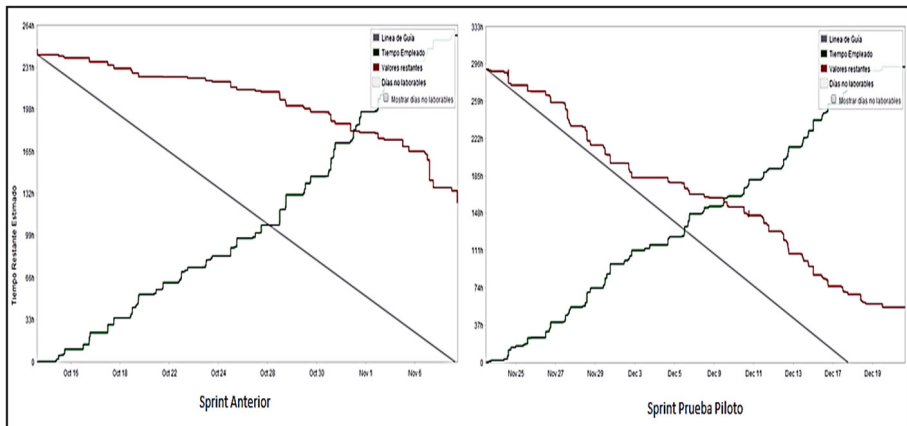


Figura 3. Comparativo gráfico Burndown Chart Sprint anterior y sprint prueba piloto

De acuerdo con las características de calidad seleccionadas en el modelo ISO, se realizó el proceso de medición con el fin de establecer la calidad del producto en el incremento del *Sprint* (no se evaluó la calidad

total del producto si no solo de las funcionalidades que se liberaron en el *Sprint*), para esto se establecieron los valores mínimos de cumplimiento con el equipo y el propietario del producto, con los resultados obtenidos se realiza el resumen de calidad (Tabla 4) donde se muestra el cumplimiento de cada una de las características seleccionadas de la norma, de acuerdo a cantidad de criterios cumplidos en la evaluación, permitiendo tener, de manera visual, un resultado global del cumplimiento de los estándares de calidad del producto establecidos.

Tabla 4 Resultados de evaluación de calidad a través de norma ISO del incremento del Sprint

Característica	Puntos evaluados	Puntos OK	Puntos NO OK	% OK
Funcionalidad	3	2	1	67%
Usabilidad	4	2	2	50%
Rendimiento	3	2	1	67%
Mantenibilidad	4	1	3	25%
Total	14	7	7	52%

Gracias a la mencionada tabla se puede obtener una perspectiva sobre la calidad actualmente manejada en el producto, basada en la evaluación de las características de la norma seleccionadas, apreciando que es necesario mejorar en aspectos como la mantenibilidad y la usabilidad, y esto a su vez se reflejará en la calidad global del producto. Otro aspecto importante es poder reflejar que a pesar de que se tenga un proceso organizado, es necesario realizar una evaluación más objetiva sobre el producto que permita establecer el estado de calidad del producto, permitiendo establecer cuáles son los puntos fuertes y débiles a los cuales debe colocarse atención.

3.2 Discusión

3.2.1 Dificultades encontradas durante el proceso. Durante todo el proceso se pudieron encontrar diferentes tipos de dificultades que deben ser tenidas en cuenta para próximas implementaciones y que probablemente permitan obtener mejores resultados del mismo, algunas de estas son:

- La falta de entrenamiento PSP formal es una de las desventajas que se encontró ya que el hecho de que los miembros del equipo no tuvieran una capacitación o curso de entrenamiento formal en PSP, hizo que el proceso fuera lento y con algunos errores en el ingreso de la información, causando re-procesos y demoras en el análisis de los datos.
- El análisis de calidad aplicando los criterios seleccionados es un proceso que requiere de recursos (Personal, Tiempo, Herramientas),

para lo cual realizarlo después de cada Sprint puede ser costoso, es necesario evaluar si es necesario modificar el intervalo de tiempo de evaluación o el número de implementaciones o desarrollos a ser tenidos en cuenta en la evaluación de calidad.

- La resistencia de algunos miembros del equipo a la implementación de la integración de *Scrum* y PSP, es un factor que debe anticiparse para evitar que se presente y contagie a los demás miembros del equipo, es necesaria una alta disposición y concientización del equipo, para que ponga todo de su parte y lograr aplicar el proceso de manera disciplinada, lo cual puede mejorar una vez cada desarrollador se da cuenta de su rendimiento y productividad.

