



Filosofía Ágil

Manifiesto

Ágil

Desarrollo ágil de Software (Agile)

- Un compromiso útil entre nada de proceso y demasiado proceso
(Fowler, 2001)

Procesos Empíricos

Aprender desde el experimento
experiencia
↳ entornos cambiantes



B) Procesos empíricos

VALORES ÁGILES

1 INDIVIDUOS E INTERACCIONES



SOBRE



PROCESOS Y HERRAMIENTAS

2 SOFTWARE FUNCIONANDO



(demoran mucho)
¡FUNCIONA!

SOBRE



DOCUMENTACION EXTENSIVA

3 COLABORAR CON EL CLIENTE



SOBRE



NEGOCIACION CONTRACTUAL

4 RESPONDER AL CAMBIO



SOBRE



SEGUIR UN PLAN

dificultad que vien

lineamiento

2001

Pbmas surgen

- tiros largos
- dificultan ante cambios
- trato equipo/cliente

⇒ 1º 4 valores



partir de valores

L a

Los 12 principios del Manifiesto Ágil

ver manifiesto
Ágil

concreto → poner en práctica

Atados a + valor

- 1 Nuestra mayor prioridad es satisfacer al cliente.

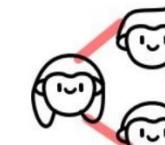
2do y 3er valor



4

- Los responsables de negocios, diseñadores y desarrolladores deben trabajar juntos día a día durante el proyecto.

1º valor



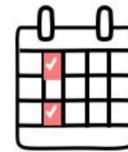
- 2 Aceptar que los requisitos cambien.

4to valor



- 3 Entregar software funcional frecuentemente.

2do valor



- 5 Desarrollamos proyectos en torno a individuos motivados.

7er valor



- 6 El método más eficiente de comunicar información es conversaciones cara a cara.

1er valor



7

- El **software funcionando** es la principal medida de éxito.

2do



8

- Los procesos ágiles promueven el desarrollo sostenible.

1er valor
3er valor



11

- Las mejores arquitecturas, requisitos, y diseños emergen de equipos auto-organizados.



9

- La **atención continua** a la excelencia técnica y al buen diseño mejor la Agilidad.



- 12 A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo y de acuerdo a esto ajustan su comportamiento.

- retroactiva (scrum)

- son obj que quicio lograr
- cultura organizacional
- necesario incorporar*



¿Qué es Ágil?

Buscar auto. aut.

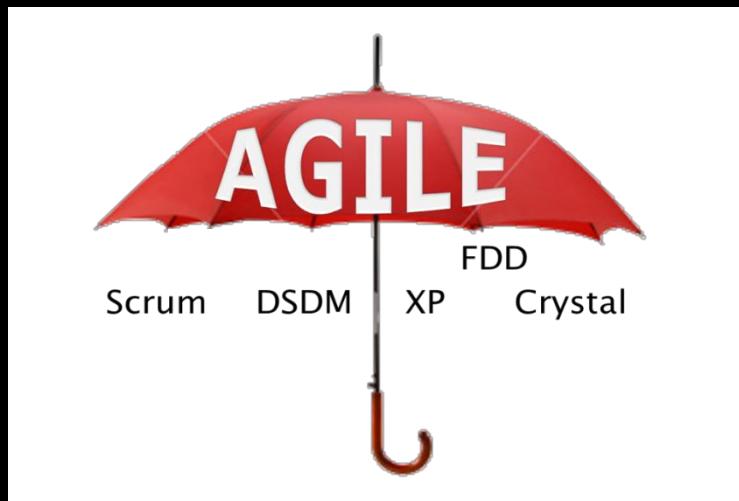
NO es una metodología o proceso Es punto de acuerdo para saber donde ir

Ágil es una ideología con un conjunto definido de principios que guían el desarrollo del producto

Valores de los equipos ágiles ...

- Planificación continua, multi-nivel
- Facultados, auto-organizados, equipos completos
- Entregas frecuentes, iterativas! y priorizadas
- Prácticas de ingeniería disciplinadas
- Integración continua
- Testing Concurrente
- Tamaño pequeño

BUSCAR

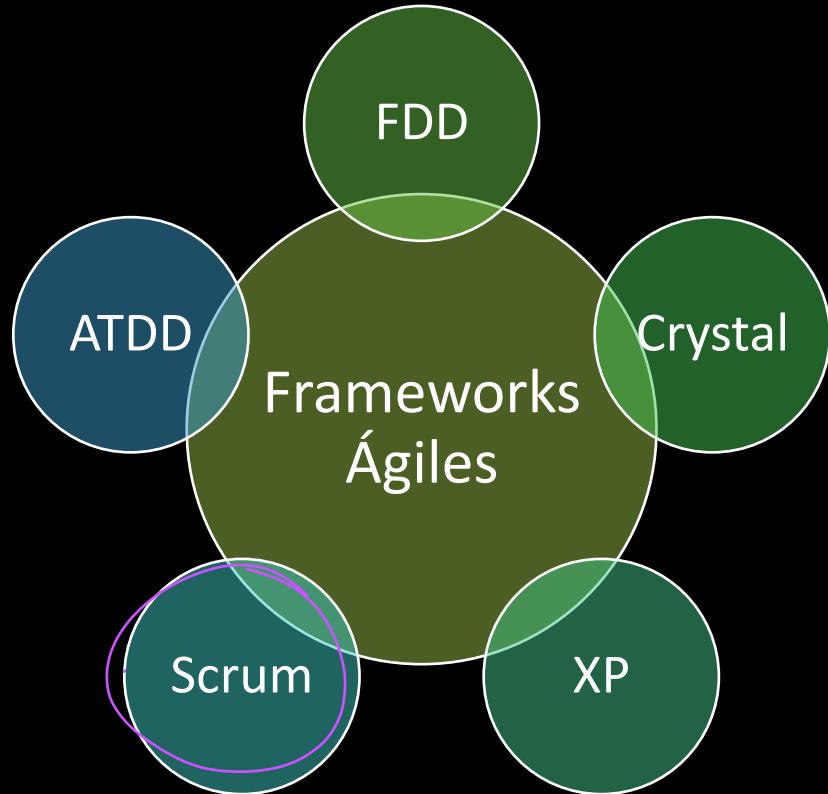


¿Pero qué significa Ágil?

