User Story: vs Caso Uso → Tecnica de Requerimiento

↳ Incluye tanto Requerimiento Func.

↳ Requerimiento Funcionales

ya tenemos descripto el q hace no hay lugar a conversacion y confirmacion.

- CU describe de forma tecnica lo que tiene en software
- Us no se enfoca en lo tecnico, dice las necesidades del cliente del negocio, no dice como se debe hacer paso a paso.

- Conversación: (es lo mas importante de la tecnica)

Tarjeta (Card)

Confirmación

Partes de la tarjeta

- Frase Verbal
- Descripción → Rol - Quien? → Que hace → por que → VALOR DE NEGOCIO
- Criterios de Aceptación
- Pruebas de usuario

Prueba de usuario vs caso de prueba

US de la mano de las metodologías ágiles, busca lograr evitar desarrollar software que no se use.
↳ Por eso el valor de Negocio → Nos guía

Usuario vs Cliente

↓ Este debería ser quien tome decisiones.

↳ es el que paga. Puede ser el usuario en algunos casos.

□ Frase Verbal: Infinitivo, que me permite saber de qué estoy hablando, algo para identificarla

□ Descripción: COMO rol QUIERO que PARA valor de negocio.

Como Cajero quiero cobrar al cliente para finalizar la compra.

! Como el que le da valor a esa persona

! Cada US tiene solo 1 valor de Negocio
↓
Si tengo más es porque estoy tratando con más de US. Debo separarlo.

Si tengo US circular
- es parte de US más grande
- El valor de negocio es más grande

□ CA: condiciones → con las cual el cliente va a acepta el US

- Son todos los criterios que puede seguir el cliente para indicar si el US está bien y le sirve.
(De alguna manera pone) Ej: Caso de Cancelar.

- Son cuestiones de Negocio, no técnicas.

- Describimos una parte de la conversación.

□ Pruebas de Usuario: Indican si el usuario Acepta o no → Una Frase o US una secuencia + Probar + Frase Verbal + Condición → Casos que el usuario puede probar.

Caso de Prueba: Sirven para examinar la calidad
Tiene pasos detallados, ejemplos puntuales
(buscamos errores en el software)
← Anterior a las Pruebas de usuario.

El usuario realiza la prueba y ve el usuario si le sirve o no en función de los criterios de aceptación.

Pruebas Usuarios:

- Probar cobrar al cliente > 500.000 → Debería decir que no. Es una prueba de Feliz → Caso Contrario Prueba Pasa. ↓ Por lo menos 1 que pase

Valoración del usuario. No tiene nada que ver con la calidad que se desarrolló el software.

- Suficientes pruebas de usuario para que el usuario esté seguro que la funcionalidad refleja la US planteada y que le sirve.
- La especificidad depende de lo crítico que sea la funcionalidad.

! Importante para el usuario

↳ Ponerse del lado del usuario. Yo como usuario como valido si acepto o no el software.

Encontrar errores rara → es muy raro → sobre todo la confiabilidad

Para trabajar con US necesitamos cumplir ciertos criterios:

Criterio de Ready. Lo debe definir cada equipo. Indica cuando la US está lista

Para trabajar, la puede tomar el equipo de desarrollo

↓

I Independiente: Cuando la estoy trabajando debe ser independiente de cualquier otra con la que estoy trabajando. Para evitar conflicto en el desarrollo.

N

U

E

S

T

(es raro verlo puro)

Negociable: Manifiesto Ágil → No está escrito en piedra los criterios de Aceptación.

↳ Negociación sobre definición contractual.

Se puede negociar durante el trabajo de la US.

Puede venir por parte del cliente o por parte del equipo de desarrollo

Dentro del alcance de una User Story. — Interesado en el negocio.

Asocar Manifiesto Agil
con Requerimientos Agil

Todo lo que está en la US se puede modificar. (Hasta cerca del desarrollo)

Una metodología Agil por equipo de desarrollo (cada equipo la adapta) → Agil es un enfoque.

¿ Los procesos se deben adaptar a equipo. } en contra del Proceso Definidos

Valuable o que de valor. Todos los US dan valor al Negocio, si no No se hace

Estimable Si yo a esto o no en un Sprint, cuanto tiempo me va a llevar
Si no lo puedo definir de entrada, hay algo mal

Small : Esta en una iteración, Puedo completar la US en la iteracion definida.

→ Testeable
→ Aceptacion
→ Integracion
} Pensar No solo en el Desarrollo

Testeable: Lo puedo probar.

- Cuando el usuario viene con la necesidad, debo pegar el PARA QUE, quizas le doy una mejor solución, Necesitamos saber el Problem de base.

Caso Práctico 1:

Guardar
- ~~Registrar~~ gastos
- Ver Gastos
- Filtrar Gasto
- Registrar Usuario
- Iniciar Sesión
- Modificar Gasto
- Registrar tipo de Gasto
- Eliminar Gasto
- Cerrar Sesión
- Asignar responsable de gastos

Tips:
- ⊗ Registrar → ✓ Como subo, como cargo, como guardo
- No necesariamente son en verbo infinitivo
-

- Nombres del lado del usuario (no tan técnico)
- Como el usuario hablaría de esta aplicación.
- Distinguir Registrar mail / Registro e inicio de sesión con redes
- Eliminar factura No sería US (ilegal)
- Hay cosas que no existen en el lenguaje del negocio. ej: Iniciar Sesión
- No porque 2 user story toquen en la misma pantalla, van a ser la misma
  - ↳ Pregunta si por sí sola aporta valor al negocio.

Tareas ← Épicas ← (US) (Forma para definir requerimientos más grande que a US).

A nivel de desarrollo se divide en Tareas.

A nivel de negocio es ATÓMICO

El rol no necesariamente se traduce a un perfil en la aplicación.

Rol ≠ Perfil

Iniciar sesión / Registrar sesión } puede ser

No es un rol

- Como Usuario quiero Guardar Gastos para llevar registros de los mismos.

Gestador miembro familiar

- Monto
- Descripción.

CA:
- Se debe mostrar el gasto con fecha actual
- Se debe permitir la fecha del gasto, debe ser pasada.
- Se debe indicar el tipo de gasto
- Se debe indicar el responsable del gasto
- Se debe tener sesión iniciada para guardar un gasto
- Se debe guardar un gasto con un tipo y responsable de gasto asociado

Se debe → Indica obligatoriedad

Se puede → Indica Opcionalidad

Que hago mientras registro y fue en otro momento?

P. US:
- Se debe probar cambiar fecha de gasto (pasa)
- Se debe probar ingresar gasto sin responsable (pasa)
- Se debe probar la asignación de fecha actual por defecto. (pasa)
- Se debe probar guardar un gasto con diferente fecha a la actual

! No asumir nada