

La masa, el ladrillo, la bota, el bocadillo...

Los pensamientos y peleas de Rodrigo Corral con el desarrollo de software

Exprimiendo Scrum: Scrum y la gestión de requisitos

productbacklog



Tradicionalmente se ha tendido a tratar la captura y la gestión de requisitos de una manera que, cada vez más, se demuestra errónea. Se ha entendido la captura de los requisitos del proyecto como una fase temprana del mismo que una vez se completaba nos daba la fotografía exacta de que necesitaba nuestro cliente. Luego la única labor del gestor del proyecto era tratar de evitar que se produjesen cambios en el conjunto de requisitos y tratar que, cuando estos se producían el cliente asumiese el coste económico de los mismos. Supongo que a todos os suena este planteamiento.

El problema que todos hemos vivido es que **el esfuerzo que supone hacer una captura detallada de todos los requisitos de un proyecto es enorme**. Tan enorme que rara vez justifica el resultado. Además otra realidad que hemos

descubierto es que **casi nunca nuestro cliente conoce sus propias necesidades** con la profundidad suficiente como para definir las a priori y a esto se une que, a menudo, durante la vida del proyecto, las necesidades y prioridades de nuestros clientes cambian. Además nuestros clientes a menudo necesitan ver lo que somos capaces de hacer y como resolvemos sus necesidades a nivel técnico para poder descubrir nuevas necesidades u oportunidades que desconocían que éramos capaces de implementar. Podemos tratar de protegernos de estos cambios estableciendo mecanismos de control, pero este enfoque a menudo nos lleva a clientes poco satisfechos, que verán el desarrollo del proyecto como algo rígido que no se adapta a sus necesidades y que si lo hace es repercutiendo costes añadidos al presupuesto del proyecto.

Vistos estos problemas, ¿qué aproximación propone Scrum?.

En Scrum los requisitos se expresan como elementos del Product Backlog. **El Product Backlog es una lista viva de requisitos funcionales y no funcionales priorizados por su valor para el cliente**. Al decir que se trata de una lista viva, dejamos claro que los requisitos que en ella aparecen y el orden de los mismos es cambiante a lo largo de la vida del proyecto. En Scrum, los requisitos se van abordando en Sprints en el orden en que aparecen en el Product Backlog.

Utilizar el Product Backlog como principal artefacto para la gestión de requisitos nos permite atajar los problemas relacionados con la gestión de requisitos que antes hemos descrito.

Aunque **construir y mantener el Product Backlog es una tarea ardua**, es mucho más simple que hacer una captura de requisitos tradicional. El motivo es simple, solo hay que expresar a grandes rasgos en que consiste el requisito, no es necesario un nivel de detalle muy elevado. Solo es necesario el nivel de detalle suficiente que nos permita estimar los requisitos y priorizarlos. A menudo se utilizan historias de usuario para expresar estos requisitos. **Los requisitos que aparecen en el Product Backlog deben ser independientes, negociables, evaluables, estimables y no demasiado grandes**. Deben ser independientes pues el orden en el que serán implementados puede cambiar. Deben

ser negociables en el sentido de que son un punto de partida para comenzar en desarrollo no un contrato cerrado. Deben ser evaluables desde el punto de vista del retorno de la inversión que proporcionan a nuestros clientes. Deben ser estimables pues es imposible priorizar algo de lo que se desconoce la magnitud. Y, por último, deben ser de un tamaño que permita estimarlos sin tener demasiadas incertidumbres sobre cuál es el alcance concreto del requisito, aunque cierto nivel de incertidumbre siempre va a existir.

