

Modelos, Herramientas y Aprendizaje Experimental para Mejora de Proceso Software

Figueroa, Liliana María

*Departamento de Informática, Facultad de Ciencias Exactas y Tecnologías
Universidad Nacional de Santiago del Estero
lmvfigueroa@yahoo.com.ar*

Ríos, Miriam

*Departamento de Matemática, Facultad de Ciencias Exactas y Tecnologías
Universidad Nacional de Santiago del Estero.
meros15@yahoo.com.ar*

Maldonado, Marilena

*Departamento de Informática, Facultad de Ciencias Exactas y Tecnologías
Universidad Nacional de Santiago del Estero
marilena@unse.edu.ar*

López, Gustavo

*Departamento de Matemática, Facultad de Ciencias Exactas y Tecnologías
Universidad Nacional de Santiago del Estero.
gustavojlopez@gmail.com*

RESUMEN: en los últimos tiempos las organizaciones orientan sus actividades hacia modelos que sugieren buenas prácticas de mejoras continuas, algunos de estos toman como base principal a las personas que llevan a cabo los procesos, aportando distintos beneficios, tanto para el éxito de la organización como para la satisfacción del cliente. Esto también se traslada en el ámbito de desarrollo del proceso software, en donde se han planteado e investigado distintos modelos de mejora del proceso software, entre los que se destaca el modelo People Capability Maturity Model (PCMM), como un modelo de cambio organizacional a través de un conjunto de buenas prácticas de gestión del capital humano. La presente ponencia surge en el marco del proyecto de investigación Optimización de la Calidad del Proceso Software con Gestión del Conocimiento, y tiene como finalidad analizar la posibilidad de que prácticas del modelo de Aprendizaje Experimental se puedan usar como estrategias para la formación y desarrollo de competencias y habilidades en el modelo PCMM; y además analizar técnicas y herramientas software que se podrían utilizar para capturar las experiencias de los integrantes del equipo de desarrollo durante los proyectos software.

1 INTRODUCCIÓN

A medida que los sistemas de software crecen y se vuelven más complejos, se crea una necesidad de un proceso de desarrollo de software que sea bien manejado y entendido, Allen (2003), Romero (2008); así lo demuestran las múltiples implementaciones de Mejoras de Procesos de Software alrededor del mundo, que aportaron varios beneficios, tales como, asegurar la calidad del producto, reducir costos y tiempo de desarrollo, maximizar la productividad, el éxito organizacional y la satisfacción del cliente,

Sulayman (2010). El beneficio más sobresaliente e importante es la calidad, ya que la calidad del producto determina el éxito de una organización, y a su vez dicha calidad está determinada por la calidad de los procesos utilizados para desarrollarlo, Habib (2008), Chrissis (2003); es por esto que las organizaciones deben dedicarle tiempo a la definición, adecuación y mejoramiento continuo de procesos de calidad, Campo (2008). Además, si se considera que la mejora de estos procesos, en la mayoría de los casos está vinculada con las personas, es decir con la fuerza laboral que llevan a cabo los procesos, las organizaciones

también deben orientar sus actividades hacia modelos que sugieran, en particular, buenas prácticas de mejora teniendo como base principal a las personas.

En este sentido, el modelo propuesto por el Instituto de Ingeniería del Software (SEI) dependiente de la Universidad Carnegie Mellon, orientado a la mejora de la capacidad de la fuerza laboral de una empresa, se conoce como People Capability Maturity Model (PCMM); donde este modelo es un conjunto probado de prácticas de gestión de capital humano que proporciona un modelo de cambio organizacional por medio de un marco de referencia evolutivo basado en sistema de prácticas de recursos humanos, Curtis (2010).

Por otro lado, el Modelo de Aprendizaje Experimental, constituye una alternativa que se adapta a las nuevas demandas de profesionales vinculados con la formación y desarrollo de competencias y habilidades, Koll (2001).

Además, las experiencias juegan también un rol principal en las actividades relacionadas con la mejora del proceso entonces existe una necesidad de recopilar experiencias y conocimientos, para lo cual en la actualidad se han desarrollado y propuesto un conjunto de técnicas y herramientas software que permiten capturar las experiencias generadas.

