

PSP Connect: Una herramienta de soporte para la metodología PSP (Personal Software Process)

Juan Pablo Mejía Duque, Sebastian Lozano Herrera

Abstract—Durante el desarrollo de software se busca que el producto tenga la mejor calidad posible, es decir, cantidad mínima de errores durante el desarrollo y una vez entregado, existen muchas metodologías que buscan aportar a la calidad de un desarrollo y *Personal Software Process* es una de ellas, pero su aplicación implica mucho trabajo extra y las herramientas disponibles complican aún más el uso de la metodología [14]. Este artículo presenta una herramienta que ha sido desarrollada teniendo en cuenta componentes de usabilidad, con el fin de que la herramienta no sea una carga adicional y el desarrollador pueda enfocarse en su proyecto y en la metodología. Para lograrlo fue necesario indagar sobre las buenas prácticas de usabilidad, aspectos que hacen una herramienta usable y formas de probar la usabilidad de la herramienta. Un método cualitativo y comparativo que se compone de 8 tareas, hecho en un grupo de personas que no conocen ninguna de herramienta de soporte de la metodología y otro grupo con personas que al menos conocen la herramienta expuesta en éste artículo. Ambos grupos de personas cumplieron las tareas en un tiempo menor cuando utilizaron *PSP Connect*, además el promedio de errores fue de 6,6 mientras que cuando usaron *Process Dashboard* el promedio de errores ascendió a 16. En conclusión se logró el objetivo de incluir usabilidad en una herramienta que soporta la metodología, el enfoque de las pruebas ayudó a garantizar la implementación de los componentes de calidad definidos por Nielsen. También es importante llevar la herramienta a una comunidad que tenga experiencia en el uso de la metodología y de *Process Dashboard* o *PSP Student Workbook*, para que las pruebas no se vean limitadas ni afectadas por componentes que son normales cuando se usa la metodología.

Index Terms—Personal Software Process, Usabilidad, Herramienta web

I. INTRODUCCIÓN

Personal Software Process, PSP por sus siglas en inglés, es una metodología creada por Watts S. Humphrey cuyo objetivo principal es generar una disciplina para que los usuarios puedan mejorar su productividad y calidad en el desarrollo de software.

Aunque la metodología ofrece muchas ventajas actualmente el uso de PSP implica un esfuerzo adicional para el desarrollador, debido a la gran cantidad de datos que se deben registrar, la abrumadora complejidad, la rigurosidad del proceso y a esto se le suma la falta de usabilidad que ofrecen las herramientas de soporte existentes para PSP como el *Process DashBoard* [1] y el *PSP Student Workbook* [2].

Este proyecto busca crear una herramienta de soporte para la metodología PSP en donde los usuarios no solo reciban las

calidades que puede brindar la metodología, sino también las características que vienen de las buenas prácticas de usabilidad y de considerar los cinco componentes de calidad de Nielsen [3] para el desarrollo de la herramienta.

Para la implementación del proyecto, se utilizó una de las tecnologías más prometedoras del momento en cuanto al desarrollo en la web conocida como Meteor.js y sobre ella desarrollamos nuestra herramienta la cual lleva el nombre de *pspconnect*.

El artículo tiene la siguiente organización: la Sección II contempla el marco conceptual. La Sección III presenta un breve resumen del desarrollo e implementación del sistema. La Sección IV presenta las pruebas y resultados. Finalmente, la Sección V presenta las conclusiones seguido por el trabajo futuro en la Sección VI.

II. MARCO CONCEPTUAL

A. Usabilidad en la web

Uno de los elementos principales desarrollados en el proyecto fue la usabilidad de la herramienta web. Para el desarrollo de este elemento fue necesario utilizar en el trabajo hecho por Jakob Nielsen siendo este una de las personas más respetadas a nivel mundial en cuanto a los temas de usabilidad en la web.

