

Requerimientos agiles (User stories)

jueves, 22 de agosto de 2024 08:51

Los requerimientos agiles son aquellos que se van descubriendo mientras avanza el proyecto

User Stories es ...

Tipos de ciclo de vida

Proceso: conjunto de actividades estructuradas con un fin común, describe como en este paso en particular hacer las cosas

- Secuenciales
- Iterativos
- Evolutivos

El ciclo de vida, marcara cuando hacer esas actividades, si puedo solaparlas, hacerlas juntas, etc.

Proceso definido vs proceso empirico

El **proceso empirico basado en experiencia** se da mediante el aprendizaje y el conocimiento. Parte de una idea del empoderamiento de los equipos de trabajo, donde son autogestionados o autogestionados, que pueden tomar sus propias decisiones, estas motivados, capacitados y tienen experiencia, estas cosas le dan la autoridad para tomar sus propias decisiones. NO TOMAN una vision de proyecto que tenga la empresa. El problema es que es complicado aplicar en casos sin experiencia, lo que va a suceder es que esta experiencia sera la propia que vayan ganando, haciendo aportes, generando experiencia y volviendo a cambiar. Estos plantean que la **experiencia no es extrapolable**, osea pertenece al equipo, si cambio de equipo o de cliente, cualquier factor de condicion y esa experiencia deja de ser valida y debo volver a tomar experiencia, no tengo garantia que el comportamiento sea el mismo en la proxima iteración

Debe tener ciclos de vida corto, con 3 pilares:

- **Integración**
- **Adaptación**
- **Transparencia**: se espera de los involucrados en el proyecto, cultura o actitud, de que el código es del equipo, se comparte es visible, toda la info es de todos

De esta forma puedo detectar errores o cosas a corregir y en la adaptación cambiar lo que haya que cambiar. La retroalimentación es la experiencia. Si no tengo experiencia, lo mejor que puede pasar es tener retroalimentación lo antes posible

Con estas condiciones, los únicos **ciclos de vida validos** para estos procesos son los **ciclo de vida iterativos**

Las iteraciones en Scrum se llaman sprint

2 enfoques para empirico

- Agile -> Scrum (Framework de gestion)

Manifiesto agile: Cosas basicas en la que estan de acuerdo para hacer **implementaciones de frameworks agiles**



Trabajamos los requerimientos agiles con una tecnica que se llama user stories, esta fue adoptada por SCRUM, pero existen otras (Otro ejemplo JBD/job to be done)

El **proceso definido** busca que todos el mundo haga las cosas de la misma forma, para utilizar la experiencia y la previsibilidad de hechos de procesos anteriores. Tengo un proceso escrito organizacionalmente que puede cubrir todo lo que hace falta para hacer software y luego en cada instancia en particular, por la gente, personas, especificidad del producto, ese proceso se adapta y esa adaptación sea que iré sacando del proceso en el proyecto que no me hace falta hacer

En las org se arma un proyecto y luego se arman guías de adaptación

Puedo adaptar el proceso y luego crear el repositorio (más complicado) o al revés

La ventaja es que al definir formas de trabajo, ayuda al recambio de gente, nuevos ingresos o personas sin experiencia, se estudia esa forma de trabajo y herramientas con formas de inducción

En estos procesos puedo elegir cualquier tipo de ciclo de vida

La gestión Ágil y la gestión real de los requerimientos, la diferencia que tienen es. El equipo va a decidir que nivel de especificación necesita el trabajo

Las mejores **disfrazo/arquitecturas** vienen de equipos **autoorganizados**



Se empezó a preguntar por que se hacia todas las características, si solo el 20% de las características eran realmente utilizadas

De la totalidad de requerimientos para el producto de software, el 50% serán requerimientos emergentes, en el otro 50% estarán los conocidos y los desconocidos

Esto nos demuestra que es imposible encontrar el 100% de los requerimientos, hay gente que identifica una parte, finje demencia y avanza sin realmente tener todos

De acá proviene el termino emergente, osea aparecen en un momento en el que no estamos esperando

Aca vemos que no es optimo que alguien externo saque los requerimientos, osea que el mejor para tomar decisiones y conocer requerimientos es alguien que esta involucrado en el proyecto y se integra al equipo

tambien un representante del negocio, lo que permitira tomar decisiones del producto que sean representativas del negocio (Product Owner) Es el factor clave en el desarrollo, si sabe lo que quiere y puede decirlo, traza grandes ventajas

Parto de una vision, una idea global corta, que describe el producto que a mi me gustaria ver. Podes empezar por lo seguro que es a ir haciendo ciclos de adaptación

Otra característica de los req agiles es que analice cuando lo necesite, no antes



Niveles de granularidad de los req

Los casos de uso se utilizan para especificar req de sw

La User stories describe una funcionalidad de negocio desde la perspectiva del usuario, se encuentra dentro de los requerimientos de negocio

La user stories se expresa en terminos del negocio. Las tiene que escribir el product owner, sea venimos con un conflicto, difícil que tome decisiones y pretendamos

La gestión ágil descansa en:



Es un contraindicador de las características que no se han desarrollado, una vez que salen, no vuelven

User stories

Cuentan con 3 partes



La mas importante es la conversacion

La mas visible son las card o tarjetas

Como **Conducir**, quiero **llevar un destino a punto de esta forma y a una hora para poder llegar al lugar deseado sin problemas**

La frase verbal es un resumen de la user story (parte remarcada en rosado), no es obligacion, el equipo decide si quiere utilizar o no. Así como tambien se debe decidir si enumerarlas

La tarjeta es la parte visible de la user story, muestra que quiere el usuario

Descripcion corta de una funcionalidad que solo la necesita un rol específico de la organización

Forma de expresar las Historias de Usuario

Como **usuario del sistema**, quiero **que el sistema me permita ingresar a la forma tal que necesito**

Cosas como el login se usan roles genericos que abarquen a todos

Pero por lo general no vamos a ser genericos, sino mas bien especificos y construye perfiles

Las User Stories son Multipropósito

- Las Historias son:
- Una necesidad del usuario
- Una descripción del producto
- Sin términos de especificación
- Vienen para una conversación
- No se usan para definir una implementación

No es una especificación de requerimiento, no alcanza la carpeta, es una descripción corta únicamente



Voy contando verticalmente, pedacitos de cada funcionalidad

La user tiene criterio de aceptación tangibles que me pueden decir si estoy cumpliendo o no y permite rechazar facilmente al product owner

Estos criterios son charlados por el product owner para saber hasta donde voy a llegar

Como **Conducir**, quiero **llevar un destino a punto de esta forma y a una hora para poder llegar al lugar deseado sin problemas**

User Story: tarjeta y Plantilla de Aceptación

De minima debe tener una que cumpla y una que no cumpla

"Debe" para dato obligatorio y "puede" para dato opcional