

Diseño de sistema para la simulación de metodologías de desarrollo de software

PAREDES-XOCHIHUA, María Petra*†, MORALES-ZAMORA, Vianney, LÓPEZ-MUÑOZ, Jesús y PEDRAZA-VARELA, Adriana.

Instituto Tecnológico de San Martín Texmelucan C.P. 74120 Puebla, Pue.

Recibido Enero 15, 2016; Aceptado Marzo 18, 2016

Resumen

Una metodología de desarrollo de software es aquella que indica los procesos se deben seguir para lograr el éxito en la creación de un sistema informático. La simulación es la acción de simular o bien representar algo experimentando o imitando ciertos aspectos de la realidad. El objetivo del presente es diseñar un sistema para la simulación de metodologías de desarrollo de software. El hecho de contemplar la metodología Scrum en esta etapa es porque considera varios procesos que otras metodologías también emplean y por lo tanto se podrán reutilizar varios aspectos de está. Además, por lo general para el desarrollo de proyectos cortos se adapta bien, ya que tiene bien determinados los periodos para cada una de las etapas que se contemplan en esta. Lo más importante es ofrecer una herramienta física y atractiva a estudiantes que en un futuro próximo se desempeñara en alguno de los roles que dictamina una metodología, permitiendo obtener la experiencia e identificando sus fortalezas y debilidades en cada sector.

Simulación, Metodología, Software, Planeación, Rubrica

Citación: PAREDES-XOCHIHUA, María Petra, MORALES-ZAMORA, Vianney, LÓPEZ-MUÑOZ, Jesús y PEDRAZA-VARELA, Adriana. Diseño de sistema para la simulación de metodologías de desarrollo de software. Revista de Sistemas Computacionales y TIC'S 2016, 2-3: 22-29

Abstract

A software development methodology is one that indicates the processes to be followed to achieve success in creating a computer system. The simulation is to simulate the action or represent something experiencing or imitating certain aspects of reality. The aim of this is to design a system for simulating software development methodologies. The fact the Scrum methodology contemplate at this stage is because it considers several processes that other methodologies used and therefore can be reused several aspects of this. Furthermore, generally for short development projects it is well suited, since it has well-defined periods for each of the stages that are contemplated by this. The most important thing is to offer students physical and attractive tool in the near future will perform in any of the roles that dictates a methodology, can benefit from this experience and identifying their strengths and weaknesses in each sector.

Simulation, Methodology, Software, Planning, Rubric

* Correspondencia al Autor (Correo Electrónico: petrypx@hotmail.com)

† Investigador contribuyendo como primer autor.

Introducción

Diseñar una herramienta didáctica que simule cada uno de los procesos que se realizan en la fabricación de software a la medida. La cual conceda que los estudiantes de las ingenierías que tengan relación con la de Sistemas Computacionales, pongan en práctica sus conocimientos teóricos sobre las metodologías ágiles de desarrollo de software y se enfrenten a cada una de las fases de manera real, desde las fases iniciales como los son: el levantamiento de requerimientos, captura de formatos, entre otras, hasta las últimas fases como la de implementación del sistema.

La cabina permitirá plantear el problema que se presenta y que por ende dará origen al desarrollo del sistema, la elección de la metodología por parte del líder del desarrollo del proyecto ya que está deberá ser la que mejor se adapte a las necesidades del proyecto y al tiempo que se tenga para el desarrollo del mismo. Dado que conforme a tiempos para cada fase se tendrán que ir registrando los entregables de cada una de estas, conforme a los roles que cada uno de los integrantes del equipo desempeña. La elección y asignación del rol será en base a un test para identificar en cual rol se desempeñaría mejor y se aplicara conforme a cada metodología. Así adquirirán experiencia en base a desempeñar y de vivir de manera real una o todas las fases del desarrollo de una metodología aplicada al desarrollo de software o sistemas de información.

Por lo tanto, los estudiantes podrán conocer cuál es el rol en el que se sienten con más capacidad para desempeñarse al egresar o bien identificar áreas de oportunidad a través de conocer sus fortalezas y debilidades al enfrentarse en el desarrollo de sistemas. Es necesario mencionar que en esta primera fase del proyecto se está trabajando con la metodología Scrum.