- No son predictivos
- Balance entre ningún proceso y demasiado proceso. La diferencia inmediata es la exigencia de una menor cantidad de documentación, sin embargo no es eso lo más importante:
- Los métodos ágiles son adaptables en lugar de predictivos.
- Los métodos ágiles son orientados a la gente en lugar de orientados al proceso.

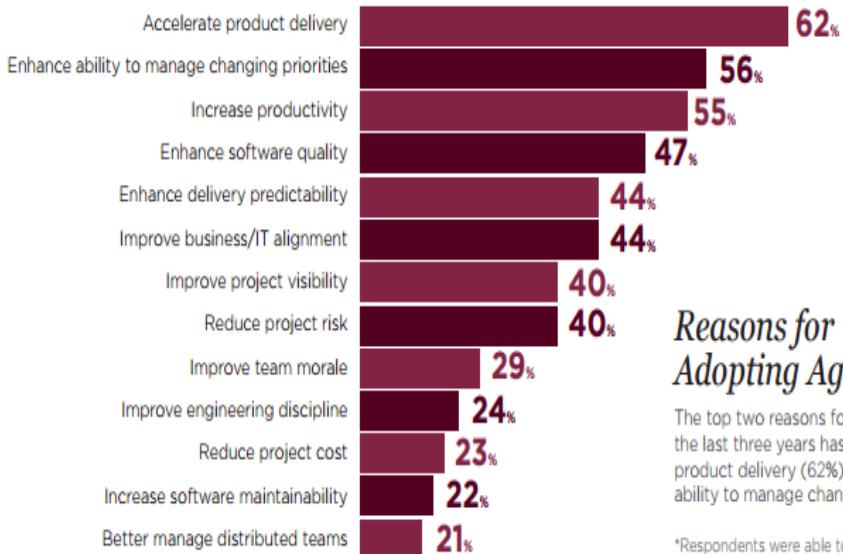
• Proceso definido en 8W:
PUD → predictivo x etapas
e →  → S } Rompe Ágil

Algunos frameworks ágiles



¿Por qué ir a Agile?

metricas que permiten avanzar



Reasons for Adopting Agile

The top two reasons for adopting the last three years has been to aid product delivery (62%) and enhance ability to manage changing priorities (56%).

*Respondents were able to make multiple selections.

Top 3 Benefits of Agile



87%

Ability to manage changing priorities



85%

Increased team productivity



84%

Improved project visibility

AGILE METHODS AND PRACTICES



Agile Methodologies Used

When asked what agile methodology is followed most closely, nearly 70% of respondents practice Scrum (58%) or Scrum/XP hybrid (10%).

Técnicas efectivas

Agile Techniques Employed

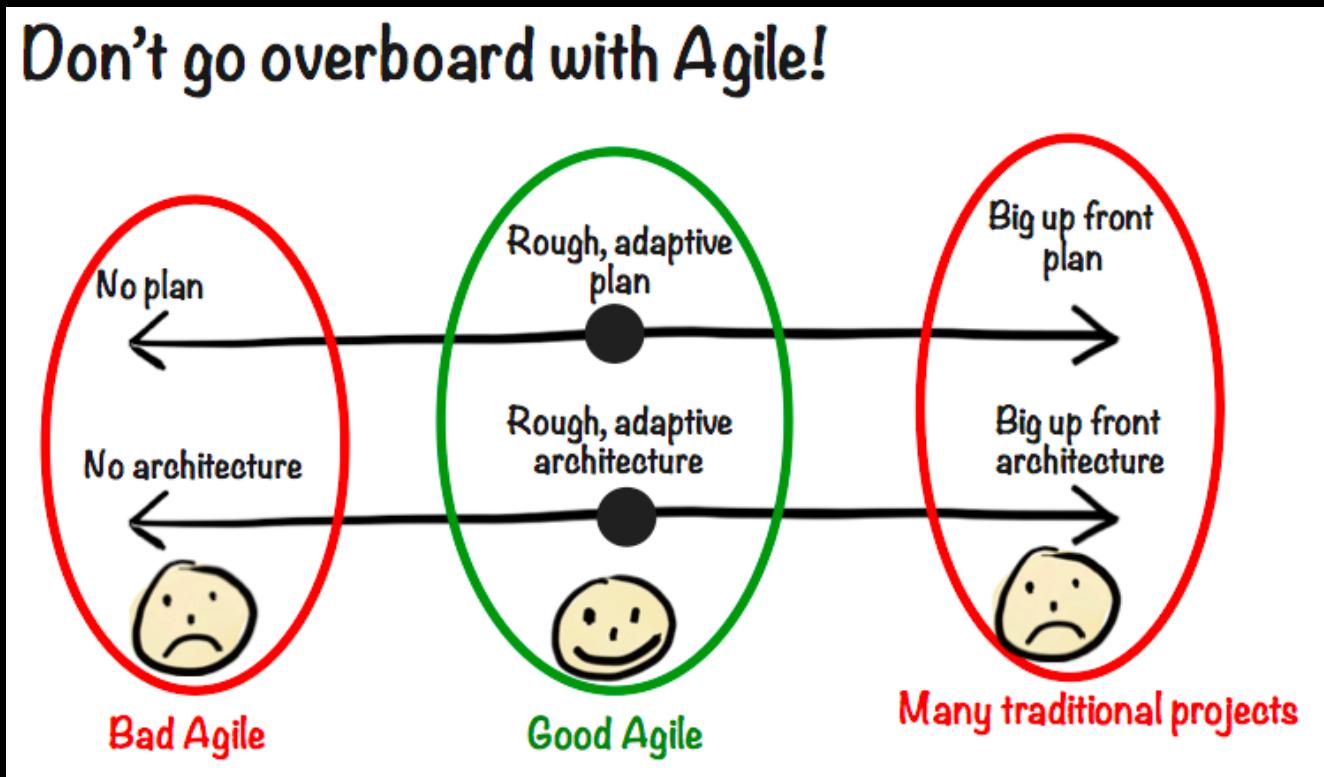
More than 39% of the respondents practiced Kanban within their organizations, up from 31% in 2014. Conversely, iteration planning dropped slightly from 71% in 2014 to 69% in 2015, likely indicating a transition to more flow-based methods such as Lean and Kanban.



