

Tipo de artículo: Artículo original
Temática: Ingeniería y Gestión de software
Recibido: 11/03/2013 | Aceptado : 6/09/2013

Implementación de métodos ágiles para la simulación de casos de uso y prototipado en el proceso de desarrollo de software

Implementing agile methods for simulating use case and prototyping in the software development process

Mónica Patiño Castro¹, Luis Merchán², César Pardo²

¹ Seguros Bolívar S.A, Colombia, CL 56 No. 2-155, Cali.

² Grupo de Investigación LIDIS. Universidad de San Buenaventura, seccional Cali, Colombia, Ave. 10 de Mayo, La Umbría, Carretera a Pance, Cali.

monica.patino@segurosbolivar.com; lmerchan@usbcali.edu.co; cjpardo@usbcali.edu.co

Resumen

Este artículo presenta el manejo de simulación de casos de uso, la creación de prototipos y generación de escenarios como parte inicial del ciclo de vida del desarrollo del software, aplicando el concepto de diseño ágil; es decir, cómo se puede llegar a una aproximación cercana de la dinámica del negocio proporcionando interacción temprana con el cliente, para continuar luego con las etapas formales del ciclo de vida.

Palabras clave: Casos de uso, diseño ágil, prototipado, simulación.

Abstract

This paper presents the use of simulation use cases, prototyping and scenario generation as an initial part of the life cycle of software development, using then concept agile design; ie how to reach a close approximation of the dynamics of the business by providing early interaction whit the client, and continues with the client, and continues with the formal stages of the life cycle.

Keywords: Agile design, prototyping, simulation, use case.

Introducción

Como es parte fundamental del ciclo de vida del desarrollo del software el encargarse de garantizar y validar que el software cumpla plenamente con los requerimientos de la aplicación, y dado que en la mayoría de los casos el cliente no es claro en la definición de los requisitos; se propone que mediante el manejo de simulaciones de los casos de uso se diseñe el modelo de un sistema real, con el propósito de presentar un determinado número de escenarios que satisfagan la necesidad del cliente, con los cuales él podrá como parte activa del proyecto identificar y describir el comportamiento y la estructura del sistema.

El proceso de simulación nos lleva a etapas de modelización de la realidad que nos permite abstraer y entender de manera sencilla los procesos, rescatando los objetos y estados importantes para la generación de cada uno de los

diferentes planteamientos o escenarios propuestos. Por medio del *Unified Modeling Language* (UML) se construirán los modelos de simulación, el cual es una herramienta que permite la evolución y refinamiento de manera ágil ayudando a que el diseño del modelo establezca una retroalimentación con el usuario sobre lo que se quiere representar; permitiendo modelar el sistema desde una perspectiva y nivel de detalle diferente indicando las principales funcionalidades del sistema.

Los prototipos construidos se deben desarrollar en poco tiempo sin utilizar muchos recursos, logrando así la eficiencia y desarrollo de varios escenarios. El enfoque del desarrollo ágil nos orienta a la representación de los aspectos del sistema que realmente se requieren y gracias a ellos se refinan los requisitos del software que se desarrollaran.

Planteamiento del problema

La presentación de los requisitos mediante un lenguaje natural en la gran mayoría de las situaciones llega con problemas de ambigüedad, escalabilidad y trazabilidad (López, 2000) y poca conexión del negocio con la implementación final del sistema.

El hecho de que la elaboración de los casos de uso parte de que exista una excelente definición y buen entendimiento de los requerimientos; si partimos de que al realizar el proceso de elaboración de los casos de uso su comportamiento se especifica simulando el flujo de eventos con el propósito de presentar al usuario uno o varios modelos cercanos a la realidad o a lo que realmente se requiere.

En el momento de la elaboración de los casos de uso comúnmente se cae en el error de realizar una descomposición funcional lo cual nos desvía de la necesidad principal o hace que se pierda el contexto, por lo tanto si modelamos con base en una serie de criterios principales podremos identificar qué valor agregado para el cliente puede tener uno u otro caso de uso simulado, según KurtBittner. “Un caso de uso debe producir valor para los actores con los que interactúa”.

Entenderemos el propósito de simular casos de uso de un sistema, experimentando con modelos de trabajo, según R.E Shannon que dice “La simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de entender el comportamiento del sistema o evaluar nuevas estrategias dentro de los límites impuestos”.

