

VirtualScrum: Un groupware para dar soporte a reuniones virtuales en Métodos Ágiles

Ing. Guillermo H. Rodríguez

ISISTAN Research Institute, UNICEN - Tandil, Buenos Aires, Argentina.

grodri@exa.unicen.edu.ar

Resumen. El desarrollo de software tradicional se ha encontrado con distintas situaciones que ponen en riesgo la aplicación de prácticas rígidas para el desarrollo de un proyecto. En la actualidad, las situaciones de inestabilidad en el contexto socio-económico, constantes cambios e incertidumbre hacen que los modelos clásicos de desarrollo queden obsoletos y poco aplicables. En este contexto, surgen los Métodos Ágiles como alternativa metodológica en el marco de un ambiente de desarrollo caótico. Este enfoque representa una óptica distinta que apunta a resolver el estancamiento de proyectos basados en el modelo tradicional y aboga por un proceso más liviano que sea adaptable a los cambios y permita la entrega periódica de porciones de software funcionales, sin perder de vista el concepto de calidad. En particular, Scrum parece ser la alternativa más fuerte, dado su gran utilización en la industria de software y por estar principalmente focalizado en la administración de proyectos.

No obstante, los Métodos Ágiles depende profundamente del factor humano y de trabajo colaborativo, sobre todo en la misma locación física. En efecto, esto parece ser una amenaza para la aplicación de esta metodología ya que existe una creciente tendencia a la distribución de desarrolladores por todo el mundo. En este trabajo se propone VirtualScrum, una herramienta que actúa como plataforma de mundo virtual para realizar reuniones a pesar de la distribución geográfica. Una encuesta realizada a un grupo de estudiantes mostró la efectividad de la herramienta que inclusive reemplazaría a las reuniones presenciales.

1 Introducción

Hasta hace poco, el proceso de desarrollo llevaba asociado un marcado énfasis en el control del proceso mediante una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Uno de los modelos principales que se encuentran en la literatura, es el modelo CMM-CMMI [CMMI-DEV 2006], el cual ha demostrado ser efectivo y necesario en proyectos de gran tamaño (respecto de tiempo y recursos), donde por lo general, se exige un alto grado de ceremonia de proceso [Letelier 2003]. Sin embargo, esta perspectiva no resulta ser la más adecuada para una gran

cantidad de proyectos actuales donde el entorno del sistema es muy cambiante, y en donde se exige reducir ampliamente el tiempo de desarrollo sin perder de vista la alta calidad. Los Métodos Ágiles (MA), en cambio, son más adecuados para proyectos de escala reducida, sumado que la ausencia de burocracia, le permite ser flexible en un ambiente con fuerte tendencia a cambios. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del “buen hacer” de la ingeniería de software, asumiendo el riesgo que conlleva [Letelier 2003]. En este escenario, las metodologías ágiles emergen como una posible respuesta para cubrir ese vacío metodológico. Por estar especialmente orientadas para proyectos pequeños, los MA constituyen una solución a medida para este entorno, aportando una elevada simplificación que, a pesar de ello, no renuncia a las prácticas esenciales para asegurar la calidad del producto.

No obstante, una de los principales problemas de la aplicación de MA en los proyectos actuales es la dificultad en la realización de las reuniones debido a que los miembros que forman parte del desarrollo de un proyecto de software se encuentran ubicados en diferentes locaciones físicas. En estas condiciones, la realización de reuniones puede ser un gasto importante de tiempo y recursos [Maher 2005].

Entendiendo que el fenómeno de desarrollo distribuido de software está creciendo de manera sostenida y que dicha tendencia no parece desaparecer [Herbsleb 2007], han surgido muchas herramientas que permiten coordinar e interactuar con las personas que trabajan en forma remota. Estudios realizados en grupos de “trabajo real” (oponiéndose a la diversión y el juego) dentro de los groupwares han demostrado la mejora de productividad de los equipos [Benford 1997], especialmente cuando los mismos están geográficamente o temporalmente dispersos. Existen herramientas 2D¹ para desarrollo de métodos ágiles pero que no soportan reuniones virtuales. Esta debilidad puede ser resuelta por un herramienta 3D para el desarrollo colaborativo mediante la cual, las personas puedan llevar a cabo reuniones cara a cara. En particular, las interfaces en 3D son una base muy popular para el software colaborativo. De hecho, la metáfora de las salas de reunión se aprobó por varias herramientas, las cuales generalmente representan a los participantes humanos por medio de *avatares* [De Lucia 2008]. La principal ventaja de esto es la posibilidad de sentirse como si se estuviese en el lugar de desarrollo, con los compañeros de equipo y con todos los artefactos inherentes al proyecto. Por otro lado, existe gran variedad de mundos virtuales 3D pero no soportan reuniones de trabajo.