A partir de esto, la finalidad de este trabajo es analizar el Aprendizaje Experimental como una alternativa de buenas prácticas para gestionar el conocimiento, destrezas y habilidades de las personas en torno a la implementación de mejoras de cambios continuos presentados en el Modelo PCMM y, además, analizar técnicas y herramientas software que se podrían utilizar para capturar las experiencias de los integrante del equipo de desarrollo durante los proyectos software, con el propósito de ofrecer a la comunidad desarrolladora de software una estrategia tecnológica que permita mejorar el proceso de desarrollo de software.

El resto del artículo está organizado de la siguiente manera. En la sección 2 se presenta un marco referencial del modelo PCMM, y se analiza qué alternativas de prácticas continuas del modelo, constituyen una estrategia válida que mejora la eficiencia del trabajo organizacional a través de la retroalimentación y aprendizaje de experiencias del grupo de trabajo. En la sección 3 se aborda el modelo de aprendizaje como una alternativa que permita aprender a partir de la experiencia. En la sección 4 se describen algunos enfoques y herramientas que posibiliten gestionar dichas experiencias. En la sección 5 se exponen observaciones realizadas a dichos enfoques y herramientas. En la sección 6 se elaboran las conclusiones y acciones futuras.

2 MODELO PCMM

El objetivo primario del PCMM es la mejora de la capacidad de la fuerza laboral. PCMM proporciona una alternativa para poner en marcha las prácticas de recursos humanos de forma progresiva. Cada nivel establecido, denominado Nivel de Madurez, define un conjunto de prácticas que han de ser llevadas a cabo con el objeto de que dicho Nivel de Madurez sea alcanzado.

PCMM es un modelo orientado de forma exclusiva a la mejora de las prácticas de recursos humanos. Donde cada nivel de madurez, excepto el primero, tiene asociado un conjunto de áreas de proceso. Un área de proceso es un agrupamiento de prácticas que, al realizarse de forma colectiva, satisfacen una serie de metas que contribuyen a la capacidad adquirida por el logro de un nivel de madurez, Curtis (2010).

El modelo PCMM tiene un total de veintidós áreas de proceso distribuidas en los cinco niveles de madurez, como se muestra en la figura 1.

5	Innovación continua Alineación del desempeño organizacional Mejora continua de la capacidad				En optimización
4	Mentoring Gestión de la capacidad organizativa Gestión cuantitativa del desempeño Activos basados en competencias Grupos de trabajo autónomos Integración de competencias				Predecible
3	Cultura de participación Desarrollo de grupos de trabajo Prácticas basadas en competencias Desarrollo de la carrera profesional Desarrollo de competencias Planificación de personal Análisis de competencias				Definido
2	Compensación Formación y desarrollo Gestión del desempeño Entorno de trabajo Comunicación y coordinación Contratación			Administrado	
1					Inicial

Figura 1. Niveles de madurez y áreas de proceso asociadas

De las prácticas propuestas en este modelo, se seleccionó la Capacitación, como una alternativa estratégica para implementar cambios continuos. La Capacitación es una herramienta fundamental que ofrece la posibilidad de mejorar la eficiencia del trabajo de la organización, permitiendo a su vez que la misma se adapte a las nuevas circunstancias que se presentan tanto dentro como fuera de la organización. Proporciona a los empleados la oportunidad de adquirir mayores aptitudes, conocimientos y habilidades que aumentan sus competencias, para desempeñarse con éxito. De esta manera, también resulta ser una importante herramienta motivadora.

La capacitación va dirigida al perfeccionamiento técnico del trabajador, para que éste se desempeñe eficientemente en las tareas asignadas y pueda producir resultados de calidad, prevenir y solucionar anticipadamente problemas potenciales dentro de la organización. A través de la capacitación se busca que el perfil del trabajador se adecue al perfil de conocimientos, habilidades y actitudes requeridas.