La usabilidad es un atributo de calidad que explica que tan fácil es usar la interfaz de usuario de un producto web. Para esto, Jakob Nielsen menciona cinco componentes de calidad [3] que deben tenerse en cuenta para cumplir con una herramienta usable y estos son:

- **Facilidad de aprendizaje.** Es la facilidad con la que los usuarios de la herramienta web puedan desarrollar tareas básicas desde su primera interacción con la plataforma.
- **Eficiencia.** ¿Cuándo un usuario ya se familiariza con la interfaz de la herramienta; que tan rápido puede desarrollar tareas en ella?
- **Recordación.** ¿Cuándo un usuario se vincula nuevamente en la herramienta luego de un tiempo de inactividad, que tan fácil es volver a trabajar eficientemente en la herramienta?

- Recuperación de errores. ¿Cuántos errores comete el usuario, que tan severos son estos errores y que tan fácil es para el usuario recuperarse de estos errores?
- Satisfacción en el uso visual de la herramienta. ¿Que tan gratificante es para el usuario usar el diseño de la herramienta?

B. La metodología PSP

Personal Software Process(PSP) surgió en 1993 como una iniciativa de Watts S. Humphrey de generar una forma disciplinada y estructurada para desarrollar software [4]. Esta metodología es el conjunto de prácticas y métodos que ayudan a los desarrolladores examinar el trabajo que se encuentran realizando. Si estos métodos y conceptos son utilizados debidamente; los desarrolladores podrían mejorar sus habilidades de planeación, estimación, compromiso y calidad de desarrollo.

PSP busca proveer a los desarrolladores con una metodología de mejora personal para el desarrollo de sus proyectos. Para esto, utiliza ciertas herramientas como métodos, formularios y scripts que le muestran al desarrollador una forma de planear, estimar y administrar su trabajo. Los objetivos del *Personal Software Process* son reducir el número de defectos en un trabajo y mejorar las habilidades de estimación y planeación para que el desarrollador logre mantener compromisos sobre su trabajo y mejorar la calidad de su producto [5].

La metodología PSP se basa en el registro de datos que deben realizarse a lo largo de sus etapas o fases. Cada etapa necesita de un insumo para cumplir su objetivo, y estas están estructuradas así: se inicia con la planeación, lo siguiente es el desarrollo que incluye la etapa de diseño y codificación, y por último está la etapa de postmortem que es el cierre de proyecto [4].

Humphrey planteó una estructura de mejora evolutiva, para ello dividió la metodología en 3 niveles que son el PSP0, PSP1 y PSP2, cada uno de estos niveles contiene scripts, formularios, estándares y métricas. A medida que se avanza en los niveles de PSP, se agregan nuevos registros de datos y nuevas formas de medir el proceso. A continuación se mencionan en detalle cada uno de los niveles.

El primer nivel (PSP0) tiene como objetivo introducir la disciplina de la metodología, es decir, el desarrollador debe acostumbrarse al registro de tiempos por etapa y al registro de correcciones de defectos. Este nivel no provee de métricas sustanciosas para el proceso, pero permite que el usuario haga un análisis basándose en los tiempos que planeó y lo que realmente le tomó, los defectos que inyectó y los defectos que eliminó.

El segundo nivel (PSP1) introduce la estimación de tiempos y tamaños. Para esto, PSP1 utiliza el método de estimación llamado *Proxy-Based Estimating (PROBE)* [6], que realiza los cálculos a partir de todos los proyectos que el desarrollador ha finalizado. Además, PSP1 incorpora el plan de mejora (PIP por sus siglas en inglés) y los reportes de pruebas.

TABLE I
COMPARACIÓN NIVELES PSP

Función	PSP0	PSP1	PSP2
Objetivo	Seguir y medir un proceso definido.	Planeación. Estimar el tamaño y el tiempo.	Revisión de calidad, diseño y código.
Métricas	Tiempo y Errores	Tamaño.	Calidad y Revisión de diseño y código.
Métodos		PROxy-Based Estimating (PROBE). Personal Improvement Plan(PIP). Estándar de Código.	Process Quality Index (PQI).
Log	Time y Defect		
Formularios	Plan Summary, Registro de tiempo, Registro de defectos, Registro de tipos de defectos.	Personal Improvement Plan(PIP). Test Report. Plan Summary (Métricas adicionales y cálculos basados en históricos).	Diseño, Plan Summary (soporta planeación y seguimiento de calidad)
Fases	Planeación. Diseño. Codificación. Pruebas. Post Mortem.		Planeación (revisión de diseño y código). Revisión Diseño. Revisión Código.

