Práctica 1
Metodologías Ágiles

Materia: Metodologías de Programación II
Alumno: Federico Almada
Profesora: Claudia Cappelletti

1) ¿Qué es lo que se conoce como “desarrollo ágil de software”?

Se conoce como desarrollo ágil de software al conjunto de métodos en el que las necesidades y soluciones evolucionan a través de colaboración y entre equipos multifunción y auto-organizados. La idea de el desarrollo ágil es promover la planificación adaptativa, desarrollo evolutivo, entrega temprana y mejora continua, promoviendo una respuesta rápida y flexible a los cambios.

2) ¿Qué es el Evolutionary Project Management (EVO)? ¿Cuándo y quién lo desarrolló? ¿Cuáles son sus principios?

Es uno de los muchos métodos de gestión de proyectos que no son en cascada (iterativos, incrementales, en espiral, de adquisición evolutiva) que se utilizan actualmente en la industria. El creador de este método es el autor Tom Gilb, que ha aplicado EVO desde 1960 en proyectos reales, tanto personalmente como a través de clientes y estudiantes.

La característica distintiva central de esta versión de EVO es su enfoque en la especificación cuantificada y el seguimiento de múltiples requisitos de rendimiento y costos durante los primeros y frecuentes pasos de EVO de entrega de resultados a las partes interesadas. EVO es el único método conocido que tiene un historial de éxito, práctico, consistente y sin fallos.

Sus principios son los siguientes:

1. Descompone por resultados de rendimiento y partes interesadas

Desglose la arquitectura y las estrategias generales, en pequeños pasos de entrega de valor (nos gustan 'semanales'), de modo que cada paso de EVO ofrezca algo de valor.

2. Realice pasos de alto riesgo a tiempo. Aprenda cómo funcionan realmente las "incógnitas"

Una posibilidad es que incluso los pasos de alto riesgo podrían ofrecer el mejor valor disponible, si tienen éxito. Hay dos "valores" aquí. El valor entregado,si tienen éxito y, alternativamente, el conocimiento de que no ofrecen valor,si fallan; desde el principio!

3. Concéntrese, en mejorar sus objetivos de rendimiento más valiosos, en primer lugar

Este principio supone que hemos cuantificado nuestros pocos objetivos de valor críticos, y que hemos seguido el progreso numérico hacia ellos en cada ciclo de EVO. En general, las cosas más valiosas que debes hacer en el siguiente paso son "lo que pueda cerrar las brechas restantes a tu nivel de meta"; para la menor cantidad de recursos o presupuesto restantes. Los ciclos de entrega de EVO pueden tener un impacto útil en dos o más objetivos simultáneos de alto valor. Los arquitectos inteligentes pensarán en múltiples dimensiones al mismo tiempo; al igual que los buenos ajedrecistas. Los pasos de Evo son

similares a una serie de movimientos de ajedrez. Ganar es alcanzar todos los niveles de meta antes de que se agote el tiempo de salida de los recursos.

4. Base su evolución temprana en los sistemas existentes y las partes interesadas

Si desea obtener la entrega de valor que va temprano, con las verdaderas partes interesadas existentes, usted tiene que hacer uso del sistema actual (por horrible que sea en este momento) utilizado por esas partes interesadas. El cambio revolucionario lleva demasiado tiempo, y fracasa. Probablemente obtendrá un resultado más revolucionario al evolucionar rápida y temprano, desde su sistema de estado actual. Y centrándose en ofrecer mejoras de valor temprano,en lugar de recrear primero la funcionalidad del sistema actual.

5. Diseño para costear dinámicamente

Todas las ideas de nivel arquitectónico (como comprar o construir, subcontratar o hacerlo usted mismo) deben ser validadas, probadas en la práctica, con respecto a sus costos reales (dinero, tiempo, talento); antes de que te hayas comprometido con ellos, y escalado. Si un componente de arquitectura falla en valor o costos excesivos, debe volcarlo rápidamente y encontrar una mejor idea que realmente funcione.