Según el autor Gore (2003), las organizaciones utilizan los programas de capacitación como una de las herramientas usuales para incorporar nuevas conductas y modificar rutinas.

Desde una visión de mejora del proceso de desarrollo del software, la Capacitación es la provisión de entrenamiento al personal vinculado al proceso de desarrollo para mejorarlo; esta estrategia de mejora también se encuentra complementada con el mentoring, otra práctica del modelo PCMM, para transferir conocimientos y experiencias en mejora de procesos software. Mentoring es el proceso por el cual una persona con más experiencia (mentor) enseña, aconseja, guía y ayuda a otras menos experimentadas (tutelados) en su desarrollo personal y profesional, invirtiendo tiempo, energía y conocimientos, Soler (2003).

3 APRENDIZAJE EXPERIMENTAL

La teoría del Aprendizaje Experimental se centra en el papel importante que juega la experiencia en el proceso de aprendizaje. Entonces el aprendizaje es el proceso por medio del cual construimos conocimiento mediante un proceso de reflexión y de dar sentido a las experiencias. En este modelo se puede identificar:

- El proceso de aprendizaje, las etapas por las que se transita cuando se aprende algo.
- Los modos en que se adquiere la nueva información y se transforma en algo significativo y utilizable.
- Los estilos individuales de aprendizaje, que son los diferentes modos en que se aborda el proceso de aprendizaje.

Además, el aprendizaje es un proceso que permite deducir conceptos y principios a partir de la experiencia para orientar la conducta a situaciones nuevas, Koll (2001).

Para que haya un aprendizaje efectivo, idealmente se debería pasar por un proceso que incluye cuatro etapas, la cual se esquematiza según Koll (2001) por un modelo de rueda llamado “Ciclo de Aprendizaje”, que consta de cuatro etapas:

- Hacemos algo, tenemos experiencia concreta.
- Luego reflexionamos sobre lo que se hizo, sobre la experiencia, estableciendo una conexión entre lo que se hizo y los resultados obtenidos, etapa denominada observación reflexiva.
- A través de las reflexiones se obtiene conclusiones o generalizaciones que son principios generales referidos a un conjunto de circunstancias más amplias que la experiencia particular, etapa de conceptualización abstracta.
- Finalmente se prueba en la práctica las conclusiones obtenida, utilizando como guía para orientar la acción en situaciones futuras, etapa de experimentación activa.

Este modelo se presenta en la fig. 2

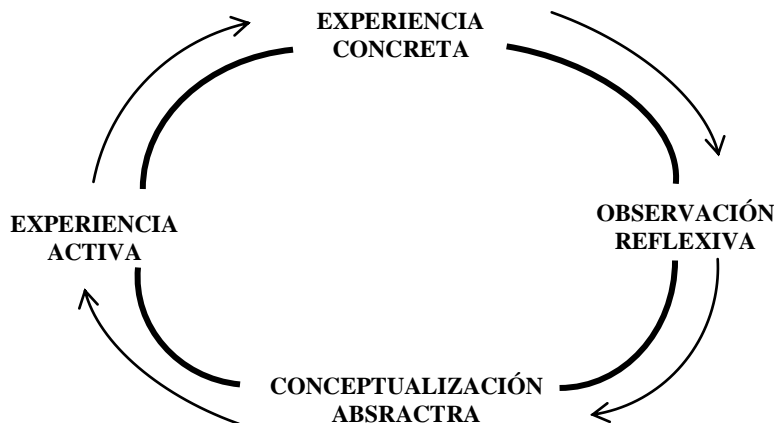


Figura 2. Modelo de rueda llamado “Ciclo de Aprendizaje”

Cada una de estas etapas tiene su propio valor debido a que cada uno de ellas genera una forma particular de conocimiento. Hacer, observar y reflexionar, desarrollar conceptos y generalizaciones y experimentar activamente con nuevas ideas, son diferentes modos de conocimientos, entonces para lograr un aprendizaje efectivo resultaría necesario pasar por todas las etapas.

