

Actualidad del estado de una nueva metodología de gestión y desarrollo de proyectos de software: Metodología Ágil basada en Telecomunicaciones MATE

Marisa Tumino, Juan Bournissen, Claudio Bracalenti, Erik Schlemper, Edgar Streuli y Silvio Kucharsky
I3@uap.edu.ar
Universidad Adventista del Plata
Instituto de Investigación en Informática – I3

Resumen

El presente trabajo tiene como objetivo primario proponer una metodología ágil que facilite el desarrollo de aplicaciones, para dispositivos móviles, utilizando, como plataforma de desarrollo, medios que permitan el trabajo a distancia y puedan, incluso, soportar desplazamientos de los implicados. El objetivo secundario es el desarrollo de un producto, que facilite la gestión de los proyectos, utilizando la metodología propuesta. Las metodologías ágiles más usuales, y desde donde se han adoptado los principios básicos del desarrollo de aplicaciones para dispositivos móviles, tales como SCRUM y *eXtreme Programming* (XP), sufren limitaciones condicionadas a las localizaciones físicas de los miembros del *stakeholders*¹.

En virtud de las condiciones demandadas por las nuevas modalidades de desarrollo *Freelance*, donde los miembros de los equipos se encuentran ubicados en diferentes puntos geográficos del mundo, se considera relevante elaborar una propuesta idónea que proporcione respuestas a estas limitaciones.

Palabras clave: Teletrabajo, Metodología ágil, Desarrollo semisincrónico distribuido, Virtualización de empresas informáticas, Telefonía móvil.

Contexto

El presente trabajo es coordinado por la Universidad Adventista del Plata, en el marco de los proyectos de investigación desarrollados

por el Instituto de Investigación en Informática (I3) de la mencionada institución educativa.

Introducción

Observaciones de las prácticas: Evaluación comparativa de herramientas

Un primer desafío que enfrentó esta iniciativa fue buscar herramientas que asistan a la metodología de trabajo distribuido.

Se han efectuado pruebas sobre un conjunto de productos, tanto pagos como gratuitos, que ofrecen una potencial solución al problema que aborda el trabajo.

Para compartir aplicaciones, escritorios y comunicarse, mediante video conferencia, se probaron las siguientes herramientas: *Adobe Connect*, *TeamViewer*, *HangOut* y *Saros*, tanto en forma individual como combinada, excepto con *Adobe Connect*, obteniendo los resultados que se describen a continuación.

Dentro de las dificultades encontradas durante las pruebas, cabe señalar las siguientes:

1. *Adobe Connect* es una herramienta orientada a conferencias con la que se puede compartir escritorio, escribir en pizarra, organizar usuarios y administrar la comunicación entre los participantes. Esta herramienta resulta útil para emplearse en exposiciones o eventos generales donde se presenta una ponencia con la participación de un auditorio remoto. Asimismo permite grabar las conversaciones y almacenar el título de los distintos bloques en que se puede fraccionar la ponencia. Esta herramienta funciona con un servidor centralizado de Adobe, lo que implica que cada integrante de la reunión mantiene una conexión con el servidor para funcionar direccionando todo el tráfico de red al mismo.

¹ **Stakeholder:** quienes pueden afectar o son afectados por las actividades de una empresa. (R. E. Freeman – “*Management: A Stakeholder Approach*” - Pitman, 1984)

1.1. Ancho de banda: tanto de subida como de bajada, debe mantenerse el mismo ancho, con lo que la conexión a internet debe asegurar una gran capacidad de ambos lados, y no como generalmente ofrecen los *Internet Service Provider* (ISP) con gran velocidad de bajada y poca velocidad de subida. De *download* se requiere un mínimo de 300 kbps y lo ideal sería obtener la misma velocidad de *upload* aunque con 100 kbps funciona correctamente en caso de no pretender transmisión de video.

1.2. Comunicación

1.2.1. Retardo: durante las pruebas la señal sonora del *stream* tuvo un retraso promedio de diez segundos, lo que permitió una comunicación fluida mientras hubo un único expositor unidireccional pero, dado el desfase, no fue apropiado cuando se debieron compartir ideas entre varios participantes.

1.2.2. Cortes: los cortes observados en la comunicación se debieron al insuficiente ancho de banda disponible. Por su parte los ISP no proveen un ancho de banda constante.

Estas consideraciones hacen de esta herramienta un medio no recomendable para ser utilizado en el trabajo diario del equipo de desarrollo distribuido donde los participantes requieren un flujo continuo y dinámico de comunicación bidireccional.