Evidentemente **antes de su implementación será necesario refinar en mayor detalle los requisitos**. En Scrum esto es algo que posponemos hasta que planeamos el Sprint en el que vamos a abordar esos requisitos en concreto. Posponer el refinado detallado de los requisitos hasta un momento cercana a su implementación nos permite que cuando realizamos esta actividad tengamos en cuenta de nuevo las necesidades del cliente que pueden haber cambiado y contar con la información que hemos ganado durante la implementación de otros requisitos, de esta manera, al contar con más información tendremos muchas más posibilidades de actuar correctamente a la hora de describir y estimar en detalle los requisitos. El fundamento para lo anteriormente expuesto es claro, es posible definir y estimar las actividades de nuestro proyecto cercanas en el tiempo con un buen margen de confianza, pero es muy complejo hacerlo con actividades cuyo momento de comienzo se encuentra lejano en el tiempo. **La actividades cercanas en el tiempo tienen una probabilidad mucho menor de sufrir cambios en su alcance que obliguen a cambiar su estimación y, en consecuencia, su planificación.** No me extendo más en esto pero si os invito a leer algunos de mis anteriores post sobre estimación: [El difícil problema de la estimación](#), [Estimación y planificación detallada](#), [Recibiendo estimaciones](#) y [Estimando \(no practicando la brujería\)](#).

Los cambios en los requisitos ocurren, por mucho que tratemos de combatirlos. En Scrum y en las metodologías ágiles en general, asumimos que estos cambios van a ocurrir y tratamos de gestionar estos cambios como oportunidades de que los clientes obtengan lo que necesitan. Evidentemente para que esto funcione la gestión del Product Backlog debe reaccionar de manera continua para reflejar las prioridades de nuestro cliente. El Product Owner (propietario del producto) es el encargado de gestionar el Product Backlog, introduciendo nuevos requisitos según se descubren y priorizándolos en función de las necesidades del cliente y el retorno de la inversión que el cliente recibirá de su implementación.

Utilizando el Product Backlog, cada vez que el cliente cambia sus prioridades o el alcance de un requisitos ve de manera explícita y visual (recordemos que se trata de una lista ordenada) que otros requisitos ven afectada su prioridad, de manera que serán implementados más tarde o incluso quedarán fuera del alcance del proyecto si este está establecido de antemano, como ocurrirá en los proyectos con precio o fecha cerrados. La clave está en que en cualquier caso, el cliente irá consiguiendo de manera paulatina que estén implementadas aquellas partes de proyecto que más valor le aportan y contará con la seguridad de que en el caso de que algo quede fuera del ámbito del proyecto serán aquellos requisitos que menor valor tienen para el.

Vemos como el Product Backlog nos permite gestionar de manera ágil y sencilla los requisitos de nuestro proyecto, refinándolos cuando tenemos información suficiente, priorizándolos según el valor que aportan a nuestro cliente.



📅 12 noviembre, 2007 👤 Rodrigo Corral González 📁 Sin categoría

11 comentarios sobre “Exprimiendo Scrum: Scrum y la gestión de requisitos”

 **anonymous**

13 noviembre, 2007 a las 4:22 am

Es realmente interesante el Product Backlog y su funcionamiento. La experiencia me dice que en todos los proyectos los requisitos cambian, evolucionan, modifican sus prioridades, se extinguen o se crean durante la vida del desarrollo. Esto ocurre por todas las razones que aparecen en el primer párrafo de tu artículo. Estoy totalmente de acuerdo con eso

Lidiar con esto no es nada fácil, sobre todo desde el punto de vista económico. Adaptar un proyecto que cuenta con un

buen análisis a nuevos requisitos o requisitos modificados no es cómodo, pero es mucho más cómodo que presupuestar un software que a lo largo de su vida de desarrollo sufrirá todos los cambios que hemos mencionado.

Me interesaría saber tu opinión sobre cómo aplicar la filosofía Scrum de captación de requisitos, que es muy ágil y pegada a la realidad a nivel funcional, sin embargo, no veo cómo puede aplicarse en un mundo en el que es necesario dar un presupuesto al inicio del proyecto y cualquier desviación del mismo será muy mal recibida por el cliente. Esto último es la realidad del día a día, tener que renegociar partes de un proyecto porque han cambiado, se incluyeron funcionalidades, etc...

