

Experiencia de la implementación de un modelo de capacitación en empresas de desarrollo de software basándose en ambientes ágiles

Carlos Enríquez Ramírez

Universidad Politécnica de Tulancingo

carlos.enriquez@upt.edu.mx

María del Pilar Gómez Gil

Instituto Nacional de Astrofísica, Óptica y Electrónica

pgomez@inaoep.mx

Resumen

Existe una importante variedad de métodos para la capacitación de ingeniería de software. Algunos se ejecutan en ambientes de desarrollo reales, donde los recién graduados o los estudiantes de los últimos niveles de pregrado pueden capacitarse de manera previa antes de enfrentarse a los ambientes reales de trabajo. Independientemente del estilo utilizado, la capacitación se debe considerar como una herramienta de mejora de los procesos de trabajo en lugar de un requisito previo para el desempeño exitoso en la parte técnica. La dinámica del desarrollo de software es cambiante, por lo tanto la mejora continua en los procesos de capacitación es necesaria para mantener el ritmo en la obtención de los conocimientos adecuados. En el presente trabajo se reporta la experiencia adquirida en la implementación de un modelo de mejora del proceso de capacitación, conocido como “Modelo de Capacitación de Ambientes Ágiles bajo Moprosoft” (CAAM), el cual está basado en esquemas ágiles de gestión de actividades. El modelo se implementó mediante un curso taller impartido a 8 pequeñas empresas mexicanas dedicadas a la construcción de software. Los participantes mostraron resultados de mejoría en promedio de 3.8% en las competencias asociadas a planeación, comunicación y uso de herramientas ágiles.

Palabras claves. Modelo de Capacitación, Metodologías Ágiles, Mejora de procesos.

Introducción

La ingeniería de software incluye modelos, estándares y métodos altamente probados experimentalmente y que el profesionalista puede aplicar. Sin embargo, resulta muy frustrante cuando recién graduados y/o desarrolladores con experiencia no consiguen adaptar estos modelos a las necesidades de la empresa y en consecuencia se dificulta enormemente su inserción en el ámbito laboral. Aun así, en la actualidad es solicitado el uso de estándares y modelos de desarrollo por muchas compañías, tanto a nivel nacional como internacional. Esto implica que las pequeñas y medianas empresas (PyMEs) deben erogar en gastos de capacitación, pues muchas veces este conocimiento no está incluido dentro de los planes y estudios universitarios, dando pauta a que sus egresados carezcan de disciplina en el uso de los mismos. Si la prioridad principal de las PyMEs es prevalecer en el mercado en vez de establecer un modelo de desarrollo, entonces se dificulta fuertemente su incorporación en ambientes competitivos.

Existe una importante variedad en métodos para la capacitación de ingeniería de software (Steve D., 2010). Algunas técnicas se ejecutan en ambientes de desarrollo reales, donde los recién graduados o los estudiantes de los últimos niveles de su preparación de pregrado, pueden capacitarse de manera previa antes de enfrentarse a los ambientes reales de trabajo (Dawson, 2000). Otro esquema de entrenamiento implica el uso de sistemas diseñados para proporcionar entrenamiento basándose en el uso de la computadora, permitiendo avanzar al siguiente nivel siempre y cuando se acredite en el que se está. La dinámica del desarrollo de software es cambiante, por lo tanto la capacitación es necesaria para mantener el ritmo en la obtención de los conocimientos adecuados como se propone en el modelo de entrada-proceso-salida propuesto en (Bushnell, 1990). Asimismo, la elaboración de software requiere de un proceso de capacitación que de manera previa estandarice los diversos escenarios donde se ubican los empleados de mayor edad (Sterns & Doverspike, 1989).

Sea cual fuere el estilo utilizado, la capacitación se debe considerar como un intento de mejorar los procesos del trabajo en lugar de un requisito previo para el desempeño exitoso en la parte técnica. Asimismo, debe recordarse que el motivo de la implementación de un mecanismo de capacitación en el área de software es el generar competencias en los procesos de madurez y no tanto en el aspecto técnico. Aunado a lo anterior, los costos generados por la implantación de cursos debe de disminuir (Garg & Varma, 2008) a fin de ser más accesible el uso de la mejora de procesos de calidad en las empresas desarrolladoras que tienen escasos presupuestos.