En este contexto, en este artículo se presenta VirtualScrum, una herramienta para el desarrollo virtual de Scrum que explota la metáfora 3D para dar soporte a reuniones y

¹ <http://agiletools.wordpress.com/2007/12/04/rally-vs-scrumworks/>

discusiones en cada parte del proceso de las metodologías ágiles. En particular, VirtualScrum se basa en la inclusión de artefactos como Product Backlog, Sprint Backlog, herramienta de planificación y asignación de tareas, soporte de Daily Meetings, entre otros. Para evaluar el funcionamiento de la herramienta, se realizaron pruebas con grupos de estudiantes que utilizaron la herramienta simulando que formaban parte de un grupo de Scrum y que debían desarrollar un prototipo de un sistema encargado por una empresa. Los resultados demostraron que es viable realizar reuniones de desarrollo a pesar que las personas no estén en el mismo lugar físico. Asimismo, a través de una encuesta realizada sobre performance y usabilidad de la herramienta, la mayoría de los estudiantes reemplazarían las reuniones cara a cara por la utilización de Virtual Scrum. El resto del trabajo se organiza de la siguiente manera. En la sección 2 se muestra el proceso de desarrollo del método ágil Scrum. La sección 3 describe conceptos de trabajo colaborativo en entornos 3D. En la sección 4 se presenta el interior del mundo virtual con sus artefactos. En la sección 5 se presentan resultados obtenidos de los casos de estudio. La sección 6 presenta trabajos relacionados. La sección 7 muestra nuestras conclusiones. Por último, se anexan dos apéndices que agregan mayor información sobre lo detallado en el informe.

2 Scrum: Una alternativa ágil para el desarrollo maduro de software

Esta perspectiva ágil está mostrando su efectividad en proyectos con requerimientos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo, sin perder de vista una alta calidad. Este enfoque ágil está revolucionando la manera de producir software, y a la vez generando un amplio debate entre sus seguidores y quienes, por escepticismo o convencimiento, no las ven como alternativa para las metodologías tradicionales [Letelier 2003]. Uno de los puntos claves de las metodologías ágiles es alivianar la rigidez de los procesos clásicos, pero priorizando la comunicación dentro del grupo de desarrollo [Beedle 2002, Schwaber 2002]. Los MA tienen en cuenta que el desarrollo de software es una actividad humana, por eso enfatizan al individuo sobre el proceso [Belchior 2007].

El objetivo de los MA es la entrega continua de software funcional y completamente verificado, mediante entregas pequeñas y periódicas. En general, estas metodologías proporcionan un ambiente propicio para que los clientes establezcan, acepten, prioricen, y verifiquen los requerimientos que ellos solicitan. Además, se confía mucho en el conocimiento de los miembros del equipo, oponiéndose, así, a la documentación exigida por los procesos pesados [Boehm 2004]. Scrum, en particular, se caracterizan por ser muy tolerantes a los cambios, aliándose a ellos. Es clave en este tipo de metodologías, un

desarrollo con verificaciones constantes, de modo de dar soporte y motivación para iteraciones futuras.

Al finalizar cada iteración, período denominado Sprint, es muy valioso concretar una reunión de retrospectiva para revisar las estimaciones realizadas y el aprendizaje del equipo, con el objetivo de reflexionar para las subsiguientes iteraciones. Las reuniones, en efecto, son uno de los principales mecanismos que permite incrementar la comunicación, la resolución efectiva de inquietudes y la construcción de consenso dentro del grupo de trabajo.

Los métodos ágiles definen la manera de llevar adelante el trabajo bajo valores y principios ágiles en pos de responder a los desafíos del desarrollo rápido y requerimientos muy cambiantes. Según Jim HighSmith [HighSmith 2004], la agilidad es la habilidad de crear y responder a los cambios en orden de salir favorecido en un contexto turbulento de negocio.

En la figura 1, puede observarse la estructura del proceso de Scrum.



Figura 1 Ciclo de vida de un proceso con la metodología Scrum².

Las reuniones son uno de los principales mecanismos que permiten incrementar la comunicación entre los miembros de un equipo, la resolución efectiva de inquietudes y la construcción de consenso, evitando roces y malentendidos provocados por la falta de la presencia física, miradas y gestos. El inconveniente de las reuniones es el costo de los recursos y la dificultad de su gestión. En las últimas dos décadas varios esfuerzos de investigación han sido dedicados a reuniones de apoyo.

² Fuente de la imagen: <http://www.proyectosagiles.org/>

3 Trabajo colaborativo en entornos 3D

Los ingenieros de software han adoptado un amplio rango de tecnologías de comunicación y colaboración para asistir a grupos en la coordinación de proyectos. Las herramientas específicamente para desarrollo colaborativo de software se clasifican en cuatro categorías [Whitehead 2007]:

Model-based: posibilita la visualización de la representación de software.

Process support: permite a los desarrolladores visualizar el seguimiento de un proceso.

Awareness tool: apunta a tener conciencia de lo que realizan los demás miembros del grupo con el fin de evitar conflictos.

Collaboration infrastructure: permite mejorar la interoperabilidad entre las herramientas de colaboración, especialmente la integración de datos.

Un ambiente 3D virtual puede permitir a otros desarrolladores saber acerca de las actividades desarrolladas por sus pares, e incluso poder celebrar reuniones cuando los miembros están distribuidos. Asimismo, un ambiente 3D puede ser usado para proveer un topología física de la estructura de la información en un gran proyecto de software, el cual permitiría un acceso más rápido a la información del proyecto [Whitehead 2007].