Finalmente, el tercer nivel (PSP2) introduce las métricas asociadas con la calidad, la revisión de código y revisión de diseño. Estas fases buscan reducir la cantidad de errores que son cometidos en el diseño y en la codificación para que cuando se realicen las pruebas los errores sean mínimos o nulos en el mejor de los casos. Para medir la calidad este nivel usa un índice de calidad del proceso (PQI por sus siglas en inglés) el cual otorga un valor que indica si el producto desarrollado es de buena calidad o no.

En la Tabla I se encuentra un resumen de los principales elementos de PSP, las columnas contienen lo que es nuevo en comparación con el proceso anterior y se sabe que cada proceso incluye al anterior.

III. IMPLEMENTACIÓN

La implementación del proyecto se dividió en dos pasos básicos: el primero, fue la selección de la tecnología en la que se desarrolló el proyecto y a su vez, la implementación misma de la herramienta web. El segundo paso fue analizar las estrategias que se aplicarían para proveer a la herramienta de una interfaz gráfica agradable con la usabilidad necesaria para cumplir con los cinco componentes de calidad definidos por Nielsen.

A. Desarrollo herramienta web

Para el desarrollo de la herramienta web, se decidió utilizar el Full-stack framework llamado Meteor.js. Este framework permite trabajar tanto en el *Front-End development* como el *Back-End development*. Meteor trae una gran ventaja al mercado frente a otros frameworks y es la uniformidad en el uso de los lenguajes de programación para el desarrollo

[9]. Aquí, el programador solo debe trabajar con los lenguajes JavaScript, HTML y CSS [9]. Fuera de esto, otra ventaja de este framework es el despliegue de datos en tiempo real, compensación de latencia, sistema de paquetes integrado y la facilidad de instalación del framework.

Luego de definir la tecnología que se utilizaría para el desarrollo de la herramienta web; se definieron los requerimientos del sistema y sobre ellos las diferentes funcionalidades que el proyecto sería capaz de realizar. Los requerimientos del proyecto fueron estructurados en un conjunto de módulos y a partir de estos se implementó la plataforma *pspconnect*. Los módulos desarrollados son:

- Gestión de usuarios. Encargado de controlar las funcionalidades de autenticación y registro de usuarios.
- Registro de datos. Encargado de recolectar y almacenar los datos necesarios para la estimación de cada usuario y de cada proyecto que se haya definido.
- Métricas. Contiene las métricas que se utilizaran para la medición de todo el proceso.
- Utilidades. Encargado de desplegar la información mediante gráficas y notificar al usuario en la herramienta sobre cualquier novedad que ocurra.
- Planeación y seguimiento. Encargado de ejecutar en la herramienta el método PROBE.
- Históricos. Contiene los datos que se consideran históricos en la metodología PSP.
- Ayuda y documentación. Contiene un tutorial básico de la herramienta para los usuarios y una comunidad en donde un usuario podrá resolver dudas.

Teniendo en cuenta los problemas de portabilidad que tienen actualmente las herramientas de soporte a la metodología PSP; se decidió desarrollar el proyecto como una plataforma web buscando así generar un fácil acceso a la herramienta para los usuarios. Para acceder a la herramienta, se debe ingresar al link www.pspconnect.co. Fuera de esto, el proyecto es una propuesta de software libre cuyo código fuente se puede descargar desde la url <https://github.com/juanpmd/PSP-Connect>.

B. Usabilidad de la herramienta web

La usabilidad es uno de los puntos más importantes en el desarrollo de la herramienta web. Tomando como referencia los documentos sobre la usabilidad y experiencia de usuario en los que se encontraron durante la investigación [10]–[13], se recolectaron diferentes puntos de enfoque en torno a los cinco componentes de calidad de Jakob Nielsen [3] y además diferentes criterios para cumplir con cada uno de estos componentes. Los principales criterios que fueron empleados en el proyecto son:

- Desarrollo de un modelo visual fácil de entender para los usuarios. Se utilizaron los mismos patrones que tienen en el momento otras plataformas buscando así que el usuario se familiarice con la herramienta rápidamente.