Es de vital importancia tener un objetivo claro para diseñar un plan de capacitación. Debe enfocarse principalmente en que los individuos sepan operar y tengan las habilidades para ejecutar correctamente sus actividades, esto implica el contar con elementos que permitan mantener actualizados los procesos que ya se llevan a cabo, además de incluir nuevos, lo que permitirá la madurez. Desgraciadamente existe una disparidad de lo que se muestra en los procesos de capacitación con lo que realmente se enseña en las universidades. Los materiales de enseñanza se quedan atrás debido en muchas ocasiones a la falta de flexibilidad de los planes de estudio (Huang P., Chen M. & Chen S., 2008); (Li, Liu, & Guo, 2008).

En este trabajo se propone que las PyMEs que tengan al menos el nivel 1 de la norma NMX-I-059-NYCE-2005 adopten un modelo de mejora del proceso de capacitación para desarrolladores de software basado en ambientes ágiles, medible para que los integrantes del desarrollo se disciplinen en ambientes de trabajo con filosofías de desarrollo ágil. Con esto se espera que las PyMEs logren mejorar sus procesos de capacitación y mantener sus proyectos dentro de un presupuesto y tiempo de desarrollo planeados. A través de esta investigación, se han definido estrategias que permiten disciplinar a los integrantes de los equipos de desarrollo en el uso de modelos y herramientas ágiles de manera incremental. El combinar metodologías ágiles con modelos de mejora de procesos, tiene como objetivo adoptar las ventajas de ambos enfoques.

En una encuesta realizada por (Mejía, Ania, & Gamboa, 2006) se considera que hay escasez de recursos humanos, principalmente de aquellos con competencias para el aseguramiento de la calidad, la alta dirección y la administración de proyectos, más que con competencias para el desarrollo de los proyectos en sí. Por otra parte, la realidad actual es que el proceso de capacitación en los desarrolladores de software es demasiado largo, incompatible muchas veces con las urgencias de entrega que provocan los desarrollos de software. Varios administradores de desarrollo y compañías de consultoría están intentando acelerar estos procesos. Por ejemplo, se cuenta con un método que permite reducir el nivel de expertos de desarrollo de software de 2 años a 4 o 6 meses, mediante el proceso de compartición de conocimiento (Bologa & Lupu, 2007).

Los factores usados para medir la efectividad de una actividad de capacitación, pueden estar dados por las categorías siguientes: cambios en el comportamiento, cambios en la actitud, adquisición de conocimiento y habilidades, medición de resultados tangibles, reacción y participación (Kirkpatrick D., 2009). Desde esta perspectiva, es posible distinguir al menos cuatro niveles de evaluación de la capacitación, basándose en la recopilación y análisis de datos (Goldstein, 1993):

1. Evaluación del proceso de capacitación, la cual está dirigida a verificar si se cumplió en la práctica el diseño de instrucción previsto. Deben considerarse aquellos aspectos que influyen en la calidad y la eficiencia del proceso de capacitación.
2. Evaluación del aprendizaje, el cual se entiende como la adquisición de competencias técnicas o psicosociales mediante experiencias sistematizadas con objetivos de aprendizaje medibles. En este punto se deben tener en cuenta las competencias a evaluar, las cuales deberán ser:
 - a) Competencias técnicas
 - i. Conocimientos
 - ii. Habilidades intelectuales
 - b) Destrezas sicomotoras

- i. Competencias psicosociales
- c) Habilidades interpersonales
- d) Disposición anímica en el trabajo

3. Evaluación del impacto de la capacitación en el desempeño laboral. Tiene por objetivo comprobar en qué medida las competencias aprendidas se reflejan en mejoras en el desempeño del trabajador. Se entiende por desempeño el comportamiento del trabajador en el marco de las normas y objetivos de la organización. Para esto se puede hacer un muestreo estadístico antes de realizar la capacitación utilizando por ejemplo, el porcentaje de los participantes no capacitados.

4. Evaluación de la rentabilidad de la capacitación. Ésta tiene por objeto determinar en qué medida los resultados de la capacitación compensaron sus costos, en cuanto a mejoras de la eficiencia o productividad en el trabajo o del clima organizacional en la empresa. Se trata, pues, de una evaluación de beneficios y costos realizada desde la perspectiva de la gerencia de la empresa.