Una vez se han identificado y elaborado los casos de uso simulados, desarrollamos los prototipos creando bocetos por medio de las herramientas disponibles en el mercado que nos brinda apoyo en su elaboración de una manera fácil y rápida pudiendo realizar modificaciones en forma sencilla e involucrar el manejo de usabilidad provista por la misma herramienta; la construcción del prototipo se debe enfocar en los aspectos del software visibles para el cliente, el cual los evaluará y así se podrán refinar.

Cuando se ha seleccionado el caso de uso basado en la simulación y posterior esbozo de prototipado, se continuará con el desarrollo formal que contempla todo el ciclo de vida del desarrollo del software.

Es importante poder llegar a un alto nivel de aceptación por parte del cliente con los que se ha modelado, lo cual nos garantiza confiabilidad y calidad para las etapas posteriores.

Marco teórico

La actual generación de las organizaciones y la evolución de las existentes, están demostrando que los requerimientos de software sean elaborados con mayor rapidez y con alta calidad satisfaciendo las necesidades de

todos los medios, dicha situación hace que se requiera de técnicas, herramientas y metodologías que faciliten el proceso en el desarrollo del software.

Para especificar el comportamiento de un sistema partimos de los Casos de Uso los cuales operan mediante la descripción de un conjunto de secuencias y acciones que el sistema debe realizar generando un resultado con características de valor.

Los casos de uso proporcionan al desarrollador el camino para comprender el requerimiento del cliente, validan la arquitectura y verifican que el sistema sea consecuente (García, 2004).

El concepto de los casos de uso fue introducido por Jacobson en 1992, el cual propone realizar una descripción de los requerimientos funcionales que nacen de las necesidades del cliente (Ceria, 1998). Con base en sus publicaciones los especialistas en métodos orientados a objetos tomaron los casos de uso como la forma de especificar el comportamiento externo de un sistema, siendo así su notación incorporada al lenguaje estándar de modelado UML (*Unified Modeling Language*) propuesto por Jacobson Rumbaugh y Booch. Su éxito radica en entender a un sistema mediante una secuencia de interacciones con los actores describiendo las funcionalidades del sistema.

La propuesta planteada es fijar el concepto como parte inicial del ciclo de vida del desarrollo del software, realizando Simulación de los casos de uso, desarrollo de prototipos y manejo de escenarios antes del desarrollo formal basados en el criterio de diseño ágil, con el propósito de obtener un excelente nivel de aceptación con el cliente estableciendo sinergia entre él y la tecnología.

Investigaciones anteriores que han incursionado con el concepto de simulación de casos de uso, nos lleva a continuar en el camino y a evaluar sus características las cuales podemos adoptar y tener en cuenta. El estudio publicado en Estados Unidos por la IEEE *Computer Society* en la 29 conferencia internacional de ingeniería de software en el año 2007, presentan una importante relación con este propósito; por medio del tema UCSIM: *A Tool for Simulating Use Case Scenarios* (Jayaraman and Whittle, 2007).

Para la ejecución de la simulación de los casos de uso y manejo de escenarios se tomará como referencia el planteamiento de la herramienta UCSIM, el cual se basa en un plugin de eclipse para la visualización y simulación de casos de uso el cual consiste en convertir la lista de casos de uso en archivos ejecutable o jerarquías UML llamadas máquinas de estado proveyendo: semántica, visión local de cada subsistema y el uso de la máquina de estados en etapas posteriores del desarrollo.

UCSIM, usa extensiones mínimas de los modelos de UML para describir los casos de uso, transforma los casos de uso automáticamente en formato ejecutable (jerárquicos UML *State Machines*) proporcionando un simulador configurable para la ejecución de la máquina de estado. En UML (Lenguaje Unificado de Modelado), una máquina de estado es aquel comportamiento que permite hacer un seguimiento de la vida de un objeto en el transcurso de un tiempo finito (Tutorial UML 2, 2007).

Otro de los planteamientos importantes que apoyan la simulación es el enfoque detallado en el paper "TOWARD A STANDARD PROCESS: THE USE OF UML FOR DESIGNING SIMULATION MODELS" donde se especifican los modelos de simulación mediante el uso de UML, (Richter and Lothar, 2012) todas las características esenciales de la estructura y las dinámicas del modelo de simulación que se construirá pueden ser descritas.

