

El proceso de desarrollo del programa de mejora UAMISoft

Humberto Cervantes Maceda, Alfonso Martínez Martínez

Universidad Autónoma Metropolitana – Iztapalapa

San Rafael Atlixco N° 186, Col. Vicentina, C.P. 09340 Del. Iztapalapa. Distrito Federal, México

{hcm,almm@xanum.uam.mx}

Resumen

Este artículo describe parte del trabajo que se está realizando actualmente en la Universidad Autónoma Metropolitana Iztapalapa (UAM-I) y que está enfocado hacia la implantación de un programa de mejora, llamado UAMISoft, que facilite la realización de proyectos de software de calidad y que permita a los alumnos participar en un proyecto de desarrollo siguiendo prácticas de ingeniería de software. Este artículo describe el proceso de desarrollo asociado a UAMISoft que está adaptado al entorno de universidad donde la disponibilidad en tiempo de los alumnos y los recursos son limitados. Como soporte del proceso se hace uso intensivo de herramientas de apoyo de fuente libre.

Summary

This article describes ongoing work at the Universidad Autónoma Metropolitana Iztapalapa (UAM-I) which is oriented towards establishing an improvement process, called UAMISoft, whose goal is to produce quality software and give students a hands-on experience inside a software project. This article focuses on the software process associated to UAMISoft. This process, which is adapted to the UAM-I's context, where time and resources are limited, relies heavily on the use of open source tools.

1. Introducción

UAMISoft es un proyecto de reciente creación dentro de la UAM-I y que tiene dos objetivos principales. El primero es de preparar profesionistas capacitados en el desarrollo de software y el segundo es de desarrollar aplicaciones de calidad que cubran, inicialmente, requerimientos internos de la UAM-I. La preparación de gente capacitada requiere que los estudiantes participen dentro de la realización de un desarrollo de software siguiendo prácticas de ingeniería de software.

Para lograr estos objetivos, el proyecto busca realizar la implantación de un programa de mejora, basado en el Modelo de Procesos para la Industria Mexicana de

Software (Moprosoft)[1], que permita generar software de calidad en proyectos de investigación y docencia en donde participen alumnos de licenciatura y posgrado. Moprosoft define un conjunto de procesos dentro de una organización con el objetivo de que esta ofrezca servicios de calidad y sea competitiva. Uno de esos procesos es un proceso de desarrollo de software que es un pilar dentro de la organización.

La etapa inicial de la implantación del proceso de mejora involucra la puesta en pie de un proceso de desarrollo que se adecue al contexto de la UAM-I. Este contexto es particular pues se deben tomar en cuenta aspectos como limitaciones en cuestiones de tiempo, la dificultad de implantar un proceso en un ambiente universitario, la dificultad de instalación de las herramientas de soporte y la falta de capacitación.

El presente trabajo describe los aspectos fundamentales del proceso de desarrollo de software asociado al proyecto UAMISoft. Estos aspectos incluyen la adaptación del Proceso Unificado Rational (RUP) [2] y el establecimiento de una serie de prácticas de desarrollo y de un entorno de soporte basado en herramientas de fuente libre.

El resto de este trabajo se estructura de la manera siguiente: la sección 2 da un estado del arte en el cuál se presentan varios procesos de desarrollo. La sección 3 presenta la metodología, que por un lado incluye la descripción del contexto propio a la UAMI y por el otro el proceso y las herramientas que lo soportan. La sección 4 describe resultados actuales y finalmente la sección 5 presenta conclusiones y perspectivas.

2. Estado del arte

En la actualidad existen un gran número de procesos de desarrollo de software. Los procesos de los cuáles el proceso de desarrollo de UAMISoft retoma conceptos están descritos a continuación.

2.1 Proceso Unificado Rational (RUP)

El Proceso Unificado de Rational [2] no es en sí un proceso de desarrollo particular, sino que es un marco de procesos que da los lineamientos para configurar un proceso iterativo que puede ir desde lo ligero (poca

ceremonia) hasta lo pesado (burocráticos). RUP es promovido por actores industriales y está basado en varias prácticas reconocidas hoy en día como “mejores prácticas” de desarrollo. Estas incluyen: desarrollo iterativo (que incluye la creación de versiones funcionales de la aplicación a lo largo del proceso), manejo de requerimientos cambiantes, arquitectura basada en componentes, modelado visual, verificación continua de calidad y control de cambios a lo largo del proceso. RUP divide el desarrollo en cuatro etapas principales: *iniciación* (comprensión de requerimientos y análisis de riesgos de factibilidad), *elaboración* (de una arquitectura ejecutable y manejo de riesgos técnicos), *construcción* (de una versión beta) y *transición* (del sistema al cliente). Esas cuatro etapas se realizan en varias iteraciones cuya cantidad depende del proyecto. RUP ha sido usado con éxito en el desarrollo de proyectos de gran envergadura y es uno de los procesos más populares hoy en día.