Además, para proveer capacidades de conciencia grupal y comunicación, muchas herramientas explotan las características de la integración colaborativa en ambientes de desarrollo. Por ejemplo, uno de los objetivos es tener la información siempre disponible a través de control de versiones. Otra ventaja es que, en este contexto, los artefactos están siempre compartidos en un entorno de desarrollo integrado [Herbsleb 2007].

En definitiva, un entorno de desarrollo 3D contribuye en que los desarrolladores se sienten familiarizados con el ambiente mediante la visualización de software que les permite percibir artefactos que naturalmente son intangibles.

Las plataformas virtuales se han convertido en una alternativa para que las personas se vinculen a pesar que no se encuentren en el mismo lugar físico. Distintos miembros de un equipo de trabajo pueden compartir sus ideas, a pesar que las distancias que los separan sean muy grandes. Esto arroja consecuencias económicas sumamente notorias para las empresas, que ahorrarían costos de traslados y organización de reuniones.

El advenimiento de mejores conexiones a Internet y computadores con mayor potencia han incentivado la creación de Mundos Virtuales en donde las personas se reúnen para interactuar, como es el caso de Second Life [Second Life 2009] o juegos online masivos (MMOG, *Massive Multiplayer Online Game*) donde miles de jugadores interactúan a la vez. Las plataformas virtuales se han convertido en una alternativa para que las personas se vinculen a pesar que no se encuentren en el mismo lugar físico. La

interacción en estos entornos trata de simular con la mayor verosimilitud posible un entorno real.

En particular, en este trabajo se focaliza en los entornos virtuales para dar soporte a reuniones de trabajo, logrando simular una oficina de trabajo. Esta forma de trabajo realizado colaborativamente encuadra en lo que se conoce actualmente como *groupware*, es decir, software colaborativo [Barret 2000]. La idea es que usuarios ubicados en distintas estaciones de trabajo distribuidas puedan conectarse, mediante una red, a un entorno virtual compartido, de modo que compartan ideas y opiniones como si estuvieran ubicados en el mismo lugar físico.

4 VirtualScrum: Un marco virtual para balancear disciplina y agilidad

En la presente sección se describe la funcionalidad provista por el *groupware* para el soporte de reuniones y la descripción de cada uno de los artefactos del proceso de desarrollo de Scrum que componen la sala.

El objetivo es que el marco en el cual se llevan a cabo las reuniones, sea lo más verosímiles posible para que sea viable poder reemplazar las reuniones cara a cara entre los miembros de un equipo, que en algunas organizaciones es bastante costoso concretarlas, dada la distribución geográfica de los desarrolladores. En la figura 2, se puede apreciar el interior de *Virtual Scrum*, escenario de reuniones virtuales para seguir un proceso de Scrum. Los cuatro participantes son miembros de un equipo de Scrum vinculados con el desarrollo de un proyecto de Software. Cada desarrollador está representado a través de su avatar. Como se puede ver, el lugar de la reunión tiene varios artefactos que simulan un entorno real de oficina o habitación de conferencia.



Figura 2 Interior de Virtual Scrum.

Para poder concretar reuniones de software, principalmente de Scrum, es necesario que la sala cuente con los siguientes artefactos para garantizar que se cumplan las prácticas y que haya comodidad a la hora de tomar parte de ella:

Visor de Product Backlog: En Scrum, los requerimientos capturados son colocados en lo que se denomina Product Backlog (ver Fig. 3). Otros componentes se pueden observar en apéndice B.



Figura 3 Product Backlog

4.1 Comparación con reuniones face-to-face

Las reuniones son un factor muy importante en la comunicación entre miembros que forman parte de un equipo. Además, los artefactos utilizados y generados en ellas poseen gran valor para fijar y recordar cuestiones analizadas. En las reuniones *face-to-face* que se realizan formando parte de una metodología ágil como Scrum, es muy frecuente el uso de post-it notes sobre un pizarrón, bocetos en rotafolios hechos con marcadores y escritos en pizarras blancas. A la hora de realizar una reunión con el personal distribuido geográficamente estas herramientas quedan obsoletas, por lo que se requiere utilizar herramientas de alta fidelidad que posean la capacidad de realizar el trabajo en forma colaborativa y distribuida. Es por eso que Virtual Scrum aparece como una alternativa sólida para dar soporte a los artefactos utilizados y generados en las reuniones.

5 Resultados experimentales

En esta sección se mostrará una prueba de concepto, su preparación y sus resultados.

5.1 Caso de estudio

Para la realización de la experiencia, se recurrió a tres grupos de estudiantes de la Universidad Nacional del Centro de la Provincia de Buenos Aires, formado cada uno por 7 personas entre 21 y 25 años. Ellos contaban con la experiencia de haber formado parte de un equipo de real de Scrum, y además, algunos de ellos tuvieron el rol de Scrum Master. El tamaño del equipo fue el que proponen los teóricos, es decir, 7 ± 2 integrantes. Cada uno contaba con su computadora personal de modo de simular la distribución geográfica y que no sea necesario que compartan la misma computadora.