Por lo tanto, antes de que la implementación empiece, los requisitos sobre el modelo de simulación pueden ser formulados fácilmente. Además, los usuarios, así como también los desarrolladores obtienen una especificación y una documentación paralela al proceso de construcción del modelo de simulación.

A través de UML se obtiene, para el desarrollo del proceso integral, una descripción de los requisitos y las características de rendimiento del modelo de simulación en cuanto a su estructura y dinámica.

Seguidamente a la simulación de los casos de uso, se desarrollaran los prototipos, los cuales por medio de modelos exploratorios ayudan a detallar con claridad la posible solución, facilitando los cambios rápidamente.

Existen gran variedad de herramientas que permiten realizar prototipos, una de ellas es “Calico” (Mangano, *et al.*, 2008), una herramienta de software de diseño, por medio del uso de una Tablet (PC), realiza un diseño ligero o esbozo informal, permitiendo de forma rápida explorar y organizar las ideas orientadas al diseño del software que lleven a la solución adecuada. La diferencia radica en que permite deshacer y hacer las operaciones diseñadas, iniciar con un bosquejo y luego agregar complejidad, agrupar elementos importantes para el diseñador haciendo que la interpretación sea más fácil.

Se busca que la herramienta de prototipado cumpla con características (Maner, 2000) esenciales para el desarrollo, entre ellas: Experimental, es decir, que permita implementar validaciones específicas; otra característica requerida es la operacional, es decir, iterativo y progresivamente refinado que nos conducen directamente al sistema real.

Podemos obtener un manejo sencillo y amigable con otras herramientas de prototipado como *Balsamiq Mockups*, el cual es un programa descrito en Flex y AIR creando rápidamente wireframes, posee una interfaz fácil de usar. Esta herramienta puede ser instalada en diferentes sistemas como Windows, Linux, Mac, posee funcionalidades básicas hasta la exportación a PNG o PDF (Balsamiq, 2012).

La necesidad de buscar el entendimiento de los procesos nos ha permitido incrementar el desempeño y optimizar los procesos mediante herramientas como simuladores (Brooks and Tobias, 2012) que se basan en un modelo del proceso real los cuales han podido potenciar sus destrezas y habilidades haciéndolos cada vez más competitivos.

La propuesta es trabajar sobre un ambiente para la construcción de simulaciones, basado en una metodología que permite integrar al usuario de una manera activa en el proceso de implementación a través de UML (del inglés, *Unified Modeling Language*), lenguaje que nos aporta el marco necesario para construir e integrar modelos (Tripani and Illescas, 1999), ofreciendo una notación estándar para describir estructuras y comportamientos de los objetos por medio de la especificación, visualización y documentación de los artefactos de un sistema.

La evolución del diseño ágil del desarrollo del software, dio inicio a mediados de los años 90, como reacción a los modelos estrictos, estructurados, lentos y poco eficientes; en el año 2001 miembros de la comunidad se reunieron en Snowbird y adoptaron el nombre de “métodos ágiles” proporcionando un entorno productivo (Calderón and Sahara, 2007), con la participación de grupos pequeños y desarrollo por iteraciones. Se encuentra como la más destacada de las metodologías ágiles a: *XP eXtreme Programming*, siendo capaz de adaptarse a los cambios de requisitos.

El éxito radica en el cumplimiento del propósito como aporte inicial en el ciclo de vida de un proyecto, usar notación para comunicar el modelo, aplicar el proceso con la simulación y prototipado bajo características y condiciones en las que se debe actuar (manejo de escenarios) y generar aceptación del cliente en las primeras etapas del desarrollo.

Características de la propuesta

El propósito que se quiere lograr con el planteamiento de simulación de casos de uso en la parte inicial del ciclo de vida del desarrollo del software, es el de construir modelos por medio de la simulación planteando diversas alternativas con el fin de elegir y adoptar la mejor para el sistema, con el objetivo de que sea la óptima o lo más cercana a la realidad planteada por el cliente.

Describir la realidad que se desea simular con el grado de abstracción apropiado para afrontar el problema que se quiere resolver mediante la simulación a implementar.

Validar la información presentada en el prototipo, que sea correcta y adecuada, ayudando al cliente a visualizar de manera concreta los casos de uso simulados, los cuales se detallaran en el software que se va a construir; y permite eliminar muchos posibles malos entendidos cuando se presente el producto final.

