**Clase 12/08 – Requerimientos Ágiles**

Requerimientos en Agile

Existen tres pilares fundamentales para los requerimientos en agile, que están a la mano de los principios del manifiesto ágil:

Diagrama, Texto

Descripción generada automáticamente

* A diferencia del enfoque tradicional, cuyo objetivo es construir software funcionando que entregue valor al cliente, el foco del enfoque en gestión ágil apunta a construir **valor de negocio**. Las herramientas de software que construimos bajo este enfoque deben ser de ayuda a cumplir con objetivos de negocio. El rector de nuestras decisiones debe siempre estar enfocado en **construir el producto correcto que entregue valor al negocio alineado a sus objetivos**.
* Los requerimientos en agile son encontrados **de a poco**. En enfoques tradicionales, era deseable obtener una definición completa de todos los requerimientos desde el principio para trabajar. El enfoque ágil comienza con **lo mínimo necesario** para poder comenzar a girar la rueda y obtener retroalimentación en el proceso, que permite ir completando iterativamente los requerimientos mientras sea necesario.
* En el enfoque ágil, el proceso de desarrollo de requerimientos se realiza **junto con el cliente**. Integra equipos técnicos con algún referente del cliente, del lado del negocio, que tiene el conocimiento del producto y que trabaja junto con el equipo.

Por lo tanto, técnicos y no técnicos trabajan y **descubren** juntos, mediante el uso de **historias** para formar la definición de producto. Este enfoque considera que la mejor forma de comunicación es la cara a cara, para poder crear la visión del producto que queremos construir.

En la actualidad, el enfoque tradicional ocasionó que los profesionales de nuestra área realicen trabajo que **no se necesita.** Esto nos lleva a que tan solo el 20% de la funcionalidad de nuestros productos sean utilizadas.

Diagrama

Descripción generada automáticamente

El enfoque ágil de requerimientos intenta equilibrar y contrarrestar esta realidad.

Es el dueño del producto es el que realmente **conoce** cuáles son sus necesidades. Por lo tanto, el enfoque ágil hace uso de este concepto fundamental, y le otorga como al dueño del producto como **principal responsabilidad** el proceso de **priorización de requerimientos**. Se define entonces lo que se conoce como **Product Backlog** que es una lista priorizada donde el dueño del producto ordena los requerimientos, organizándolos en orden descendente de prioridad. Esta lista se ordena **verticalmente** para obligar al dueño del producto a determinar la prioridad. El dueño del Product Backlog es el dueño del producto.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

El **talón de Aquiles de éxito de los enfoques agiles**, es que se pueda implementar una **gestión ágil de requerimientos**. Esto indefectiblemente requiere que el **cliente** tenga la voluntad y ganas para trabajar como **dueño del producto** y que además sea alguien **que tenga conocimiento sobre el producto, que tenga capacidad de tomar decisiones y que esté disponible para el equipo**, para responder dudas en los tiempos requeridos.

Los enfoques ágiles, cuando logran trabajar de esta forma, **compensan** lo que para los enfoques tradicionales era obligatorio. Por ejemplo, mediante los enfoques ágiles deja de ser necesaria la necesidad de tener una documentación formal precisa, porque se compensa mediante la comunicación cara a cara con el cliente.

Los requerimientos ágiles utilizan un enfoque “justo a tiempo” donde el producto no estará especificado al cien por cien desde el principio, los requerimientos serán encontrados y descriptos a medida que se encuentran. No pueden llegar ni antes, ni después.

**En síntesis**, una **gestión ágil de requerimientos** requiere que técnicos y no técnicos trabajen juntos, entendiendo las necesidades del negocio para transformarnos en **expertos del dominio**. Luego, junto con el usuario, debemos descubrir cual es la solución o la mejor forma de satisfacer esas necesidades. Esto produce el surgimiento del Product Backlog, que requiere de un equipo motivado y competente, y permite determinar en una única iteración qué características serán entregadas. Con esto **entregamos frecuentemente valor a los stakeholders** y obtenemos retroalimentación que nos ayuda a refinar el Product Backlog.

