

Procesos Ágiles para el Desarrollo de Aplicaciones Web

Paloma Cáceres, Esperanza Marcos
Grupo Kybele
Departamento de Ciencias Experimentales e Ingeniería
Universidad Rey Juan Carlos
C/ Tulipán, s/n, 28933 – Móstoles, Madrid (España)
{p.caceres/cuca}@escet.urjc.es

Resumen. Internet se está convirtiendo en un importante medio de comunicación y las aplicaciones Web han aparecido como instrumentos imprescindibles para la divulgación de información, así como para la dotación de servicios a los usuarios de la red. Por este motivo se ha acrecentado la necesidad de publicar información en la Web y, además, siempre es necesario que dicha información esté disponible para “ayer”. Uno de los problemas con los que nos encontramos en el desarrollo de Sistemas de Información Web (SIW), es que aún no existe una metodología universalmente aceptada que permita guiar al desarrollador en el proceso de desarrollo. Por otra parte, las metodologías tradicionales no son siempre válidas para este tipo de desarrollos debido, no sólo a que no proporcionan todas las técnicas y notaciones requeridos para modelar un SIW, sino también, y quizá como uno de los mayores inconvenientes, debido a la burocracia y rigidez de las mismas. Las metodologías clásicas de Ingeniería del Software son, en general, “pesadas”, y no facilitan el desarrollo rápido de aplicaciones. Por esta razón, se hace necesario una metodología que ayude al diseñador en el proceso de desarrollo de aplicaciones Web. Y la tendencia actual en el proceso de desarrollo, apuesta por metodologías ágiles. Lo que aún se hace más necesario para el desarrollo de aplicaciones Web.

Palabras clave: Proceso Web, metodologías de desarrollo Web, metodologías ágiles.

1. Introducción

La aparición de aplicaciones y sitios Web proporciona la explotación de otros mercados y servicios antes impensables como el comercio electrónico, la enseñanza virtual, etc., y esto conlleva un importante crecimiento en el desarrollo del software sobre dicha tecnología. Ahora bien, desde el punto de vista de la ingeniería del software es importante dotar de los mecanismos adecuados, para que la realización de este tipo de aplicaciones satisfaga las necesidades tanto de los usuarios como de los clientes que contratan el desarrollo de este tipo de aplicaciones. Pero actualmente no existe una metodología universalmente aceptada, que guíe en el proceso de desarrollo de aplicaciones Web.

En cualquier caso, existen criterios universalmente aceptados acerca del desarrollo software. Por ejemplo, y según afirma Jacobson et al. (2000), el modelo de proceso más adecuado para el desarrollo de software es un proceso iterativo e incremental, puesto que a diferencia de otros modelos de proceso, como por ejemplo el modelo en cascada, permite la obtención de diversas versiones del producto software antes de la entrega final del mismo y la depuración y validación progresiva del mismo, lo que sin duda redundará en un software más satisfactorio para usuarios y cliente. Además y según indica Conallen (2000), con este tipo de proceso es posible añadir o modificar requisitos que no han sido detectados con anterioridad. Aún no existe ninguna propuesta universalmente aceptada para el desarrollo Web, pero Fraternali (2000) indica que una posible solución al desarrollo adecuado de aplicaciones Web, sería combinar los ciclos de vida tradicionales con las propuestas de diseño para el desarrollo de aplicaciones hipermedia. De hecho, algunos de los trabajos existentes, relacionados con la tecnología hipermedia y Web, combinan el tratamiento de esas características especiales, con el uso de un modelo de proceso iterativo e incremental (Atzeni et al., 1998; Fraternali, 1998; Isakowitz et al., 1995; Schawbe and Rossi, 1995; Lowe and Hall, 1999). En cualquier caso los métodos clásicos no son adecuados para el desarrollo de aplicaciones Web, puesto que no contemplan determinadas características específicas de este tipo de aplicaciones, Lowe and Hall (1999). Por otra parte, las metodologías tradicionales generalmente imponen un proceso de desarrollo demasiado pesado y burocrático según afirma Fowler (2001), lo que impide un desarrollo ágil y rápido para este tipo de aplicaciones.

Como reacción a estas metodologías clásicas, recientemente ha aparecido un nuevo paradigma de ciclo de vida del software. Son las metodologías y procesos de desarrollo denominados ágiles, que garantizan un proceso de desarrollo suficiente pero no excesivo.

El objetivo de este artículo es la justificación de este tipo de procesos para el desarrollo de aplicaciones Web.