Reflejar de manera lógica la visión del sistema a través de las interfaces elaboradas en etapas tempranas (Zapata and González, 2008).

Reducir los riesgos que se presentan cuando se desarrolla sin tener una visión clara de lo que el cliente quiere, los cuales se resuelven con el grado en el que el modelo logra describir el comportamiento, con su precisión y con la facilidad de que sean comprendidos.

Modificar los modelos de simulación de manera rápida con el fin de analizar diferentes políticas o escenarios.

Mejorar el sistema por la vía de la simulación, es más recomendable que hacerlo directamente en el sistema real; es mucho más sencillo comprender y visualizar los métodos de simulación que los métodos puramente analíticos.

Permitir involucrar al cliente o usuario final desde la etapa más temprana del desarrollo, alcanzando sinergia entre los involucrados y la tecnología.

Materiales y métodos

Experimentación

Descripción del proceso

Como primera medida, se debe tener una descripción breve del problema, de los procesos que la organización lleva a cabo, identificar las actividades que se deben desarrollar, las posibles interfaces y con que cuenta actualmente la organización para su implementación.

Elaborar Simulación de los casos de uso

Con base en la información que se ha obtenido, y haciendo uso de las herramientas que actualmente nos provee la tecnología, se da inicio al desarrollo de los casos de uso; comenzando así la etapa de modelización de la necesidad del cliente para llegar a la realidad requerida, involucrando activamente al cliente o al usuario.

Por tanto, definimos los límites del sistema y los objetivos, verificando que estos no cambien durante el desarrollo del proceso; logrando identificar cuellos de botella o posibles restricciones.

Una de las herramientas para tomar como referencia es la planteada por UCSIM, la cual por medio de un plugin de eclipse permite la visualización y simulación de casos de uso convirtiendo la lista de éstos en archivos ejecutables o máquinas de estado proporcionando un simulador configurable.

Otra de las formas por la que se puede optar es la especificación de los modelos de simulación mediante el uso de UML.

Los diagramas UML se utilizaran para asignar aspectos relevantes o de gran importancia, como son: La interacción con los usuarios, estructuras estáticas y comportamiento dinámicos. Nos centramos en los casos de uso (Jacobson 1992) (Zapata and González, 2008), mostrando la funcionalidad que ofrece el sistema futuro desde la perspectiva del usuario. Como consecuencia obtenemos una descripción gráfica del modelo de simulación, la cual será utilizada como parte de la documentación requerida en todo proceso de software.

Diversas herramientas nos permiten elaborar casos de uso, siendo esta una técnica muy bien aceptada dada la facilidad de entendimiento por los clientes. Entre ellas se destacan: Eclipse y *Enterprise Architect*, a continuación se mencionan algunas características:

Tabla. Características de algunas herramientas

#	Eclipse	Enterprise Architec
1	Es un entorno de desarrollo integrado (IDE), de código abierto multiplataforma de desarrollo. Posee distintos plug-ins entre los que se destaca My-eclipse	Soporta los 13 diagramas de UML: Diagramas de comportamiento: Casos de Uso, Actividades, Estados, Secuencias y Diagramas de Estructura: Paquetes, Clases, Objetos, Composición, Despliegue.
2	Software de desarrollo Open-Source y multiplataforma que proporciona robustez. Recientemente ha creado Eclipse Modeling Project cubriendo casi todas las áreas de ModelDrivenEngineering.	Posee un sólido núcleo UML 2.4.1 y una interfaz de usuario intuitiva, fácil de construir y mantener, repositorio eficiente y escalable. Cuenta con el apoyo de SOLUS S.A Empresa de servicios, tecnología y desarrollo de aplicaciones
3	La arquitectura de plugins de eclipse permite integrar diversos lenguajes sobre su IDE.	Arquitectura dirigida por modelos permitiendo transformar modelos simples en complejos.
4	Permite introducir otras aplicaciones útiles durante el proceso de desarrollo como: Herramientas UML, editores visuales de interfaces.	Con cerca de 300.000 usuarios en todo el mundo, a un precio asequible; con 12 años de desarrollo continuo ha logrado una excelente suite de herramientas, compatible con muchos estándares.
5	Permite la libertad de elegir el entorno del proyecto. Extendible a otros lenguajes de programación, bases de datos, Subversion.	Enterprise Architect es una plataforma de modelado, diseño y administración colaborativa basada en UML 2.4.1.
6	Proporciona un framework que hace más fácil crear, integrar las herramientas de software. Posee un conjunto de extensiones que apoyan Java, Bases de datos, UML, AJAX.	EA es una herramienta que abarca integralmente el ciclo de vida, cubre el desarrollo de software desde los requerimientos, pasando por las etapas de análisis, modelos de diseño, testing y finalmente el mantenimiento.
7	Desarrollado inicialmente por IMB, es ahora desarrollado por la Fundación Eclipse organización que fomenta una comunidad de código abierto.	Posee validación de modelos, soporte para MDA (Arquitectura dirigida por modelos) y documentación.