En el ámbito de desarrollo del proceso software, el conjunto de conocimientos que se van generando no es un proceso individual, sino un conjunto relacionado de procesos individuales que se fortalecen a partir del aprendizaje que da la experiencia. En este ámbito, sus miembros aprenden desde la experiencia, sí:

- Están involucrados y comprometidos con el equipo y sus propósitos, creando nuevos conocimientos e identificando desafíos (experiencia concreta).
- Se involucran en la reflexión y la conversación acerca de las experiencias del equipo y realizan observaciones para asegurar que todo el conocimiento disponible haya sido tratado (observación reflexiva).
- Piensan críticamente acerca de cómo trabaja el equipo y presentan nuevas teorías, conciben planes o modelos y sitúan eventos abstractos en explicaciones concretas y sencillas (conceptualización abstracta).
- Toman decisiones, realizan acciones y experimentan diversos enfoques y estrategias para resolver problemas (experimentación activa).

Cuando las experiencias no están gestionadas en las organizaciones, generan problemas tales como:

- No existen registros que puedan almacenar experiencias y conocimientos generados durante la ejecución del desarrollo de procesos software.
- Los empleados no tienen una retroalimentación continua de los éxitos o fracasos de desarrollos anteriores.

Por lo tanto, para hacer frente a estos problemas, se debe tener en cuenta una combinación de consideraciones que permitan realizar un cambio de la cultura organizacional y disponer de una infraestructura tecnológica.

4 ENFOQUES Y HERRAMIENTAS PARA GESTIONAR EXPERIENCIAS DE MIEMBROS DE EQUIPOS SOFTWARE

4.1. Fábrica de experiencia

El enfoque fábrica de experiencia o Experience Factory es una infraestructura física y/o lógica

que apoya los proyectos de desarrollo. Apunta esencialmente a la captura, análisis y empaquetado de experiencias de todo tipo adquiridas durante el desarrollo de proyectos software con el objetivo de reutilizar esa experiencia en nuevos proyectos de desarrollo de software, Basili, Caldeira & Rombach, 1994.

Este enfoque divide los esfuerzos de desarrollo de software en dos unidades con responsabilidades separadas de desarrollar proyectos de software y capturar experiencias.

Basili, Lindvall & Costa, 2001 explican que la implementación física de una fábrica de experiencia es un sistema de administración de experiencia compuesto de contenido, estructura, procedimientos y herramientas.

- El contenido (que pueden ser datos, información, conocimientos o experiencias).
- La estructura (que es la forma en que está organizado el contenido) constituyen lo que se denomina Base de Experiencia.
- Los procedimientos son instrucciones acerca de cómo manejar la base de experiencias, y las herramientas soportan la gestión del contenido y la ejecución de los procedimientos.

En una fábrica de experiencia sus valores centrales son que, para mejorar, los empleados necesitan aprender de experiencias pasadas y, para que los empleados aprendan, la organización necesita crear un ambiente de aprendizaje, Basili, Lindvall & Costa, 2001.

4.2. Bases de experiencia software

El cuerpo de conocimientos de una base de experiencia son típicamente de distintos tipos (know-how, know-why, know-what) y utiliza diferentes esquemas de representación, tales como modelos explícitos, experiencias documentadas o lecciones aprendidas, así como conocimientos tácitos y habilidades más o menos estructurados poseídos por las personas, Ruhe & Bomarius, 2000.

Para implementar con éxito una base de experiencia, existen cuatro factores claves Conradi, Lindvall & Seaman, 2000:

- Cambio cultural: es importante que las personas provean conocimiento a la base de experiencia y que también hagan uso del conocimiento que esté disponible en ella.
- Estabilidad: relacionada con la habilidad para gestionar los cambios de manera controlada.
- Valor para el negocio: la base de experiencia se percibe como un elemento exitoso, si ésta provee un valor concreto y demostrable para el negocio.

- Implementación incremental: si la implementación y la introducción de una base de experiencia se realizan en pequeños incrementos y en estrecha conexión con sus futuros usuarios recibiendo de estos una retroalimentación continua, ambas instancias se consideran exitosas.