Para identificar el rol que desempeñara cada integrante del grupo se desarrolló un sistema experto que a través de un test se identifica qué rol es el más óptimo. La selección de los roles debe ser determinada por las habilidades de cada integrante las cuales deben llegar a la solución por el camino más corto para garantizar la mayor eficiencia en el funcionamiento.

Con este sistema se podrá tener una mayor precisión, rapidez, objetividad y confiabilidad en el proceso.

Revisión de literatura**Metodología SCRUM**

Scrum es un framework para trabajar en equipo en una serie de interacciones. Las fases en las que se divide y define un proceso de SCRUM son las siguientes:

1. El ¿Quién? y el ¿Qué?: identifica los roles de cada uno de los miembros del equipo y define su responsabilidad en el proyecto.
2. El ¿Dónde? y el ¿Cuándo?: que representan el Sprint.
3. El ¿Por qué? y el ¿Cómo?: representan las herramientas que utilizan los miembros de Scrum. (Lara, 2016).

Roles en Scrum – ¿Quién? y ¿Qué?

El equipo de Scrum consiste en tres diferentes roles, los cuales se pueden observar en la figura 1, y se describen a continuación:

Propietario del Producto (Product Owner): es el responsable de gestionar las necesidades que serán satisfechas por el proyecto y asegurar el valor del trabajo que el equipo lleva a cabo. Las actividades que realiza y aporta al equipo son:

- Identificación de necesidades o historias de usuario.
- Administrar y ordenar las necesidades.
- Autorizar el producto de software al finalizar cada iteración.
- Maximizar el retorno de inversión del proyecto. (Lara, 2016).

Scrum Master: es el responsable de asegurar que el equipo siga las bases de Scrum. Sus principales funciones son:

- Ayuda a que el equipo y la organización adopten Scrum.
- Liderar el equipo buscando la mejora en la productividad y calidad de los entregables.
- Ayudar a la autogestión del equipo.
- Gestiona e intenta resolver los impedimentos con los que el equipo se encuentra para cumplir con las tareas del proyecto. (Lara, 2016).

Equipo de desarrollo: El equipo está formado por los desarrolladores, que convertirán las necesidades del Product Owner en un conjunto de nuevas funcionalidades, modificaciones o incrementos del producto software final. El equipo de desarrollo tiene características especiales:

- Auto-gestionado: el mismo equipo supervisa su trabajo.

- Multifuncional: no existen compartimientos estancos o especialistas, cada integrante del equipo puede encargarse de tareas de programación, pruebas, despliegue, etc.
- No distribuidos: es conveniente que el equipo se encuentre en el mismo lugar físico.
- Tamaño óptimo: un equipo de desarrollo Scrum (sin tener en cuenta al ProductOwner y al Scrum Master) estaría compuesto por al menos tres personas.

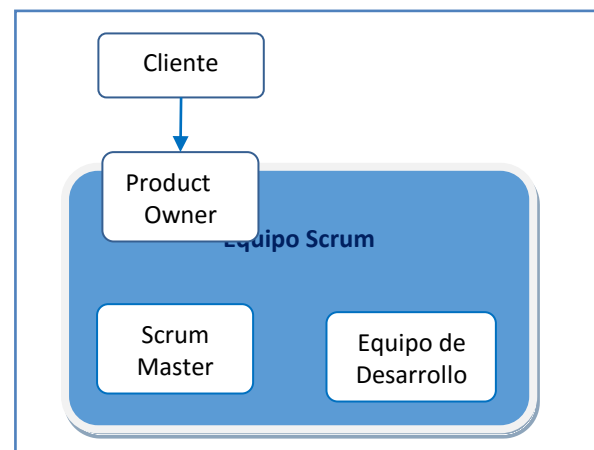


Figura 1 Equipo Scrum.

El Sprint – ¿Dónde? ¿Cuándo?

El Sprint es la unidad básica de trabajo para un equipo Scrum. Esta es la característica principal que marca la diferencia entre Scrum y otros modelos para el desarrollo ágil. Es una simple iteración llevada a cabo por los miembros del equipo. Un equipo puede completar varios sprints durante el desarrollo del proyecto.