Buscamos: Arg. · pequeñas

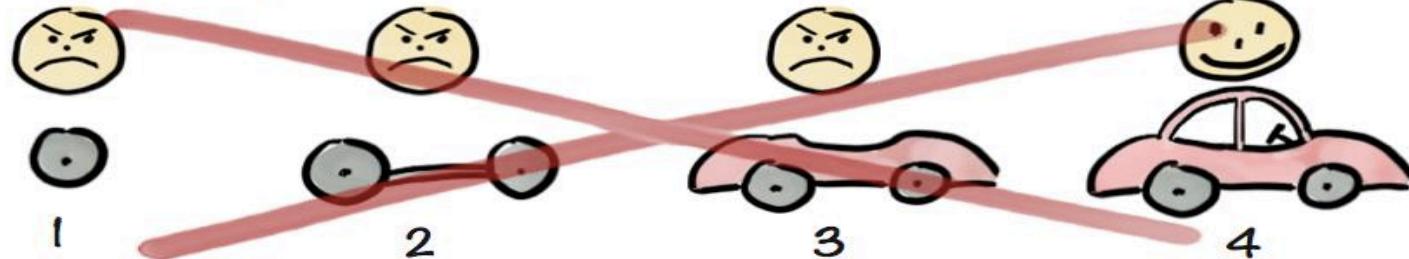
Ser Ágil no es ser indisciplinado.

no ausencia de P, implica organización

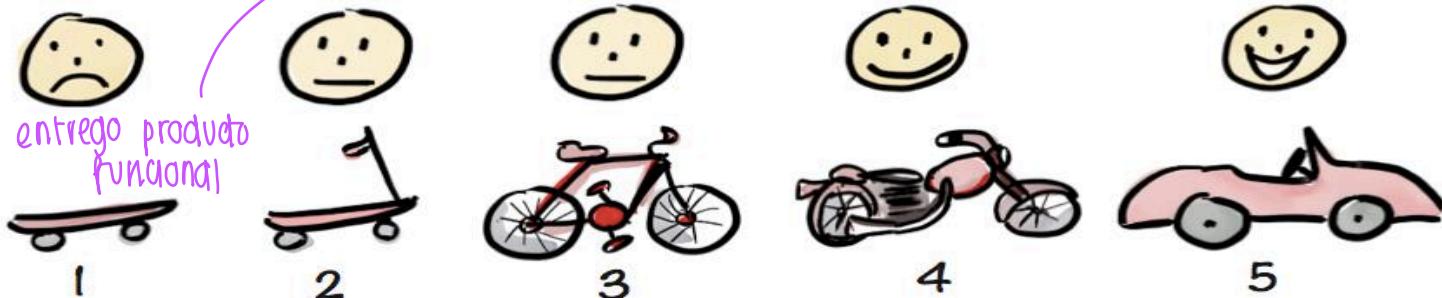


Entonces hacemos todo por pedacitos y somos agiles!!!?? NO!

Not like this....



Like this!