2. *Team Viewer* es una aplicación utilizada para mantener videoconferencias entre múltiples personas, compartiendo el escritorio del organizador de la reunión. La naturaleza de la imagen compartida puede ajustarse para máximo rendimiento o máxima calidad. En las pruebas realizadas con las videoconferencias, se pudo apreciar eventualmente leves retrasos y/o desacoples entre imagen y sonido, con tendencia a pérdidas parciales, distorsiones, reverberancias y acoples del sonido, oscilando los resultados desde un buen desempeño hasta la incapacidad de comunicarse en forma comprensible.

La herramienta funciona muy bien para uso compartido de escritorios, permitiendo a los diferentes participantes tomar el control del ordenador del expositor, aunque no simultáneamente, lo que permite el desarrollo en parejas en forma distribuida.

3. *HangOut* consiste en una funcionalidad de *google+*, la red social de Google, y actúa en

forma similar a *google talk*, diferenciándose en que *HangOut* permite realizar conferencias entre múltiples personas. Además permite compartir documentos, aplicaciones (propias de *google+* o *apps* específicas) y escritorios, aunque con una calidad inferior a la que ofrece *TeamViewer* y sin la posibilidad de obtener el control remoto. En pruebas realizadas se ha podido observar que *HangOut* responde con retrasos y algunas pérdidas cuando el tráfico de red aumenta.

4. *Saros* nace como un *plugin* (Plug-in) para el entorno de desarrollo Eclipse, que permite el desarrollo colaborativo de aplicaciones. Los cambios efectuados por uno de los integrantes del equipo se reflejan en las versiones locales de los desarrolladores. Esta herramienta permite la transferencia de un proyecto completo pero se recomienda utilizar una herramienta de versionado de software a los efectos de mantener disponible todas las versiones de los archivos que forman parte del historial del proyecto.

Pruebas combinadas: *TeamViewer*, *HangOut* y *Saros*

En este punto se señala que el objetivo fundamental de efectuar pruebas con la combinación de herramientas, fue probar su rendimiento ante una demanda de uso más exigente de las capacidades de la red.

En primera instancia se realizaron pruebas con *TeamViewer* y *HangOut* para luego evaluar el desempeño conjunto de *HangOut* y *Saros*, ambas herramientas corriendo al mismo tiempo en dos computadoras distintas. Una de ellas fue una máquina de escritorio con altos recursos, en lo que a hardware se refiere, la otra fue una *netbook* con limitaciones en tanto a sus capacidades gráficas y de procesamiento.

Se procedió a probar la comunicación por medio de voz ip en las circunstancias mencionadas y como fruto de las experiencias puede destacarse que resulta más recomendable la utilización de *Saros*, para compartir el código, y *HangOut*, que tiene la ventaja de indicar si las personas involucradas se encuentran conectados a *Google+*, *Google talk* o *Gmail*, para compartir la imagen y la voz. Por su parte se ha prescindido del *Adobe Connect*

por no reunir los requerimientos mínimos que exige esta modalidad de trabajo. Bajo estas condiciones la programación en parejas opera con eficiencia dado que la tecnología permite una comunicación fluida con la posibilidad de compartir pantallas según diferentes opciones.

Evaluación comparativa entre metodología ágil centralizada y la distribuida

A partir del análisis de estas experiencias, y con la visión enfocada en una nueva metodología ágil distribuida, se procedió intencionalmente a efectuar una evaluación comparativa del desempeño de los desarrolladores en ambas modalidades metodológicas: centralizada y distribuida.

Para tal efecto, uno de los desarrolladores del equipo integró simultáneamente dos *stakeholders* utilizando en uno de ellos la modalidad centralizada de metodología ágil y en el otro la modalidad distribuida que se propone en el presente trabajo. El propósito de esta práctica fue analizar comparativamente ambas modalidades a los efectos de extraer conclusiones conducentes a mejorar esta propuesta. Del análisis surgieron las siguientes apreciaciones:

1. *Ronda*: las reuniones diarias (*Rondas*) que se llevan a cabo por internet son más breves, precisas y organizadas que las que se mantienen en forma presencial, puesto que suelen ser más concretas que en el caso de reuniones físicas donde se da lugar a distracciones y charlas informales no pertinentes. Adoptar las prácticas de reunión diaria o *rondas* es una tarea extremadamente importante al comunicarse de forma remota.

2. Espacio físico: el trabajar desde el domicilio personal puede representar tanto una ventaja como una desventaja, dependiendo de las circunstancias. En una oficina de desarrollo por lo general se cuenta con mejor infraestructura, orden e higiene, dado que comúnmente en los llamados *software factory* se cuenta con personal dedicado a esta tarea. En el caso de los desarrolladores *freelance* estas condiciones no siempre están garantizadas, lo que suele ocasionar pérdidas de concentración. El trabajo desde el domicilio personal conlleva comodidades que por lo general superan a las

que se encuentran en las oficinas de desarrollo y que pueden influir positivamente en el confort del desarrollador.