- Todos los mensajes desplegados en la herramienta deben estar hechos en un lenguaje entendible para los usuarios. El texto utilizado en la herramienta no debe incluir código de programación o palabras técnicas.
- Mantener la consistencia en los datos para evitar confusiones por parte del usuario sobre cualquier texto de información u acción.
- La herramienta contará con una opción de documentación y ayuda la cual estará siempre a la mano de los usuarios. Esta documentación contiene un foro en donde el usuario puede realizar preguntas a toda la comunidad; además, existe un tutorial que muestra cómo funcionan algunas funcionalidades básicas de la herramienta.
- El despliegue de diálogos solo contendrá la información más relevante posible. Esto permite que los usuarios gasten menos tiempo leyendo información de la herramienta.
- Siempre existirá una opción para regresar a la vista principal de la herramienta. Esto busca que el usuario siempre tenga la facilidad de volver a las opciones básicas de la plataforma y pueda elegir otra acción en cuestión de segundos.
- La herramienta contara con un flujo visual que va acorde al mercado actual. Buscamos desarrollar nuestra herramienta web con la misma estructura visual utilizada en el mercado actual para evitar así confusiones para el usuario.
- La herramienta informará siempre sobre el estado en el que se encuentra el usuario después ejecutar una acción sobre ella.
- Mantendremos la interfaz simple y limpia de manera tal que la herramienta cargue rápidamente en el navegador del usuario.
- Resaltar la posición del usuario en los inputs de la herramienta. Esto ofrece una mayor comodidad al usuario para saber siempre en que parte de la herramienta se encuentra evitando cometer errores.
- Mostrar un Breadcrumb que le permita al usuario saber en qué parte de la herramienta se encuentra y/o que acción se encuentra realizando.
- Anticipar todas las acciones que un usuario pueda ejecutar en la herramienta web y prevenir errores que puedan generarse sobre estos estados.
- Uso adecuado de los colores en la herramienta web junto con el tamaño del texto y la densidad de píxeles.

IV. PRUEBAS Y RESULTADOS

Para las pruebas de la herramienta, fue necesario definir el tipo de retroalimentación que mejor se ajustara a nuestras necesidades y así mismo se definieron las métricas que servirían para cuantificar el proceso. Por esto se decidió trabajar con las pruebas cualitativas llamadas *respuesta verbal* [15] cuyo objetivo es realizar pruebas de usabilidad personalizadas con los usuarios y sobre ellas anotar el comportamiento del usuario, expresiones corporales, expresiones faciales, comentarios y opiniones de la herramienta.

Luego de definir el método que se utilizaría para realizar las pruebas de usabilidad, se establecieron las métricas que se

emplearían en las pruebas de *pspconnect*. Estas métricas son [15]:

- Éxito: ¿El usuario logró completar la tarea asignada?.
- Tasa de error: ¿Cuántos errores cometió el usuario realizando la tarea?. Para este punto, los errores se consideraron como el número de páginas a las que el usuario accedía antes de llegar a la página correcta.
- Tiempo de completitud: El tiempo que le tomo al usuario completar la tarea.
- Dificultad de la tarea: ¿Qué tan difícil fue realizar la tarea?. Para este punto, el usuario calificaria las tareas en una escala del 1 al 5, siendo uno muy fácil y 5 muy difícil.

Luego de definir los elementos a considerar en las pruebas, se formuló un formato que debía ser ejecutado con cada usuario utilizando el *Process Dashboard* y posteriormente la herramienta *pspconnect*. El propósito de esto es hacer un contraste entre las dos herramientas y poder determinar cuáles fueron las funcionalidades que se lograron mejorar en *pspconnect* frente a las herramientas ya existentes. Las tareas que se asignaron y sobre las cuales se hizo el análisis fueron:

- Crear Proyecto con nivel PSP 0.
- Crear iteración.
- Registrar tiempo.
- Finalizar etapa de planeación.
- Añadir tiempo usando el modal.
- Registrar defecto.
- Eliminar iteración.