PRACTICO



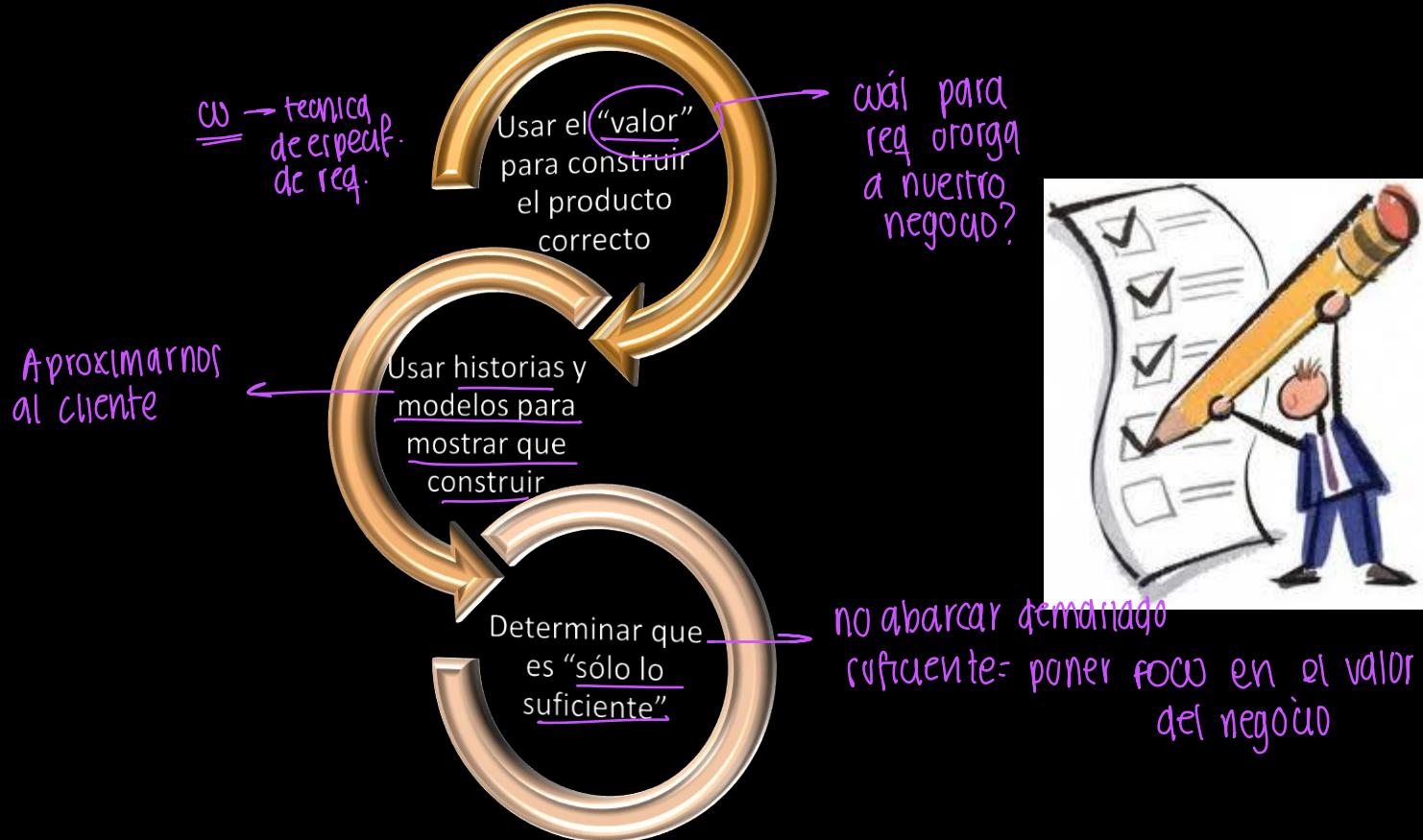
PROBLEMA / NECESIDAD/ OPORTUNIDAD

[PONGA AQUÍ EL
CAMINO MÁS EFECTIVO]



SOLUCIÓN / ENTREGABLE

Requerimientos en Agile



El costo del tradicional BRUF

Medimos features de los productor del mercado

↳ la utilidad es muy

Los productos pequeña
"Exitosos" comparada
"también tienen" lo que es
un desperdicio ↓
significante desperdicio

Excel →

↓ USA
FOCO en lo que quiere ☺



Fuente: Jim Johnson of the Standish Group, Keynote Speech XP 2002

COMO PONEMOS FOCO?

Gestión Ágil de Requerimientos de Software

Los requisitos cambiantes son una ventaja competitiva
si puede actuar sobre ellos

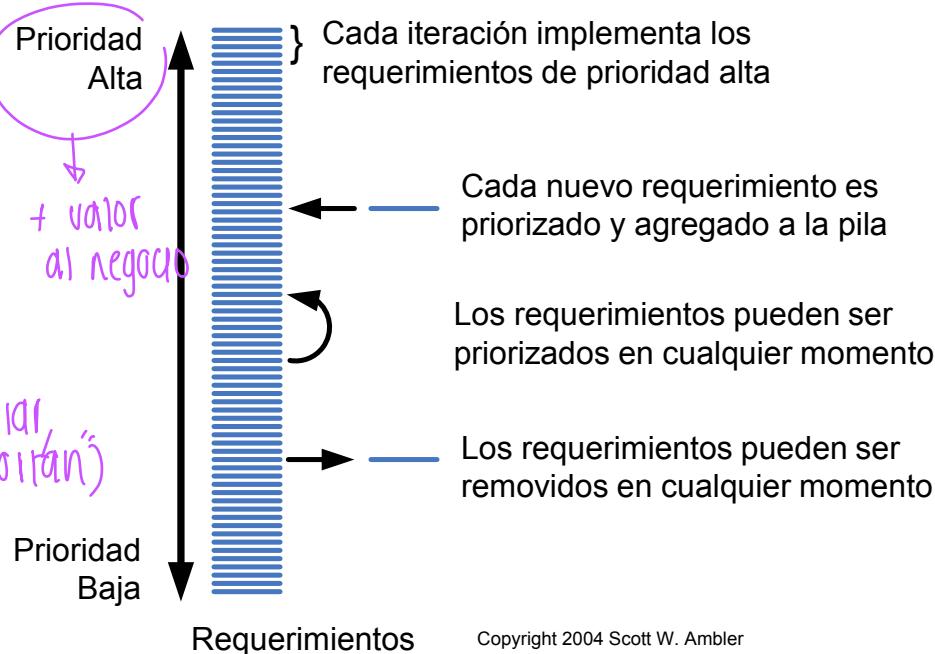
Agil ayuda a gestión
de cambio,
prioridades pueden cambiar

⇒ trabajamos sobre
los requerimientos
que estén encima

(“no pierdo tiempo en
algo que puedo cambiar,
eliminar o no importar”)

FOCO

lista ordenada → prioridad





just in time

Analice cuando lo
necesite, no antes

2
El cara-a-cara permite que fluya información vocal, subvocal, gestual con realimentación rápida.

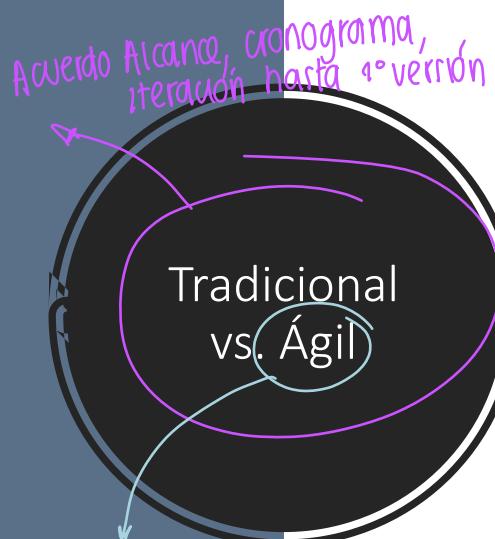


*“Valor es la obtención de beneficio **tangible** o **intangible**”*

Masa Maeda, Serious LeAP

“El valor lo asociamos a la utilidad, beneficio o satisfacción que le ofreces a los usuarios finales por cada funcionalidad completa que le entregas”

