

App Web Scrum

Javier Jesús León Silvestre

EPSEVG – Ingeniería Técnica en Informática de Gestión

Resumen

En el presente documento se expone la herramienta implementada para gestionar proyectos software basados en la metodología ágil Scrum.

Una metodología ágil se basa en la iteración e incrementación del desarrollo donde los requisitos evolucionan a lo largo del tiempo.

Una vez analizada la metodología Scrum, podemos observar que se compone básicamente de dos artefactos, tres roles y cuatro eventos.

Posteriormente, escogeremos la arquitectura y tecnologías que mejor se adapten para implementar la herramienta.

Finalmente, con los componentes analizados de Scrum y las tecnologías que utilizaremos, plantearemos los requerimientos de la solución para poder ser diseñada e implementada.

1. Introducción

El motivo por el cual se escogió este proyecto, nació a raíz de un par de cursos web. Ambos cursos hicieron una breve introducción a las metodologías ágiles y más concretamente a Scrum. Empecé a indagar y pude observar que es una metodología en auge y que funciona bastante bien en aplicaciones web y móvil.

Me pareció interesante hacer este tipo de propuesta sobre todo por dos motivos:

- Adquirir conocimiento sobre una metodología que puede ser de utilidad en un futuro.
- Ofrecer una herramienta de gestión para este tipo de proyectos como consecuencia de la fuerte demanda.

Los objetivos planteados que debe realizar la aplicación son:

- Gestionar los componentes de Scrum.
- Generar informes con los datos de los proyectos.
- Generar logs automatizados para garantizar un correcto funcionamiento.
- Generar emails con la información de los logs.
- Garantizar la seguridad de los datos mediante copias de seguridad automatizadas de la base de datos.

2. Conceptos básicos

Cualquier sistema informático tiene un ciclo de vida el cuál se compone de las siguientes:

- Análisis de los requerimientos.
- Diseño.

- Implementación.
- Pruebas.
- Mantenimiento.

Las metodologías del desarrollo del software ofrecen el marco de desarrollo para estructurar, planificar y controlar el proceso de desarrollo.

Cada metodología tiene su propia manera de plantear el desarrollo de software. En función del planteamiento podemos clasificar las metodologías como:

- Modelo en cascada: Framework lineal.
- Prototipado: Framework iterativo.
- Incremental: Combinación del Framework lineal e iterativo.
- Espiral: Combinación del Framework lineal e iterativo.
- RAD: Rapid Application Development. Framework iterativo.

El modelo en cascada, es el más básico de todos los modelos. La filosofía que plantea es que el desarrollo de todas las etapas del ciclo de vida se deben seguir secuencialmente.

Las metodologías ágiles, dependiendo del tipo, son una combinación entre prototipado, incremental, espiral y RAD.

Las metodologías ágiles se basan en el desarrollo de software en fases cortas. El software desarrollado en una unidad de tiempo se llama iteración.



Fig. 1. Tradicional vs. Agile

La filosofía de las metodologías ágiles valoran:

- A las personas por encima de las herramientas.
- Priorizan el software funcional a la documentación.
- Colaboración con el cliente en vez de contratos.
- Los cambios son bienvenidos en vez de seguir un plan.

Tenemos diferentes metodologías ágiles, vamos a ver un ejemplo de éstas:

- LEAN: desarrollo ágil orientado a conseguir exactamente lo que quiere el cliente. Se centra principalmente en la optimización de procesos.
- XP: desarrollo ágil principalmente para equipos pequeños. Se basa en implantar lo más rápidamente posible el software, eliminando la documentación y los procesos tradicionales.

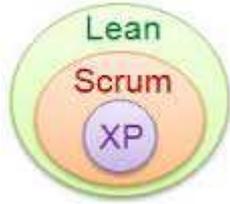


Fig. 2. Comparación metodologías ágiles

3. Scrum

Scrum es un marco de trabajo (Framework) para la gestión y desarrollo de software basado en un proceso iterativo e incremental utilizado en entornos de desarrollo ágil de software.

Se fundamenta en la teoría empírica de control de procesos. Esto significa:

- El resultado de los procesos no está definido a priori.
- El control del proceso se realiza mediante inspección y adaptación del proceso en base a los resultados obtenidos. Además, se fomenta la transparencia para deducir las mejoras de forma compartida.