Luego de entrenarse con la herramienta, aprender la ubicación de los distintos elementos dentro del mundo virtual y comunicarse mediante la herramienta de chat, los miembros se consolidaron como grupo y llegaron a reconocerse como tal, se les propuso realizar un proyecto simulando que un cliente se lo solicitaba. El proyecto se trataba de realizar un sitio Web para una empresa encargada de promocionar y vender productos de pastelería.

5.2 Análisis de métricas de Scrum

Las métricas se agrupan en indicadores que sirven para análisis en la performance del proceso de software [Mahnic 2007]³. Se apunta a cumplir con el objetivo “Información

³ [Mahnic 2007]. Mahnic, V., Zabkar, N. Introducing CMMI Measurement and Analysis practices into Scrum-based software development process. International Journal of Mathematics and Computers in Simulation. Eslovenia. 2007.

oportuna sobre los resultados de los proyectos” del área de Métricas y Análisis de CMMI.

Los indicadores son los siguientes:

- *Work Effectiveness(WE)*
- *Schedule Performance Index (SPI)*
- *Cost Performance Index of labor costs (CPI)*

WE se refiere al ratio entre el decremento de trabajo remanente y la cantidad de trabajo realizado. En el plano ideal, el decremento de trabajo remanente entre los días d1 y d2 de un Sprint, debería ser mayor o igual a la cantidad de trabajo realizado en ese intervalo. En efecto, el valor objetivo de este indicador es 1 o más; sin embargo, valores muy superiores a 1 hablan de una planificación pobre.

SPI se refiere al ratio entre el valor ganado (valor de las tareas completadas) y el valor de planificación (la estimación inicial). El valor objetivo de este indicador es mayor o igual a 1. Valores muy superiores a 1 indican que el proyecto avanza más de lo previsto.

CPI se refiere al ratio entre el valor ganado y los costos actuales. El valor objetivo es mayor o igual a 1, indicando que el costo de completar el trabajo es justo lo planeado o menos de lo previsto.

Fórmulas de los indicadores:

Earned Value (ER)

$$ER_{d,j} = \frac{\sum_{i=1}^{d-1} WS_{i,j}}{\sum_{i=1}^{d-1} WS_{i,j} + WR_{d,j}}$$

ER_{d,j} es el ratio entre la cantidad de trabajo realizado en la tarea j el día i, WS_{i,j}, con i=1,2,...,d-1 y el total de trabajo WS_{i,j} + WR_{d,j} (realizado y remanente).

Schedule Performance Index (SPI)

$$SPI_d = \frac{\sum_{j=1}^n ER_{d,j} * WR_{init,j}}{\sum_{j=1}^n WR_{init,j}} * \frac{SL}{DE}$$

Donde WR_{init,j} denota la estimación inicial del trabajo remanente para la tarea j, SL es la longitud del Sprint, y DE el número de días transcurridos desde el comienzo del Sprint.

Cost Performance Index (CPI)

$$CPI_d = \frac{\sum_{j=1}^n ER_{d,j} * WR_{init,j} * CEH_j}{\sum_{i=1}^{DE} \sum_{j=1}^n WS_{i,j} * CEH_j}$$

Usando la métrica de trabajo realizado, se puede calcular el costo laboral actual multiplicando las horas transcurridas y el costo de una hora de ingeniería (CEH, Cost

Engineering Hour) para todas las tareas del Sprint Backlog. Con respecto a CEH, se tomó un valor consensuado de 10.

Los resultados arrojados en la experiencia simulada fueron:

Uno de los grupos pudo capturar 9 requerimientos a partir de la narrativa.

A continuación, se muestra el tiempo de realización que le dieron a cada tarea y la estimación inicial. A su vez, la cuarta columna muestra el valor de ER para cada tarea. Se decidió que el día de medición fue el día 5, una vez comenzado el Sprint. Para simplificar, se determinó que las tareas comenzaron a realizarse todas el mismo día (ver Tabla 1):

Tarea	Tiempo de realización (días)	Estimación Inicial (puntos de importancia)	ER _{tarea}
1	14	100	0,35
2	14	10	0,35
3	7	150	0,71
4	14	100	0,35
5	14	100	0,35
6	14	5500	0,35
7	14	100	0,35
8	14	100	0,35
9	7	100	0,71

Tabla 1 Distribución y valor ganado de las tareas.

Para el día 5 luego de comenzado el Sprint, se calcularon los siguientes parámetros:

$$SPI_5=2.18$$

$$CPI_5=9.98$$

Como se puede observar, los valores dieron mayor a 1, por lo que las estimaciones han sido buenas con tendencia a ser sobreestimadas. Tal es el caso de la tarea 6 que fue estimada con un valor muy alto (5500). Este valor fue puesto porque el equipo no contaba con experiencia previa en la utilización de la tecnología Web, por lo que le asignaron un alto puntaje a esta tarea.

5.3 Análisis de las encuestas realizadas

A partir de la encuesta realizada (ver apéndice A.) a cada miembro que participó de la experiencia de usar Virtual Scrum, se procesaron los datos y se determinaron 4 *clusters* a modo de indicadores. El modelo de la encuesta se puede observar en el apéndice. En general, la aceptación fue buena lo que permitió alcanzar las expectativas propuestas.