Sesiones de Entrenamiento.

Para desarrollar esta investigación, se realizó un análisis de diversos trabajos de capacitación propuestos en la literatura (Bushnell, 1990); (Barbazette, 2006). De particular interés resulta el modelo propuesto por (Buckley R. Caple J., 2009), ya que representa un enfoque sistemático de entrenamiento, identificando los elementos para el funcionamiento de un programa de capacitación genérico. En el modelo se incluyen, a grandes rasgos, características de conocimiento, habilidad y desarrollo de aptitudes, mismas que forman parte de una capacitación basada en competencias.

Para la validación del modelo se seleccionaron empresas, en donde se implementó CAAM. Los desarrolladores de estas empresas involucrados en el experimento contestaron el cuestionario como tarea inicial, antes de realizarse el experimento de

implementación, y un cuestionario al finalizar el experimento. Con el objetivo de incorporar habilidades consideradas como ágiles, estas son:

- Planeación de actividades para ambientes ágiles.
- Comunicación con el cliente del producto
- Uso de herramientas ágiles.
- Uso de herramientas de pruebas.

Los cuestionarios se diseñaron siguiendo el método de escalamiento de Likert (Hernández, Fernández, & Baptista, 2007, pág. 341), tomando en cuenta las variables involucradas en la hipótesis. Se aplicó el modelo a empresas para la obtención de datos que posteriormente se trataron mediante técnicas estadística descriptiva para obtener el comportamiento. Las fases propuestas para dirigir el experimento se conformó por los siguientes pasos:

Fase I: Formulación de las pregunta de investigación, construcción de un conjunto de premisas e hipótesis.

Fase II: Diseño e implementación del modelo CAAM.

Fase III: Diseño del instrumento de encuesta y la recolección de los datos mediante el mismo.

Fase IV. Desarrollo de taller de Capacitación en *Scrum*-Moprosoft, el que se llevó a cabo en la ciudad de Puebla e Hidalgo contando con una participación de 26 participantes de empresas tales como: Synergy, Zutri, BTOB Consultores Eviciti, como se muestra en la fig.1.



Figura 1. Taller de capacitación de *Scrum* en la ciudad de Puebla.

El mecanismo que se llevo a cabo en el taller fue bajo los siguientes puntos:

- Inducción a las metodologías ágiles.
- Facilitación de conceptos ágiles en la práctica
- Fase de independencia para la utilización de *Scrum* en algún proyecto de la empresa.
- Mapeo de actividades propias de Moprosoft en *Scrum*

Fase V: Análisis de los datos obtenidos en la fase III.

Las actividades del área clave de CAAM, principalmente se basan en el uso de competencias como una mejora constante de la capacidad, para que pueda llevarse a cabo las tareas y responsabilidades asignadas. El objetivo de CAAM es asegurar que todos los individuos tienen las competencias requeridas para realizar sus tareas haciendo uso del área de proceso, mediante un conjunto de prácticas relacionadas que cuando se realizan colectivamente satisfacen la capacidad adquirida en el logro de un nivel de madurez. Las funciones a lograr mediante la aplicación del área de proceso del modelo son:

- Planeación de actividades para ambientes ágiles.
- Comunicación con el cliente del producto
- Uso de herramientas ágiles.
- Uso de herramientas de pruebas.

En la tabla 1 se muestra las principales funciones asociadas al modelo, y la evolución que presentan a lo largo de los niveles por los cuales esta compuesto el modelo de CAAM.

Tabla 1 Descripción de niveles para la funcionalidad de CAAM.

