

ELECCIÓN DE ENFOQUES METODOLÓGICOS PARA EL DESARROLLO DE SOFTWARE EN CURSOS UNIVERSITARIOS

Ramón Ventura Roque Hernández rvHernandez@uat.edu.mx
 Adán López Mendoza aLopez@uat.edu.mx
 Carlos Manuel Juárez Ibarra cJuarez@uat.edu.mx

Universidad Autónoma de Tamaulipas
 Facultad de Comercio, Administración y
 Ciencias Sociales de Nuevo Laredo



Facultad de Comercio
 Administración y
 Ciencias Sociales
 Nuevo Laredo
 Universidad Autónoma de Tamaulipas

TEMÁTICA DE PARTICIPACIÓN: PRÁCTICAS EDUCATIVAS INNOVADORAS

Resumen

En este trabajo se presenta una propuesta para la aplicación y selección de metodologías para el desarrollo de software en los cursos universitarios. También se muestran experiencias de la aplicación práctica de esta propuesta en la evaluación de las metodologías ágiles denominadas Programación Extrema (XP) y Scrum. Se observó que ambas contribuyeron a la obtención exitosa de software funcional. Los resultados, además de demostrar en la práctica la teoría existente, muestran perspectivas importantes que constituyen un antecedente valioso para la enseñanza del desarrollo ágil de aplicaciones móviles en la universidad.

Palabras clave

- Educación superior.
- Desarrollo de software.
- Enfoques metodológicos.
- Aplicaciones móviles.
- Programación.

Objetivos

- Desarrollar una propuesta pedagógica para la selección y aplicación de enfoques metodológicos para el desarrollo de aplicaciones móviles en los cursos universitarios.
- Presentar resultados preliminares de la aplicación del modelo propuesto.

Fundamentación pedagógica

Las materias de programación son percibidas como difíciles y suelen tener importantes índices de reprobación (Roque, Salinas, López, Mota, y Flores, 2014), (Zhang, Zhang, Stafford, y Zhang, 2013).

Elaborar software de buena calidad y con buenas prácticas en el entorno universitario debe ser una prioridad, ya que es aquí donde se gestan muchas soluciones plenamente funcionales para Pequeñas y Medianas Empresas (PYMES) y otros organismos. El uso de un enfoque metodológico para desarrollar software ayuda a mantener el orden en las actividades y a disminuir la probabilidad de fracaso del proyecto. Sin embargo, existen muchos enfoques que podrían adoptarse; la elección de uno u otro, es un tema que ha sido abordado anteriormente pero que todavía se plantea como un problema de investigación.

Kropp y Meier (2013) destacan la enseñanza de los enfoques y prácticas ágiles en la universidad debido a su importancia y utilidad en el mundo empresarial mientras que Schroeder, Klarl y Kroib (2012) ponen de manifiesto la importancia del lado humano del desarrollo de software y presentan los cursos prácticos en laboratorio como medio para enseñar las prácticas ágiles en las escuelas.

Se debe tener presente que en el entorno universitario se prepara a los desarrolladores de software profesionales del futuro con conocimientos, habilidades y valores que les apoyarán en la creación de software de calidad una vez que hayan graduado y se desempeñen en su campo laboral.