Un Sprint inicia con un equipo que se compromete a realizar el trabajo y finaliza con la demostración de un entregable. El tiempo mínimo para un Sprint es de una semana y el máximo es de 4 semanas.

Durante el desarrollo de un Sprint se llevan a cabo ciertos eventos, estos reciben el nombre de Eventos Scrum. Estos son:

a. Reunión de Planificación del Sprint (Sprint Planning Meeting)

Todos los involucrados en el equipo se reúnen para planificar el Sprint. Durante este evento se decide qué requerimientos o tareas se le asignará a cada uno de los elementos del equipo. Cada integrante deberá asignar el tiempo que crea prudente para llevar a cabo sus requerimientos. De esta manera se define el tiempo de duración del Sprint. (Lara, 2016).

b. Scrum Diario (Daily Scrum)

Estas reuniones se deben realizar diariamente con un máximo de 15 minutos. Siempre en el mismo horario y lugar. En ellas, cada miembro del equipo deberá responder tres simples preguntas:

- ¿Qué hiciste ayer?
- ¿Qué tienes planeado hacer hoy?
- ¿Qué obstáculos encontraste en el camino?

Las reuniones sirven para que todos los miembros del equipo se apoyen entre ellos.

c. Refinamiento del Backlog

El Product Owner revisa cada uno de los elementos dentro del Product Backlog con el fin de esclarecer cualquier duda que pueda surgir por parte del equipo de desarrolladores. También sirve para volver a estimar el tiempo y esfuerzo dedicado a cada uno de los requerimientos.

d. Revisión del Sprint (Sprint Review)

Los miembros del equipo y los clientes se reúnen para mostrar el trabajo de desarrollo de software que se ha completado. Se hace una demostración de todos los requerimientos finalizados dentro del Sprint.

En este punto no es necesario que todos los miembros del equipo hablen. Pueden estar presentes pero la presentación está a cargo del Scrum Master y el Product Owner.

e. Retrospectiva del Sprint (Sprint Retrospective)

En este evento, el Product Owner se reúne con todo su equipo de trabajo y su Scrum Master para hablar sobre lo ocurrido durante el Sprint. Los puntos principales a tratar en esta reunión son:

- Qué se hizo mal durante el Sprint para poder mejorar el próximo
- Qué se hizo bien para seguir en la misma senda del éxito
- Qué inconvenientes se encontraron y no permitieron poder avanzar como se tenía planificado

Herramientas Scrum – ¿Por qué? ¿Cómo?

Para poder definir las respuestas a estas preguntas, se hace uso de ciertas herramientas que Scrum nos provee. Estas son:

a. Backlog de Producto

Esto puede referirse a todo elemento que sea parte del proyecto. Puede ser un bug, una referencia o parte de un requerimiento

b. Historias de Usuario

Es un elemento especial del Backlog de producto. Son llamados Historias porque en ellos se proporciona información sobre cómo debe ser el comportamiento del requerimiento que se está trabajando.

c. Backlog del Sprint

Es el conjunto de elementos tomados del Backlog del producto que fueron priorizados, medidos y aceptados en las reuniones de Sprint Planning.

d. Panel de Tareas

El panel de tareas muestra todas y cada una de las tareas que tienen asignadas cada uno de los miembros del equipo. Esta tabla se divide en tres columnas que representan el estado de la tarea:

- Por hacer
- Haciendo
- Terminado

Diseño del sistema

El sistema deberá cubrir los siguientes módulos como requerimientos (ver figura 2):

- Identificación del proceso de Scrum, ya que es necesario contar con una herramienta que identifique cada iteración, a la cual se le denomina Sprint.
- Simulación de eventos formales en Scrum, dado que esta prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación.