3. Indumentaria: en las oficinas los trabajadores suelen verse obligados a utilizar indumentaria formal, lo que puede generar incomodidad en algunos casos y dificultar la concentración en períodos prolongados de tiempo.

4. Aprovechamiento de las comunicaciones: en una oficina la comunicación con otros miembros del equipo surge de forma espontánea y ágil. En el caso de la comunicación vía internet se requiere mayor coordinación, lo que provoca una pérdida en el dinamismo o espontaneidad. La comunicación de voz es esencial a la hora de programar de a pares en forma distribuida y *HangOut* cubre este requisito. Por su parte, la compartición de código demanda una realimentación instantánea de los pares y el uso de *Saros* facilita esta modalidad al mostrar inmediatamente los cambios efectuados por el par desarrollador.

5. *Debug*: el proceso de *debugging* utilizado en la metodología ágil distribuida es más rápido que el que se da en la metodología propuesta por *eXtreme Programming* (XP). Al operar cada desarrollador con un equipo propio, pueden ejecutar más pruebas en la misma unidad de tiempo o encarar la búsqueda del error desde perspectivas diferentes, lo que potencia el rendimiento.

6. Capacidad de los desarrolladores: el aprendizaje de un lenguaje o de una tecnología nueva depende de la capacidad autodidacta de cada desarrollador. Si bien en la metodología XP un desarrollador más experimentado puede enseñar al mismo tiempo que desarrolla, el otro miembro del equipo está más concentrado en *ver lo que el otro hace* que en *verificar si lo hace bien*. En la metodología distribuida se propone trabajar de a pares pero cada uno con su equipo personal. Esto implica una ventaja a la hora de visualizar más directamente en su propia pantalla las acciones ejecutadas por su par y, simultáneamente, disponer del equipo para efectuar otras tareas paralelas, como por ejemplo ejecutar pruebas o buscar información mediante foros u otros medios disponibles. El hecho de que cada desarrollador cuente con su

propio equipo brinda la capacidad de investigar mientras el par continúa trabajando, ya sea evaluando el código, probándolo o incluso desarrollándolo.

Líneas de investigación y desarrollo

Los temas que se desarrollan se encuadran dentro de las siguientes líneas de investigación del I3:

1. Ingeniería del software: (a) Ingeniería del software orientada al web, (b) Métodos avanzados de producción de software y (c) Usabilidad y accesibilidad de aplicaciones y sistemas web
2. Dispositivos Móviles en Aplicaciones Empresariales.

Resultados y objetivos

A partir de los análisis de las diferentes bibliografías consultadas y de las experiencias de desarrollo de los diferentes miembros del equipo de investigación, se proponen las siguientes pautas de trabajo distribuido para la Metodología Ágil basada en Telecomunicaciones (MATE):

1. El ciclo productivo de un entregable debe estar comprendido entre 15 y 60 días. Este producto representará una Entrega de Valor Agregado (EVA) al entregable anterior.
2. El producto se entrega debidamente testeado, por lo que cada ciclo productivo debe contemplar el tiempo necesario para ejecutar todos los tipos de pruebas, incluyendo las pruebas beta.
3. El trabajo se deberá desarrollar a distancia en modalidad semi-sincrónica, con al menos 4 horas diarias de trabajo de programación en pareja, sincrónico, y el resto de testeo individual, asincrónico.
4. El mejor lugar desde donde el cliente, en caso de existir, puede participar, es su propio ámbito de trabajo.
5. Para una mejor organización del trabajo se definen los siguientes roles, constituyendo todos ellos el *stakeholders* en el sentido aceptado del vocablo en el campo de conocimientos de las ciencias informáticas:

- a. El dueño de la idea.
- b. El facilitador.

c. El oráculo.

d. Los desarrolladores.

El dueño de la idea constituye una interfaz que vincula al “cliente” con el “equipo”.

El facilitador por su parte es el líder de equipo de desarrollo quien comienza y termina las reuniones virtuales, determina el orden del día, enuncia los puntos acordados, fija posiciones con el dueño de la idea, gestiona los recursos y acuerda la extensión de cada EVA en función de lo propuesto por los desarrolladores.

El oráculo² gestiona toda la información concerniente al proyecto, tal que cualquier integrante del *stakeholders* tenga un rápido acceso a la misma desde cualquier punto.

Los desarrolladores determinan la extensión de las tareas y tienen a su cargo el desarrollo del código y su correspondiente testeo y depuración.

