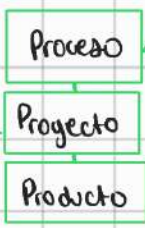


PERSONAS



que tiene que estar todo completo/Definido, qué métricas se van a usar, qué herramientas se van a usar.

Definidos — completitud

Empíricos — filosofía... herramientas

Agile — surge en el software

Lean — Nace en la industria automotriz

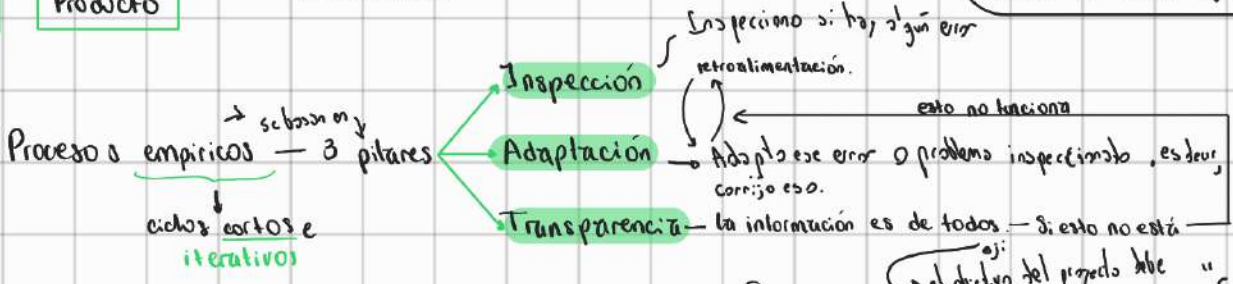
“Lean = liviano”

Ej: PUD, RU.

Proceso Unitario de Desarrollo

Leer Manifiesto ágil

Todo proceso empírico se basa en ciclos de vida iterativos cortos



Filosofía Ágil

NO genera procesos completos

Frameworks — Da pautas. “buenas prácticas”

- Scrum
- DSM
- XP — Xtreme Programming
- ATDD
- Cystal
- FDD

“Scrum no es un proceso empírico o una metodología sino que es un framework hablando técnicamente”

— la mejor comunicación es la CARA A CARA



Se valora más lo personas que los procesos y herramientas

Se valora más lo que está arriba

Product Owner — Representante del cliente

PO — Responsable de la existencia de las HUs. → “User stories”

Dueño del proyecto. No técnico, de negocio

Hasta aquí.

PO es un rol que está en el negocio, no es del equipo de dev.

Alguien que conoce suficientemente el producto o la idea del mismo y que sepa para qué sirve en el negocio.

#masmujeresux @valeuche

12 Principios de agilidad

- Nuestra mayor prioridad es **satisfacer al cliente.**
- Aceptar que los requisitos **cambian.**
- Entregar software funcional **frecuentemente.**
- Los responsables de negocios, diseñadores y desarrolladores deben **trabajar juntos** día a día durante el proyecto.
- Desarrollamos proyectos en torno a **individuos motivados.**
- El método más eficiente de comunicar información es **conversaciones cara a cara.**
- El **software funcionando** es la principal medida de éxito.
- Los procesos ágiles promueven el **desarrollo sostenible.**
- La **atención continua** a la excelencia técnica y al buen diseño **mejora** la Agilidad.
- La **simplicidad** es esencial.
- Las mejores arquitecturas, requisitos, y diseños emergen de **equipos auto-organizados.**
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo y de acuerdo a esto **ajustan su comportamiento.**

compromisos. Haceme cargo y tomar decisiones donde no hay un jefe.

Requerimientos

Emergentes
Conocidos
No conocidos

Significa que aparecen mientras se está haciendo el proyecto.
50%
Un 50% surge de así.