Pablo Lischinsky, Agile Trainer & Consultant, Entrepreneur



hacemos entregar
q 3 semanas lo
que alcancemos
en las prioridades



tiempo : se acuerda en equipo. Sunum no tiene P definido pero entre 2/4 semanas

ciclo de vida en cascada → req cambian

FIJO

REQUISITOS

RECURSOS

TIEMPO

ambos iteraciones
pero ↗
T → plan del Alcance
R → recursos y tpo
A → equipo fijo
TQO
mueve el alcance.



Req se modifican
despues.

ESTIMADO

RECURSOS TIEMPO

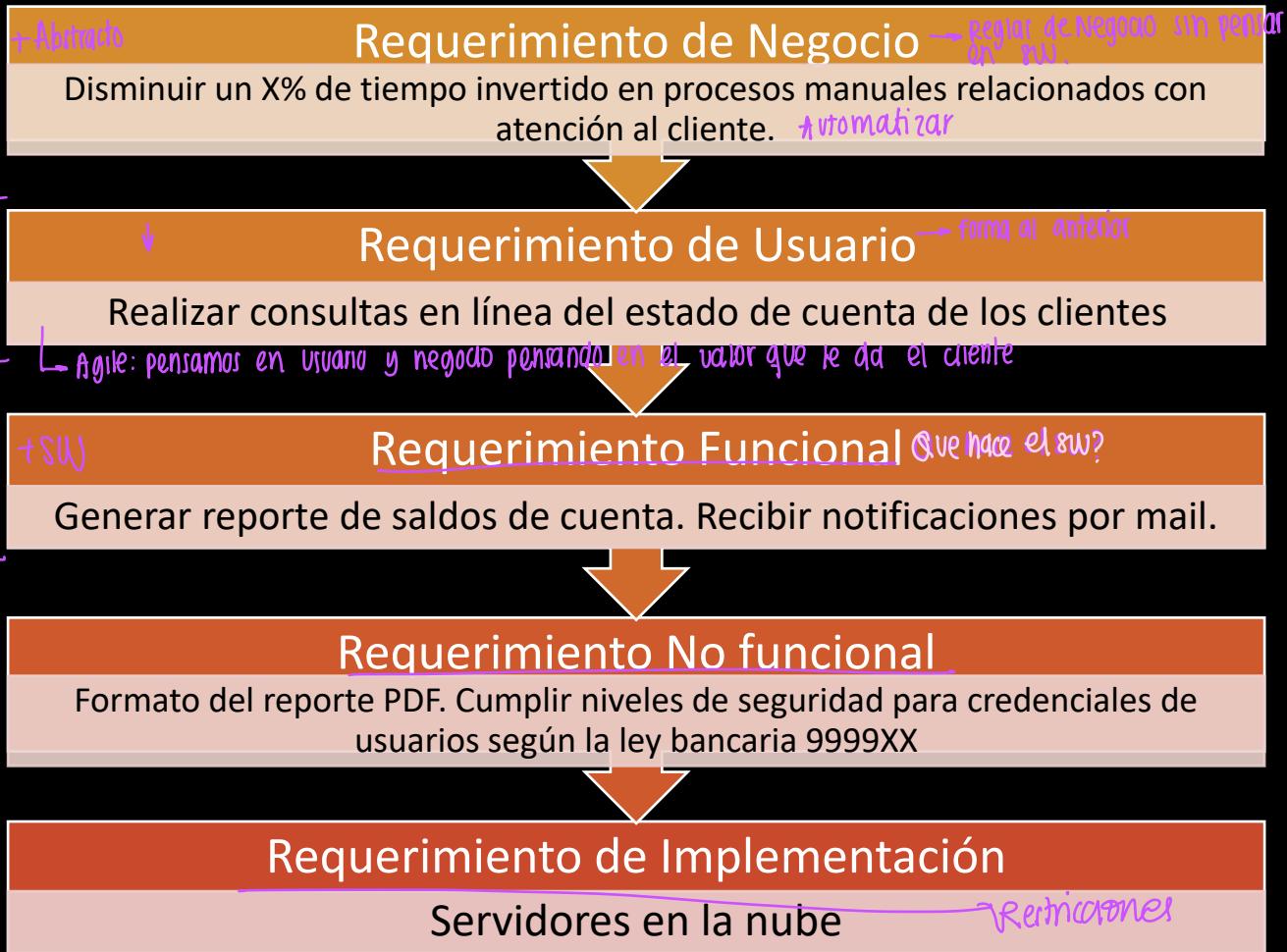
→ entregar temprano y
frecuentes → según prioridades
mi cliente percibe predicibilidad

(A) → me comprometo → (R) y tiempo

ALCANCE

Trabajamos pero
el cambio se hace una
vez que termino la iteración

Tipos de Requerimientos



Algo que necesito
resolver

Requerimientos
de Negocio

Requerimientos de
Usuario

Requerimientos de Software

Dominio
de la
Solución



Dominio
del
Problema



EN RESUMEN...



ENTENDIENDO LA
NECESIDAD Y NEGOCIO...



DESCUBRIENDO LA SOLUCIÓN
DE FORMA COLABORATIVA...

B
E
D
C



JUNTO A UN EQUIPO
MOTIVADO Y COMPETENTE...

A
N



ENTREGAMOS
FRECUENTEMENTE VALOR A
LOS STAKEHOLDERS.