6. Diseño para el rendimiento dinámicamente

La misma idea de "diseño a valor", como la anterior, también se aplica a la gestión de todos los aspectos de rendimiento y calidad (los valores que queremos ofrecer). Necesitamos validar, en la práctica, los múltiples impactos medidos de nuestras arquitecturas y estrategias en los principales valores que buscamos, antes de escalar y comprometernos. Lo que no ofrece el valor esperado, debe ser abandonado temprano. Los arquitectos más imaginativos necesitan encontrar una arquitectura de entrega de mejor valor y demostrarlo.

7. Invertir en una arquitectura abierta desde el principio

Vas a tener algunas decepciones cuando pruebes algunas ideas de arquitectura en una etapa temprana. No se puede arriesgar a un alto costo, en tiempo y dinero, por tener que deshacerse de las malas ideas recién probadas.

Por lo tanto, su arquitectura técnica y organizativa debe permitir cambios de bajo costo de arquitectura nueva e mejor e imprevista. Esto significa nuevos proveedores, socios y componentes técnicos. Usted, frente a él, tendrá que lidiar con este problema a largo plazo, pista: "deuda técnica", por lo que también podría asegurarse de que puede cambiar las cosas fácilmente en el corto plazo.

La facilidad técnica de cambio se puede diseñar en el sistema. Se puede cuantificar por adelantado, como un objetivo de calidad crítica 4. Incluso se puede diseñar simultáneamente en, años después de que debería haber sido, como Confirmit 4,pero ¿por qué esperar?

8. Motive a su equipo recompensando los resultados

No premiar, con elogios o pagos, mero esfuerzo,mero trabajo,mero código completado. Dígale a la gente: ¡el esfuerzo es irrelevante! ¿Entregaron o no entregaron resultados medibles, placenteros y planificados a las partes interesadas! Delira sobre los resultados reales. ¡Destaca el equipo que los entregó! Pague a los contratistas por el valor realmente entregado, no por el trabajo hecho, las horas gastadas.

9. Priorizar los cambios por valor, no colocar en cola

No importa cuando surge una idea de cambio, o mejora. La decisión, en cuanto a qué hacer en el próximo paso de Evo, debe basarse en el valor (a la relación costo) de la idea.

Una de las principales ventajas incorporadas de Evo es exactamente que podemos hacer el mejor "movimiento" ágil posible, inmediatamente - incluso si acabamos de ver la oportunidad recientemente (por ejemplo, durante la última semana). No estamos vinculados a las decisiones aprobadas por la comisión del año pasado. El trabajo de la comisión es aprobar previamente que el proyecto tome las decisiones, que dan la máxima competitividad en todo momento.

10. Aprende rápido, cambia rápido, adáptese a la realidad rápidamente

Este principio anterior es un buen resumen del método Evo. La victoria es para los vencejos.

Fuente: About Agile Alliance | Agile Alliance

3) Investigue sobre la Alianza Ágil en su sitio web oficial www.agilealliance.org. ¿Qué es? ¿Cuáles son sus objetivos y actividades?

Agile Alliance es una organización global sin fines de lucro fundada en el Manifiesto para el Desarrollo ágil de software. Apoya a las personas y organizaciones que exploran, aplican y amplían valores, principios y prácticas ágiles.

Lleva a cabo una variedad de actividades para construir una comunidad global inclusiva, avanzar en la amplitud y profundidad de Agile y proporcionar valor a los miembros. Estas actividades incluyen:

- Conferencias que reúnen cara a cara a la comunidad agile
- Un sitio web lleno de información sobre Agile y la Comunidad Ágil
- Membresía que proporciona acceso a valiosos recursos creados por miembros de la comunidad
- Iniciativas que abordan ámbitos específicos de interés en la Comunidad Ágil y prestan apoyo a los Grupos Comunitarios Locales

Su objetivo es apoyar a las personas y organizaciones que exploran, aplican y amplían valores, principios y prácticas ágiles. En apoyo de esta misión, Agile Alliance quiere garantizar un entorno respetuoso, seguro e inclusivo para todos cuando y dondequiera que participen en las actividades de Agile Alliance.