Para la ejecución de las pruebas, se trabajó con proyectos de nivel PSP0 debido a que la mayoría de los usuarios que desarrollaron las pruebas no tenían un amplio conocimiento sobre la metodología. Además, el objetivo de las pruebas de usabilidad no es evaluar la aplicación misma de la metodología PSP. En total se realizaron 10 pruebas y se dividió la muestra en dos grupos con el fin de probar si los componentes de calidad se implementaron de manera adecuada. Los grupos en los que se dividió la población fueron:

- Grupo A. Personas que no conocían ninguna de las dos herramientas (grupo de cinco personas).
- Grupo B. Personas que utilizaron la herramienta *pspconnect* de manera previa y no conocían *Process Dashboard*. (grupo de cinco personas).

En la Tabla II, III y IV esta la comparación de los resultados promedios obtenidos sobre las pruebas realizadas en el grupo A.

Los resultados permiten deducir que la herramienta presenta una mejora significativa en tiempos y en la completitud de las tareas. Así mismo, se evidencia que los errores cometidos en *pspconnect* son mucho menores en comparación con los que se cometieron en el *Process Dashboard*. Las Tablas de resultados sobre el Grupo A evidencian que se cumplieron los siguientes componentes de calidad:

- Facilidad de aprendizaje.
- Eficiencia.

TABLE II
COMPARACIÓN DE TIEMPOS(MINUTOS:SEGUNDOS).

	PSP Connect	Process Dashboard
Crear Proyecto	00:22	02:08
Crear Iteración	00:08	01:16
Registrar tiempo	00:42	00:24
Finalizar etapa de planeación	00:37	01:14
Añadir tiempo usando modal	00:23	01:27
Registrar defecto	00:23	00:27
Eliminar iteración	00:06	00:21

TABLE III
ERRORES PROMEDIO POR TAREA.

	PSP Connect	Process Dashboard
Crear Proyecto	0	7,2
Crear Iteración	1	3
Registrar tiempo	2,4	1,6
Finalizar etapa de planeación	2,2	2,4
Añadir tiempo usando modal	0,4	4,4
Registrar defecto	0	0,6
Eliminar iteración	1,2	1,8

- Recuperación de errores.

Para evaluar la satisfacción en el uso visual de la herramienta, se utilizó la retroalimentación de cada usuario preguntando en ella los elementos que más se dificultaron en la prueba y a la vez los elementos que se tornaron más sencillos de desarrollar. Sobre las respuestas obtenidas, se encontraron resultados positivos frente al concepto visual de la herramienta cumpliendo con así con la satisfacción en el uso visual de la herramienta.

Para asegurar que se cumplió con el componente de calidad de recordación fue necesario analizar por separados los resultados de las personas que ya habían utilizado la herramienta en pruebas anteriores. Estas fueron las pruebas realizadas sobre el Grupo B y los resultados están en las Tablas V, VI y VII.

Las primeras pruebas de usabilidad realizadas en la herramienta mostraron que uno de los elementos más complicados de utilizar era el registro de tiempos. La actividad que más tiempo tomó fue registrar el tiempo en las etapas del proyecto, mientras que la actividad con más errores fue el finalizar la etapa de planeación. Dentro de lo que los usuarios sugirieron para mejorar la herramienta está:

- Además de la notificación debería resaltarse dónde está el error.
- Los items del menú deberían tener un orden de importancia.
- La etapa actual debería ser más distintiva que las demás.
- Los íconos que están al lado izquierdo del nombre de la etapa deberían reflejar el estado de la misma.
- El registro de tiempo asistido debería ser más llamativo.
- Debería poder verse el tiempo que llevo registrando sin

TABLE IV
TASA DE COMPLETITUD.

	PSP Connect	Process Dashboard
Crear Proyecto	100%	20%
Crear Iteración	100%	40%
Registrar tiempo	80%	80%
Finalizar etapa de planeación	80%	60%
Añadir tiempo usando modal	80%	60%
Registrar defecto	100%	100%
Eliminar iteración	100%	100%

TABLE V
COMPARACIÓN DE TIEMPOS(MINUTOS:SEGUNDOS), PRUEBAS 1 Y 2.