Decisión y Coordinación

En el gráfico de la figura 5, se puede observar que para una alta mayoría de los encuestados, el proceso de trabajo, toma de decisiones y discusión fue eficaz y eficiente.

No obstante, los problemas de usabilidad presentes en la herramienta provocaron cierto ruido en la comunicación entorpeciendo el camino hacia la toma de decisiones. Por otro lado, hubo una porción de los encuestados que le pareció muy satisfactorio el proceso de discusión mediado por el groupware.

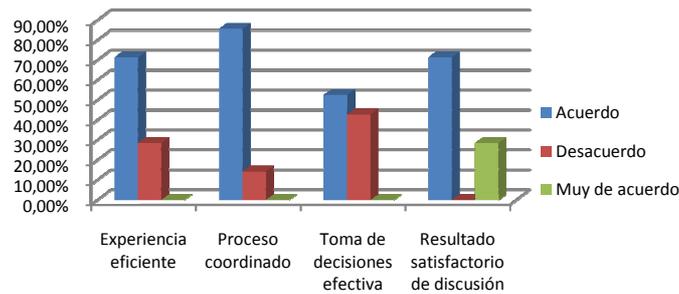


Figura 4 Distribución según opinión sobre toma de decisiones y coordinación.

Usabilidad

En la figura 6, se puede apreciar que hay cuestiones de usabilidad que afectan a la comodidad para utilizar el groupware. Estas cuestiones de usabilidad se pensaron como trabajo futuro. No obstante, hubo una buena aceptación de la herramienta sumado que la mayoría podría llegar a reemplazar a una reunión *face to face*, en caso que las mismas no sean viables o se dificulten concretarlas. Es notable ver que casi el 90% de los encuestados usaría la herramienta siempre que no sea posible el contacto cara a cara.

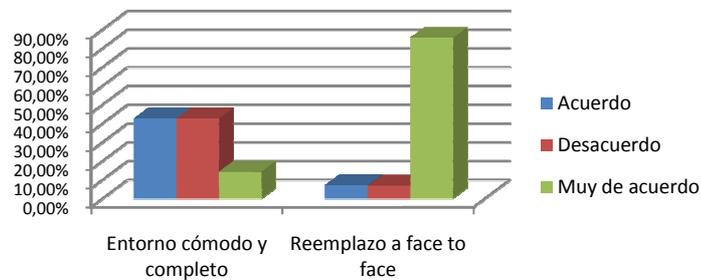


Figura 5 Distribución del grado de usabilidad del sistema.

Asimismo, sobre el tema de usabilidad, se indagó a cada miembro si volverían a usar Virtual Scrum para los fines que fue creado, y los resultados, según la figura 7, arrojaron que el 14,3 % volvería a usar la herramienta alguna vez y el 85,7 % usaría la herramienta siempre y cuando no sea viable concretar la reunión cara a cara. Se puede observar que el groupware atrae a los miembros de desarrollo y parece ser una

alternativa fuerte a la hora de una reunión virtual con miembros distribuidos geográficamente.

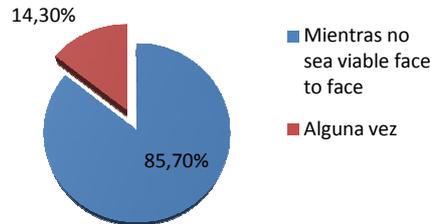


Figura 6. Opiniones sobre usar la herramienta nuevamente.

Compromiso

En el gráfico de la figura 8, se puede observar que hay un alto porcentaje de desacuerdo que asciende casi al 50% que no se siente responsable en la correctitud de la solución. Claramente, esa actitud es producto de la falta de seguridad y comodidad dentro de la herramienta provocada por los errores de usabilidad. No obstante, se puede apreciar los grados de compromiso para con la solución lograda con la herramienta son altos. Se puede concluir que, si se realizan mejoras en usabilidad, habrá mayor compromiso y actitud responsable en el proceso de toma de decisiones mediado por el groupware.

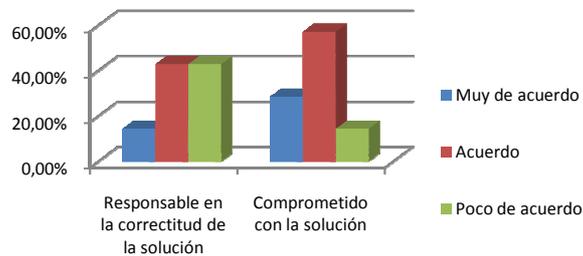


Figura 7 Grados de compromisos asumidos en la interacción con la herramienta.

Calidad de trabajo

En la figura 9, se puede notar que, aproximadamente el 70% de los usuarios que participaron de la experiencia de usar la herramienta, opinan que están satisfechos con la calidad del resultado arrojado en la interacción con el groupware. En efecto, se puede concluir que los errores de usabilidad, no afectan de manera considerable a la calidad del proceso de interacción entre los miembros.