Cada integrante del *stakeholders* puede ejecutar más de un rol al mismo tiempo. La idea es que estos roles puedan ser desempeñados desde cada uno de los ámbitos de los integrantes del equipo, incluyendo al cliente (de existir).

Mecánica de trabajo propuesta

Una vez concebida una idea, a partir de la cual se pretende construir una aplicación, se constituye un equipo de análisis formado por los desarrolladores conducidos por el dueño de la idea. Se trabaja sobre el análisis del producto, intentando enunciar las funcionalidades deseadas y haciendo abstracción de su dificultad. Se elabora un documento de precedencia de funciones, en virtud del potencial de generación de utilidades de cada una, acordado con el cliente o el dueño de la idea. Los miembros del equipo de desarrollo dividen las funciones en tareas estimables, en términos de tiempo y esfuerzo, y se estipulan los plazos para la realización de las mismas. Este procedimiento se realiza una única vez por cada EVA.

Semanalmente, preferentemente todos los lunes, se realizan reuniones donde intervienen todos los involucrados, con excepción del

2 Oráculo (RAE) Persona a quien todos escuchan con respeto y veneración por su mucha sabiduría y doctrina.

cliente o dueño de la idea. En estas instancias el facilitador conduce la revisión de la asignación semanal de tareas.

En la *ronda* diaria, y en forma breve, cada par de desarrolladores y el facilitador registran el reporte de avances y de dificultades. En esta oportunidad se revisan las tareas pendientes. En caso de que la distancia física imponga una diferencia horaria tal que impida la reunión simultánea de todos los pares, el facilitador mantendrá reuniones dedicadas con cada grupo de pares que gocen de husos horarios compatibles.

La jornada de trabajo no debe superar las 8 hs. diarias. Los pares productivos deben poder trabajar al menos 4 hs en forma sincrónica por lo que sus locaciones geográficas y costumbres deben permitirlo. Se trabaja en forma colaborativa y simultánea por captura de escritorio. El tiempo de trabajo independiente se dedica a la revisión de código, prueba y depuración.

Aplicación de la MATE

Las recomendaciones propuestas para la nueva MATE se están aplicando actualmente en el desarrollo de una aplicación de gestión de la propia metodología, similar a la *Kunagi*, que asista a la administración del tablero, con las tareas que conforman los sprint, y que ostente la particularidad de poder ejecutarse en todo dispositivo móvil con pantalla táctil. Esta aplicación dispondrá además de los gráficos de avance, a modo de asistentes visuales del trabajo concluido y del que se encuentra pendiente. La interfaz gráfica facilitará el arrastre de las notas dentro del tablero con la facultad de actualizar los estados de cada tarea.

Formación de recursos humanos

En esta línea de investigación participan 3 docentes y 3 alumnos de la carrera de Sistemas de Información de la Universidad Adventista del Plata.

Referencias

Ableson, F. (2011). *Android: Guía para Desarrolladores* (2º Ed.) - Madrid: Anaya Multimedia.

Blanco, P., Camarero, J., Fumero, A., Warterski, A., Rodríguez, P. (2009). *Metodología de desarrollo ágil para sistemas móviles: Introducción al desarrollo con Android y el iPhone*. Universidad Politécnica de Madrid. Documento recuperado de http://www.adamwesterski.com/wp-content/files/docsCursos/Agile_doc_TemasAnv.pdf

Carvajal Riola, J. C. (2008). *Metodologías ágiles: herramientas y modelo de desarrollo para aplicaciones Java EE como metodología empresarial*. Tesis final de Master.

Fantasia, J., Ferrochio, D., Maldonado, C., Martínez, E., Trujillo H. (2006). Desarrollo de una metodología utilizable en la construcción de aplicaciones de tecnología Móvil. Recuperado de http://sedici.unlp.edu.ar/bitstream/handle/10915/20785/Documento_completo.pdf?sequence=1

Gamma E., Helm, R., Johnson, R. y Vlissides, J. (2003). *Patrones de Diseño*. Madrid: Pearson Educación.

Letelier, P. y Penadés, M. C. (2006). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia. Documento recuperado de <http://www.willydev.net/descargas/masyxp.pdf>

Ray, J. y Johnson, S. (2010). *Desarrollo de Aplicaciones para Iphone*. Madrid: Anaya Multimedia.

Scalone, F. (2006). Estudio comparativo de los modelos y estándares de calidad del software. Tesis de Maestría. Recuperado de <http://laboratorios.fi.uba.ar/lsi/scalone-tesis-maestria-ingenieria-en-calidad.pdf>

Wake, W. C. (2000). *Extreme Programming Explored*. Recuperado de <http://faculty.ksu.edu.sa/MubarakRashed/Books/Extreme.Programming.Explored.pdf>