Función	Nivel 0	Nivel 1	Nivel 2	Nivel 3
Integración Moprosoft-Scrum	No se cuenta con conocimiento	Se inicia con la identificación de las actividades que se realizan en cada una de las fases de Moprosoft cuando se realiza la gestión de un proyecto usando Scrum.	Se inicia el uso de Scrum, cuidando que se cumpla con el mayor número posible de especificaciones de Moprosoft en la categoría de operación.	Se realiza una medición del esfuerzo en el uso de los procesos de gestión, es decir, en el logro de las competencias a este nivel.
Implementación de filosofía ágil	No se cuenta con conocimiento ó se cuenta pero no se aplica	Planear los principales instrumentos ágiles que se usarán en la capacitación.	Establecer una planeación de 40 horas para el uso de: planeación ágil, historias de usuario y estimación de cada uno de los <i>Sprints</i> .	Medición del esfuerzo y competitividad en el uso de la filosofía ágil, en el marco de Moprosoft.
Proceso de construcción de código	No se cuenta con documentación para guiar	Conocimiento de estructuras de tipo TDD	Entrenamiento del desarrollo de código mediante el uso	Competitividad en la construcción del código

	el proceso de capacitación		TDD.	mediante TDD. Obtención de métricas de productividad y esfuerzo.
--	----------------------------	--	------	---

Cabe hacer notar que la medición de las prácticas claves es importante para tener un parámetro que indique de manera cuantitativa si se están logrando el objetivo de CAAM mediante el uso de las métricas que sirven de control del trabajo a implementar.

Proceso de Implementación de CAAM

En esta sección se describen los pasos que se ejecutan, por medio de actividades concretas, para la implementación de CAAM. Para diseñar este proceso de implementación, se tomó como base la especificación revisada en (Jaap, Heijstek, & Linders, 2008); donde se explica la planificación de la implementación desde el proyecto, producto, integración del proceso y medición. La planeación de proceso desarrolla la capacidad para definir, implementar y mejorar conjuntos de procesos en una organización.

Los pasos a seguir para implementar CAAM son:

Paso 1.- Para la implementación del modelo de entrenamiento propuesto en este trabajo (CAAM) se necesita detectar el grado de conocimiento de la aplicación de las técnicas ágiles por parte de los participantes. Para esto se aplica a los desarrolladores de la empresa un cuestionario inicial, que consta de 11 preguntas, por medio del cual se determina el conocimiento de las prácticas de manera intuitiva o la carencia de las mismas.

Paso 2.- Determinación del grado de conocimientos. Aquí se deben identificar las competencias ágiles necesarias a usar en el ambiente de trabajo de la empresa donde está implementándose CAAM.

Paso 3.- Evaluar que tanto hay que avanzar. Implica definir cuantos cursos/horas de trabajo son necesarios y cada cuando se deben de evaluar.

Paso 4.- Definición de la dinámica de los cursos. Esto se realiza mediante la impartición de los materiales por parte del entrenador(a) y resolviendo casos prácticos, teniendo cuidado de considerar las competencias a cubrir. Se realiza una medición del logro de las competencias a nivel individual y en equipo. El seguimiento del curso se efectúa mediante la planeación de las actividades teóricas y prácticas, esto permite pasar del nivel cero al siguiente (uno) de Moprosoft, debido que en un inicio no se contaba con un modelo de planeación ágil y en el siguiente del nivel del modelo de CAAM se propone como característica principal el planear y el conocimiento de las competencias a lograr.

Paso 5.- A la finalización del curso se contesta un cuestionario de 11 preguntas para determinar la nueva actitud que adopta el/la capacitado(a) en el desarrollo de software. Este cuestionario mide la adquisición de habilidades asociadas a: trato al cliente, tamaño del equipo, cumplimiento de las planeación, tiempos de decisión.

Paso 6.- Hacer reuniones con los equipos de trabajo para “hacer suyos” los estilos/actividades, ajustar los roles de las personas de acuerdo a lo aprendido en los cursos tomados.

Paso 7.- Ajustar el plan de cursos realizados en el paso 3 de acuerdo a la filosofía de CAAM.

Análisis de datos

De la hipótesis planteada en este trabajo la “mejora de las competencias en los desarrolladores” mediante un modelo de procesos de capacitación. Se efectuaron tres cursos con un total de 32 asistentes de los cuales para fines del reporte solamente se tomaron los resultados emitidos por 27, debido a la inconsistencia de los datos restantes. Mediante la aplicación de un cuestionario de 11 preguntas como fase de pre-capacitación y la misma cantidad de preguntas se realizaron en una siguiente fase de post-capacitación para verificar el grado de aceptación por parte de los participantes. La finalidad de ambas encuestas fue el encontrar la asociación de las respuestas a las características principales de las competencias que se desean desarrollar, que son: la planeación de actividades para ambientes ágiles, la comunicación con el cliente, el uso de herramientas ágiles y el uso de herramientas de pruebas.