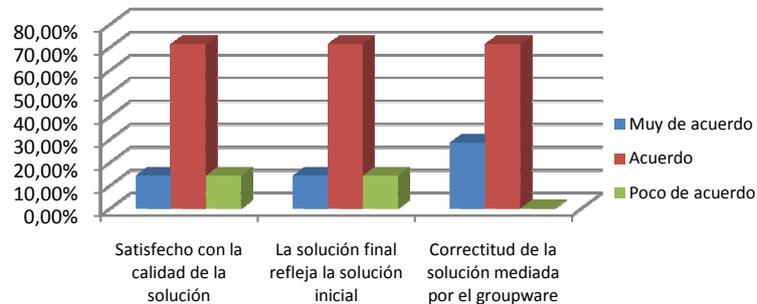


Figura 8. Opiniones sobre la calidad de la solución obtenida con la herramienta.

6 Trabajos relacionados

A continuación, se muestran algunos trabajos donde se utilizan o crean mundos virtuales para diversos propósitos.

El mundo virtual más conocido actualmente es Second Life [Second Life 2009], creado por la empresa Linden. Second Life, a diferencia de lo que muchas personas piensan, no es un juego, sino un mundo virtual de propósito general. Provee un SDK (*Software Development Kit*) donde personas particulares pueden crear elementos dentro del mundo, con los cuales los *avatares* pueden interactuar. No obstante, el mundo virtual no presenta hasta el momento una propuesta para realizar reuniones de desarrollo de software.

Wonderland [Wonderland 2009], por otra parte, debido a su naturaleza *Open Source*, provee un *toolkit* que permite adaptarse a las necesidades de una compañía o proyecto en particular. El proyecto Wonderland está totalmente desarrollado en java y liberado bajo la licencia GPL para la creación de mundos virtuales 3D colaborativos. Dentro de estos mundos, los usuarios pueden comunicarse con audio de alta fidelidad, compartir aplicaciones de escritorio y documentos y llevar a cabo un negocio real. Por otro lado, existen herramientas 2D para desarrollo colaborativo en métodos ágiles como CollabNet TeamForge⁴ que permite visualizar y desarrollar artefactos de software en un entorno distribuido, pero no admite la interacción virtual de los desarrolladores ni tampoco la posibilidad de desarrollar reuniones. Otra herramienta que se desempeña en 2D es ScrumWall⁵ de Microsoft, en la cual es posible visualizar artefactos del desarrollo del proyecto basado en Scrum a través de una interfaz web. De todos modos, no presenta alternativa para la realización de reuniones entre los miembros del equipo. VirtualScrum, al igual que los mundos existentes, permite el desarrollo de una metáfora en 3D pero con

⁴ http://www.b-vision.com/products_03_ctf.php

⁵ <http://www.scrumwall.com/>

la desventaja que no soporta comunicación vía reconocimiento de voz, sino sólo mediante chat. No obstante, cuenta con la ventaja de presentar artefactos propios para la realización de una reunión para desarrollar software. Esta última característica parece ser una limitación presente en los mundos virtuales existentes.

El objetivo de VirtualScrum es extender la funcionalidad provista por los mundos virtuales existentes para soportar reuniones, particularmente las que se realizan practicando el método ágil Scrum, permitiendo interactuar de manera similar a la realidad. Asimismo, se pretende ofrecer dentro del groupware, las facilidades de desarrollo provistas por las herramientas 2D existentes, incluyéndolas como artefactos del mundo virtual.

7 Conclusiones

En este trabajo se presentó un groupware capaz de soportar reuniones virtuales para proyectos basados en métodos ágiles. Asimismo, se analizó una aproximación para hacer frente a la problemática del software distribuido en un contexto donde es muy costoso para las organizaciones concretar reuniones presenciales en forma periódica.

Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptables, sobre todo en proyectos pequeños y con requerimientos cambiantes. Una de las cualidades más destacables en una metodología ágil es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo.

Dadas las condiciones actuales del desarrollo de software en que los miembros trabajan en forma de distribuida, resulta poco factible aplicar métodos ágiles sin ningún tipo de soporte. Uno de los principales problemas del desarrollo distribuido es la falta de comunicación constante y la poca frecuencia con las que se celebran reuniones, dados los costos que implica realizarlas.

Por estos motivos, Virtual Scrum propone facilitar varias de las cuestiones mencionadas, donde la aplicación de metodologías ágiles parece obstaculizarse. El trabajo presentado cumple satisfactoriamente con el objetivo de facilitar la realización de reuniones en un mundo virtual. Los resultados de la experiencia realizada muestran que el uso del groupware desarrollado fue efectivo y, además, una gran mayoría volvería a usar la herramienta, e incluso reemplazaría a una reunión *face to face*, en caso que ésta

se vea imposibilitada se ser concretada. No obstante, existen mejoras a realizarse que influirían en la performance de reuniones. Actualmente, se está trabajando en manejo de audio VOIP y reconocimiento de voz, como así también, en la integración dentro del mundo virtual con herramientas de desarrollo 2D tales como el gestor de documentos SVN, el issue-tracker Mantis y el entorno de desarrollo integrado Eclipse.