Fuente: Code of Conduct | Agile Alliance

4) Según el Manifiesto Ágil, ¿cuál es la mejor forma de comunicación en un proyecto de desarrollo de software?

La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.

Fuente: Los 12 Principios del Manifiesto Ágil – Make the world a better place (doinglobal.com)

5) ¿Qué es el "timeboxing"?

La técnica del timebox consiste en fijar el tiempo máximo para conseguir unos objetivos, tomar una decisión o realizar unas tareas, y hacer lo mejor que podamos en ese intervalo. De esta manera, en lugar de ponerse a trabajar en algo hasta que esté hecho, de antemano se acuerda que sólo se dedica un tiempo limitado. La consciencia de esta limitación temporal favorece la priorización de objetivos/tareas y fuerza la toma de decisiones.

Fuente: Timebox – Proyectos Ágiles (proyectosagiles.org)

6) ¿Cuál es la clave de un método de desarrollo adaptativo? ¿Cuál es su opuesto y en qué se diferencian?

La clave del método de desarrollo adaptativo es adaptarse rápidamente a las cambiantes realidades. Cuando las necesidades de un proyecto cambian, un equipo adaptativo cambia también.

Su opuesto es el desarrollo predictivo que básicamente se centra en el análisis y la planificación del futuro y los riesgos conocidos.

7) Investigue sobre el modelo y documentación JBGE ("Just Barely Good Enough"). ¿A qué apunta?

Just-Barely-Good-Enough (JBGE) es un método de documentación que asempleo. Su simplicidad es su fuerza. Su facilidad le da profundidad y usabilidad, su evolución le da longevidad. Básicamente documenta, ya sean procesos, estándares, fragmentos de código, cómo hacer, lo que necesitas, etc. No se deje restricciones por plantillas. Simple, fácil de leer. Busque comentarios, pida a otros que actualicen, que evolucionen con el tiempo.

Fuente: (99+) Just Barely Good Enough, is good enough | LinkedIn

8) Detalle algunas críticas negativas a las metodologías ágiles.

Algunas críticas negativas frecuentes son:

Insuficiente nivel de conocimientos

El conocimiento y la formación son imprescindibles para gestionar proyectos a través de los métodos ágiles. Asistir a alguna conferencia sobre Kanban, participar en un seminario de Scrum o matricularse en un curso ágil buscando obtener una certificación para iniciarse en el mundo ágil no basta para estar preparado. La práctica no es igual a la teoría y, si a pesar de todo se está en posición de responsabilizarse de una iniciativa de este tipo, merece la pena buscar mentores con experiencia en la implementación ágil.

Problemas de localización

Pese a que el trabajo con equipos remotos es cada vez más frecuente, en lo que concierne a este tipo de proyectos puede no ser la mejor opción. La cercanía entre los desarrolladores favorece la resolución de problemas y minimiza su aparición, además de permitir aumentar el rendimiento.

Síndrome del Burn Out

Trabajar con métodos ágiles puede aumentar el estrés de algunos de los desarrolladores. Esta circunstancia puede detectarse a tiempo de poner medidas, por lo que hace falta mantener los ojos bien abiertos antes de que los resultados se vean afectados y la desmotivación se extienda al resto del equipo.

Falta de participación del cliente

A pesar de su entusiasmo por trabajar en base a métodos ágiles, a la hora de la verdad, el cliente no lo pone tan fácil para mantener el ritmo de comunicación necesario. Dejarlo todo claro desde el principio acerca de la importancia de su involucración puede asegurar que el cliente será capaz de comunicarse con el equipo de desarrollo sobre una base constante.