Si la respuesta es que inicialmente no presupuestemos un proyecto, sino que presupuestemos los primeros 2 Sprints y luego veremos el resto a medida que se refinan y se vayan abordando, me parece una idea algo inaplicable, pues un cliente siempre necesita saber cuánto le va a costar lo que pide y en cuánto tiempo. No creo que le convenza mucho que le digamos que es un presupuesto abierto.

Me gustaría saber tu opinión al respecto de este particular.

Saludos.

 **anonymous**

13 noviembre, 2007 a las 10:55 am

Navegante, no en todas las situaciones se puede utilizar Scrum, no es válido para cualquier necesidad de desarrollo.

En el caso que planteas, cuando es necesario presentar un presupuesto cerrado, es mejor utilizar el método «tradicional» de intentar analizar todo al inicio.

Scrum suele funcionar muy bien para desarrollos internos y para productos «empaquetados», pero no tan bien para desarrollos a medida donde haya que cerrar precio por anticipado.

He trabajado para clientes que aceptaron pagar por horas, a cambio de la flexibilidad que les proporcionaba no tener que decidir todo lo que necesitaban al principio. Pero es complicado que lo acepten.

 **anonymous**

13 noviembre, 2007 a las 12:11 pm

José Alberto, estoy de acuerdo con tu opinión. Pensaba justo que una aproximación así podría funcionar muy bien cuando el desarrollo es interno con el equipo de desarrollo en plantilla.

Se me antoja complicado aplicar este enfoque para un cliente tradicional en un escenario tradicional proveedor-cliente.

Lo de cobrar por horas a cambio de la flexibilidad es una buena alternativa, en algunos casos será aceptada dependiendo de la flexibilidad del cliente también, me parece interesante, también he usado ese método alguna vez.

Gracias José Alberto

 **rcorral** 

13 noviembre, 2007 a las 8:36 pm

Scrum no está para nada reñido con las ofertas a precio cerrado. Todo lo contrario. Existen numerosísimas experiencias que así lo atestiguan.


El problema de las ofertas a precio cerrado es un problema en primera instancia de estimación y en segunda instancia de comunicación proveedor-cliente, pero no es un problema que convenga abordar desde la gestión severa del cambio en los requisitos, que es el enfoque tradicional.

Si tratamos de limitar la capacidad de nuestros clientes para cambiar los requisitos, estaremos limitando sus posibilidades para conseguir el resultado que necesitan. Y en última instancia tendremos un proyecto que termina en tiempo y presupuesto, pero que no le sirve al cliente, con la consecuencia de que en última instancia tendremos un cliente insatisfecho.

En lugar de esto, siempre podemos establecer una colaboración y una comunicación clara con nuestros clientes y dejar que cambien los requisitos haciendo ver claramente que todo cambio tiene un impacto. Usando el Product Backlog, este impacto se hace evidente y esta comunicación es mucho más sencilla.

Evidentemente no con todos los clientes es esto posible, pero si que es posible con una gran mayoría de ellos, con muchos más de los que a priori prodría parecer posible.

Un saludo y gracias por los comentarios!!!

 **quintas**

14 noviembre, 2007 a las 9:18 am

Rodrigo, un artículo estupendo. Bien sabes que yo he sufrido como el que más el análisis de requisitos y lo tedioso de su captura, empezando por encontrar un o varios interlocutores válido en la empresa.

La aproximación de SCRUM es muy realista y práctica. El cliente siempre va a querer cambios y no suelen ser caprichos sobre todo porque en un proyecto mediano o grande su empresa o modelo de negocio cambia mientras desarrollas la solución.