2.2 PSP y TSP

PSP y TSP son encarnaciones del Capability Maturity Model (CMM) [3] adaptadas al contexto de un desarrollador individual y de un equipo pequeño de desarrollo respectivamente. Una de las ideas fundamentales de CMM es de buscar la mejora de un proceso de desarrollo bien establecido a través de la recolección de métricas. Hoy en día, CMM se ha vuelto prácticamente un estándar para medir el nivel de madurez de una organización de desarrollo.

2.3 Metodologías ágiles

Las metodologías ágiles de desarrollo hicieron su aparición de manera relativamente reciente y fueron propuestas por un grupo de personas que buscaban limitar la burocracia de los proyectos. Las metodologías ágiles comparten principios como:

- Favorecer la entrega de software funcional sobre la entrega de documentos.
- Dar más importancia a los desarrolladores (no verlos como piezas reemplazables)
- Adaptarse continuamente al cambio, usando iteraciones muy cortas.

Una de las metodologías ágiles de desarrollo más populares hoy en día es la programación extrema (XP) [4]. Dos de las características de XP son la realización de pruebas unitarias antes de la escritura del código y la programación por parejas.

3. Metodología

Esta sección describe primeramente el contexto de la UAM-I que justifica las decisiones tomadas en cuanto al proceso y posteriormente describe el proceso de desarrollo y las herramientas de soporte asociadas.

3.1 El contexto de la UAM-I

La Universidad Autónoma Metropolitana se caracteriza por tener una organización de cursos trimestral, lo cual limita los alcances que se pueden lograr en cuanto a la realización de un proyecto dentro de un curso debido al poco tiempo del que se dispone. Sin embargo, durante la formación de un alumno de nivel licenciatura, existen dos etapas en las cuáles el alumno puede dedicarse más de un trimestre a desarrollar un proyecto: el servicio social y el proyecto de investigación. Estas dos etapas duran 6 meses cada una y la primera se caracteriza por requerir una dedicación de 20 horas a la semana mientras que la segunda requiere cerca de la mitad de horas, pero privilegia los aspectos de investigación. El proceso UAMISoft está pensado para alumnos de licenciatura o de posgrado, estos últimos disponen de más tiempo (3 trimestres) para el desarrollo de un proyecto. Una dificultad adicional a tomar en cuenta es que al entrar en un proyecto de desarrollo, los alumnos (sobre todo en licenciatura) tienden a tener una formación limitada en cuanto a procesos y tecnologías de desarrollo y su nivel de inglés no es muy elevado. Además, requieren de capacitación de acuerdo al contexto del proyecto, lo cual limita el tiempo efectivo dedicado al desarrollo (hasta en un 50% en el caso de proyectos terminales). Por último, dentro de la UAM-I se dispone de recursos limitados para adquirir herramientas de soporte.

3.2 El proceso de desarrollo dentro del proyecto UAMISoft

El proceso de desarrollo de software de UAMISoft está pensado para ser flexible y para adaptarse al contexto particular de la UAM-I. Por ello, este proceso se ha construido como una instancia ligera del RUP, que involucra poca ceremonia. Sin embargo, este proceso busca incorporar algunos aspectos de las metodologías ágiles, pues favorece el desarrollo continuo de prototipos y la realización de pruebas, y del PSP/TSP en aspectos estratégicos y en el mejoramiento continuo del proceso.

El proceso de desarrollo de UAMISoft define un conjunto de prácticas que permiten que un sistema sea construido gradualmente por distintos alumnos que disponen de tiempo limitado:

Arquitectura Modular: La arquitectura de un sistema está construida siguiendo el principio de la alta cohesión y bajo acoplamiento. Un módulo solamente se comunica con otros a través de interfases bien definidas. Por otro lado se utiliza el patrón inversión de control donde los distintos módulos son instanciados por una entidad principal que los conecta, lo cual permite realizar un desacoplamiento total.

Desarrollo en paralelo: Cada módulo es visto como un proyecto individual y puede ser compilado y distribuido independientemente. Por otro lado, cada

módulo es la responsabilidad de un estudiante y, dependiendo del tiempo de dedicación, un estudiante es responsable de uno o más módulos.

Reutilización de componentes: Antes de proceder a la realización de un módulo “desde cero” se hace una búsqueda en internet para tratar de localizar algún componente reutilizable (preferentemente de fuente abierta) que pueda simplificar el desarrollo del módulo. Esto permite, entre otras cosas, reducir el tiempo de desarrollo.