Desarrollo de prototipos

Diseño o primer modelo que opera como el acercamiento a lo que el cliente necesita, esbozado en la etapa temprana del ciclo de vida del desarrollo del software. Con técnicas de cuarta generación permitir que con diversos lenguajes se elaboren rápidamente prototipos, ensamblando componentes de software, re-utilizándolos, agrupando elementos, haciendo y re-haciendo el diseño sin ocasionar traumatismo en la elaboración del producto, conduciéndonos a la solución adecuada ya que con la participación activa del cliente se logra encontrar y entender de mejor manera cual será el resultado y la forma en que debe responder el sistema ante la interacción del usuario con el software.

Es importante dejar claro que el prototipo construido es un mecanismo para la definición de los requisitos que son la base para las siguientes etapas del desarrollo y con el tiempo puede evolucionar.

Diseño ágil y grado de aceptación

El desarrollo ágil como experiencia recogida, minimiza los riesgos en el desarrollo, permite que exista comunicación con el equipo de trabajo.

Como el planteamiento de XP (Calderón and Sahara, 2007), el cual difiere con definir todos los requisitos al comienzo del proyecto y posteriormente dedicar mucho tiempo en controlar los cambios en los requisitos. Poder medir el grado de aceptación o de rechazo del producto de software en la primera etapa es lograr justificar las ventajas que ofrecerá el producto final.

Validación de la simulación

Como lo que se ha modelado es una abstracción del sistema real, se debe llegar a obtener la validación de la simulación, lo cual hace referencia a realizar la construcción de un modelo correcto, es decir que sea una excelente representación de lo que el cliente espera sea su sistema final.

La validación se logra en la medida en que se va construyendo hasta conseguir su aceptación; si se descubren fallas que nos indique el no estar siguiendo el camino correcto, se debe en corto tiempo corregir, incluir nuevos aspectos relevantes o excluir los sobrantes y efectuar una nueva revisión.

Es importante que exista confianza en el modelo por quienes harán uso del sistema el cual inicia con la simulación.

Resultados y discusión

Seleccionada la herramienta que para la propuesta se han planteado dos alternativas: La primera es usar MyEclipse para la elaboración de los casos de uso en el sistema del cliente y UCSIM para la simulación (ver figura 1).

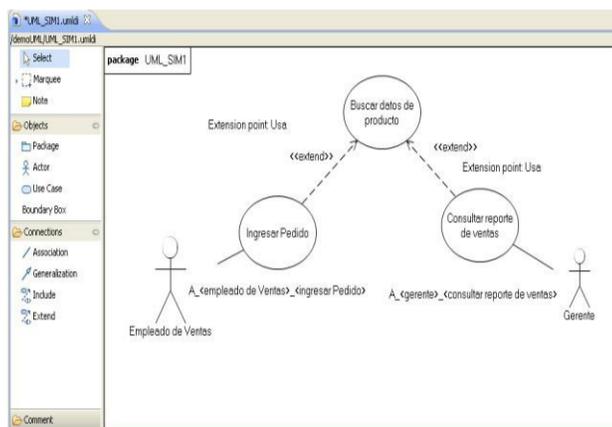


Figura 1. Caso de uso de un sistema de pedidos.

La facilidad que nos provee la herramienta, permite que el cliente pueda comprender las acciones o funcionalidades que el sistema y los interesados deben realizar.

Con base en este resultado, pasamos a la elaboración del diagrama de estados, tomando cada nodo especificado, con el propósito de modelar su comportamiento durante su tiempo de vida, identificando los posibles estados por los cuales puede atravesar en el proceso y llegaremos a los eventos de respuesta manejados, controlados y esperados (ver Figura 2).