El resto del artículo se estructura de la siguiente forma: en la sección 2 se comentan las diferencias entre las metodologías tradicionales y las metodologías ágiles; en la sección 3 se justifica la necesidad de un proceso ágil para el desarrollo de aplicaciones Web; por último en la sección 4 se comentan las conclusiones.

2. Metodologías tradicionales versus Metodologías Ágiles

Las metodologías tradicionales imponen una disciplina de trabajo sobre el proceso de desarrollo del software, con el objetivo de conseguir un software más eficiente y predecible. Para ello, se hace un especial hincapié en la planificación total de todo el trabajo a realizar y una vez que esta todo detallado, comienza el ciclo de desarrollo del producto software. Este planteamiento está basado en el resto de disciplinas de ingeniería, a pesar de que el software no pueda considerarse como la construcción de una obra clásica de ingeniería.

Con estas metodologías se lleva trabajando desde hace tiempo y no ha habido en ningún caso ninguna experiencia traumática acerca de su uso. Pero aún así, han recibido diversas críticas, y la más común hace referencia a su carácter excesivamente burocrático, y como afirma Fowler (2001), este hecho ha llevado a identificarlas como *metodologías monumentales*. Además, las metodologías tradicionales no se adaptan adecuadamente a los cambios, por lo que no son métodos adecuados cuando se trabaja en un entorno, donde los requisitos no pueden predecirse o bien pueden variar.

En contraposición a estas metodologías clásicas, en los últimos años ha aparecido un nuevo grupo de metodologías, que se identifica como *metodologías ágiles*. Aportan como novedad, nuevos métodos de trabajo que apuestan por una cantidad apropiada de proceso. Es decir, ni se pierden en *una excesiva cantidad de cuestiones burocráticas* ni defienden tampoco *la falta total de proceso*. Buscan el equilibrio en la relación proceso/esfuerzo.

Las diferencias existentes entre ambos grupos de metodologías surgen por un enfoque y objetivos diferentes. Y como principales diferencias, Fowler (2001) identifica las siguientes:

- Las metodologías ágiles son adaptativas mas que predictivas. Las metodologías tradicionales potencian la planificación detallada de prácticamente todo el desarrollo software a largo plazo. Pero cuando se produce un cambio, toda esta planificación puede venirse abajo. Sin embargo, las metodologías ágiles proponen procesos que se adaptan y progresan con el cambio, llegando incluso hasta el punto de cambiar ellos mismos.
- Las metodologías ágiles están orientadas al personal mas que orientadas al proceso. Intentan trabajar con la naturaleza del personal asignado al desarrollo, mas que contra ellos, de tal forma que permiten que la actividad de desarrollo software se convierta en una actividad grata e interesante.

Existen diversas metodologías que coinciden en llamarse metodologías ágiles. Y aunque entre ellas comparten muchas características tienen también diferencias significativas. A continuación se presentan algunas de las metodologías ágiles más representativas.

- Extreme Programming (XP) - La programación extrema (Beck, 1999; Mc Breen, 2000) concede una gran importancia a las pruebas del software (testing). Aunque la mayoría de los procesos las tienen en cuenta, generalmente lo contemplan de una forma demasiado ligera y superficial. Sin embargo, XP lo toma como base para el desarrollo y cada programador que escribe código también escribe los casos de prueba. Estos forman parte del proceso continuo de generación de código y se integra continuamente con ello, lo que garantiza una plataforma estable para el futuro desarrollo. Sobre dicha plataforma se genera un proceso de diseño evolutivo, que es la base del sistema y que se enriquece con cada iteración. Nunca se generan diseños futuros. Según afirma Fowler (2001), el resultado es un proceso de diseño que combina adecuadamente la disciplina con la adaptabilidad.
- Open source - Open source apuesta por la distribución de trabajo entre diferentes equipos, al igual que ocurre con la mayoría de los procesos adaptativos. La mayoría de los proyectos open source cuentan con supervisores de código. Estos supervisores de código, son las únicas personas autorizadas para realizar un cambio en el repositorio del código fuente. Por otra parte, el resto del personal puede realizar cualquier cambio en el código base. Sin embargo, el supervisor del código es la persona responsable de coordinar y de mantener la consistencia del diseño del software. Una de las principales ventajas de los desarrollos open source es que la depuración es altamente paralelizable, aunque un gran número de personas puedan verse involucradas. Cuando se soluciona un error, se envía la

solución al supervisor de código, lo que garantiza que alguien realiza la modificación de forma fiable mientras otra parte del personal se dedica a las tareas de depuración.