Pruebas e integración continua y automática: Dentro de un proyecto, las pruebas unitarias se realizan de manera continua y automática. La integración de los distintos módulos también se realiza de manera continua y automática. La integración depende de la realización exitosa de las pruebas unitarias lo que garantiza que la integración se hace con módulos sin defectos. Los resultados de pruebas unitarias son, además, útiles para realizar mediciones de calidad.

Control de versiones: El código de la aplicación se almacena en un repositorio que permite llevar el control de versiones. El repositorio está organizado de tal manera que, además de contener el código de la versión en desarrollo, contiene una serie de “fotografías” de las distintas versiones ejecutables que se van realizando. Esto permite poder mostrar en todo momento la última versión 'estable' del desarrollo.

Documentación continua: Aunque el proceso pone énfasis en la producción de software funcional, también le da importancia a la generación de documentación de forma continua y colaborativa. Los alumnos son motivados a editar continuamente la documentación del proyecto, lo cual permite además generar en forma rápida su informe final o su tesis.

3.3 Herramientas de soporte

Las distintas prácticas que propone el proceso de desarrollo UAMISoft requieren de una cierta disciplina para ser llevadas a cabo en su totalidad. Para garantizar que estas prácticas se realizan de forma efectiva, el enfoque que toma UAMISoft es de simplificar al máximo su realización para los desarrolladores. Esta simplificación se logra a través del uso de herramientas de soporte. Estas herramientas se dividen en dos áreas: el control de proyecto y documentación (DotProject y DokuWiki) y las herramientas asociadas al desarrollo (en este caso en Java).

A continuación se describen las distintas herramientas de soporte que se usan en el proyecto. Hay que subrayar uno de los criterios de selección de las herramientas es que sean de fuente libre, o que tengan licencias académicas, lo cual permite reducir enormemente los costos de implantación.

DotProject: La herramienta DotProject (<http://www.dotproject.net>) permite realizar todas las actividades asociadas al manejo de proyecto desde un navegador web. Este enfoque de cero instalación hace que su uso sea muy simple. Esta herramienta permite llevar un control sobre la calendarización (usando diagramas de Gantt), pero además provee otras facilidades como foros de discusión y un sistema de “tickets”, que es en sí una herramienta de control de los problemas (bug-tracking) en la aplicación. En UAMISoft, todos los participantes de un proyecto tienen acceso al sitio DotProject asociado y son responsables de actualizar la calendarización que corresponde a sus actividades.

DokuWiki: La herramienta DokuWiki (<http://wiki.splitbrain.org>) es un Wiki, es decir un sitio Web en donde se pueden editar las páginas directamente desde un navegador de forma colaborativa. El Wiki permite cubrir la necesidad de realización de documentación continua de manera sencilla.

El uso de un Wiki tiene varias ventajas que incluyen:

- Facilidad de uso: sólo requiere de un navegador.
- Control de acceso: el Wiki permite llevar un control de usuarios y restringir el acceso público o las modificaciones a la documentación.
- Uniformidad en la documentación: el uso de un Wiki permite que toda la documentación del proyecto se almacene de manera uniforme y que sea fácilmente modificable.
- Control de versiones: el Wiki provee un mecanismo sencillo de control de versiones que permite revertir cambios indeseables.

Dentro de UAMISoft, cada proyecto tiene un área dedicada dentro del Wiki, en donde se almacena toda la documentación (requerimientos, documentos de apoyo, etc...) usando un templete común. Además de esto, el Wiki alberga una base de conocimiento común a todos los proyectos que contiene información sobre las distintas tecnologías que se usan en el desarrollo.

Un aspecto importante de usar un Wiki es que permite que nuevos alumnos que se integran a un proyecto puedan obtener conocimientos rápidamente sobre el proyecto y las herramientas. El hecho de que la documentación esté en español permite además colmar parcialmente las deficiencias de idioma.

Ant: La herramienta Ant (<http://ant.apache.org>) permite automatizar el proceso de construcción de un proyecto en Java. El proceso de construcción incluye tareas como la compilación, el empaquetamiento, la generación de documentación, etc... Ant funciona a partir de scripts portables escritos en formato XML que describen las reglas de construcción de un proyecto.

Dentro de los proyectos de UAMISoft, cada módulo lleva un script Ant asociado, lo cual le da el carácter de proyecto independiente. Por otro lado existe un script

principal que permite automatizar el proceso de integración de la aplicación, y que facilita la integración continua.

Subversion: La herramienta Subversion (<http://subversion.tigris.org/>) es un sistema de control de versiones que mejora CVS. Este sistema permite realizar principalmente el control de versiones del código, ya que el de documentos es llevado por el Wiki.