- La disciplina de la captura de *Bugs* fue tal vez una de las dificultades más críticas en el proceso de PSP, que debió corregirse en varias ocasiones, debido a que es una práctica adquirida con el tiempo.

- Las pruebas de calidad llevaron demasiado tiempo, en especial las de rendimiento, además sacar los indicadores de cumplimiento de las diversas características fue un factor que no era muy claro y tuvo que hacerse a juicio de expertos inicialmente. Es necesario establecer una política de calidad clara para saber cuáles son los lineamientos a cumplir y además de esto tener información histórica que permita evaluar el grado de mejora de calidad.

4. Conclusiones

Se puede evidenciar que *Scrum* y PSP son elementos dentro del proceso de desarrollo que se complementan entre sí, a través de las ventajas y características que cada uno de los modelos ofrece, *Scrum* como marco de trabajo basado en pilares como la transparencia, inspección y adaptación como lo indica Sutherland & Schwaber (2013, 4), permitiendo realizar una gestión eficiente del proceso de desarrollo de software desde la entrada del requerimiento hasta su salida, por otro lado a nivel de equipo, es atractivo por su simplicidad, la práctica, los artefactos y reglas que se generan en si son pocos y fáciles de aprender según Schwaber (2004, xvii), adaptándose a proyectos con poca o alta complejidad, mientras que PSP provee un marco de trabajo personal que permite al desarrollador planear, medir, controlar y gestionar la manera en que trabaja como lo indica Humphrey (2000, ix), brindando soporte para la mejora de la calidad personal.

La utilización de las métricas para estimación de tamaño y tiempo en PSP, son una herramienta efectiva y que ayuda al proceso de planeación de *Scrum*, dando al equipo una mayor precisión ya que se basa

en datos históricos reales provenientes de la información ingresada por cada miembro del equipo.

La integración de *Scrum* y PSP, es una herramienta que busca impactar positivamente a la calidad del producto, pero aun así no es garantía de que el producto sea de calidad; es por esto que la integración del modelo de calidad ISO para el producto es una herramienta potente que permite complementar el proceso para establecer y medir los parámetros de calidad que la organización establezca.

Uno de los factores de éxito de la integración de *Scrum* y PSP fue el uso de herramientas que ayuden a minimizar el impacto de la implementación de las actividades definidas para el proceso, de tal manera que no centren el esfuerzo de los miembros del equipo en la recolección de la información y gestión del proceso.

GQM es un modelo que permite definir métricas de manera fácil y clara, que permitan establecer la meta y como llegar a ella. En el presente proyecto se utilizó para la definición de las métricas de calidad de la norma ISO, pero puede extenderse al resto del proceso en una mejora futura del mismo.

Como se pudo evidenciar en los resultados de la evaluación de calidad de producto a través de la ISO:25000 la mantenibilidad y usabilidad no obtuvieron buenos valores de resultado, lo cual se puede corregir mejorando las técnicas de planeación y diseño en las fases de PSP de tal manera que permita mejorar las puntuaciones en estos aspectos de calidad.

Referencias Bibliográficas

- ABRAN, Alain; DESHARNAIS, Jean-Marc; QUTAISH, Rafa Al & HABRA, Naji (2005). An Information Model for Software Quality Measurement with ISO Standards [online]. In: International Conference on Software Development, SWDC'05 (27/05-01/06/2005) Reykjavik (Iceland): University of Iceland. Proceedings of the SWDC'05. p. 104-116. <http://www.researchgate.net/publication/228663257_An_information_model_for_software_quality_measurement_with_ISO_standards/file/9fcfd5089494a89177.pdf> [consult: 23/10/2014]
- BOURQUE, Pierre & FAIRLEY, Richard E. (eds) (2014). SWEBOK version 3.0: Guide to the Software Engineering Body of Knowledge [online]. Piscataway (NJ, USA): IEEE. 346 p. ISBN: 978-0769551661 <<http://www.computer.org/portal/web/swebok/swebokv3>> [consult: 08/10/2014]
- CHRISSIS, Mary Beth; KONRAD, Michael D. & SHRUM, Sandy (2011). CMMI for Development: Guidelines for Process Integration and Product Improvement. 3 ed. Massachusetts (USA): Pearson Education. Serie in Software Engineering. 688 p. ISBN 978-0-321-71150-2
- DAUGHTREY, Taz & CARROLL, Sue (2007). Fundamental Concepts for the Software Quality Engineer vol 2. Wisconsin (USA): American Society for Quality, ASQ. 362 p. ISBN 0-87389-521-5
- DEEMER, Pete & BENEFIELD, Gabrielle (2007). The Scrum Primer: An Introduction to Agile Project Management with Scrum – Version 1.04 [online]. Bangalore (India): Goodagile India Private Ltd. 16 p. <<http://www.rallydev.com/documents/scrumprimer.pdf>> [consult: 09/10/2014]