3. Desarrollo de aplicaciones Web con procesos ágiles

Desgraciadamente en la actualidad, debido generalmente al exceso de burocracia de las metodologías tradicionales se ha pasado, en la mayoría de los casos, a no utilizar ningún método de trabajo específico y a trabajar “a destajo” con el único y erróneo objetivo de ahorrar tiempo y dinero. Aplicar cierto grado de disciplina ayudará en el proceso de desarrollo y siempre es mejor utilizar un proceso ágil que ninguna otra cosa. Sin embargo, un proceso ágil es adecuado para el desarrollo de cierto tipo de aplicaciones, por lo que no se deberían utilizar estos métodos para cualquier tipo de desarrollo software, Fowler (2001).

Los procesos ágiles son una buena elección cuando trabajamos con requisitos desconocidos o variables. Si no existen requisitos estables, no existe una gran posibilidad de tener un diseño estable y de seguir un proceso totalmente planificado, que no vaya a variar ni en tiempo ni en dinero. En estas situaciones, un proceso adaptativo será mucho más efectivo que un proceso predictivo. Por otra parte, los procesos de desarrollo adaptativos también facilitan la generación rápida de prototipos y de versiones previas a la entrega final, lo cual agradará al cliente. Pero la mayor barrera que habrá que salvar será convencer al cliente de que no existen una planificación y una forma fija de hacer las cosas. En cualquier caso, lo que se garantiza es un menor riesgo ante la posibilidad de cambios en los requisitos. Porque los cambios existen, y los procesos adaptativos permitirán estos cambios lo que en definitiva, garantizará que el producto final sea el deseado por el cliente. Según afirma Booch (2001), todas estas razones son las que hacen que los procesos indicados por las comunidades de Extreme Programming y de Open Source hayan suscitado tanto interés.

En general, las aplicaciones Web cumplen la mayor parte de las características mencionadas en el párrafo anterior, por lo que la utilización de procesos ágiles podría ser beneficioso para este tipo de desarrollos. La necesidad del cliente que contrata un desarrollo Web es que su producto esté disponible en la red lo más pronto posible. Si no se contempla esta necesidad, la aplicación no resultará un producto satisfactorio para el cliente. Puesto que los procesos ágiles permiten obtener versiones de producto previas a la versión final, si se aplican adecuadamente estos procesos el cliente podrá disponer de forma rápida de alguna versión intermedia. Además el ciclo de desarrollo de la mayoría de los sitios y aplicaciones Web es extremadamente corto, Overmyer (2000). Esto implica que generalmente no se aplique ningún tipo de proceso, pero sin duda y como se mencionaba anteriormente más vale un proceso ágil que nada. Por otra parte, los desarrollos Web se perciben como desarrollos sencillos y los desarrolladores son sometidos a una gran presión de trabajo para terminar lo más pronto posible. Esta forma de trabajar va a implicar sin duda alguna modificaciones. Luego sería conveniente garantizar un proceso de desarrollo adaptable a los cambios. Otra cuestión fundamental a tener en cuenta es que las aplicaciones Web se desarrollan sin conocer los perfiles de los usuarios finales de las mismas, o lo que es lo mismo sin conocer los requisitos de usuario del sistema. Sin lugar a dudas esto implicará cambios en los requisitos inicialmente detectados, lo que nos lleva de nuevo a la elección de un proceso adaptativo. Por lo tanto, podríamos concluir que este tipo de procesos son especialmente aplicables al desarrollo de aplicaciones para la Web.

En este sentido, estamos trabajando en la definición de una metodología para el desarrollo de aplicaciones Web (MIDAS¹) cuyo modelo de proceso sea iterativo, incremental y adaptativo. Este modelo ha sido definido en base a las características de aplicaciones Web (Cáceres y Marcos, 2000) y de las nuevas tendencias sobre procesos ágiles (Fowler, 2000). Se ha propuesto un proceso iterativo para garantizar la realimentación de información y de requisitos una vez iniciado el desarrollo, así como la validación continua del sistema de información Web. Esto permite que cada iteración contemple ciclos de desarrollo completos y cortos, y obtener así rápidamente, una nueva versión con mejoras sobre las versiones anteriores. Se ha propuesto un proceso incremental con la finalidad de obtener el sistema final tras la realización de diferentes etapas. Cada etapa está ligada a la realización de un determinado tipo de tareas. El final de cada etapa proporciona además, una versión estable del software, que gracias a

¹ Este trabajo se está llevando a cabo dentro del proyecto MIDAS, parcialmente financiado por la CICYT y la CE (referencia 2FD97-2163).

esas tareas concretas, cuenta con unas características específicas. Esto permite entregas al cliente de forma rápida y ágil. El proceso es además adaptativo lo que permite que pueda ser aplicado a diferentes aplicaciones Web. De esta forma, cada tipo de aplicación conllevará la ejecución de un determinado conjunto de tareas y pasará por unas determinadas etapas .