**Por último**:

* **Los cambios son la única constante**: Asumimos que los cambios existirán todo el tiempo.
* **Stakeholders, no son todos los que están:** Debemos entender quienes son los involucrados. Los involucrados no son únicamente los que están en la etapa de descubrimiento de requerimientos. **Involucrados son todos los que tienen algo para decir respecto del producto**, estén o no estén presente.

El dueño del producto actúa como en **representante** de todos los que tienen algo para decir sobre el producto. El dueño del producto es **uno solo**.

* **Siempre el usuario dice lo que quiere cuando recibe lo que pidió:** El usuario tiene algo para decir cuando **recibe** algo para ver. Es por este motivo que los enfoques ágil tienen la necesidad de mostrar una iteración lo antes posible.
* **No todas las técnicas ni herramientas sirven para todos los casos**.
* **Lo importante no es entregar una salida, lo importante es entregar un resultado, una solución que de valor**.

Tradicional vs Ágil

Diagrama

Descripción generada automáticamente

En el enfoque ágil, se busca que el producto se defina conforme haga falta “justo a tiempo”, difiriendo las decisiones en la mayor medida y tomando decisiones en el último momento responsable (postergando la toma de decisiones lo mayor posible hasta que se disponga de la información más relevante y actualizada para hacerlo, de forma de evitar decisiones prematuras o basadas en suposiciones).

Esto nos lleva a que, a diferencia del enfoque tradicional que deja fijo los requerimientos (y esto indefectiblemente requiere que los requerimientos se definan de forma completa y de antemano) para a partir de ahí derivar recursos y tiempo para llevarlos a cabo; el enfoque ágil **deja fijo el tiempo** con **iteraciones de duración fija** y **recursos fijos** donde se tiene un equipo fijo con una determinada capacidad que se estima con los recursos de infraestructura y comunicaciones necesarios. Entonces, teniendo fijo el tiempo y los recursos, puedo decir **cuánto del producto puedo construir con el equipo en ese tiempo** y para ello se **acuerda** cuánto es lo que el equipo puede entregar de software **funcionando** y luego se **repite** en la próxima iteración.

Tipos de Requerimientos

Imagen que contiene Diagrama

Descripción generada automáticamente

Los requerimientos a los cuales el enfoque ágil atiende son los **requerimientos de negocio y requerimientos de usuario**. Los requerimientos de negocio son los que le interesan al usuario y el que le da valor al negocio. Los requerimientos de usuario, que son identificadas mediante historias de usuario **por el dueño del producto**, son **el medio por el cual** el usuario cree que logrará el valor de negocio.

Mediante la comunicación cara a cara es fundamental que nosotros advirtamos si el requerimiento de usuario planteado será o no capaz de satisfacer el requerimiento de negocio.

Principios ágiles relacionados a los requerimientos ágiles



Principalmente, el éxito de la gestión ágil radica en los principios (4) y (6).

Historias de Usuario

Una historia de usuario es una descripción corta de una necesidad que tiene el usuario respecto del producto de software. Es una técnica que nos permite encontrar requerimientos del usuario. La historia del usuario representa algo que el usuario nos quiere contar, que considera importante para su negocio.

Componentes de una historia de usuario

* Conversación: Son las discusiones, debates y conversaciones que ocurren entre los miembros del equipo, técnicos y no técnicos, para comprender completamente la historia de usuario. La conversación no se guarda, porque tal y como establece el principio, la mejor conversación es la cara a cara.
* Tarjeta: Es la parte visible de la conversación. Es una breve descripción que se centra en expresar el valor que la funcionalidad proporcionará al negocio. Describe la característica del producto que se espera que el usuario reciba.
* Confirmación: Son las pruebas de usuario que se identifican necesarias para que el dueño del producto acepte la característica de software que construyo a partir de la historia de usuario.

La definición del **valor de negocio** es muy importante ya que le permite al dueño del producto **priorizar el Product Backlog** a la historia de usuario en el contexto de todas las demás.