	Prueba 1	Prueba 2
Crear Proyecto	01:00	00:20
Crear Iteración	01:10	00:05
Registrar tiempo	09:12	00:30
Finalizar etapa de planeación	08:05	00:25
Añadir tiempo usando modal	03:12	00:20
Registrar defecto	04:17	00:20
Eliminar iteración	00:51	00:16

necesidad de detener la toma de tiempo.

- El tiempo registrado en cada etapa debería ser más notorio.
- Debería informarse al usuario que al registrar un defecto se esta tomando el tiempo que tomó para su corrección.
- Al finalizar un proyecto debería aparecer la fecha de finalización.
- Las notificaciones de error a advertencia deberían durar más tiempo en pantalla.
- Las etiquetas de los botones y opciones deberían ir más acorde a su funcionalidad.
- Debería haber una mejor distinción entre las preguntas de la comunidad que fueron cerradas y las que siguen abiertas.

Luego de aplicar las recomendaciones sugeridas por los usuarios en las pruebas de usabilidad; las Tablas II, III y IV muestran una mejora significativa en cuanto al tiempo tomado, errores cometidos y la tasa de completitud registrada sobre la ejecución de las tareas de la prueba. La reducción de errores en todas las tareas evidencia que se logró reducir la complejidad vista en las herramientas de soporte a la metodología PSP actuales y además de que los usuarios se adaptaron fácilmente a la herramienta tras tres meses de inactividad sobre la misma.

V. CONCLUSION

Este trabajo presentó el proyecto *pspconnect* cuyo objetivo es servir como una herramienta de soporte para la metodología PSP y a su vez cumplir con los cinco componentes de calidad definidos por Nielsen.

Los resultados obtenidos en las pruebas de usabilidad evidenciaron el éxito que tuvo la herramienta *pspconnect* frente a las herramientas de soporte a la metodología ya existentes.

TABLE VI
ERRORES PROMEDIO POR TAREA.

	Prueba 1	Prueba 2
Crear Proyecto	0	0
Crear Iteración	0	0
Registrar tiempo	6	1
Finalizar etapa de planeación	8	1,2
Añadir tiempo usando modal	0	0
Registrar defecto	3	0
Eliminar iteración	0	0

TABLE VII
TASA DE COMPLETITUD.

	Process Dashboard	PSP Connect
Crear Proyecto	100%	100%
Crear Iteración	100%	100%
Registrar tiempo	10%	95%
Finalizar etapa de planeación	20%	70%
Añadir tiempo usando modal	80%	100%
Registrar defecto	100%	100%
Eliminar iteración	100%	100%

Estas pruebas nos permitieron medir los cinco componentes de calidad definidos por Nielsen y con ello se aseguro el cumplimiento de estos componentes en la herramienta.

Sobre las pruebas de usabilidad se obtuvo tambien resultados positivos frente al concepto visual utilizado en la herramienta web y con ello se logro cumplir el objetivo de presentar una herramienta que fuera fácil de usar y que fuera agradable visualmente. Nuestro publico objetivo fueron estudiantes activos de la carrera ingenieria de sistemas y a su vez de personas laborando sobre este campo de trabajo. En total, logramos realizar 10 pruebas de usabilidad en donde obtuvimos una retroalimentacion tanto de las funcionalidades de la herramienta como de la usabilidad de la misma.

Como resultado del proyecto, se desarrollo la plataforma *pspconnect* que se encuentra disponible en www.pspconnect.co y cuyo código fuente se puede descargar desde la url <https://github.com/juanpmd/Thesis>. Esta herramienta es una propuesta de software libre e incluye las funcionalidades de PSP0, PSP1 y parcialmente PSP2.

Durante el desarrollo de *pspconnect*, se logró profundizar los conocimientos frente a la ingeniería de software y el desarrollo en la web. Con ello se aprendió todo lo referente a la funcionalidad de la metodología PSP y a su vez sobre el proceso que requiere desarrollar una herramienta web pensada para los usuarios.