Referencias

- [Barret 2000] Barret S., Tangney, B. Wilde: Supporting Change in Groupware.
- [Beedle 2002 ,Schwaber 2002]. Beedle, M., Schwaber, K. Agile Software Development with Scrum. 2002.
- [Belchior et.al 2007]Belchior, A., Soares Furtado, F., et.al. Mapping CMMI Project Management Process Areas to Scrum Practices.
- [Benford 1997] Benford, S., Colebourne, A., O'Brien, J., Rodden, T., and Snowdon, D. Phoenix. Informing the Design of Collaborative Virtual Environments. AZ. 1997. ISBN: 0-89791-897-5.
- [Boehm 2004] Boehm, B., Turner, R. Balancing Agility and Discipline: A Guide for the Perplexed. Addison Wesley.2003.
- [CMMI-DEV 2006] CMMI for Development, V1.2 model, CMU/SEI-2006-TR-08. Software Engineering Institute, 2006.
- [De Lucia 2008] De Lucia, Francese, Passero, Tort. SLMeeting: Supporting Collaborative Work in Second Life. 2008.
- [Herbsleb 2007] James D. Herbsleb. Global Software Engineering: The Future of Socio-technical Coordination. School of Computer Science. Carnegie Mellon University. 2007.
- [HighSmith 2004]. HighSmith, J. Agile Project Management. 2004.
- [Letelier 2003]. Letelier, P., Canós, J., Penadés, M.C. Metodologías Ágiles en el Desarrollo de Software. Univ. Politécnica de Valencia. 2003.
- [Maher 2005] Mary Lou Maher. DesignWorld: A Tool for Team Collaboration in High Band Virtual Environments. 2005.
- [Schwaber 2004] Schwaber, K. "Agile Project Management with Scrum".2004.
- [Second Life 2009] <http://www.secondlife.com>.2009
- [Whitehead 2007] Whitehead, J. "Collaboration in Software Engineering: A Roadmap". University of California, Santa Cruz. 2007.
- [Wonderland 2009] <https://wonderland.dev.java.net/index2.html>.2009

APÉNDICE A: Encuesta realizada a los miembros que participaron del experimento

1. Nombre :
2. Tamaño del equipo:
3. Género: Mujer: Varón:
4. Edad: <20 21-25 26-30 >30

Por favor responda las siguientes preguntas en base a la experiencia universitaria y laboral:

5. ¿La experiencia de trabajo mediante groupware resultó eficiente?

- Fuertemente desacuerdo
- Desacuerdo
- Acuerdo
- Fuertemente de acuerdo

6. ¿El proceso de desarrollo le pareció coordinado?

- Fuertemente desacuerdo
- Desacuerdo
- Acuerdo
- Fuertemente de acuerdo

7. ¿El proceso de toma de decisiones le pareció efectivo?

- Fuertemente desacuerdo
- Desacuerdo
- Acuerdo
- Fuertemente de acuerdo

8. ¿El entorno de trabajo le pareció cómodo y completo?

- Fuertemente desacuerdo
- Desacuerdo
- Acuerdo
- Fuertemente de acuerdo

9. ¿El resultado de la discusión del grupo fue satisfactorio?

- Fuertemente desacuerdo
- Desacuerdo
- Acuerdo
- Fuertemente de acuerdo

10. ¿Qué tan satisfecho está con la calidad de la solución del grupo?

- Para nada satisfecho
- Poco satisfecho
- Satisfecho
- Completamente satisfecho

11. ¿En qué medida la solución final refleja la solución al problema inicial?

- Para nada acertado
- Poco acertado
- Acertado
- Completamente acertado

12. ¿En qué medida se siente comprometido con la solución del grupo?

- Muy desinteresado
- Desinteresado
- Comprometido
- Muy comprometido

13. ¿En qué medida está seguro que la solución alcanzada es correcta utilizando este medio?

- Fuertemente desacuerdo
- Desacuerdo
- Acuerdo
- Fuertemente de acuerdo

14. ¿En qué medida se siente personalmente responsable de la correctitud de la solución?

- Nada responsable
- Poco responsable
- Responsable
- Fuertemente responsable

15. ¿En qué medida el groupware reemplaza las reuniones cara-a-cara?

- Para nada
- Poco
- Mucho
- Completamente

16. ¿Volvería a usar el groupware para alguna reunión de trabajo?

- Nunca más
- Alguna vez
- Mientras no sea viable cara a cara
- Siempre

APÉNDICE B: Variedad de artefactos presentes en Virtual Scrum

A continuación, se presenta la gama de artefactos presentes en el mundo virtual:

- *Visor de Product Backlog*: En Scrum, los requerimientos capturados son colocados en lo que se denomina Product Backlog. Éste, es una lista de los requerimientos en formato de post-it note (Ver figura 10).



Figura 10 Product Backlog

- *Visor de Sprint Backlog*: En el Sprint Backlog, como se ve en la figura 11, se encuentran las tareas que el equipo se comprometió a realizar durante el transcurso de ese sprint. Como en el componente anterior, las tareas van pasando por un estado de TO DO, DOING y DONE. Cada encargado de la tarea es responsable de asignarle un estado a la tarea.