- FEILER, Peter & HUMPHREY Watts (1992), Software Process Development and Enactment: Concepts and Definitions [online]. Technical Report CMU/SEI-92-TR-004, Massachusetts (USA): Carnegie Mellon, Software Engineering Institute. 7 p. <<http://www.sei.cmu.edu/reports/92tr004.pdf>> [consult: 02/03/2015]
- GUOPING, Rong; DONG, Shao & HE, Zhang (2010) Scrum-PSP: Embracing Process Agility and Discipline. In: 17th Asia Pacific Software Engineering Conference, APSEC'10 (01-03/12/2010), Sydney (Australia): IEEE. Proceedings of the ASPEC' 10. Washington D.C. (USA): IEEE Computer Society. p. 316-325. ISBN: 978-0-7695-4266-9
- HUMPHREY, Watts S. (2000). The Personal Software ProcessSM (PSPSM) [online]. Technical Report ESC-TR-2000-022. Pittsburgh (PA, USA): Carnegie Mellon, Software Engineering Institute. 41 p. <<http://www.sei.cmu.edu/reports/00tr022.pdf>> [consult: 01/10/2013]
- HUMPHREY, Watts S. (2005). PSP(sm): A Self-Improvement Process for Software Engineers. Boston (USA): Pearson - Addison-Wesley Professional. 368 p. ISBN: 0-321-30549-3
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, ISO (2011). ISO/IEC 25023: Systems and software engineering systems and software quality requirements and evaluation (SQuaRE) measurement of system and software product quality. ISO/IEC JTC 1/SC 7. Geneva (Switzerland): International Organization for Standardization.
- ISO (2005). ISO/IEC 25000: Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE). Geneva (Switzerland): International Organization for Standardization
- MENGJIAO, Shen; GUOPING, Rong & DONG, Shao (2013). Integrating PSP with agile process: a systematic review [online]. In: 2nd International Conference On Systems Engineering and Modeling, ICSEM-13 (19-20/04/2013) Beijing (China): Computer Science and Electronic Technology International Society and Atlantis Press. Proceedings of ICSEM-13. Paris (France): Atlantis Press. p. 0805-0811. ISBN: 978-94-91216-42-8< http://www.atlantis-press.com/php/download_paper.php?id=5660> [consult: 09/10/2014]
- PARDO, César; PINO, Francisco J.; GARCÍA, Félix & PIATTINI, Mario (2009). Analizando el apoyo de marcos SPI a las características de calidad del producto ISO 25010 [en línea]. En: REICIS, Revista Española de Innovación, Calidad e Ingeniería del Software, Vol. 5, No. 2 (sep) Barcelona (España): Asociación de Técnicos de Informática, ATI. p. 6-16. ISSN: 1885-4486 <<http://www.redalyc.org/articulo.oa?id=92217153003>> [consulta: 23/10/2014]
- SCHWABER, Ken (2004). Agile Project Management with Scrum. Washington (USA): Microsoft Press. Series: Developer Best Practices. 192 p. ISBN: 978-0735619937
- SUTHERLAND, Jeff & SCHWABER, Ken (2013). The Scrum GuideTM: The Definitive Guide of Scrum: The Rules of the Game [online]. Bern (Switzerland): Scrum.org. 16 p. <<http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>> [consult: 01/12/2013]
- SUTHERLAND, Jeff; JAKOBSEN, Carsten Ruseng, & JOHNSON, Kent (2008). Scrum and CMMI Level 5: The Magic Potion for Code Warriors [online]. In: 41st Annual Hawaii International Conference on System Sciences (07-10/01/2008). Waikoloa (HI, USA): IEEE. Proceedings. Washington DC (USA): IEEE Computer Society. p. 272-278. ISSN: 1530-1605. <https://systematic.com/media/282221/scrum_and_cmmi_level_5_the_magic_potion_for_code_warriors.pdf> [consulta: 09/10/2014]
- THE STANDISH GROUP (2013). Chaos Manifesto 2013: Think Big, Act Small [online]. Boston (MA, USA): The Standish Group. 48 p. <<http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf>> [consulta: 01/03/2013]
- VAN SOLIGEN, Rini & BERGHOUT, Egon (1999). The Goal/Question/Metric Method: A practical guide for quality improvement of software development. London (UK): McGraw-Hill. 260 p. ISBN: 978-0077095536