Desarrolladores protagonistas

Muchas veces puede suceder que algunos miembros del equipo se consideren superiores al resto, por su nivel de experiencia o conocimientos, y esto les anime a pensar que pueden tomar decisiones finales por sí mismos sin necesidad de consultar al resto y saltándose todos los principios del agile development. Para evitarlo, nada como recordar las reglas.

Aumento del riesgo

Puede suceder que el ritmo del trabajo y el entusiasmo por los logros alcanzados haga perder la aversión al riesgo que se necesita en cualquier proyecto. Si bien la innovación y la proactividad son atributos necesarios en toda iniciativa de IT, si no se emplean técnicas de gestión de riesgos dentro del desarrollo ágil los problemas no tardarán en aparecer.

Falta de controles de calidad: en la misma línea, la calidad no puede obviarse

De nada sirve avanzar a toda velocidad si el producto no reúne las condiciones necesarias. De hacerlo así, además de surgir problemas en algún punto del desarrollo, los clientes terminarán descontentos.

Fuente: Los 8 inconvenientes de los métodos ágiles a los que deberás enfrentarte | OBS Business School

9) En cuanto a las pruebas, ¿cuál es la diferencia entre los métodos ágiles y el de cascada?

La cascada es una metodología de desarrollo de software que a menudo puede ser bastante rígida, mientras que la metodología Ágil es conocida por su flexibilidad.

El proceso de desarrollo de software se divide en diferentes etapas en el modelo cascada, mientras que la metodología Ágil separa el ciclo de vida del desarrollo del proyecto en sprints.

Según el modelo cascada, tenemos que completar el desarrollo del software como un solo proyecto. Luego lo dividimos en diferentes fases. Cada fase ocurre solo una vez durante el proyecto. La metodología Ágil, por otro lado, puede verse como una colección de muchos proyectos pequeños diferentes. Proyectos que no son más que iteraciones de las diferentes etapas destinadas a mejorar la calidad general del software con comentarios de los usuarios o del equipo de control de calidad.

Completamos todas las fases de desarrollo del proyecto, como diseño, desarrollo, pruebas, etc. en el modelo cascada una vez, mientras que como parte de la metodología Ágil utilizamos un enfoque de desarrollo iterativo. Por lo tanto, la planificación, el desarrollo, la creación de prototipos y otras fases del desarrollo de software pueden ocurrir más de una vez durante un proyecto Ágil.

Si bien la metodología de cascada es un proceso interno y no requiere la participación del usuario, el enfoque de desarrollo de software de Ágil se centra en la satisfacción del usuario y, por lo tanto, en la participación de los usuarios durante la fase de desarrollo.

Una de las principales diferencias entre la metodología de desarrollo Ágil y cascada es su propio enfoque de calidad y pruebas. En el modelo cascada, la fase de "prueba" viene después de la fase de "construcción", pero en el método ágil, generalmente realizamos las pruebas al mismo tiempo que la programación o al menos durante la misma iteración que la programación.

Fuente: Diferencias importantes entre la metodología Agile y Waterfall (itpedia.nl)

10) Para los creadores del Manifiesto Ágil, ¿cuál es el principal factor de medición de avance de un proyecto? ¿Qué otras métricas de avance se suelen usar tanto en métodos ágiles como tradicionales?

- Agility Index Measurements (AIM) asigna puntaje a los proyectos sobre una serie de factores de agilidad.

- Agility Measurement Index (AMI) asigna puntaje a los desarrollos en cinco dimensiones: duración, riesgo,innovación.

Otras técnicas se basan en objetivos mensurables o la velocidad del proyecto. Existen evaluaciones para determinar si un equipo está utilizando prácticas ágiles, como las pruebas Nokia o Karlskrona. La aplicación práctica de estas mediciones es objeto de debate.