Por último

FOCO: valor del req



Los cambios son la única constante.



→ involucradon no presentes
Stakeholders: no son todos los que están.



Siempre se cumple eso de que: “El usuario dice lo que quiere cuando recibe lo que pidió”.

↳ quien pide es quien mide



No hay técnicas ni herramientas que sirvan para todos los casos.



Lo importante no es entregar una salida, un requerimiento, lo importante es entregar, un resultado, una solución de “valor”.

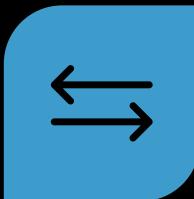
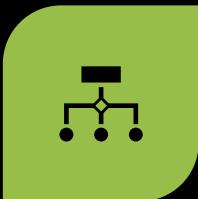
Relación con
los principios

Principios Ágiles relacionados a los Requerimientos Ágiles

ser flexible a
los cambios

Negocio y usuario

→ técnica: Userstories.



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

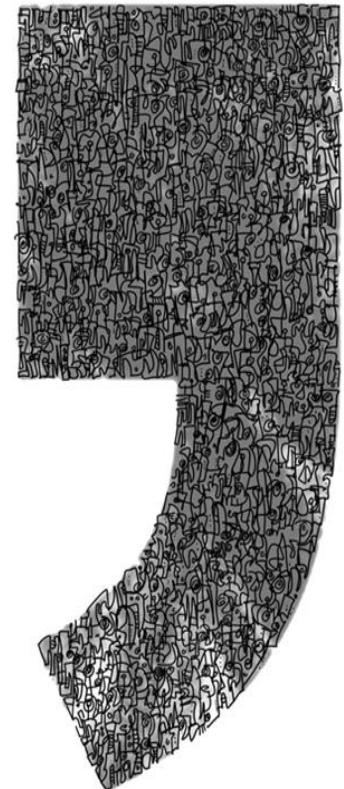
contar historia en función del requerimiento

User Stories

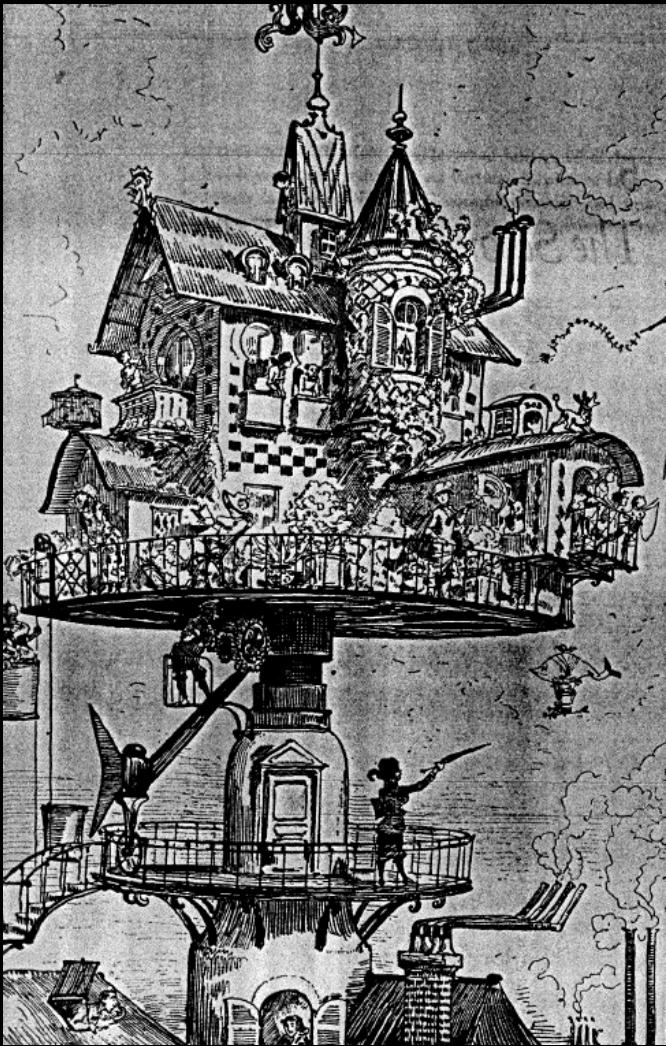
“....se las llama “stories” porque se supone que Ud. cuenta una historia. Lo que se escribe en la tarjeta no es importante, lo que Ud. habla, si!.

--- Jeff Patton, InfoQ,

create conversation.