A lo largo de los últimos años se han desarrollado diversas herramientas software que recurren a los enfoques arriba mencionados, fábrica y base de experiencia software. Una acotada selección de las mismas se presenta a continuación:

- *Software Experience Center.(SEC)*

El SEC “Software Experience Center”, Schneider, von Hunnius & Basili, 2002, basado en el concepto de la Experience Factory, Basili, Caldeira & Rombach, 1994, tiene como objetivo operacional proveer a las unidades de negocio de los conceptos de una organización que aprende y un prototipo de una base de experiencia. Proporciona apoyo en todas las actividades, desde la educación de experiencias hasta hacer disponible esas experiencias para la tarea de software entre manos.

El SEC ha mejorado muchos procesos en la organización y a logrado que el aprendizaje por medio de la experiencia se convierta en una parte más natural de la vida diaria de las unidades de negocio, Schneider, von Hunnius & Basili, 2002.

- *Experience Engine.*

El software Experience Engine, Johansson, Hall & Coquard, 1999, se sustenta en el conocimiento tácito a pesar de ser una variante de Experience Factory, Basili, Caldeira & Rombach, 1994, basada en experiencias almacenadas en bases de experiencia.

A partir de la definición de dos nuevos roles en la organización, el comunicador de experiencia “experience communicator” y el agente de experiencias “experience broker”, se logró hacer accesible el conocimiento tácito a un grupo mayor de personas. El primero es una persona que posee un conocimiento profundo acerca de uno o más temas, mientras que la misión del agente de experiencias es conectar al comunicador con la o las personas que tienen un problema. El comunicador no resuelve el problema sino que educa y asiste al poseedor del mismo acerca de cómo resolverlo, Johansson, Hall & Coquard, 1999.

- *Knowledge Dust to Pearls*

El enfoque Knowledge Dust to Pearls, Basili & Seaman, 2002, provee mecanismos que facilitan la iniciación rápida de una base de experiencia. Mediante la idea de capturar las partículas de

conocimiento “knowledge dust” que los empleados utilizan e intercambian diariamente en sus actividades e, inmediatamente y con mínimas modificaciones, hacerlas disponibles a través de toda la organización. En forma paralela, estas partículas de conocimiento son analizadas, sintetizadas y transformadas en perlas de conocimiento “knowledge pearls” que representan elementos de conocimiento más sofisticados, refinados y valiosos, Basili & Seaman, 2002.

4.3. Técnicas para captura de experiencia a partir de proyectos software

4.3.1. Análisis post mortem de proyectos

La técnica del análisis *post mortem* de proyectos, Birk, Dingsoyr & Stalhane, 2002, se emplea con el propósito de capturar experiencias y sugerencias de mejora a partir de proyectos completados o cuando en el proyecto se haya alcanzado un hito significativo. Estos autores, delimitan la técnica a proyectos de software y sostienen que en cada proyecto de este tipo, los miembros del equipo ganan nuevo conocimiento y experiencia que pueden ser beneficiosos tanto para futuros proyectos como para el propio desarrollo profesional de los miembros del equipo.

La aplicación de esta técnica consiste de tres fases:

i) Preparación: en esta fase se recorre la historia del proyecto para obtener un mejor entendimiento de lo que ha ocurrido y se revisan todos los documentos disponibles, tales como planes del proyecto, la estructura de desglose de tareas (“work breakdown structure”) y los reportes de revisión. En esta fase también se determinan los objetivos específicos para el análisis a realizar.

ii) Recolección de datos: esta fase está dedicada a obtener la experiencia relevante del proyecto consistente no sólo de los problemas o aspectos negativos que deberían haberse evitado sino también de los aspectos exitosos. Una vez identificados los temas importantes, se debe priorizarlos antes de proceder al análisis. El establecimiento de prioridades asegura que se traten en primera instancia los aspectos de mayor significación.

iii) Análisis: en esta fase un moderador conduce una sesión de retroalimentación para discutir e intercambiar ideas y puntos de vista sobre los temas identificados en la fase anterior y finaliza con la elaboración de un informe con las conclusiones a las que se llegó y las recomendaciones que correspondan.