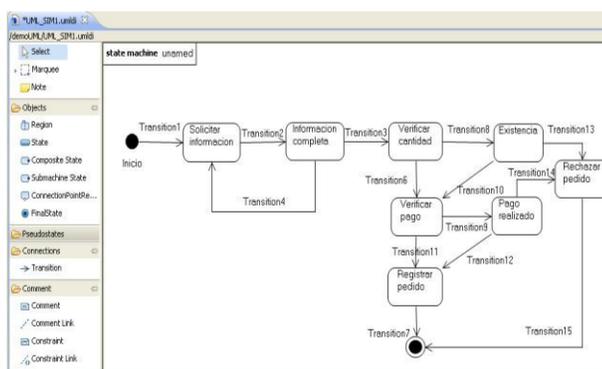


Figura 2. Diagrama de estado.

Con el usuario se evalúa la propuesta y si no llena las expectativas, se plantea conjuntamente otra versión y se simula un nuevo diagrama de caso de uso y de estado; pero si el resultado es el esperado y satisface la necesidad, también se

puede simular con base en criterios mejorados un nuevo caso de uso; el anterior no se desecha, lo que se pretende entonces es presentar las diversas posibilidades que se acercan más a la realidad requerida.

Con la herramienta UCSIM se pueden igualmente tomar un conjunto de máquinas de estado (Jayaraman and Whittle, 2007), permitiendo el simulador verlas para incluirle eventos, manejando de esta manera varios escenarios de prueba como lo podemos observar en la siguiente figura; estos pueden ser personalizados por el usuario desde el mismo momento de la ejecución.

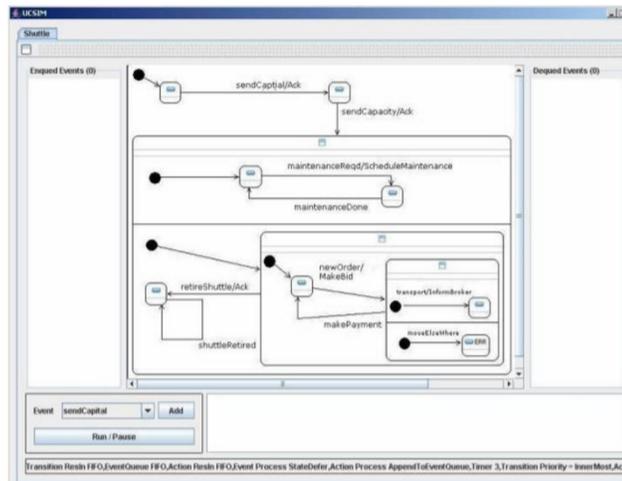


Figura 3. Simulación de máquina de estado.

Como segunda alternativa, se propone el uso de *Rational Software Modeler* (RSM) (IBM, 2012), con la cual se pueden realizar bocetos de diagramas UML modelando con alto nivel de abstracción de forma rápida y sencilla (ver Figura 4).

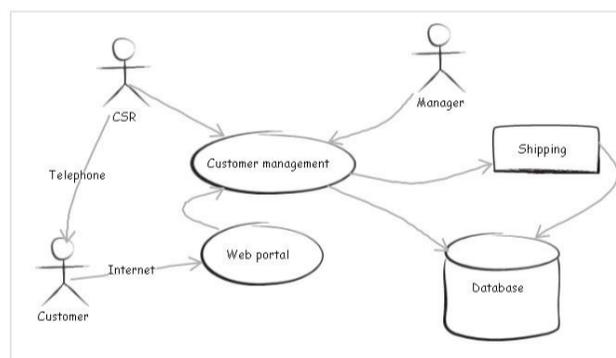


Figura 4. Scketching: Dibujar diagramas sencillos.

Los elementos del boceto pueden contener otros elementos los cuales pueden ser agrupados o se pueden relacionar entre ellos. Cuando se haya completado el diagrama, la herramienta permite convertir los bocetos o elementos del esquema en elementos de UML, pasando a un modelo más específico que representa el modelo del sistema que se puede simular y depurar.