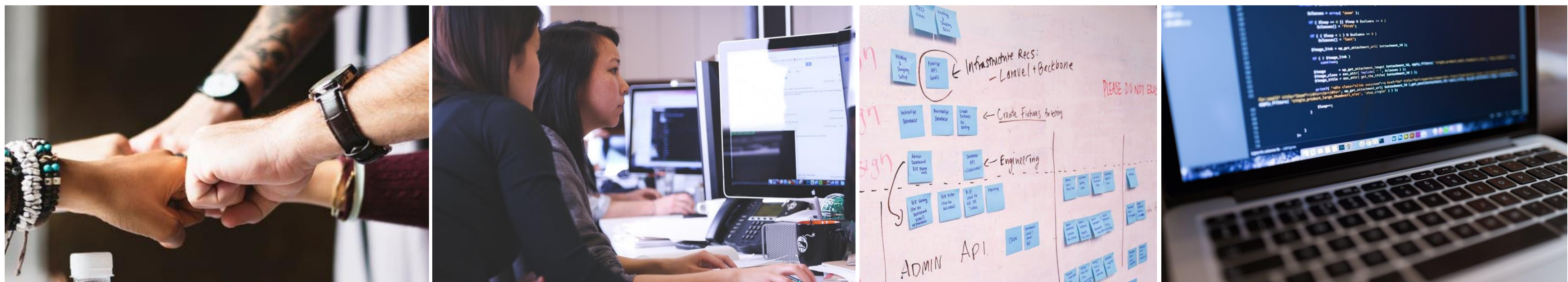
Propuesta pedagógica



Resultados

Se seleccionaron dos enfoques metodológicos: Programación Extrema y Scrum. Ambos se pusieron en práctica en el desarrollo de una aplicación móvil en un curso universitario. Se recabaron las percepciones de los estudiantes a través de cuestionarios y entrevistas no estructuradas y se analizaron las experiencias obtenidas. El análisis permitió la elaboración de la siguiente tabla de ventajas y problemas que reportaron los estudiantes.

Programación Extrema (XP)		Scrum	
Ventajas	Problemas	Ventajas	Problemas
El software se terminó rápidamente y con calidad.	La mala organización inicial del equipo.	Todo el equipo participó.	La toma de decisiones fue lenta.
La detección y prevención de errores fue eficaz.	Al principio, los integrantes no se conocían entre sí y eso dificultó la interacción.	Las diferentes habilidades de cada uno permitieron sacar adelante muchos inconvenientes.	Algunos integrantes no tuvieron suficiente paciencia en los momentos decisivos.
Hubo mucha comunicación y convivencia entre los integrantes.	Hubo problemas en el manejo de las herramientas utilizadas.	Se avanzó rápidamente en las actividades del proyecto.	Hubo problemas técnicos con el programa y el entorno de desarrollo.
Los requerimientos se entendieron fácilmente.	Los integrantes tenían diferentes enfoques para el diseño y solución de problemas.	El trabajo se dividió estratégicamente.	Los integrantes socializaron poco.
Se unificaron diferentes perspectivas para el desarrollo en equipo.	Se presentaron problemas en la integración de código.	El equipo decidió cómo organizarse.	En ocasiones, las diferentes formas de pensar impedían llegar a acuerdos.



Conclusiones

- Se presentó una estrategia para analizar y elegir las maneras más apropiadas para desarrollar software en los cursos universitarios.
- Se mostraron resultados preliminares de la aplicación de la estrategia propuesta utilizando XP y Scrum. Ambos enfoques produjeron a tiempo aplicaciones móviles funcionales y de calidad.
- Las prácticas de XP favorecen la interacción profesional y la convivencia personal, aspectos que habría que mejorar en Scrum.
- Los factores técnicos, humanos y de gestión son importantes en los desarrollos de software realizados en las aulas universitarias.

Referencias

- Kropp, M., y Meier, A. (2013). Teaching Agile Software Development at University Level. *IMVS Fokus Report*, 15-20.
- Roque Hernández, R. V., Salinas Escandón, J. M., López Mendoza, A., Mota Martínez, S., y Flores Rosales, O. (2014). Análisis de proyectos de Programación Orientada a Objetos utilizando métricas de software como medio para determinar necesidades didácticas. *Investigación y Ciencia*, 61, 20-26.
- Schroeder, A., Klarl, A., Mayer, P., y Kroib, C. (2012). Teaching agile software development through lab courses. *Global Engineering Education Conference (EDUCON) 2012* (págs. 1177-1186). Marrakech: IEEE.
- Zhang, X., Zhang, C., Stafford, T. F., y Zhang, P. (2013). Teaching introductory programming to IS students: the impact of teaching approaches on learning performance. *Journal of information systems education*, 24(2), 147-155.