User Stories

User Storie

Una user storie contiene una descripción corta de una funcionalidad valuada por un usuario o cliente de un sistema. Se llaman "stories" porque se supone que Ud. cuenta una historia, es decir lo que se escribe en la tarjeta no es lo más importante, sino la conversaciones entre los clientes y desarrolladores a cerca de la historia.

Los user stories son:

- Una necesidad de usuario
- Una descripción del producto
- Un ítem de planificación
- Token para una conversación
- Mecanismo para diferir una conversación

Estructura de una User Storie

Como **<< rol de usuario >>**
yo puedo **<< actividad >>**
de forma tal que **<< valor de negocio >>**
Rol de usuario: representa quien está realizando la acción o quien recibe el valor de la actividad.
Actividad: representa la acción que realizará el sistema.
Valor de negocio: comunica porque es necesaria la actividad, es decir de qué forma agregar valor al negocio.

Componentes de una User Storie

1. **Tarjeta:** Una descripción de la historia, utilizada para planificar y como recordatorio.

2. **Conversación:** Discusiones acerca de la historia que sirven para desarrollar los detalles de la historia.

3. **Confirmación:** Pruebas que expresan y documentan detalles y que pueden usarse para determinar cuándo una historia está completa.

Detalles de la User Storie

Los detalles de la historia se obtienen de conversaciones entre el dueño del producto y el equipo. Sin embargo, si es necesario más detalle puede proveerse como adjunto en forma de algoritmo, planilla de cálculo, prototipo o lo que sea. Puede obtenerse en el tiempo con discusiones adicionales con los involucrados. *Criterios de aceptación de una US*

Los criterios de aceptación son condiciones de satisfacción ubicadas en el sistema. Son las condiciones que deben cumplirse para determinar que la historia fue desarrollada completamente, de lo contrario, los desarrolladores no tendrían un parámetro para definir si la misma fue cumplimentada o no.

Pruebas de aceptación de una US

Expresan detalles resultantes de las conversaciones entre clientes y desarrolladores; suelen usarse para completar detalles de la US y se ven como un proceso de 2 pasos:

- Notas en el dorso de la US.
- Pruebas muy completas utilizadas para demostrar que la historia se ha hecho correctamente y se ha codificado en forma completa.

Las mismas deben escribirse antes que la programación empiece (las escribe el cliente o al menos especifica que pruebas se utilizaran para determinar si la historia ha sido desarrollada correctamente).

¿Qué NO es una User Storie?

Las user stories NO son especificaciones detalladas de requerimientos (como los UC), son expresiones de intensión, "es necesario que haga algo como esto… ".

- No están detalladas al principio del proyecto
- Necesita poco o nulo mantenimiento
- Pueden descartarse después de la implementación

- Junto con el código sirven de entrada a la documentación que se desarrolla incrementalmente después.

Dividiendo User Stories
Las User Stories frecuentemente se derivan de épicas (**epics**) y características (**features**), conceptos vagos y grandes de algo que queremos hacer para un usuario. No hay una rutina definida para dividir User Stories, sólo lineamientos generales:
- Que sea una pieza de valor para el usuario.
- Que tenga un corte vertical a través del sistema.
- Que entre en una iteración.

Investment Themes (Temas de Inversión)
Representan un conjunto de iniciativas o propuestas de valor que conducen la inversión de la empresa en sistemas, productos, aplicaciones y servicios con el objetivo de lograr una diferenciación en el mercado y/o ventajas competitivas.
Los THEMES son una combinación de inversión en:
- Inversión en ofertas de productos existentes, mejoras, soporte y mantenimiento.

- Inversión en nuevos productos y servicios que mejorarán los beneficios y/o ganarán porciones de mercado en el período actual o al corto plazo.

- Inversión a futuro en productos y servicios avanzados.

- Inversión hoy, pero que no dará beneficios hasta dentro de unos años.

- Inversión en reducción (estrategia de retiro) para ofertas existentes que están cerca del final de su vida útil.

Epics
Son iniciativas de desarrollo de gran escala que muestran el valor de los temas de inversión; son requerimientos de alto nivel que se utilizan para coordinar el desarrollo, son estimadas, priorizadas y mantenidas en el backlog; son planificadas antes de la planificación del release y descompuestas en características (features).
- **Epics de negocio:** son funcionales o de experiencia de usuario.
- **Epics arquitectónicas:** usadas para implementar cambios tecnológicos que deben realizarse a elementos significativos.

Spikes
Las spikes son un tipo especial de historias de usuario (invento de XP), usado para quitar el riesgo e incertidumbre de una US y otra faceta del proyecto; se clasifican en técnicas y funcionales y pueden utilizarse para:
- Inversión básica para familiarizar al equipo con una nueva tecnología o dominio.
- Analizar un comportamiento de una historia compleja y poder dividirla en piezas manejables.
- Ganar confianza frente a riesgos tecnológicos, investigando o prototipando
- Ganar confianza frente a riesgos funcionales, donde no está claro como el sistema debe resolver la interacción con el usuario para alcanzar el beneficio esperado.

Spikes técnicas
Usadas para investigar enfoques técnicos en el dominio de la solución (evaluar performance potencial, decisiones de hacer o comprar y evaluar la implementación de cierta tecnología).
Spikes funcionales
Usadas cuando hay cierta incertidumbre respecto de cómo el usuario interactuará con el sistema, usualmente son mejor evaluadas con prototipos para obtener retroalimentación de los usuarios involucrados.

INVEST Model

 Independent: Calendarizable e implementables en cualquier orden.

 Negotiable: el "qué" no el "cómo".

 Valuable: Debe tener valor para el cliente.

 Estimable: Para ayudar al cliente a armar un ranking basado en costos.

 Small: Deben ser "consumidas" en una iteración.

 Testeable: Demostrar que fueron implementadas.

Estimaciones de Software

Definición de Estimación

Una estimación es una predicción que tiene como objetivo predecir la completitud y administrar los riesgos. Se relaciona con los objetivos del negocio, compromisos y control.

Errores en las estimaciones

 Información imprecisa acerca del software a estimar o acerca de la capacidad para realizar el proyecto

- Demasiado caos en el proyecto (mal definido el proyecto)

- Imprecisión generada por el proceso de estimación.

- Una de las fuentes de error más común es omitir actividades necesarias para la estimación del proyecto tales como, requerimientos faltantes, licencias, reuniones, revisiones, etc.

Consideraciones

 Momentos apropiados para estimar:

- Al inicio del proyecto
- Luego de la especificación de requerimientos
- Luego del diseño

- Una de las actividades más complejas en el desarrollo de software, luego de la definición del software.

- Por definición una estimación no es precisa, la mayor cantidad de veces nos equivocamos al estimar.

- Existe un universo de probabilidades asociado a las estimaciones

- Estimar no es planear y planear no es estimar

- Las estimaciones son la base de los planes, pero los planes no tienen que ser lo mismo que lo estimado.

- A mayor diferencia entre lo estimado y lo planeado, mayor es el riesgo.

- Las estimaciones no son compromisos

- Pasos: Estimaciones - WBS - Calendarización