El modelo será simulado utilizando la perspectiva de la ejecución, obteniendo comportamientos con posibilidad de realizar correcciones para mejorar el modelo. Se puede ejecutar hasta un punto específico o seleccionar un área determinada y obtener resultados rápidamente. Los resultados pueden ser vistos como mensajes históricos producidos durante la ejecución del modelo, permitiendo realizar documentación de lo que ha ocurrido, flujo de eventos, evaluando los escenarios.

La tecnología nos ha permitido llegar a que el manejo gráfico es mucho más relevante, sencillo y amigable que el manejo descriptivo o de sólo texto; nos permite llevar secuencias y adición de comentarios que aclaren o detallen el proceso que se está representando.

Continuando con el proceso, hacemos uso de una herramienta de prototipado, entre ellas se encuentran algunas muy sencillas y fáciles de manejar, dependiendo del nivel del proyecto se pueden seleccionar la más pertinente. A continuación se detalla el manejo que se efectuaría con alguna de ellas. La primera es: Balsamiq Mockups (Balsamiq, 2012), presentando prototipos de diseño para web; permitiendo realizar borradores rápidos dando forma a los modelos por medio de varias pantallas que serán posteriormente enlazadas; con el objetivo de presentarle al usuario la solución al sistema de manera interactiva sin complicaciones. Estos bocetos serán la base para las etapas siguientes del proceso formal.

El usuario puede formar parte activa de la creación del boceto dada la facilidad de manejo de la herramienta sin requerir que tenga conocimiento técnico; comunicando eficientemente sus ideas y poder hacer acuerdos de aspectos importantes para el desarrollo como la distribución de los elementos, jerarquías y navegación.

Al igual el desarrollador puede entregar rápidamente una propuesta de la solución sin que se tome mucho tiempo en su elaboración. La Figura 5 nos muestra parte de las opciones que presenta la herramienta para el desarrollo.

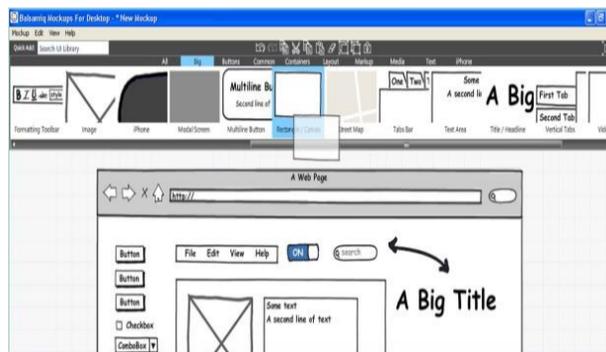


Figura 5. Herramienta BalsamiqMockups.

Otra de las herramientas para manejo de prototipado es CALICO, la cual se encuentra en la tercera versión y continúa su desarrollo; permite realizar bocetos de modelamiento en diseño temprano, con base en una pizarra electrónica o Tablet apoyando el esbozo informal con una rápida exploración y organización de las ideas de diseño (Mangano, *et al.* 2008).

Se plasman las ideas en un espacio (lienzo) amplio o abierto, dibujando todo lo que se piensa (Figura 6), estos pueden ser modificados y refinados detallando los diagramas con características específicas que den mayor claridad a lo que el cliente necesita (Figura 7).



Figura 6. CALICO, Bosquejo básico [5].

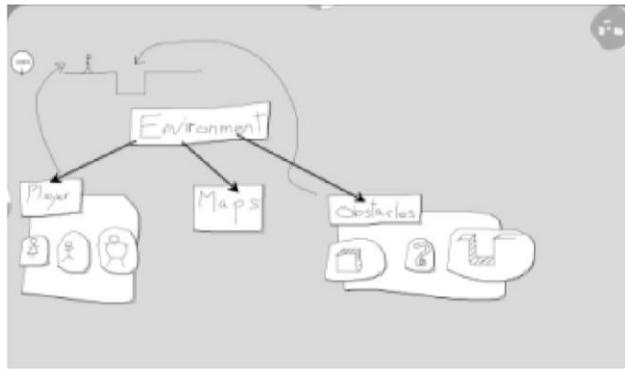


Figura 7. CALICO, Ampliación de características[5]

Conclusiones

Para muchas situaciones o proyectos de desarrollo de software, la simulación es el único medio para lograr una solución, permitiendo proyectar los resultados que el sistema debe realizar.

Todo el proceso descrito, nos brinda aproximaciones que permiten establecer los límites e identificar objetos relevantes y relaciones entre diferentes actividades, proyectos o procesos; comunicándonos la solución final y facilitando que los cambios se manejen rápidamente.