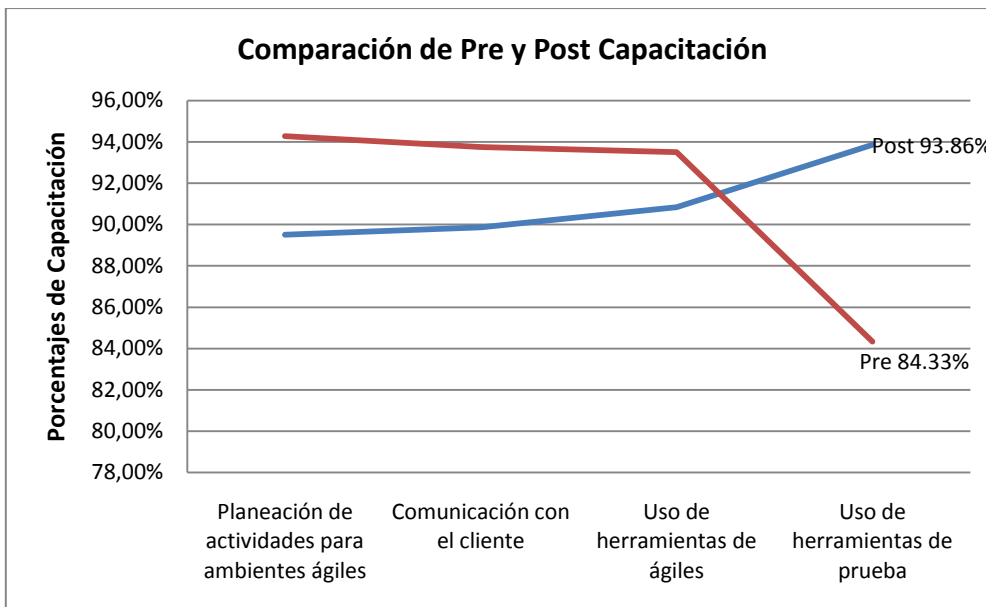


Figura. 1 Gráfico de comparación de resultados de encuestas.

Se observa en el gráfico de la Fig. 1 una mejora en algunos de los rasgos de las competencias a adquirir mediante el curso de *Scrum*. En el caso de la característica de planeación de actividades para ambientes ágiles se observa una mejora del 4.78% puntos con respecto al primer cuestionario. De manera similar, en la característica "comunicación con el cliente" se tiene una mejora del 3.89% puntos y en la de "uso de

herramientas ágiles" una de 2.66% puntos, al compararse el cuestionario aplicado antes de la capacitación con respecto al aplicado después de capacitación. Por último puede notarse que el rasgo de "probar en ambiente ágiles" es un concepto que no tuvo una mejora, sino al contrario, desciende en un 9.53% puntos. Esto indica que es necesario desarrollar una mejor estrategia al enseñar el concepto de "probar primero," comparado al tradicional esquema "codificar-probar". Esto es una buena oportunidad para realizar el paso número 6 del proceso de CAAM, en el cual se indica "Hacer reuniones con los equipos de trabajo para "hacer suyos" los estilos/actividades, ajustar los roles de las personas de acuerdo a lo aprendido en los cursos tomados" y poder determinar cuales son los conceptos no adquiridos.

Conclusión

En el desarrollo de la capacitación se generó la experiencia en la elaboración de competencias en ambientes de gestión ágil para guiar el éxito de una planeación, una mejor comunicación con el cliente, el uso de herramientas ágiles y desarrollo de pruebas ágiles. Conducir a diferentes roles de desarrollo de software mediante la generación de proyectos particulares. Ayudar a iniciar las cortas iteraciones de los *Sprints*. Generar un modelo que controle los procesos de la capacitación. Se estableció que el modelo CAAM cumple de manera parcial a los intereses de capacitación ágil en empresas con modelos que cuentan o no cuentan con un control de procesos, mediante el seguimiento de cada uno de los procesos establecidos.

