Scrum

El proceso de desarrollo de software Scrum para equipos pequeños

En el entorno actual de desarrollo de software, los requisitos a menudo cambian durante el ciclo de vida del desarrollo del producto para satisfacer las demandas cambiantes del negocio, creando dolores de cabeza interminables para los equipos de desarrollo. Discutimos nuestra experiencia en la implementación del proceso de desarrollo de software de Scrum para abordar estas preocupaciones.

En AG Communication Systems, los equipos de desarrollo de software varían en tamaño de dos a varios cientos de individuos. Intuitivamente, el proceso de desarrollo apropiado para equipos muy grandes no funcionará bien para equipos pequeños y viceversa. En nuestra organización, diversidad de procesos significa adoptar un enfoque flexible de los procesos de desarrollo para que cada equipo pueda aplicar lo que funciona mejor. Al experimentar con el proceso de desarrollo de software de Scrum, encontramos que los equipos pequeños pueden ser flexibles y adaptables para definir y aplicar una variante apropiada de Scrum. Este artículo describe nuestra experiencia en la implementación de este proceso.

¿Por qué Scrum?

Como miembros del Grupo de Tecnología de Software, nuestro grupo es responsable de introducir nuevas tecnologías y procesos en nuestra organización en AG Communication Systems en Phoenix, Arizona. Investigamos nuevos enfoques y patrocinamos su introducción y crecimiento. También realizamos chequeos de proyectos de desarrollo para proyectos en curso y postmortem para los terminados.

En nuestros postmórmenes y chequeos periódicos, hemos notado algunos problemas recurrentes para nuestros proyectos de Oportunidad de Nuevos Negocios (NBO). La Figura 1 enumera algunos comentarios surgidos de proyectos en nuestra organización que enfrentaron desafíos significativos.

En el nuevo mercado de telecomunicaciones donde opera nuestra compañía, el cambio es abrumador. Los desarrolladores de software siempre se han quejado de cambios en los requisitos, pero en los enfoques tradicionales supusieron que entenderían los requisitos antes de pasar a la siguiente fase. En el entorno actual, sin embargo, los requisitos del proyecto pueden ser poco claros o desconocidos, incluso a medida que el proyecto se pone en marcha. De hecho, es posible que el mercado no esté definido, incluso podría ser que nadie entienda claramente el producto en desarrollo.

La mayoría de los equipos de desarrollo responden con "¡Haz que el caos se vaya! ¡Denos mejores requisitos! "Desafortunadamente o no, el caos es la realidad en este nuevo entorno de negocios, ya que los desarrolladores de software debemos aprender a satisfacer las necesidades del cliente y convertir este caos en nuestra ventaja. Al ver que algunos de nuestros proyectos NBO fueron más exitosos que otros, nos esforzamos por examinar los datos post mortem para aprender los secretos de esos proyectos exitosos. Estos son algunos comentarios de equipos exitosos:

■ Hicimos la primera pieza y luego la reestimamos, ¡aprende a medida que avanzas!

■ Hicimos una breve reunión diaria. Sólo los que tenían una necesidad asistieron.

■ El documento de requisitos era de alto nivel y estaba abierto a la interpretación, pero siempre podíamos reunirnos con el ingeniero de sistemas cuando necesitábamos ayuda.

Alrededor de este tiempo, descubrimos los sitios Web de Scrum (www.controlchaos.com/safe y www.jeffsutherland.org) y leímos un documento presentando patrones para usar el proceso de desarrollo de software de Scrum de la Conferencia 1998 del Lenguaje de Patrones de Programas. Desde el

En los sitios web de Scrum aprendimos que Scrum es un proceso para la creación incremental de software en entornos complejos. Scrum proporciona controles empíricos que permiten que el desarrollo ocurra tan cerca del borde del caos como la organización en desarrollo puede tolerar.

El proceso de desarrollo de software de Scrum descrito en este artículo se desarrolló en una colaboración entre los métodos de desarrollo avanzado y el software VMARK. Ambas compañías estaban reportando una productividad revolucionaria. Este enfoque se superponía significativamente con lo que vimos en postmortem de proyectos exitosos. Nuestra organización ha utilizado durante mucho tiempo los patrones con éxito, no sólo para el diseño sino también para la organización y el proceso.2,3 Muchos patrones existentes soportados

Lo que aprendimos acerca de Scrum y vi en los datos post mortem (vea la barra lateral ¿Qué es un patrón?). Por ejemplo, Scrum defiende el uso de equipos pequeños, no más de 10 miembros del equipo. El patrón de "tamaño de la organización" de Jim Coplien recomienda equipos de 10 personas, mientras que Fred Brooks argumenta que

"El pequeño equipo afilado, que por consenso común no debe exceder a 10 personas" .5 Debido a que Brooks, Coplien y postmortem de equipos exitosos apoyan este y muchos otros principios de Scrum, nos sentimos confiados acerca de emprender un proyecto piloto.