Users stories → Nivel más alto que los cu o nivel de negocio

Requerimientos Ágiles

Usar el "valor" p/ controlar el producto correcto

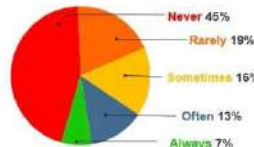
Usar historias y modelos para mostrar que construyes

Determinar que es "suficiente"

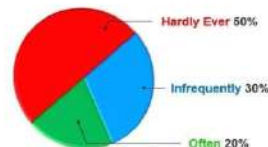
Priorización sobre qué es lo más importante
Si funciona, ya está. No desperdiciar el tiempo

Los productos "exitosos" también tienen un desperdicio significativo

Features / Functions Used in a Typical System
Standish Group Study Reported at XP2002



Features / Functions Used in a Typical System
Standish Group - Updated May 22, 2017



User story, apunta a describir una decisión del usuario

Just in time

Analice cuando lo necesites

No necesitamos todos los requerimientos al inicio

Vamos a entregar los req. que necesita el cliente ahora, no hace falta detallar todos los Req. para entregarlo

Cura a cura

co-localizados — (+) calidad

17% — contenido

83% — contenido

comunicación

"co-localizados" → Significa q estamos trabajando en el mismo lugar

Req. de negocio

Req. de usuario

H0.

Req. de software

Func.
No Func.

El equipo está fijo

gente fija

iteraciones de duración fija y no se cambia
e.g. (solo un mes).

Fijo

Requisitos

Recursos

Tiempo

Dirig. por un plan

Dirig. por valor

Recursos

Tiempo

Alcance

Agil

Agil → El esfuerzo representa el 70% del costo del producto



1- LA PRIORIDAD ES SATISFACER AL CLIENTE A TRAVÉS DE RELEASES TEMPRANOS Y FRECUENTES (2 SEMANAS A UN MES)



2- RECIBIR CAMBIOS DE REQUERIMIENTOS, AUN EN ETAPAS FINALES



4- TÉCNICOS Y NO TÉCNICOS TRABAJANDO JUNTOS TODO EL PROYECTO



6- EL MEDIO DE COMUNICACIÓN POR EXCELENCIA ES CARA A CARA



11- LAS MEJORES ARQUITECTURAS, DISEÑOS Y REQUERIMIENTOS EMERGEN DE EQUIPOS AUTOORGANIZADOS

User Stories

Es un PRODUCTO.
NO tarea.

Leer — Fred Brooks — "No Silver Bullet - Essence and Accidents of Software Engineering". IEEE Computer.

Esto entra en el proceso. Es un capítulo

Partes de una User Story

Son las 3C:

- Conversation
- Card
- Confirmation

Conversación

Tarjeta

Confirmación

esto 2 es lo que queda plasmado en lo que se dijo en la conversación

Evidencias:

Tarjetas bibliográficas:

- En el frente — Tarjeta
- dorso — confirmación.

Tarjeta ejemplo

Tiene que ver con el what

Nombre corto

Formas de expresar las HUs

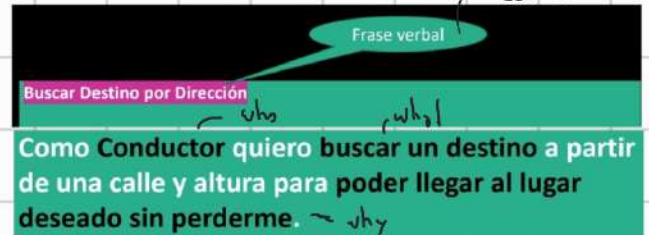
Como < nombre del rol > — who 3 partes fundamentales,

yo puedo < actividad > — what ¿qué necesito es la más imp.

de forma tal que < valor de negocio que recibo > — why

No hay how.

No se dice cómo se va a hacer en la tarjeta.



(+) Criterios de aceptación — Oatreda

(+) notas u observaciones.

Las HUs describen partes del producto entero

Product Backlog — Organiza / Prioriza HUs — PO. es el responsable de priorizar

Ordena las HUs según prioridades

HUs: User Stories

Las pruebas de aceptación nos tienen que demostrar que se cumplen los criterios de aceptación

Las User Stories son Multipropósito

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

Pruebas de accep ≠ criterios de accep.

En metodologías Ágiles, todo varía.

Una HU es grande en función a si se puede realizar en un ciclo de iteración.