4.3.2. Revisiones post-proyecto

El autor Harrison, 2003, considera la realización de revisiones post-proyecto (post-project reviews) como una forma de proveer un mecanismo formal para transferir la experiencia de un equipo de proyecto a una memoria corporativa una vez que se ha completado un proyecto y mientras esas experiencias están aún frescas en las mentes de los participantes. La experiencia capturada se vuelca a un repositorio de lecciones aprendidas cuyo propósito es facilitar la organización, mantenimiento y diseminación del conocimiento capturado. El repositorio está basado en tecnología web y dispone de una interfase basada en formularios para que los suministradores de lecciones aprendidas puedan agregar nuevas experiencias al mismo.

4.4. Herramientas colaborativas para compartir experiencias: Wikis, wikis semánticos y ontologías

Un wiki es esencialmente una colección de sitios web conectados por vínculos de hipertexto, Schaffert, S., 2006a. Mientras que, los *wikis* semánticos son la combinación de wikis con tecnologías de la web semántica, Schaffert, Westenthaler & Gruber, 2006b.

Tanto a nivel personal como organizacional, los *wikis* se están transformando en herramientas populares de Gestión del Conocimiento, Oren, et al., 2006, y pueden verse como una plataforma liviana (“lightweight”) para intercambiar artefactos reutilizables dentro y entre proyectos software, y entienden que también pueden ser considerados como formas de memorias organizacionales o fábricas de experiencias.

En otro sentido, una ontología define los términos y relaciones básicas que componen el vocabulario de un área tópica, así como las reglas para combinar términos y relaciones para definir extensiones a ese vocabulario, Corcho, O. et al., 2006.

• *Riki*

Los autores Rech, Bogner & Hass, 2007, reportan el desarrollo de un wiki, denominado Riki, que utiliza tecnologías de razonamiento basado en casos (“case-based reasoning”) y ontologías para proveer un marco formal y consistente para describir conocimientos y experiencias, y cuyo

propósito es implementar un sistema de documentación orientada a la reutilización de conocimientos sobre proyectos software.

La ontología y las plantillas de documentos (“templates”) enriquecen el contenido de Riki con semántica que permite a los usuarios aumentar el conocimiento incorporado en Riki con información adicional y experiencias documentadas, así como compartir experiencias y reutilizar conocimiento relativo a proyectos, Rech, Bogner & Haas, 2007.

• *Wikitología*

Wikitología, Klein, Hoecht & Decker, 2005, es un enfoque que aúna técnicamente wikis y ontologías, y cuyo propósito es que el conocimiento sobre Ingeniería de Software pueda ser constantemente mantenido y cultivado por los ingenieros de software. Este enfoque, a largo plazo, debería resultar en una calidad superior de los sistemas software debido a una mejor disponibilidad de los conocimientos relativos a Ingeniería de Software.

• *MASE*

El autor, Chau & Maurer, 2005, describen la utilización de wikis en una herramienta, denominada MASE, orientada a mejorar los mecanismos de compartición de experiencias entre diferentes equipos de desarrollo software en una organización. Esta herramienta permite a los usuarios registrar información de manera informal y no estructurada, así como también definir las tareas de un proyecto y almacenar información sobre las mismas en base a formatos específicos (estructurados). La herramienta, además, provee capacidades de búsqueda de texto en cualquiera de las páginas wiki y apoya el trabajo colaborativo tanto en forma sincrónica como asincrónica.

5 OBSERVACIONES A LOS ENFOQUES, TÉCNICAS Y HERRAMIENTAS PRESENTADAS

En la tabla siguiente se presentan observaciones realizadas al análisis de los enfoques, técnicas y herramientas presentadas:

Enfoques/ técnicas / herramientas	Características
Fabrica de experiencias y sus derivaciones que hacen uso de repositorios de experiencias	<ul style="list-style-type: none"> • Es el enfoque más comprehensivo para capturar las experiencias en el ámbito de la Ingeniería de Software • Su marco de trabajo establece cuales son las actividades de Gestión de Conocimiento que es necesario realizar (educación, análisis,