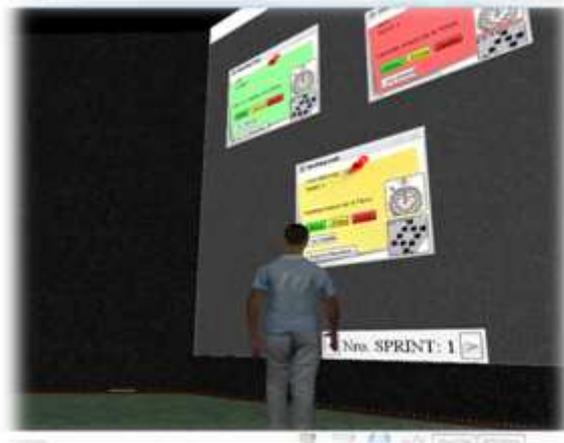


Figura 11 Sprint Backlog del sprint 1

- *Soporte de Planning Game*: La alternativa más fuerte en Scrum a la hora de estimar es la denominada *Poker Planning*. El modelo consta de 8 cartas con los números representados más abajo, los cuales los equipos emplean como unidad de esfuerzo, es decir, como días de trabajo de cada par de programadores que trabajan con tareas de tamaño máximo de 10 días. El funcionamiento es muy simple: cada participante dispone de un juego de cartas, y en la estimación de cada tarea, todos vuelven boca arriba la combinación que suma el esfuerzo estimado. Cuando se considera que éste es mayor de 10 horas (o del tamaño máximo considerado por el equipo para una tarea), se levanta la carta “infinito”. Los números que siguen la serie de Fibonacci son: $\frac{1}{2}$, 1, 2, 3, 5, 6, 7, 8, *infinito*. En *Virtual Scrum*, la planificación sigue la idea de *Poker Planning*, sólo que de manera digital. Existe una interfaz, en la figura 12, para cada miembro del equipo para que pueda emitir su voto numérico.



Figura 12 Estimación con Poker Planning en la sesión de planning

- *Visor de filminas*: En las reuniones es muy común presentar los temas fundamentales en formato de filminas con el objetivo de atrapar mejor la atención, lograr mejor explicación didáctica y dejar plasmado en un documento lo charlado. En la figura 13, puede verse un ejemplo.



Figura 13 Visor de filminas para dar soporte a presentación de temas y charlas

- *Browser Web*: Es posible que en una reunión sea necesario realizar una búsqueda en algún sitio Web, visitar un sitio particular del tema, entrar a la

casilla de mail para buscar algún dato e inclusive entrar a las herramientas Web que forman parte del desarrollo. La figura 14 ilustra un caso particular.



Figura 14 Navegador web para realizar búsquedas de información

- *Calendario*: Para organizarse las reuniones, programar eventos, planificar entregas y estimar tareas, es fundamental la existencia un calendario real, como el que se puede ver en la figura 15. *Virtual Scrum* posee un calendario, en cual es posible navegar por los meses y avanzar y retroceder de año.



Figura 15 Calendario navegable para organizar y planificar eventos

- *Agenda*: Este componente resulta ser el complemento al calendario anteriormente mencionado. En la agenda de la figura 16, es posible visualizar los eventos programados para un día en particular. Cada evento se puede ver en detalle con toda su descripción.



Figura 16 Visualización de la agenda con los eventos diarios

- *Chat*: Para permitir la comunicación instantánea entre todos los miembros, más allá del rol, *Virtual Scrum* brinda soporte de un chat, según se aprecia en la figura 17. El objetivo es que todos estén comunicados ante cualquier comentario o necesidad y en todo momento.



Figura 17 Conversación llevada a cabo entre miembros del equipo mediante un chat

- *Soporte de Daily Meeting*: El objetivo de esta reunión es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad, al poner de manifiesto puntos en que se pueden ayudar unos a otros, según el formato de *daily meeting* que propone Scrum (Ver figura 18).

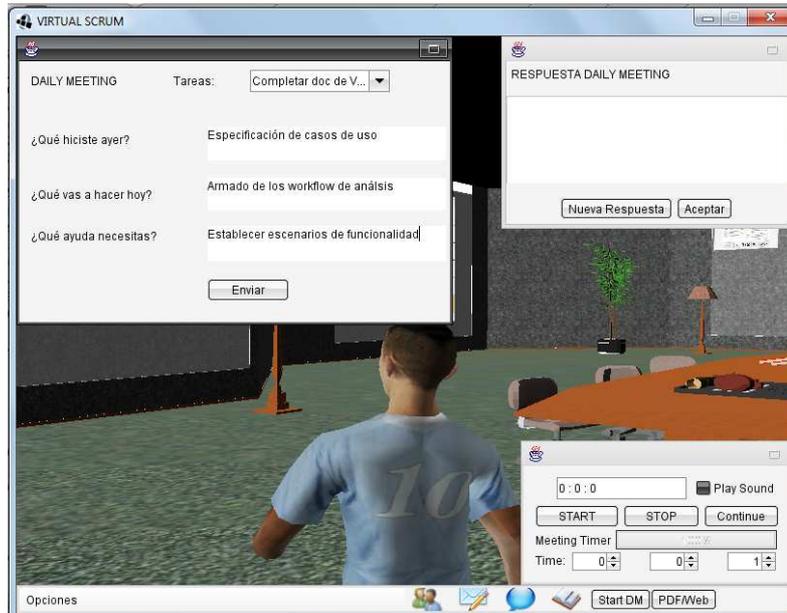


Figura 18 Daily Meeting