JUnit: La herramienta JUnit (<http://www.junit.org>) es un marco de trabajo que permite crear pruebas unitarias para código en Java. Las pruebas unitarias se realizan automáticamente desde el script de Ant como parte del proceso de construcción. Por otro lado, JUnit permite generar reportes que permiten realizar medidas de defectos con objetivo de mejorar el proceso.

Eclipse: La herramienta Eclipse (<http://www.eclipse.org>) de IBM es un entorno de desarrollo integrado extensible a base de plugins. El hecho de que sea integrado significa que Eclipse es capaz de incorporar diversas herramientas que son provistas por unidades de extensión que son los plugins.

Dentro de UAMISoft, se usan plugins que permiten integrar todas las herramientas descritas anteriormente (a excepción de las que son accesibles vía navegador). La ventaja de concentrar todas las herramientas en una misma aplicación es que el desarrollador tiene todas las herramientas a la mano reunidas en una misma ventana. Actualmente se está explorando la utilización de una nueva herramienta que permite realizar modelado visual dentro del entorno de desarrollo (UML Studio de Omondo <http://www.omondo.com>); esto permitirá cubrir el proceso completo de desarrollo sin salir de Eclipse.

4. Resultados

La implantación del proceso de UAMISoft comenzó alrededor de Diciembre 2004 dentro del curso de Ingeniería de Software. Los alumnos que participaron en el curso y que tenían poco conocimiento de Java al iniciar terminaron creando una aplicación compleja que sólo fue posible gracias al trabajo obtenido al aplicar el proceso de desarrollo de software y a las herramientas de soporte descritas aquí. En ese curso se observó que las herramientas que más dudan en usar los estudiantes de manera inicial son las de documentación y de pruebas unitarias, ya que al principio les cuesta entender el beneficio que les pueden aportar y prefieren dedicarse a codificar. Esta situación cambia conforme avanza el proyecto y posteriormente todas las herramientas se usan eficazmente.

Hoy en día, el proceso de desarrollo de software descrito en este artículo se está empleando en la

realización de tres proyectos de desarrollo distintos. El primero es llevado a cabo por un equipo de 5 alumnos de licenciatura y es una aplicación para el área de Procesamiento de Señales e Imágenes Biomédicas. Esta aplicación se construye como un conjunto de plugins por encima de la plataforma Eclipse. El segundo proyecto corresponde al proyecto terminal de 3 estudiantes y es una aplicación de tipo “tres capas” para administrar registros de estudiantes en un programa experimental. Finalmente, el tercer proyecto, que está siendo realizado por un equipo de 4 alumnos de maestría, está relacionado con la implementación de un estándar de comunicación de imágenes médicas (DICOM). Estos tres proyectos se encuentran actualmente en las etapas de concepción o elaboración y el hecho de que se tenga un proceso común está permitiendo a los autores coordinar al conjunto de los alumnos de manera eficaz.

5. Conclusiones y perspectivas

Este artículo presentó el proceso de desarrollo del proyecto UAMISoft que está adaptado al contexto de la UAM-I. Las bases de este proceso fueron establecidas dentro de un curso de ingeniería de software que dio resultados exitosos y hoy en día se está aplicando en tres proyectos distintos. Estos proyectos permitirán obtener retroalimentación para mejorar el proceso de desarrollo. En el futuro, se buscara incorporar el proceso descrito en este artículo como pilar de una organización definida mediante el modelo de procesos MoproSoft. Esto permitirá llevar a cabo proyectos de manera eficaz, cubriendo la necesidad de dar experiencia a los alumnos en cuanto a la participación dentro de un proyecto de software y permitiendo desarrollar software de calidad para uso dentro o fuera de la universidad. Se espera que la experiencia de esta iniciativa pueda ofrecer elementos para su implantación en otras instituciones educativas.

6. Bibliografía

[1] Oktaba, H. et Al., "*Moprosoft: Modelo de Procesos para la Industria de Software*," Documento en línea (www.lania.mx/biblioteca/manuales/moprosoft/V%201.1%20DocumentoBase.pdf), Mayo 2003.

[2] Kruchten, P. "The Rational Unified Process: An Introduction", Addison-Wesley Professional; 3a edición, Diciembre 2003.

[3] Carnegie Mellon University, "*The Team Software Process (TSP) and the Personal Software Process (PSP)*", documentación en línea (<http://www.sei.cmu.edu/tsp/>), Junio 2005

[4] Beck, K. y Andres, C. "*Extreme Programming Explained*," Addison-Wesley Professional; 2a edición, Noviembre 2004