Enfoques/ técnicas / herramientas	Características
	generalización, empaquetado y diseminación) <ul style="list-style-type: none"> • Presenta la carencia de no prescribir cómo deben llevarse a cabo esas actividades, así como el estar basados, la mayoría, en la estrategia de codificación de conocimientos, Chau & Maurer, 2005.
Análisis post mortem y similares	<ul style="list-style-type: none"> • Presentan el inconveniente de ser extemporáneos respecto del momento de ocurrencia de las experiencias que se pretenden capturar. • Desafortunadamente, la mayoría de los ingenieros de software no tienen tiempo para “finalizar” un proyecto cuando ya están siendo reasignados a otros, y no registran en la mayoría de los casos sus experiencias, Desouza, Dingsoyr & Awazu, 2005.
Wikis y wikis semánticas	<ul style="list-style-type: none"> • Las mismas apuntan a apoyar el trabajo colaborativo de gestión documental para facilitar la reutilización de los conocimientos plasmados en esos documentos.

Tabla 1. Características y observaciones a fábrica de experiencia, análisis post mortem y wikis

Otras observaciones:

*Con respecto a la fábrica de experiencias y el análisis post mortem ambos se centran en el resultado o producto final de la adquisición de experiencia, esto es, en la experiencia misma adquirida, pero no establecen ningún mecanismo relacionado con el propio proceso de adquisición de esa experiencia; no delimitan, a priori, los tipos de conocimientos y de experiencias a capturar.

5. CONCLUSIONES Y ACCIONES FUTURAS

Si bien el modelo PCMM describe las prácticas de gestión de recursos humanos que una organización debe implementar para lograr un determinado nivel de madurez, el modelo no describe la manera en que esas prácticas deben implementarse en la organización; es decir, el modelo indica QUÉ debe implementarse, pero no la manera en CÓMO hacer esta implementación. Por este motivo, en este trabajo se propone el modelo de Aprendizaje Experimental como una alternativa para que profesionales informáticos puedan capacitarse en la formación y desarrollo de competencias y habilidades, a partir de sus experiencias.

Por lo tanto, este enfoque de aprendizaje es una buena práctica de mejora de capacitación, que requiere desde la organización una cultura colaborativa para implementar esta estrategia de capacitación.

Las herramientas y técnicas de captura descriptas permiten, incorporar, preservar, aplicar, diseminar en una organización, la experiencia y el

conocimiento adquiridos en proyectos software pasados y que pueden utilizarse para mejorar las prácticas en proyectos futuros. Por lo tanto, conocer los distintos tipos de técnicas, herramientas de captura de experiencias da una oportunidad de mejorar su calidad.

Por otro parte es oportuno aclarar que esta propuesta de investigación se encuentra aun en un estado inicial, resultando importante indagar la posibilidad de que la misma constituya una alternativa válida de buenas prácticas de formación y desarrollo de recursos humanos en el ámbito de las organizaciones que están preocupadas por la mejora y calidad de sus procesos de desarrollo de proyectos software.

Como trabajo futuro se plantea:

- Implementar el Aprendizaje Experimental a través de herramientas de nuevas tecnologías de aprendizaje.
- Completar esta primera aproximación del modelo de mejora de proceso software basado en prácticas de recursos humanos en el marco de esta investigación, con prácticas que complementen las distintas áreas de proceso del modelo PCMM.
- Estudiar y analizar herramientas para capturar conocimientos y aprendizajes basados en la experiencia que los miembros de los equipos de proyecto adquieren “durante” la realización de sus actividades de proyecto, para evitar la pérdida de los mismos.