4. Conclusiones y líneas futuras

Desde hace algunos años, diferentes autores vienen indicando que las metodologías tradicionales no son totalmente adecuadas para todos los desarrollos software. Las razones son diversas, pero quizá las principales son la falta de flexibilidad de su proceso de desarrollo y la necesidad de generar una cantidad excesiva de documentación.

Los clientes que contratan un desarrollo software para la Web, plantean una serie de necesidades que no se satisfacen con este tipo de metodologías. Recientemente han aparecido las metodologías denominadas ágiles, que adaptan el proceso de desarrollo al desarrollo software específico y que permiten fácilmente incorporar y contemplar cambios en los requisitos. Es por ello que parece adecuado, el uso de este tipo de procesos para el desarrollo Web.

En resumen, parece que el proceso adecuado para este tipo de desarrollo software sería un proceso ágil además de iterativo e incremental; es decir, un proceso que se adapte al desarrollo de los diferentes tipos de aplicaciones Web (BD Web, aplicaciones de comercio electrónico, etc.), permitiendo la incorporación de nuevos requisitos y sin exigir una excesiva generación de documentación.

MIDAS (Cáceres y Marcos, 2001) es un marco metodológico, adaptable a diferentes tipo de aplicaciones, para el que se propone un proceso de desarrollo de este tipo. Actualmente estamos validando el proceso de desarrollo aplicándolo a diferentes tipos de aplicaciones para la Web. A partir de estas experiencias, se irán incorporando mejoras al trabajo realizado.

Referencias

- Atzeni, P; Mecca, G; Merialdo, P; (1998) Design and Maintenance of Data-Intensive Web Sites. Atzeni, Mecca y Merialdo. *Advances in Database Technology*. Ed. Sheck, Saltor, Ramos, Alonso. Proceedings of the 6th. Conference on Extended Database Technology (EDBT'98). Springer Verlag, Valencia.
- Beck, K. (1999). Embracing Change with eXtreme Programming. *Computer* , vol. 32, nº 10, pp.70-77.
- Booch, G. (2001). Developing the Future. Software Solutions. *Communications of ACM*, vol. 44 (3), pp. 119-121, March 2001.
- Cáceres, P., Marcos, E (2000) El Proceso de Desarrollo en las Aplicaciones Web. *Actas de las V Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2000)*. Ed. Delgado, C., Marcos, E., Universidad de Valladolid, Valladolid, España.
- Cáceres, P., Marcos, E (2001) Las metodologías de desarrollo y la mejora de calidad de las aplicaciones Web. *IV Encuentro para la Calidad en las Tecnologías de la Información y las Comunicaciones*. Ed. Nuno Olivera, J., Brito e Abreu, F., Piattini, M., Lisboa, Portugal, pp. 3-9.
- Conallen, J. (2000). Building Web applications with UML. Addison Wesley.
- Fowler, M. (2001). The New Methodology. Retrieved May 2001 from <http://www.martinfowler.com/articles/newMethodology.html>.
- Fraternali, P; Paolini, P; (1998) A conceptual Model and a Tool Environment for Developing more Scalable, Dynamic and Customizable Web Applications. *Advances in Database Technology*. Ed. Sheck, Saltor, Ramos, Alonso. Proceedings of the 6th. Conference on Extended Database Technology (EDBT'98). Springer-Verlag, Lecture Notes in Computer Science, Valencia.
- Fraternali, P. (2000). Tools and Approaches for Developing Data-Intensive Web Applications: a Survey. Retrieved July 2000 from the World Wide Web: <http://toriisoft.com>
- Isakowitz, T., Stohr, E., Balasubramanian, P. (1995). RMM: A Methodology for Structured Hypermedia Design. *Communications ACM* , 58(8), pp. 34-43, August.
- Jacobson, I., Booch, G., Rumbaugh, J. (2000). *El proceso unificado de desarrollo del software*. Addison Wesley.
- Lowe, D.; Hall, W. (1999). *Hypermedia & the Web. An Engineering Approach*. J. Wiley and Sons.
- McBreen, P. (2000). Applying the lessons of eXtreme Programming. *The Proceedings of the 34th. International Conference on Technology of Object Oriented Languages and Systems (TOOLS34)*.
- Overmyer, S. P. (2000). What's Different about Requirements Engineering for Web Sites? *Requirements Engineering*, Vol. 5, pp. 62-65, Springer-Verlag, London.
- Schwabe, D., Rossi, G. (1995). The Object Oriented Hypermedia Design Model. *Communications of the ACM* , 38(8), 45-46.