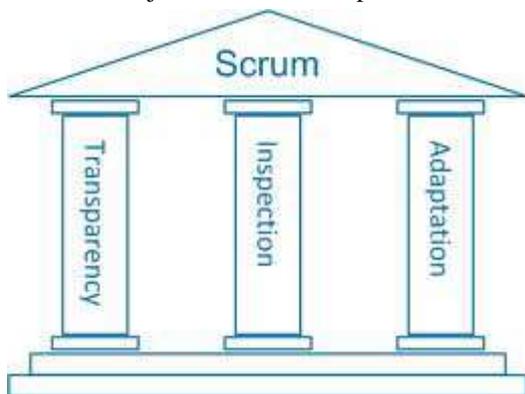


Fig. 3. Principios de Scrum

Scrum se compone de roles, eventos y artefactos. Veamos cada uno de ellos en detalle.

Scrum tiene los siguientes roles:

Product Owner

- Representa los intereses del cliente.
- Define los requisitos y marca las prioridades.
- Valida la implementación de los requisitos al final del sprint

Scrum Master

- Se asegura que Scrum se adapta correctamente.
- No es jefe de proyecto ni se encarga de los artefactos.

Equipo de desarrollo

- Se encargan de transformar las tareas de la pila del producto en software funcional a la finalización de cada sprint.
 - Son auto-organizados.
 - Son multifuncionales.
 - Son equipos pequeños, alrededor de 8 personas.
- Scrum Team

- La conjunción del Product Owner, Scrum Master y el equipo de desarrollo.

Una vez vistos los roles, veamos los eventos o reuniones que nos ofrece Scrum:

Sprint Planning

- Refinar la estimación de las historias asignadas al sprint.
- Comunicar el alcance de sprint al Product Owner.
- La duración está restringida a ocho horas para sprint de un mes.
- Se realiza al comienzo de cada sprint.

Daily Scrum

- En el mismo sitio, lugar y de pie.
- Duración de 15 minutos como máximo.
- El objetivo es sincronizar al equipo con tres preguntas básicas: ¿Qué hiciste ayer?, ¿Qué harás hoy?, ¿Qué problemas has tenido?

- Se realiza diariamente.

Sprint Review

- Intervienen el equipo de desarrollo y el Product Owner.
- El objetivo es revisar y validar el resultado del sprint.
- Duración de cuatro horas para sprint de un mes.
- Se realiza al final de cada sprint.

Sprint Retrospective

- El objetivo es que el equipo de desarrollo se inspeccione a sí mismo con el fin de obtener unas mejoras para el siguiente sprint.
- Duración entre tres y cuatro horas para sprint de un mes.

- Se realiza entre el Sprint Review y el Sprint Planning.

Finalmente, tenemos los artefactos de Scrum. Los artefactos de Scrum representan los documentos que aportan valor de negocio y proporcionan transparencia, inspección y adaptación. Hay dos únicos artefactos:

- Product Backlog: es una lista ordenada por valor de negocio de todas las funcionalidades que podrían ser necesarias para realizar el producto. Varía en función del tiempo, es decir, se pueden incorporar o desestimar funcionalidades según el Product Owner crea conveniente.
- Sprint Backlog: es un conjunto del Product Backlog seleccionado para el sprint, es decir, un subconjunto del Product Backlog que se ejecuta en cada iteración. El “dueño” es el equipo de desarrollo. Proporciona transparencia para todo el equipo. Sólo el equipo puede modificar el contenido del Sprint Backlog. En este

punto, es donde se analizan las historias de usuario y se descomponen en tareas.

El concepto de incremento se puede definir como la suma de todos los elementos del Product Backlog completados durante un sprint y durante todos los sprints previos. Al final de cada sprint, la nueva parte del incremento debe estar calificada como “Hecho”, lo cual significa que está validada para añadir las funcionalidades al ProductBacklog.

El concepto de “Hecho” varía según cada equipo. Son las condiciones que establece el equipo para dar el trabajo por válido.

Además de los artefactos de Scrum, se utilizan otras herramientas de control, que no son propias de Scrum, pero si ayudan a aplicar Scrum de forma más eficaz. Entre estas herramientas cabe destacar la pizarra Kanban, tiene el objetivo de obtener una visión global del trabajo que se está realizando diariamente y sirve como punto de unión del Scrum Team. Consiste en una pizarra de tres columnas, existen más variantes, que indica las tareas que se van a realizar, las que se están realizando y las que se han realizado.

Por otra parte, dos herramientas que se utilizan de forma habitual en Scrum son las gráficas de trabajo pendiente (Burndown charts). El objetivo de éstas es representar el progreso actual del equipo frente a un progreso teórico, son generadas a nivel de Product Backlog y de Sprint Backlog, de esta forma el equipo sabe en todo momento si se encuentra por encima o por debajo del trabajo previsto y tomar decisiones sobre qué desarrollar en cada momento.