Bibliografía

Alfonso, M., & Botío, A. (2005). Resources for Agile Software Development in the Software Engineering Course. Alicante: Proceedings of the 18th Conference on Software Engineering Education & Training.

Barbazette, J. (2006). *Training needs assesment*. Portland: Pfeiffer.

Bologa, R., & Lupu, A. R. (2007). Accelerating the Sharing of Knowledge in Order To Speed Up the Process of Enlarging Software Development Teams - a Practical Example. *Proceedings of the 6th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases.*, 90-95.

Buckley R. Caple J. (2009). The theory & practice of training. En R. B. Caple Jim, *The theory & practice of training* (págs. 24-33). British: Kogan Page.

Bushnell, D. (1990). Input, process, output: A model for evaluating. *Training and Development Journal*, 41-43.

Chuan-Hoo, T., & Week-Kek, T. (2010). Designing an Information Systems Development Course to Incorporate Agility, Flexibility, and Adaptability. *Communications of the Association for Information Systems*, 172-192.

Cuellar C. (2010). Valores y Capacitación. *Software gurú*, 12-13.

Dawson, R. (2000). Twenty dirty tricks to train software engineers. *Software Engineering, 2000. Proceedings of the 2000 International Conference*, 209 - 218 .

Dingsøyr, T., Nerur, S., Balijepally, V., & Brede, N. (2012). A decade of agile methodologies: Towards explaining agile software development. *The Journal of Systems and Software*, 1213-1221.

Garg, K., & Varma, V. (2008). Software Engineering Education in India: Issues and Challenges. . *Software Engineering Education and Training, 2008. CSEET '08. IEEE 21st Conference*, 110-117.

Goldstein, I. L. (1993). *Training in organizations: Needs Assesment, Development, and Evaluation*. Pacific Grove, CA: Brooks/Cole.

Hernández, S., Fernández, C., & Baptista, P. (2007). *Metodología de la Investigación*. México: McGraw-Hill.

Huang P., Chen M. & Chen S. (2008). The practice training in the Software Engineering Education. *The 9th International Conference for Young Computer Scientists*, 2636-2640.

Jaap, C., Heijstek, A., & Linders, B. (2008). *CMMI Roadmaps*. Carnegie Mellon University.

Kirkpatrick D. (2009). *Evaluating Training Programs: The four levels*. San Francisco, California: Berrett-Koehler.

Li, P., Liu, Z., & Guo, X. (2008). Research on the Diversification Training Mode of Software Talents Based on University-Enterprise Cooperation. *International Conference for Young Computer Scientists*, 2417-2421.

Mejía, M., Ania, I., & Gamboa, R. (31 de 12 de 2006). *Diagnóstico de la Industria de Servicio de Software en México*. Recuperado el 17 de 04 de 2011, de AIS Electronic Library (AISeL): <http://aisel.aisnet.org/amcis2006/497/>

Prochazka, J., Kokott, M., Chmelar, M., & Krchnak, J. (2011). Keeping the spin – from idea to cash in 6 weeks. *International Conference on Global Software Engineering*, 124-130.

Reichlmayr, T. (2003). The Agile Approach in an Undergraduate Software Engineering Course Project. *Proceedings of the 33rd ASEE/IEEE Frontiers in Education Conference*, 5-8.

Rogers, A. W., Campbell, H. R., & Pak, R. (2001). A systems approach for training older adults to use technology. En N. Charness, D. C. Parks, & B. A. Sabel, *Communication Technology and Aging: Oportunities and challenges for the future* (págs. 187-202). New York: Springer Publishing.

Stephenson, W. (1953). *The study of behavior, Q technique and its methodology*. Chicago: University of Chicago.

Sterns, H. L., & Doverspike, D. (1989). Aging and the training and learning process in organization. En Golstein, *Training and development in work organizations: Frontiers of industrial and organizational psychology* (págs. 299-332). San Francisco, CA: Jossey Bass.

Steve D. (2010). Learning, Training and Development in Organizations. En *Learning, Training and Development in Organizations* (págs. 67-95). New York: Routledge.

Wingreen, S., & LeRouge, C. (2009). Structuring Training for IT Professionals and the Firm: An Application of the Q-Methodology. *IJGMS*, 53-67.