Aunque la mayoría de nuestros equipos de NBO eran pequeños, algunos habían crecido hasta superar el límite de 10. Pensamos inicialmente que podríamos simplemente dividir equipos más grandes en colecciones de subconsultores más pequeños, cada uno no mayor que 10. Encontramos que cuando los sub- Las interfaces bien definidas, esto funciona. Cuando la superposición es considerable y las interfaces poco comprendidas, los beneficios no son tan grandes. Claramente, esto no es un enfoque para estructuras de equipo grandes y complejas, pero encontramos que incluso equipos pequeños y aislados en un proyecto grande podrían hacer uso de algunos elementos de Scrum. Esta es la verdadera diversidad de procesos.

Al realizar un chequeo del proyecto, pensamos que Scrum podría abordar algunos de los problemas que enfrenta el equipo en particular. Después de describir el enfoque tal como lo entendíamos, el equipo decidió probar Scrum y se convirtió en nuestro primer proyecto piloto. No entendimos mucho sobre el enfoque, así que aprendimos como fuimos adelante, al igual que el equipo. El proyecto piloto y cada proyecto siguiente nos enseñaron más acerca de lo que funcionaba para cada grupo y qué componentes responderían a necesidades particulares.

La barra lateral "Informes de experiencia" describe tres de nuestras experiencias en la implementación del enfoque Scrum.

Melé

Un scrum es un equipo de ocho personas en Rugby. Todo el mundo en el paquete actúa junto con todos los demás para mover el balón por el campo.

Para los que saben el rugby, la imagen es clara. Los equipos trabajan como unidades estrechas e integradas, con cada miembro del equipo desempeñando un papel bien definido y todo el equipo centrándose en un solo objetivo. En los equipos de desarrollo, cada miembro del equipo debe entender su rol y las tareas para cada incremento. Todo el equipo debe tener un enfoque único. Las prioridades deben estar claras. Como describimos ahora, el proceso de desarrollo de Scrum facilita este enfoque de equipo.

¿Cómo funciona Scrum?

Scrum es un proceso de desarrollo de software para equipos pequeños. Como lo han demostrado Coplien y Brooks, los equipos pequeños que trabajan de forma independiente son más eficaces.4,5 La investigación de la dinámica social basada en el bienestar apoya este punto de vista. El fenómeno llamado "loafing social", identificado ya en 1927, midió empíricamente la contribución individual a un esfuerzo grupal6. Usando un indicador unido a una cuerda, el estudio encontró que los individuos que tiran de la cuerda promediaron 138.6 libras de presión. Sin embargo, los grupos de tres promediaron sólo 2,5 veces la tasa individual, y los grupos de ocho promediaron menos de cuatro veces la tasa individual.

Como en todos los proyectos, debe haber una fase de planificación inicial. Durante esta fase, el equipo del proyecto debe desarrollar una arquitectura e identificar a un arquitecto jefe. Durante el desarrollo, el equipo debe estar preparado para hacer cambios en esta arquitectura, pero necesitan un plan, una arquitectura y un arquitecto jefe al principio. El arquitecto jefe define la visión del proyecto de desarrollo basado en esta arquitectura y asegura la coherencia de la visión a lo largo de todas las fases de desarrollo. El patrón de Coplien "Architect Controls Product" apoya esta comprensión: "El papel de Arquitecto debe aconsejar y controlar las funciones de Desarrollador y comunicarse estrechamente con los desarrolladores".

Después de la planificación inicial, una serie de fases de desarrollo cortas, o sprints, entregan el producto de forma incremental. Un sprint típicamente dura de una a cuatro semanas. Una fase de cierre suele completar el desarrollo del producto. El equipo rastrea todos los

Tareas, capturándolos en una lista llamada backlog. El backlog impulsa la actividad del equipo. Antes de cada sprint, el equipo actualiza el backlog y reprioriza las tareas: cada equipo se registra para una serie de tareas y luego ejecuta un sprint. Los miembros individuales del equipo están de acuerdo en completar las tareas que consideran factibles durante cada sprint. El equipo define la granularidad de las tareas según corresponda, especificando que una tarea debe completarse en menos de una semana. Las tareas más grandes son más difíciles de definir e informar.