@gapingvoid



Hace ref. → especificar req +
dificil ↓

(N) Averdo

contexto; como? ?
minimizar pbmas

No
Silver Bullet

→ leer

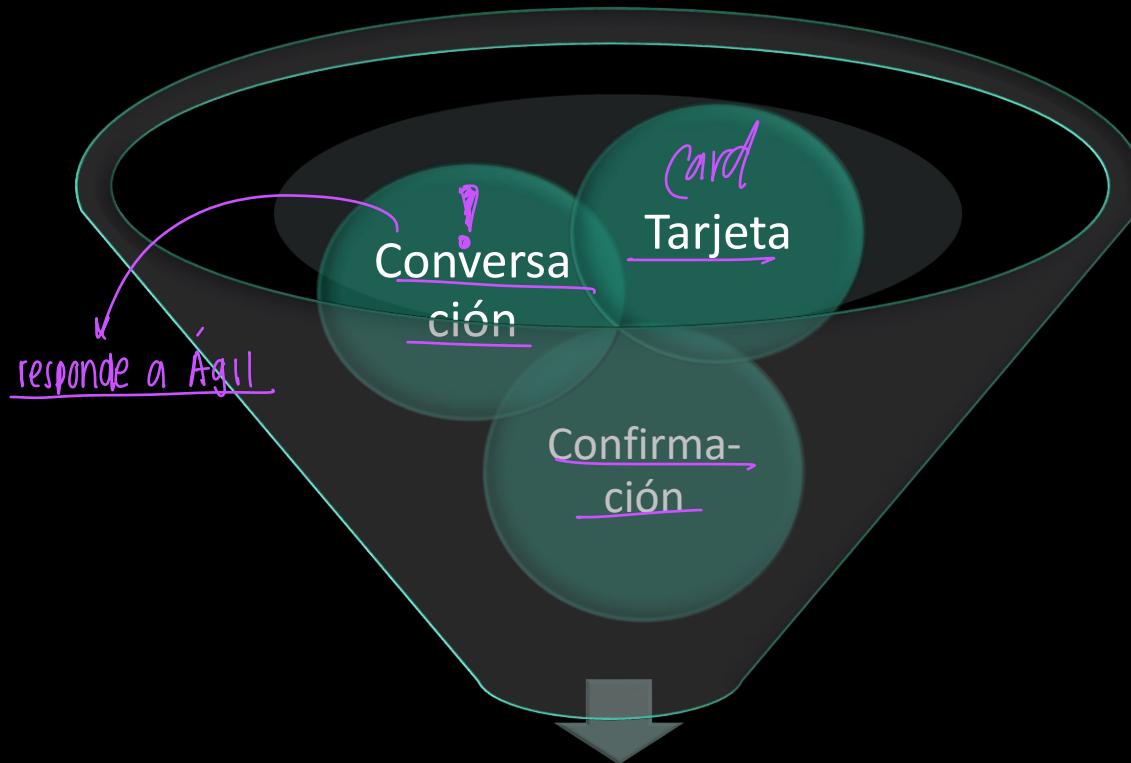
La parte más difícil de construir un sistema de software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan difícil como establecer los requerimientos técnicos detallados... Ninguna otra parte del trabajo afecta tanto el sistema resultante si se hace incorrectamente. Ninguna otra parte es tan difícil de rectificar más adelante”

Fred Brooks - “No Silver Bullet - Essence and Accidents of Software Engineering”. IEEE Computer, Abril de 1987.

¿Cuáles son las partes de una User Story?

TECNICA

3C



User Story

Forma de expresar las Historias de Usuario

1º frase verbal

2º Como <nombre del rol>, yo puedo <actividad> de forma tal que <valor de negocio que recibo>

llegar a destino lo antes posible.

concretamente
Acción.

Comunica porque es necesaria la actividad

mejora la prioridad

As who, I want what so that why.

GPS

Representa quién está realizando la acción o quién recibe el valor de la actividad.

Representa la acción que realizará el sistema

User Story: un ejemplo de tarjeta

Buscar Destino por Dirección

Frase verbal

→ nombre.

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

Las User Stories son Multipropósito

pero no es todo

- Las historias son: frase sirve p/

- Una necesidad del usuario (lo que necesita, expectativa)
- Una descripción del producto (charlas sobre esto)
- Un ítem de planificación (pila del backlog)
- Token para una conversación ..
- Mecanismo para diferir una conversación
 - ↳ tenemos que hablar

* Kent Beck coined the term user stories in Extreme Programming Explained 1st Edition, 1999



El Product Owner Prioriza las historias en el Product Backlog

dueño del producto

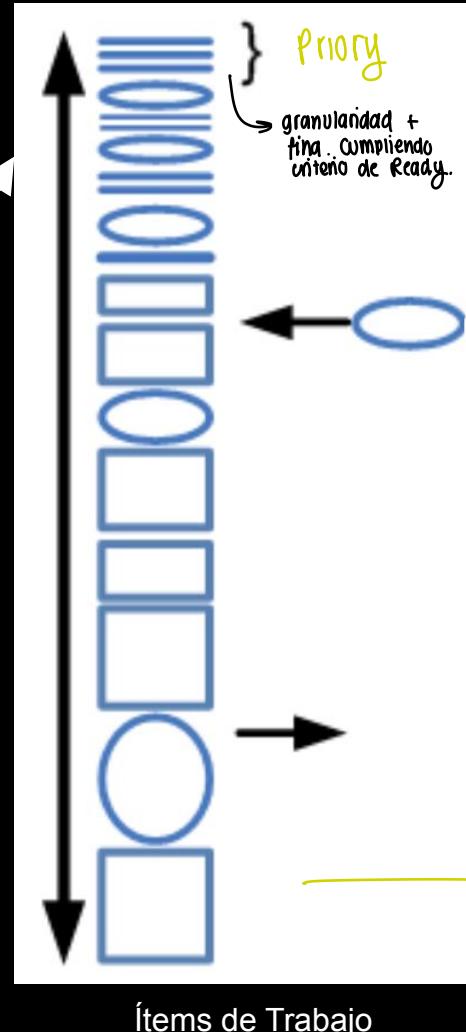
Modelado con mayor detalle

Modelado con menor detalle

listo de N

Prioridad Alta

→ Priorización de necesidades



Cada iteración implementa los ítems de trabajo de mayor prioridad

→ lo que va es del producto
Pueden haber spikes, users, epicas, temas. Bug puede ir

Tareas no pueden porque son del equipo.

Cada nuevo requerimiento es priorizado y agregado a la pila

Los ítems de trabajo pueden ser re-priorizados en cualquier momento

Los ítems de trabajo pueden ser removidos en cualquier momento

No necesario

Caract.

User stories: Porciones Verticales

→ ítem de planificación de valor concreto



planteo un
conjunto de
implementaciones

↓
xq necesito una
funcionalidad en la
iteración

Donde impacta
“fuertemente”
esto??