Modelado de Roles — Importante. Los roles se describen en una tarjeta.

Las HUs son personas ficticias, significa que le damos que entregar al PO (Product Owner) software que esté funcionando.

Rol de Usuario: Reclutador Interno

Nos es un experto en computadoras, pero bastante adepto a utilizar la Web. Utilizará el software con poca frecuencia pero muy intensamente. Leerá anuncios de otras compañías para averiguar cuál es la mejor palabra para sus anuncios. La facilidad de uso es importante, pero más importante es que lo que aprenda, lo pueda recordar meses después.

Se describe el rol, sus características

Usuarios Representantes (Proxies)

Tipos de usuarios representantes:

- Gerentes de Usuarios
- Gerentes de Desarrollo
- Alguien del grupo de marketing
- Vendedores
- Expertos del Dominio
- Clientes
- Capacitadores y personal de soporte.

evitarlos

cuando se puede, evitarlos.

PO proxies.

Son una formulación de las cosas que el PO nos va a exigir que estén implementadas en esa HU.

Criterios de Aceptación

¿Cuáles son buenos?

Definen una intención

Independientes a la implementación

Nos sirve también para definir los pruebas de aceptación

- Definen límites para una user story (US)
- Ayudan a que los PO respondan lo que necesitan para que la US provea valor (requerimientos funcionales mínimos)
- Ayudan a que el equipo tenga una visión compartida de la US
- Ayudan a desarrolladores y testers a derivar las pruebas.
- Ayudan a los desarrolladores a saber cuando parar de agregar funcionalidad en una US

Buscar Destino por Dirección

Como **Conductor** quiero **buscar un destino** a partir de una calle y altura para **llegar al lugar deseado sin perderme**.

Criterios de Aceptación:

- La altura de la calle es un número.
- La búsqueda no puede demorar más de 30 segundos.

se puede ser más fino. ej: un número int no float

Ejemplo de Tarjeta.

Pruebas de aceptación de HU

Son un acuerdo con el PO

Expresan detalles resultantes de la conversación

Complementan la HU

Proceso de dos pasos:

1. Identificar el dorso
2. Diseñar las pruebas

Buscar Destino por Dirección

Como **Conductor** quiero **buscar un destino** a partir de una calle y altura para **poder llegar al lugar deseado sin perderme**.

Criterios de Aceptación:

- La altura de la calle es un número.
- La búsqueda no puede demorar más de 30 segundos.

Pruebas de Usuario

- Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura existente (pasa).
- Probar buscar un destino en un país y ciudad existentes, de una calle inexistente (falla).
- Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura inexistente (falla).
- Probar buscar un destino en un país inexistente (falla).
- Probar buscar un destino en País existente, ciudad inexistente (falla).
- Probar buscar un destino en un país y ciudad existentes, de una calle existente y demora más de 30 segundos (falla).

Definición de listo - Definition of Ready - DoR

Es una forma de controlar la calidad de lo que hacemos.



Se refiere si está listo la HU. Si está listo, se manda a producción.

INVEST Model

(difícil de implementar)



- Cada HU se puede implementar en cualquier orden, es independiente.
- Significa que la HU está escrita en términos de "qué" quiero y no "cómo" hacerlo.
- Significa que puedo establecer el valor del negocio en esa HU.
- Una HU si es muy compleja o tiene muchos casos que no sabemos cómo hacer, no es estimable. Le debo poder asignar un valor numérico.
- La HU se debe iniciar y terminar en esa iteración.

User Stories — Que no es

- No son especificaciones detalladas de requerimientos (como los casos de uso)
- Son expresiones de intención, "es necesario que haga algo como esto..."
- No están detallados al principio del proyecto, elaborados evitando especificaciones anticipadas, demoras en el desarrollo, inventario de requerimientos y una definición limitada de la solución.
- Necesita poco o nulo mantenimiento y puede descartarse después de la implementación.
- Junto con el código, sirven de entrada a la documentación que se desarrolla incrementalmente después.

Niveles de Granularidad / Abstracción.



No es una clasificación universal y puede ser flexible.