Finalmente tenemos las historias de usuario, utilizadas en metodologías ágiles para describir las funcionalidades que serán útiles para el usuario, en este caso el Product Owner. Se describen en lenguaje de negocio, no técnico, y se basan en el principio I.N.V.E.S.T. lo que quiere decir que deben ser lo más independientes entre éstas, negociables para poder ser debatidas, valoradas por el Product Owner, estimables, pequeñas para poder ser estimadas con cierta precisión y validez, y verificables para acordar unos criterios de aceptación que verifican la funcionalidad esperada.

4. Trabajos relacionados

Se ha realizado un estudio sobre qué tipos de proyectos están relacionados con Scrum. Básicamente se pueden catalogar en tres tipos de proyectos.

El primer tipo aplica la metodología Scrum sobre proyectos de software, principalmente en entornos web. Es el más común de encontrar aunque en algunos casos no siguen las reglas que marca Scrum ni aparecen todos los componentes que debieran. Se puede decir que aplican Scrum parcialmente.

En el segundo tipo, se realiza un estudio teórico de metodologías software incluyendo Scrum. Analizan exhaustivamente las diferentes metodologías a diferentes niveles, y en ocasiones, se realizan una práctica teórica para cada metodología

Por último, utilizan Scrum para la implantación de una metodología a nivel de institución. Tiene el objetivo de definir una metodología de trabajo basada en Scrum para

mejorar el rendimiento, eficiencia y satisfacción tanto de los equipos de desarrollo como de los clientes.

5. Mi Propuesta

Antes de realizar la aplicación es necesario evaluar y justificar la metodología, la arquitectura y las tecnologías que se van a utilizar para llevar a cabo dicha herramienta.

Desde el punto de vista metodológico, se puede pensar a priori que siendo un proyecto que trata sobre la metodología Scrum se le aplicará la misma metodología. Además, como anteriormente se ha citado, Scrum funciona correctamente con aplicaciones que varían a lo largo del tiempo como pueden ser aplicaciones de entorno web. Dicho esto, la razón por la cual no se ha adoptado esta metodología es la ausencia de roles y eventos que carece este proyecto, es decir, no hay un equipo de desarrollo y no hay las reuniones de Scrum. Por lo tanto, se ha aplicado una metodología en cascada para poder llevar a cabo el susodicho.

A nivel de arquitectura, se ha adoptado una arquitectura MVC (Modelo Vista Controlador). Es una arquitectura bastante recomendable para aplicaciones web gracias a que separa la lógica de negocio de la vista. Como la vista es susceptible a varios cambios, ya sea por decisión de un cliente o porque la tecnología evoluciona rápidamente, nos aseguramos que, al tener una aplicación totalmente desacoplada, basta con modificar la vista sin que afecte al modelo de la aplicación. Otro valor añadido es el de poder hacer pruebas unitarias ya que este tipo de arquitectura separa las responsabilidades y facilita este tipo de pruebas.

A nivel de tecnología, en cuanto a la capa de presentación se refiere, se ha designado la conjunción de HTML5, CSS3 y JavaScript porque ofrecen una usabilidad e interacción con el usuario excelentes, además son soportadas independientemente del dispositivo que las utilice ya sea un ordenador, un móvil o tablet. Desde el lado del servidor, se ha optado por implementar la lógica sobre el Framework .NET puesto que la plataforma está orientada a la productividad aunque en su momento se barajó la posibilidad de implementarlo sobre Java. Como servidor de datos, se ha utilizado para persistir los datos SQLServer probablemente uno de los dos servidores más potentes que existen hoy en día. Como se ha citado anteriormente, la aplicación realizará seguimientos mediante la generación de logs, el Framework de Apache denominado Log4NET será el encargado de realizar todo el trabajo una vez configurado. Finalmente para la generación de informes, se ha utilizado la herramienta CrystalReports de SAP, una herramienta muy potente y compatible con el Framework de .NET.

6. El Framework .NET

El Framework de .NET se basa en el CLR (Common Language Runtime), éste es el núcleo de todo el Framework y básicamente permite la ejecución de las aplicaciones implementadas de .NET.

Una aplicación nunca se ejecuta directamente sobre el procesador, sino que se ejecuta sobre este procesador virtual el CLR. Además el mismo Framework nos ofrece de manera nativa un conjunto de librerías ya implementadas que podemos utilizar para desarrollar las aplicaciones.

El proceso de compilación se realiza en Just-In-Time, es decir, que el CLR compila sobre la marcha el código generado por la aplicación (código intermedio MSIL) y lo traduce en instrucciones nativas del procesador, prácticamente igual que Java.

Para poder realizar la aplicación, se ha utilizado parte del Framework que nos ofrece .NET para implementar la herramienta web, esta parte del Framework que se utiliza para entornos web es denominado ASP.NET.