Las historias de usuario son multipropósito

* Describen una necesidad de usuario.
* Una descripción del producto.
* Son un ítem que nos sirve para planificar lo que conformará a la iteración.
* Son un recordatorio de que tenemos que conversar con el usuario respecto de las características que el especificó.
* Son un mecanismo para diferir responsabilidades y decisiones.

Las historias de usuario **no son una especificación de requerimientos de software**. No reemplazan la ERS, no reemplazan los casos de uso.

Las historias de usuario son porciones verticales

Las historias de usuario se diseñan para representar funcionalidades que le entreguen valor al cliente. La única forma de que una historia de usuario entregue valor de negocio al usuario es mediante porciones verticales, están diseñadas para completarse en su totalidad en una iteración, por lo que **todas las capas** del sistema deben estar implementadas y funcionando para las características que esa historia representa.

Gráfico, Tabla, Gráfico de rectángulos

Descripción generada automáticamente

Las historias representan funcionalidades completas. Al usuario no le entrega valor tener toda la interfaz gráfica, si la misma no tiene funcionalidad; o toda la base de dato, si con ella no puede interactuar. Una historia debe tener la interfaz gráfica, lógica y base de datos necesaria (y únicamente lo necesario) para poder entregar una funcionalidad completa que entregue valor al cliente.

Proxies

Existen diversas situaciones en la cual, la persona que debería actuar de dueño de producto, por varios motivos no puede ocupar ese lugar; sea porque no tiene tiempo de hacerlo o lo que sea. En estos casos se deben buscar alternativas.

Las alternativas son **siempre perfiles de negocio** **no-técnicos**. El dueño del producto se debe cubrir con un rol **de negocio**, **no-técnico**. A estos roles que reemplazan al dueño del producto se los conoce como **proxies**.

No son ideales de seleccionar como lo es el dueño del producto, por lo que solo deben ser seleccionados en casos específicos.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Criterios de aceptación

Los criterios de aceptación son información concreta que tiene que servirnos para saber si lo que hemos implementado es correcto o no. El dueño del producto aceptará o no la historia de usuario en base a estos criterios de aceptación, que permitirán definir si es o no lo que el dueño del producto necesita.

Los criterios de aceptación se escriben en lenguaje de usuario:

* Deben ser verificables y especificados. Deben indicar qué es “válido” que es “mucho” que es “poco”.
* Deben ser independientes a la implementación.
* Deben definir una intención del usuario, no una solución a esa intención.

Ejemplo válido: El usuario debe elegir al menos una cuenta para operar.

Ejemplo inválido: El usuario debe seleccionar una cuenta de la lista desplegable ubicada en la esquina superior derecha de la pantalla.

Definición de Listo (DoR)

Es una medida de calidad que construye el equipo en su conjunto para poder determinar que la historia de usuario está en condiciones de entrar en un entorno de desarrollo, de implementación.

El modelo de calidad **de base** es el INVEST. La definición de listo de un equipo, como mínimo, debe cumplir este modelo base, pero puede incluir otras que el equipo considere.

* Independent: El dueño de producto tiene la libertad de priorizar las historias de usuario para poder desarrollarlas en cualquier orden.
* Negotiable: La historia de usuario debe estar escrita en términos de qué necesita el usuario y no de cómo será implementado.
* Valuable: La historia debe tener valor de negocio.
* Estimatable: Asignar un número a la historia de usuario para poder compararla con otra en términos de complejidad o peso.
* Small: Deben ser consumidas en una única iteración. Esta característica depende del equipo, las historias deberían ser lo suficientemente pequeñas para poder ser consumidas en una iteración, pero no tanto como para que no sume valor de negocio.
* Testable: Se debe poder demostrar que la historia de usuario efectivamente se implementó cumpliendo los criterios de aceptación definidos.

Adicionalmente el equipo puede solicitar más requisitos como: solicitar un prototipo de la historia, solicitar las reglas de negocio asociadas, las clases del diagrama de clases que se verán afectadas, entre otras.