- Planificación de la iteración al proceso Scrum, la cual tiene 4 horas máximo donde el equipo elabora la lista de tareas de la iteración necesaria para desarrollar los requisitos a que se ha comprometido
- Registro de la Ejecución de cada iteración, en el que se tendrá que proporcionar un resultado completo, un incremento de producto que sea potencialmente entregable, de manera que cuando el cliente (Product Owner) lo solicite sólo sea necesario un esfuerzo mínimo para que el producto esté disponible para ser utilizado.
- Registro de los resultados revisión del Sprint.
- Registro de los resultados de aplicar la retrospectiva del Sprint.
- Identificación y asignación de roles.
- Simulación de las actividades del Product Owner. Dentro del sistema el superusuario o administrador simulará ser el dueño del producto, este dará aportaciones y la problemática al Scrum master, el superusuario tendrá la oportunidad de calificar al master Scrum. También podrá observar por medio de cámaras web las actitudes de todo el equipo.
- Simulación de las actividades del Scrum Master. En este módulo se debe proporcionar un ejemplo y asignarse entre sus diferentes desarrolladores asignándoles tareas en un determinado tiempo, realizar la evaluación de cada desarrollador por medio del rubricas.

- Registro de las actividades que el desarrollador notificara que ha concluido, para que se prosiga con su revisión.
- Chat, se desarrollara un chat que permita la comunicación instantánea entre el equipo Scrum. Y al finalizar cada sesión, se deberá guardar el respaldo en un archivo txt.
- Monitoreo del equipo Scrum, para observar las actitudes y comportamientos de cada desarrollador, para lo que se incorporaran cámaras web en cada equipo de cómputo.
- Rubricas de evaluación, serán diseñadas para calificar a cada integrante del equipo Scrum de acuerdo a su desempeño en cada actividad
- Visualizar los documentos Scrum:
- Crear las historias de usuario que se emplean en el Product backlog en Scrum
- Gráfica que mostrara públicamente la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint documento necesario Burn down en Scrum
- Crear el Sprint backlog es un documento detallado donde se describe cómo el equipo va a implementar los requisitos durante el siguiente sprint, en cuanto a las tareas se dividen en horas duración no superior a 16 horas.
- Registro de usuarios, que permita establecer los tipos de usuarios que ingresaran al sistema, y restricciones que se realizan.



Figura 2 Requerimientos del sistema

Resultados del diseño

El diseño del sistema se ha realizado considerando las necesidades que se presentan en las asignaturas de Ingeniería de Software y Desarrollo de Proyectos de Software del plan de estudios de las ingenierías en sistemas computacionales e Ingeniería en Tecnologías de la Información y Comunicación. Este podrá ser empleado por profesores y estudiantes de las ingenierías antes mencionadas, así como las afines.

De la figura 3 a la 8 se pueden observar algunas de las interfaces que se desarrollaran para el sistema, en estas se trata que sean usables para cualquier usuario.