6. REFERENCIAS

Allen, P., Ramachandran, M, & H. Abushama, Prisms: an approach to software process

- improvement for small to medium enterprises, in Proc. Third Int Quality Software Conf, 2003, pp. 211-214.
- Basili, V. & C. Seaman, *The experience factory organization*, IEEE Software, 19, 30-31, 2002.
- Basili, V., G. Caldeira & H. Rombach, *The experience factory*, en Marciniak, J. (ed.) Encyclopedia of software engineering, J. Wiley & Sons, New York, 1994.
- Basili, V., M. Lindvall & P. Costa, *Implementing the experience factory as a set of experience bases*, International Conference on Software Engineering and Knowledge Engineering (SEKE '01), Buenos Aires, 2001.
- Birk, A., T. Dingsoyr & T. Stalhane, *Postmortem: never leave a project without it*, IEEE Software, 19, 43-45, 2002.
- Campo, L. F., Modelos de capacidad y madurez y la industria del software en Colombia, Revista Generación Digital, vol. 7, pp. 22-25, 2008.
- Chau, T. & F. Maurer, *A case study of wiki-based experience repository at a medium-sized software company*, University of Calgary, Department of Computer Science, 2005.
- Conradi, R., M. Lindvall & C. Seaman, *Success factors for software experience bases: what we need to learn from other disciplines*, ICSE '2000, Limerick, 113-119, 2000.
- Corcho, O., et al., *Ontological engineering. Principles, methods, tools and languages*, en: Calero, C., Ruiz, F., Piattini, M. (eds.): Ontologies for software engineering and software technology, Springer, Berlin, 1-39, 2006.
- Curtis, B., Hefley W.E. & Miller S. A, People CMM. A Framework for Human Capital Management. 2nd. Ed. 2010, Boston, MA: AddisonWesley
- Chrissis, M. B., Konrad, M. & Shrum, S., CMMI: Guidelines for Process Integration and Product Improvement, Pearson, Ed. Addison Wesley, 2003.
- Gore, E., Conocimiento Colectivo, Ediciones Gráfica, pag. 19-20, Buenos Aires, 2003.
- Habib, M., Ahmed, S., Rehmat, A., Khan, M. J. & Shamail, S., Blending six sigma and Cmmi an approach to accelerate process improvement in smes, in Proc. IEEE Int. Multitopic Conf. INMIC 2008, 2008, pp. 386-391.
- Harrison, W, *A software engineering lessons learned repository*, Proceedings of the 27th Annual NASA, 2003.
- Johansson, C., P. Hall, & M. Coquard, *Talk to Paula and Peter; they are experienced. The experience engine in a nutshell*, Proceedings of the 11th International Conference on Software Engineering and Knowledge Engineering, Learning Software Organizations, Methodology and Applications, 171-185, 1999.
- Klein, B., C. Hoecht & B. Decker, *Beyond capturing and maintaining software engineering knowledge. Wikitology as shared semantics*, Knowledge Engineering and Software Engineering Workshop, 28th German Conference on Artificial Intelligence, Koblenz, Alemania, 2005.
- Koll, D.A., *Experiential Learning Theory Bibliography 1971-2001*, Boston, Ma. McBer and Co, 2001.
- Oren, E. et al., *Semantic wikis for personal knowledge management*, Lecture Notes in Computer Science, 4080, 509-518, 2006.
- Rech, J., C. Bogner & V. Haas, *Using wikis to tackle reuse in software projects*, IEEE Software, 24, 99-104, 2007.
- Ruhe, G. & F. Bomarius, *Learning software organizations*, Springer, Berlin, 2000.
- Soler, R., *Mentoring. Estrategia para el desarrollo de recursos humanos*, 2003, Madrid: Gestión 2000.
- Schaffert, S., *IkeWiki: a semantic wiki for collaborative knowledge management*, International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, (WETICE'06), 388-393, 2006a.
- Schaffert, S., R. Westenthaler & A. Gruber, *IkeWiki: A userfriendly semantic wiki*, 3rd European Semantic Web Conference (ESWC'06), 2006b.
- Schneider, K., J. von Hunnius & V. Basili, *Experience in implementing a learning software organization*, IEEE Software, 19, 46-49, 2002.
- Sulayman, M. & E. Mendes, *Quantitative assessments of key success factors in software process improvement for small and medium web companies*, in SAC '10: Proceedings of the 2010 ACM Symposium on Applied Computing (SAC). Sierre, Switzerland, March 22-26, 2010.