En el desarrollo de la herramienta se encontraron una variedad de problemas en torno al uso y ejecución de los paquetes Meteor debido a una extensa actualización que realizó el framework en Marzo de 2016. Este cambio convirtió varios de los paquetes que se estaban utilizando en la plataforma

en elementos obsoletos lo cual retraso considerablemente el desarrollo del proyecto.

Además, se presentó un retraso en la implementación de la herramienta debido a los problemas y la complejidad que llevaba desplegar un proyecto Meteor en la web. Debido a esto, se tuvo que buscar una alternativa diferente a los comandos de despliegue que ofrece el framework de manera nativa y se encontró el sistema *MUPX* con el cual logramos concluir la tarea.

Finalmente, esta herramienta estará disponible para toda la comunidad PSP cuyo primer objetivo posterior al proyecto es extender su uso a todos los usuarios que se encuentran dentro de la comunidad Colombiana *Colombian SEI-PSP Developers Community*.

VI. TRABAJO FUTURO

Este proyecto obtuvo una gran aceptación por parte de los usuarios encuestados y sobre él existen aún un conjunto de mejoras que se pueden realizar en futuros proyectos; algunas de estas mejoras son:

- Generar una unión entre la plataforma de soporte a la metodología PSP y un curso guiado sobre el funcionamiento mismo de esta metodología. Este es un elemento importante ya que pudimos observar en las pruebas del proyecto que el manejo de la herramienta está aún muy ligado al conocimiento que tienen los usuarios sobre la metodología PSP.
- Desarrollar la metodología *Team Software Process*. Este elemento podría darle un enfoque positivo a la herramienta y a su vez una mayor completitud a las funcionalidades que se brindan en este proyecto basándose con ello en los componentes de calidad definidos por Nielsen.
- Crear la opción de migrar los datos que los usuarios han recolectado de manera previa en otras herramientas de soporte a la metodología PSP como el *Process Dashboard* y el *PSP Student Workbook*. Con esto se busca que los usuarios no tengan que comenzar nuevamente su proceso permitiendo así que continúen su proceso en la nueva herramienta *pspconnect*.
- Promocionar el uso de la herramienta web *pspconnect* en toda la comunidad PSP y a su vez en diferentes lugares como las universidades o cursos de ingeniería de software.

REFERENCES

- [1] Tuma Solutions, Process Dashboard, <http://www.processdash.com/>. Online: 2016-09-26
- [2] Software Engineering Institute, PSP Academic Material, <http://www.sei.cmu.edu/tsp/tools/academic/index.cfm>. Online: 2016-10-12
- [3] Jakob Nielsen, *Usability 101: Introduction to Usability*, 2012
- [4] Marsha Pomeroy-Huff, Robert Cannon, Timothy A Chick, Julia Mullaney and William Nichols. *The Personal Software ProcessSM (PSPSM) Body of Knowledge, Version 2.0*. Technical report, DTIC Document, 2009
- [5] Watts S. Humphrey, *Using a defined and measured Personal Software Process*. In *IEEE Software*, page 77-88. IEEE, 1996
- [6] Watts S. Humphrey, *A Discipline for Software Engineering*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995
- [7] Watts S. Humphrey, *The personal software process (PSP)*. 2000
- [8] Beni Suranto, *PSP and PQI, How do they improve individual software process?*, 2015
- [9] Building pure JavaScript applications (Meteor.js), <https://www.airpair.com/javascript/posts/building-pure-javascript-applications-meteorjs>. Online: 2016-02-22
- [10] Jakob Nielsen, *10 usability heuristics for user interface design*. Fremont: Nielsen Norman Group., 1995
- [11] Jakob Nielsen, *Extreme Usability: How to Make an Already-Great Design Even Better*, 2008
- [12] Bill Scott and Theresa Neil, *Designing web interfaces: Principles and patterns for rich interactions*, " O'Reilly Media, Inc.", 2009
- [13] Bruce Tognazzini, *First principles of interaction design (revised & expanded)*, 2014
- [14] Jouni Lappalainen, *Tool Support for Personal Software Process*, Springer Berlin Heidelberg, 2005
- [15] Chris Bank, *The guide to usability testing*, 2015
- [16] Automated Usability Testing: A Case Study, http://uxpamagazine.org/automated_usability_testing/. Online: 2016-12-22