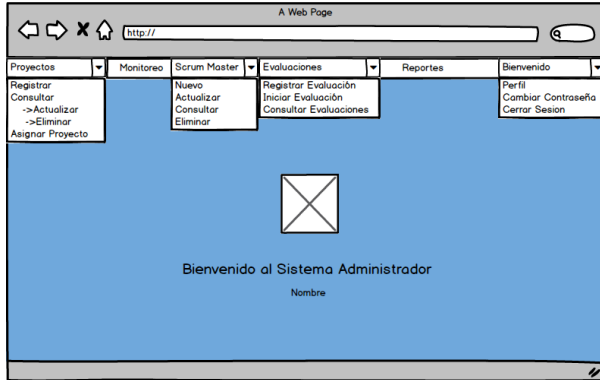


Figura 3 Pantalla de baja fidelidad de Bienvenida al Sistema

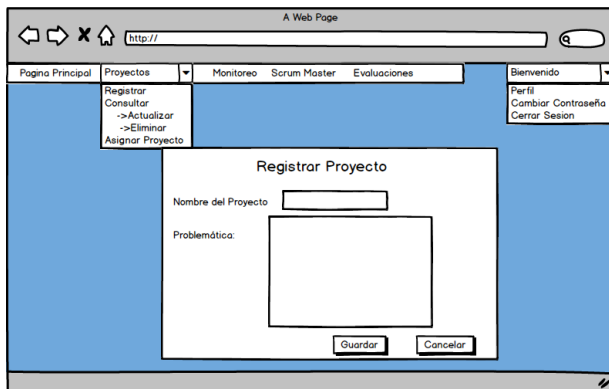


Figura 4 Pantalla de baja fidelidad de Registro de Proyecto

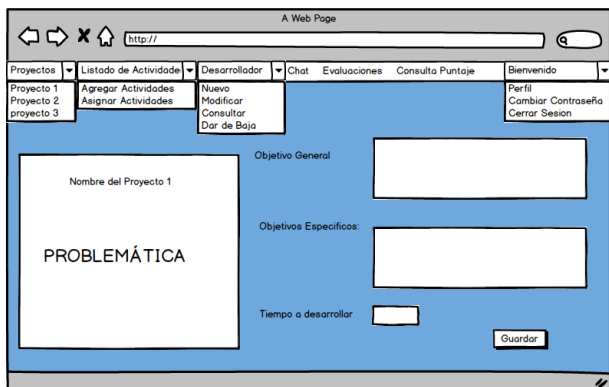


Figura 5 Pantalla de baja fidelidad de Objetivos del Sistema



Figura 6 Pantalla de baja fidelidad de Asignación de proyecto

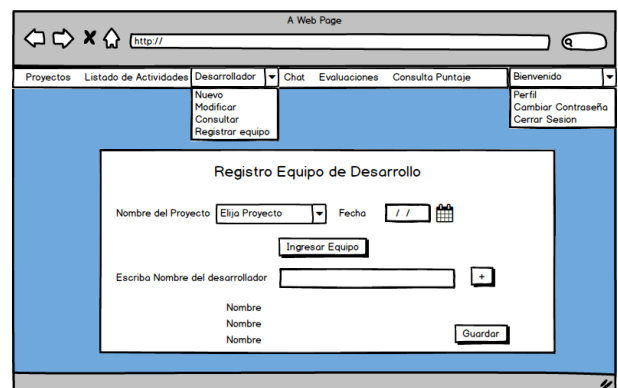


Figura 7 Pantalla de baja fidelidad de Registro Equipo de Desarrollo

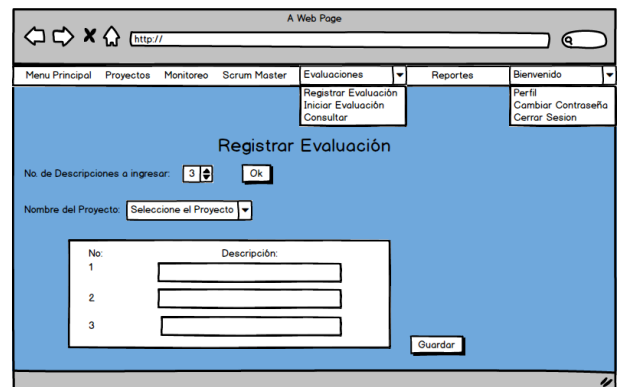


Figura 8 Pantalla de baja fidelidad de Registro Evaluación

Conclusiones

Derivado a que se presenta un bajo índice de inserción de los egresados de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico Superior de San Martín Texmelucan en los sectores de desarrollo de software y la demanda de software a la medida, que requiere el sector productivo de la región, se identificó que es necesario fortalecer el área de desarrollo de software mediante el uso de una metodología que sea aplicable de manera rápida en simulación de ambientes reales, que genere conocimiento empírico y que le permita al alumno poner en práctica sus habilidades, conocimientos y competencias obtenidas en el proceso de formación académica.

Existen metodologías para la formación del perfil profesional en el desarrollo de software, es por ello que para el diseño se consideró Scrum dado que es una metodología ágil, además que el número de integrantes en los equipos es reducido y por lo tanto es viable aplicarlo en los problemas reales y cortos durante un semestre. Así los estudiantes podrán tener la enseñanza a través de ejemplos reales, que permita identificar sus diferencias, ventajas, desventajas y beneficios de la aplicación.

Referencias

Coos Bu R. (1992). Simulación un enfoque práctico. México. LIMUSA.

Pressman, R.S. (2008). Ingeniería del Software un enfoque práctico. 6ª. Edición México: MC GRAW HILL.

Lara W. (01 de Junio de 2016) Cómo funciona la metodología de trabajo Scrum. Recuperado <https://platzi.com/blog/guia-Scrum/>

Godoy, Diego Alberto. (12 de Julio de 2016) Diseño de un Simulador Dinámico de Proyectos de Desarrollo de Software. Recuperado <http://sedici.unlp.edu.ar/handle/10915/44915>