De las diferentes opciones que nos ofrece el Framework ASP.NET, se ha utilizado la parte conocida como MVC. Esta parte, aunque tiene la curva de aprendizaje más elevada, es la que nos ofrece un mayor control sobre la aplicación.

Básicamente consiste en una tabla de ruteo. Esta tabla está asociada a los controladores que reciben la petición, los controladores invocan al modelo y una vez realizada la operación, éstos renderizan la vista que tienen asociada.

7. Framework Log4NET

Es un componente que nos ofrece Apache. Permite llevar a cabo el proceso de seguimiento de una aplicación de manera fácil. Es open-source y también está disponible para Java y PHP por ejemplo.

Únicamente tiene tres componentes:

- **Loggers:** generan los mensajes de error.
- **Appenders:** son los destinatarios de los mensajes de log generador por un logger.
- **Layout:** Estilo utilizado para mostrar los mensajes de log en un appender específico.

8. Trabajo Futuro

En cualquier aplicación siempre se pueden incorporar mejoras.

Como posibles mejoras que puedan añadirse a la herramienta tenemos la autenticación y autorización de usuarios. De esta manera, podemos limitar o restringir el acceso a la aplicación. Por otra parte, sería interesante agregar las gráficas de trabajo pendiente para poder tener un mayor control y predicción sobre los proyectos. Y finalmente, para mejorar la usabilidad y aumentar la transparencia, se podría incorporar una pizarra Kanban para tener una visión general del estado en que se encuentran las tareas.

9. Conclusiones

Scrum es muy adecuado en desarrollos internos de producto donde existe confianza entre cliente y proveedor, no se requieren grandes inversiones iniciales y donde existe un equipo con unos objetivos comunes. Los entornos idóneos para aplicar Scrum son entornos muy volátiles y que siempre requieren variar los requisitos, principalmente, entornos web y móvil. No es recomendable aplicar dicha metodología en proyectos que establecen una relación contractual entre el cliente y proveedor. También donde el plazo, alcance y recursos están fijados de antemano, requieren una inversión inicial elevada y diferentes equipos con objetivos propios.

Desde el punto de vista personal, ha sido una experiencia muy positiva. Primero por los objetivos cumplidos y segundo por el aprendizaje que se ha adquirido.

10. Agradecimientos

Me gustaría agradecer este proyecto a mi directora Àngels Hernández que me ha guiado durante el proyecto y a todos los profesores, tanto de la universidad como de los cursos que he realizado por ofrecerme todos los conocimientos necesarios que han sido de gran utilidad para realizar este proyecto.

Bibliografía

- [1] *CODE PROJECT* www.codeproject.com
- [2] *STACK OVERFLOW* www.stackoverflow.com
- [3] *MSDN LIBRARY* www.msdn.microsoft.com
- [4] *C# CORNER* www.c-sharpcorner.com
- [5] *VIDEO TUTORIAL YOUTUBE* www.youtube.com
- [6] *ONLINE WEBS TUTORIAL* www.w3schools.com
- [7] *JQUERYUI* www.jqueryui.com
- [8] *SLIDE HOSTING SERVICE* www.slideshare.com
- [9] *WIKIPEDIA* www.wikipedia.org
- [10] *SCRUM ORG* www.scrum.org
- [11] *Scrum Guide, Ken Schwaber, Jeff Sutherland, 2013*
- [12] *Curso Introducción Scrum, Bcelona Activa, 2014*
- [13] *CRYSTAL REPORTS* www.crystalreports.com
- [14] *Curso MVC, Jordi Ariño, PUE, 2014*
- [15] *Curso WebForms, Jordi Ariño, 2013*
- [16] *Proyecciones, J.M. Merenciano, EPSEVG, 2013*
- [17] *EXPRESIONES REGULARES* www.regular-expressions.info
- [18] *CONNECTIONSTRINGS* www.connectionstrings.com
- [19] *ENTITYFRAMEWORK TUTORIAL* www.entityframeworktutorial.net
- [20] *APACHE LOGGING SERVICE* <http://logging.apache.org/log4net>
- [21] *Curso base de datos, Carles Romero, PUE, 2014*
- [22] *Herramienta de auditoría automática de los procesos de una organización, Ricardo Meana de la Llave, Pamplona, 2010*
- [23] *Implantación de la metodología ágil Scrum en el mantenimiento de sistemas CRM, Jordi Moreno, UAB, 2013*
- [24] *Automatización de sistemas de desarrollo ágil, Héctor Mudarra, UAB, 2010*
- [25] *Scrum Manager Gestión de Proyectos, Juan Palacio, Safe Creative, 2011*
- [26] *Aplicación de gestión (ERP) de comedores en centros educativos, Arnau Timoneda, EPSEVG, 2011*