Durante un sprint, no se permiten cambios fuera del equipo.

¿Qué sucede durante un sprint?

Un sprint produce un producto visible, utilizable y de entrega que implementa una o más interacciones del usuario con el sistema. La idea clave detrás de cada sprint es ofrecer funcionalidad valiosa.

Cada incremento de producto se basa en incrementos previos. El objetivo es completar las tareas con la fecha de entrega del sprint. Un sprint es timeboxed de desarrollo, lo que significa que la fecha de finalización de un sprint no cambia. El equipo puede reducir la funcionalidad entregada durante el sprint, pero la fecha de entrega no puede cambiar.

Durante el sprint, el equipo celebra reuniones frecuentes (por lo general diarias) de Scrum. Estas reuniones responden a la observación hecha por Brooks: "¿Cómo llega un proyecto a ser un año de retraso? Un día a la vez. "5 Cuando el equipo

Viene junto para una reunión corta, diaria, cualquier resbalón es inmediatamente obvio a cada uno. Las reuniones involucran a todos los miembros del equipo, incluyendo aquellos que trabajan en telecomutas. Las reuniones sirven a un propósito de construcción de equipos y traen incluso a contribuyentes remotos, haciéndolos sentir parte del grupo y hacer su trabajo visible para el resto del equipo.

¿Cómo planifica y estima?

A menudo, el grupo de marketing o el cliente establecerá la fecha de lanzamiento. Con una fecha de lanzamiento establecida, los grupos de desarrollo y marketing deben trabajar juntos para proporcionar las características con el valor más alto para la primera versión del producto. Marketing debe dar prioridad a las características, mientras que el grupo de desarrollo de productos proporciona estimaciones para el esfuerzo. La comercialización y el desarrollo de productos deben estar de acuerdo con el conjunto objetivo de características. Si el grupo de desarrollo de productos no puede entregar las funciones solicitadas, los grupos deben negociar un conjunto reducido de características.

En las características de negociación de la versión, la administración debe identificar a los desarrolladores disponibles para el desarrollo de funciones. El número de desarrolladores para cada equipo debe cumplir con la recomendación de Scrum: no más de 10. Por supuesto, usted puede tener varios equipos. Para completar las negociaciones con el departamento de marketing, el grupo de desarrollo de productos debe tener los desarrolladores necesarios comprometidos con el proyecto para el período de tiempo identificado.

Una vez concluidas las negociaciones entre los grupos de marketing y desarrollo de productos, el backlog se asigna a sprints en orden de prioridad. El grupo de desarrollo de productos también establece el entorno de desarrollo de destino y determina los riesgos asociados con el mismo. Mientras que el departamento de marketing entiende lo que el cliente necesita y el valor para el cliente, los miembros del desarrollo del producto entienden el riesgo de nuevas tecnologías, herramientas o procesos de software. Identifican y abordan estos riesgos en la toma de decisiones que conduce a la asignación de funciones a sprints. En general, se ocupan de artículos de riesgo en los primeros sprints para dar tiempo a recuperarse si surgen dificultades técnicas.

La planificación procede con relativa rapidez porque los supuestos iniciales seguramente cambiarán a medida que los sprints proporcionen funcionalidad incremental.

¿Quién lidera el equipo?

El maestro Scrum lidera las reuniones Scrum, identifica el backlog inicial que se completará en el sprint y empíricamente mide el progreso hacia el objetivo de ofrecer este conjunto incremental de funcionalidad del producto. El maestro de Scrum garantiza que todo el mundo progrese, registre las decisiones tomadas en la reunión y rastrea los elementos de acción, y mantiene las reuniones de Scrum cortas y enfocadas.

El maestro de Scrum trabaja constantemente para reducir el riesgo del producto a través de la entrega incremental de características, la respuesta rápida a los obstáculos de desarrollo y el seguimiento continuo de la entrega de artículos de backlog.

¿Qué sucede durante una reunión Scrum? Cada miembro del equipo debe responder a tres

Preguntas

1. ¿Qué has completado, en relación con el registro de seguimiento, desde la última reunión de Scrum?

2. ¿Qué obstáculos tuvieron en tu camino para completar este trabajo?

3. ¿Qué cosas específicas planeas lograr, en relación con el retraso, entre ahora y la próxima reunión de Scrum?