Conocer el proceso o las funcionalidades mediante abstracciones visuales presentado modelos y diseños cercanos a la realidad del cliente permite que los diseñadores no se desvíen del foco y se queden en actividades que no son la necesidad del cliente.

Al tener contacto o participación del cliente al inicio del proceso (presentando los modelos de la solución) producirá efectos beneficiosos sobre la calidad del diseño y del desarrollo en la etapa formal del ciclo de vida; obteniendo un alto grado de aceptación por parte del cliente.

La metodología propuesta orienta a que el desarrollo sea consecuente con la necesidad, realizando posibles cambios a tiempo y de manera ágil, proporcionando que la aplicación sea simple, funcional y estandarizada; haciendo que el proceso cumpla con el objetivo de ser evolutivo.

Trabajos futuros

Para posteriores investigaciones, se deberá evaluar el grado de complejidad de los proyectos sobre los cuales se aplicará el proceso simulación de casos de uso, prototipado y diseño ágil en la etapa inicial del ciclo de vida del desarrollo del software; con el propósito que las organizaciones puedan estimar la duración de esta etapa dentro del ciclo de vida, en acuerdo con el cliente y las áreas de tecnología.

En trabajos futuros relacionados con el sistema de simulación de casos de uso y manejo de prototipos, se puede implementar que el desarrollo aplique en diferentes plataformas como PDAs o teléfonos inteligentes al igual que el uso de pantallas táctiles lo cual facilitarían el manejo e interactividad con los desarrolladores y el cliente.

Considerar en los siguientes estudios para la elaboración en el nivel cero de los casos de uso, se permita con base en ellos armar en etapas posteriores los casos de prueba o Deck de pruebas, orientando al usuario a identificar y describir las acciones a probar en el sistema para obtener los resultados esperados validando y controlando las acciones o eventos.

Referencias

- BALSAMIQ M.2.1.19. Septiembre 2012. [en línea] Disponible en: [\[http://www.balsamiq.com/products/mockups\]](http://www.balsamiq.com/products/mockups).

- BROOKS R. J., Y TOBIAS A. M. Choosing the Best Model, Level of Detail, Complexity, and Model Performance, Simulación, 2012.
- CALDERÓN A., SAHARA D. Métodos ágiles, Universidad nacional de Trujillo Escuela de informática, Perú, 2007.
- CERIA S.: Casos de Uso un método práctico para explorar requerimientos, Universidad de Buenos Aires, 1998.
- GARCÍA M. Introducción a los casos de uso, Universidad de Sonora, México, Diciembre, 2004.
- IBM. Rational Software Modeler. Septiembre 2012. [en línea] Disponible en: [<http://ibm.com>].
- JAYARAMAN K. P., WHITTLE J., UCSIM: A Tool for Simulating Use Case Scenarios, Information & Software Engineering, George Mason University Fairfax, VA 22030, 2007.
- LÓPEZ O., LAGUNA M. A., MARQUES J. M. Generación automática de casos de uso para Desarrollo de software basado en reutilización, Instituto tecnológico de Costa Rica. Universidad de Valladolid, 2000.
- MANER W. Prototipado, Área de Ingeniería de proyectos, Departamento de Diseño y fabricación, Centro politécnico Superior, Universidad de Zaragoza, España, Traducido Febrero 2000.
- MANGANO N., BAKER A., y ANDRÉ VAN DER HOEK. A Prototype Sketching Tool for Modeling in Early Design, Universidad de California, Irvine, 2008.
- RICHTER, H., LOTHARM. Toward a Standard Process: The use of UML for Designing Simulation Models, Resumen Paper, 2012.
- TRIPODI G., ILLESCAS G. Construcción de modelos de simulación orientados a alentar la participación de usuarios finales utilizando UML, Universidad Nacional del centro de la provincia de Buenos Aires, Tandil, Argentina, 1999.
- Tutorial UML 2, Sparx Systems Pty Ltda., 2007.
- ZAPATA C. M., GONZÁLEZ G. Especificación formal en OCL de reglas de consistencia entre los diagramas de clases y casos de uso de UML y el modelo de interfaces, Revista Ingenierías Universidad de Medellín, Vol.7, no.12, p.169-191. 2008.