El único problema (que pasa con cualquier otra metodología) es lidiar con la picaresca del cliente (muy típica en este país) de incluir nuevas funcionalidades que afectan estructuralmente a todo el proyecto (no es difícil y por mucho que hayamos abstraído la solución a veces no es suficiente). Entonces entra el gran problema de estimar y decirle al cliente que vale, pero aumentaría los costes y los tiempos del proyecto y ahí es cuando comienzan todos los problemas de confianza.

Pero creo que merece la pena adoptar SCRUM para la mayor parte de proyectos software. Quizás la única condición necesaria es que el cliente se involucre en el proceso y en el seguimiento.

Un abrazo.

 **anonymous**

20 noviembre, 2007 a las 8:42 pm

Estoy de acuerdo con muchas de las personas que indicaron que es un poco complicado el no dar un precio estimado al cliente cuando se esta por empezar a desarrollar un proyecto, además de que es muy complicado involucrar al cliente cuando este no esta en la misma empresa en la que se desarrollara la aplicación, pero definitivamente, es conveniente utilizar esta metodología para desarrollar agilmente software, pero por experiencia me parece que es aplicable solo para proyectos de una institucion para la misma institucion y no asi para clientes externos, ademas de estimar el precio inicial de la aplicacion.

 **anonymous**

8 febrero, 2008 a las 10:57 pm

El artículo que propones me resulta muy interesante, pero quisiera preguntarte ¿cómo insertarías o aplicarías la Ingeniería de Requisitos como disciplina en Scrum?

 **rcorral** 

9 febrero, 2008 a las 9:00 pm

Frank, precisamente es lo que explico en este post. El product backlog es el mecanismo de gestión de requisitos en Scrum. En cuanto a la captura de requisitos las técnicas son las habituales: entrevistas con el cliente, JAD, observación en la sombra, encuestas a usuarios etc... Y sobre la especificación de requisitos, cada vez se utilizan más las historias de usuario como aproximación a la especificación ágil de requisitos.

 **anonymous**

12 febrero, 2008 a las 12:11 am

En el post haces referencia al concepto «Historias de Usuario» como aproximación a la especificación ágil de requisitos. ¿Qué entender por «Historias de Usuario»? y como se insertan en una metodología ágil como Scrum.

Scrum presenta 3 artefactos principales:

Product Backlog, Sprint Backlog y Reportes. La gestión de requisitos sólo se presenta en el primero de los artefactos o existe alguna entre todos que contribuya a la Gestión de los Requisitos. Quisiera que la respuesta la dieras desde el punto de vista de la Ingeniería de Requisitos como disciplina de la Ingeniería de Software.

Saludos

 **anonymous**

18 febrero, 2008 a las 4:53 am

Lo cierto es que las metodologías ágiles han ganado en demasía en el mundo de la industria del software. Para el caso del proceso ingenieril de captura de requisitos, existe una realidad que favorece este proceso en las metodologías livianas como SCRUM y es el caso de la tendencia cada vez mayor de un acercamiento de los clientes al equipo que desarrolla el producto que ellos mismo van a emplear. Sin embargo, es interesante el caso en que los clientes, ya sean por razones de políticas de la institución que desarrolla el software o por las que rigen la dinámica de trabajo del cliente, no puedan estar en frecuente comunicación con el equipo de desarrollo. Para estos casos, creo que es preciso combinar el método “tradicional” de identificación de los requerimientos y el propio método ágil. Es decir, de alguna u otra manera no perder la posibilidad de darle al cliente la oportunidad de ajustar en tiempo de desarrollo, sus requisitos del software que él mismo va a utilizar. Esto, entre otras cosas permitiría un mayor grado de satisfacción en el cliente y seguridad para el equipo de desarrollo, en cuanto a lo que se está construyendo, si no es lo más exacto al menos lo más próximo a lo que el usuario realmente quiere, cuestión de marcada problemática en el mundo del desarrollo de software.

 **anonymous**

18 enero, 2010 a las 8:30 am

Juan Palacio , David Alfaro y un servidor grabamos un podcast este pasado sábado sobre el papel que en