La reunión de Scrum debe durar de 15 a 30 minutos, lo que proporciona suficiente tiempo para abordar los obstáculos, pero no permite tiempo para una lluvia de ideas de una solución. Todas las discusiones distintas de las respuestas a las tres preguntas se aplazan a una reunión posterior que involucra solamente a los realmente afectados por la discusión. Los objetivos de la reunión de Scrum incluyen:

■ enfocar el esfuerzo de los desarrolladores en los elementos de la cartera,

■ comunicar las prioridades de los elementos de la cartera a los miembros del equipo,

■ Mantener a todos informados sobre el progreso y los obstáculos del equipo,

■ resolver los obstáculos tan pronto como sea posible,

■ seguir el progreso en la entrega de la funcionalidad de la reserva, y

■ abordar y minimizar el riesgo del proyecto. ¿Qué sucede al final de un sprint?

Al final de un sprint, el equipo produce un incremento que se basa en incrementos previos. El equipo puede recortar la funcionalidad pero debe cumplir con la fecha de entrega prometida.

Después de cada sprint, todos los equipos del proyecto se reúnen con todas las partes interesadas, incluyendo la administración de alto nivel, los clientes y los representantes de los clientes. Toda la información nueva del sprint recién terminado se informa. En esta reunión, todo puede ser cambiado. El trabajo se puede agregar, eliminar, o reprioritized. La entrada del cliente modela las actividades de establecimiento de prioridades. Los artículos que son importantes para el cliente tienen la prioridad más alta.

Los nuevos planes y estimaciones se realizan siguiendo el mismo proceso que se discute en "¿Cómo planifica y estima?". Las asignaciones se hacen a los equipos para el próximo sprint. A medida que los sprints terminan, las estimaciones se hacen mejores cuando los planificadores ven lo que cada equipo ha producido en sprints anteriores. Con los pequeños incrementos de tiempo, los planificadores deben ser cuidadosos en su estimación, porque, como Brooks afirma, "La extrapolación de tiempos para el trazo de cien yardas muestra que un hombre puede correr una milla en menos de tres minutos".

Debido a que cada sprint produce un incremento visible, utilizable, la entrega del producto puede tener lugar después de cualquier sprint. De hecho, la entrega de un tipo tiene lugar y la retroalimentación de los negocios y el lado de la comercialización y de los clientes es una reacción a la entrega del incremento actual. Típicamente, la entrega final tiene lugar después de una fase de recapitulación que se ejecuta como un sprint final.

La organización puede tomar una decisión muy importante al final de un sprint: si continúa el desarrollo del producto. Teniendo en cuenta lo que se dio durante el último sprint y el estado actual del mercado, la reunión de las partes interesadas debería abordar la pregunta "¿Debe este proyecto continuar?" Esta es una decisión de negocios y debe hacerse después de considerar todas las cuestiones técnicas y de marketing.

Como resultado de la interacción de pequeños equipos en pequeños ciclos de desarrollo enfocados,

■ el producto se convierte en una serie de trozos manejables,

■ progreso, incluso cuando los requisitos no son estables,

■ todo es visible para todos,

■ mejora la comunicación con el equipo,

■ el equipo comparte éxitos en el camino y al final,

■ los clientes ven la entrega a tiempo de incrementos,

■ los clientes obtienen retroalimentación frecuente sobre cómo funciona realmente el producto,

■ se desarrolla una relación con el cliente, se crea confianza y el conocimiento crece, y

■ Se crea una cultura donde todos esperan que el proyecto tenga éxito.

La mayoría de los elementos de Scrum no son nuevos. Se trata básicamente de un desarrollo incremental, en tiempo real, con un giro añadido: las frecuentes reuniones donde se hacen las tres preguntas. El modelo espiral de Barry Boehm aborda ciertamente el elemento de riesgo en el desarrollo de software.7 Las ideas esenciales detrás del modelo en espiral son exactamente las mismas que las de Scrum, ¡aceleradas! Un sprint típicamente dura de una a cuatro semanas y un proyecto entero solamente algunos meses. Durante ese corto tiempo, el cliente original podría salir y un nuevo llegar - con nuevas restricciones. Además, la tecnología podría cambiar de debajo del equipo.

Hemos descubierto que Scrum es apropiado para proyectos en los que no podemos definir los requisitos por adelantado y se esperan condiciones caóticas a lo largo del ciclo de vida del desarrollo del producto. En AG Communication Systems, continuamos desarrollando el enfoque a medida que nuestros grupos de desarrollo ganan experiencia con él.