Story 1	Story 2
GUI	
Business Logic	
Database	

Modelado de Roles



tiene caract.
de acuerdo quién es?
podemos descubrir us

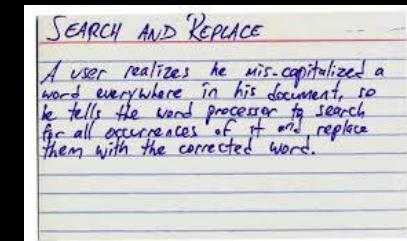


1) Modelado de Roles: Tarjeta de Rol de Usuario

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.

↓ descripción del rol → acostumbrado
frecuencia
prioridades } caract.



Modelado de Roles: Técnicas Adicionales

2)

- Personas — de personas part.

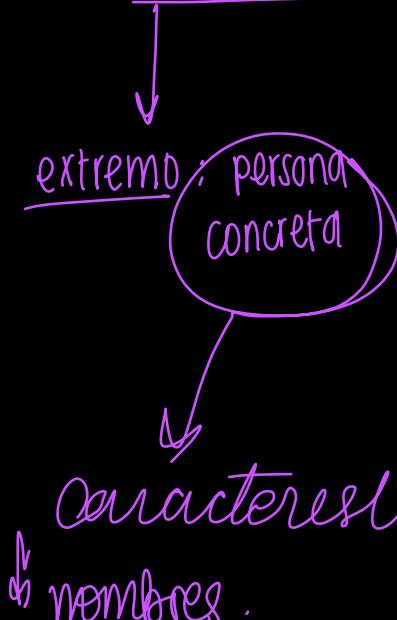
"Imaginar"
↳ quien usa
↳ RU

↓
tipo concreto
↓
Identificar como
va a usar

Mario trabaja como reclutador en el departamento de Speedy Networks, una fábrica de componentes de red de alta gama. El ha trabajado para Speedy Networks por 6 años. Mario tiene un arreglo de horario flexible y trabaja desde casa cada viernes. Mario es muy fuerte con las computadoras y se considera a sí mismo un usuario avanzado de los productos que usa. La esposa de Mario, Kim, está terminando su Doctorado en Química en la Universidad de Stanford. Dado que Speedy Networks ha estado creciendo consistentemente, Mario siempre está buscando ingenieros.



3) Modelado de Roles: Técnicas Adicionales – Personajes Extremos



Diseño de un PDA para:

- El Papa
- Una mujer de 20 años con muchos novios
- Un traficante de drogas

Tanto la mujer como el traficante desearán mantener agendas separadas en caso de que la vea la policía o un novio. El Papa probablemente tenga menos necesidad de discreción pero querrá un tamaño de fuente más grande.

Usuarios Representantes (Proxies)

evitar genéricos → porque necesito materializar

- 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.

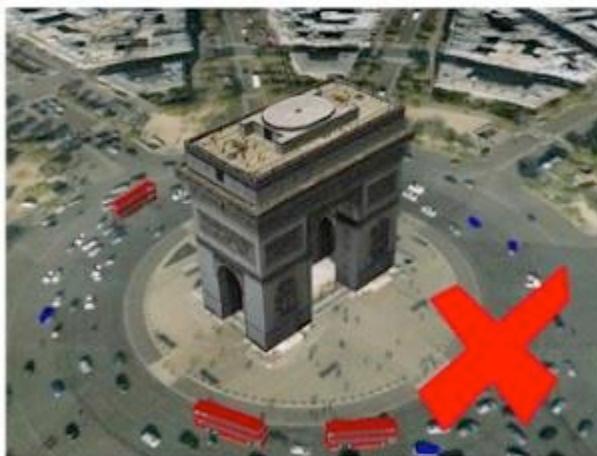
No son ideales como
los usuarios verdaderos
...EVITELOS !!!



① Frase Card conversacion confirm

② Criterios de Aceptación de User Stories

↳ confirmación → lo que tiene que cumplir el producto para saber si ✅ o ❌
↓ Product Owner



Criterios de Aceptación de Historias de Usuario

Cómo definimos?

- 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 que tiene valor y que no
- Ayudan a desarrolladores y testers a derivar las pruebas.
- Ayudan a los desarrolladores a saber cuando parar de agregar funcionalidad en una US

Cómo?



User Story: un ejemplo de tarjeta

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.

son como RNF pero no son
o sea cuando yo pruebe tiene
que cumplir con eso

Wantos + ⇒

dentro prueba

¿Cuáles son los Criterios de Aceptación buenos?

Alto nivel no te dice que hacer?

- Definen una intención, no una solución
 - Ej.: El usuario debe elegir al menos una cuenta para operar
- Son independientes de la implementación
- Relativamente de alto nivel, no es necesario que se escriba cada detalle

!@



¿Y los detalles?

¿Dónde van?



- Detalles como:
 - El encabezado de la columna se nombra “Saldo”
 - El formato del saldo es 999.999.999,99
 - Debería usarse una lista desplegable en lugar de un Check box.
- Estos detalles que son el resultado de las conversaciones con el PO y el equipo puede capturarlos en dos lugares:
 - Documentación interna de los equipos
 - Pruebas de aceptación automatizadas

↙ No tarjeta. Si automático prueba ya lo pongo

derivadas de los criterios

③

Pruebas de Aceptación de User Stories

de usuario final ej: testing in context

versión del usu
prueba o hace.

lo que imaginarios que
va a probar el usuario

Front of Card

Back of Card

ITB

As a student I want to purchase
a parking pass so that I can
drive to school

Priority: ~~Must~~ Should

Estimable: 4

Copyright 2005-2009 Scott W. Ambler

No busco defectos sino que
busco probar que funcione si ingreso algo
que no sea un nro por ejemplo.

van detrás, necesito que
haga lo que me pidió

Pruebas de Aceptación de Historias de Usuario

Expresan detalles resultantes de la conversación

Complementan la User Story

Proceso de dos pasos:

1. Identificarlas al dorso de la US.
2. Diseñar las pruebas completas



User Story: Tarjeta y Pruebas de Aceptación

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:

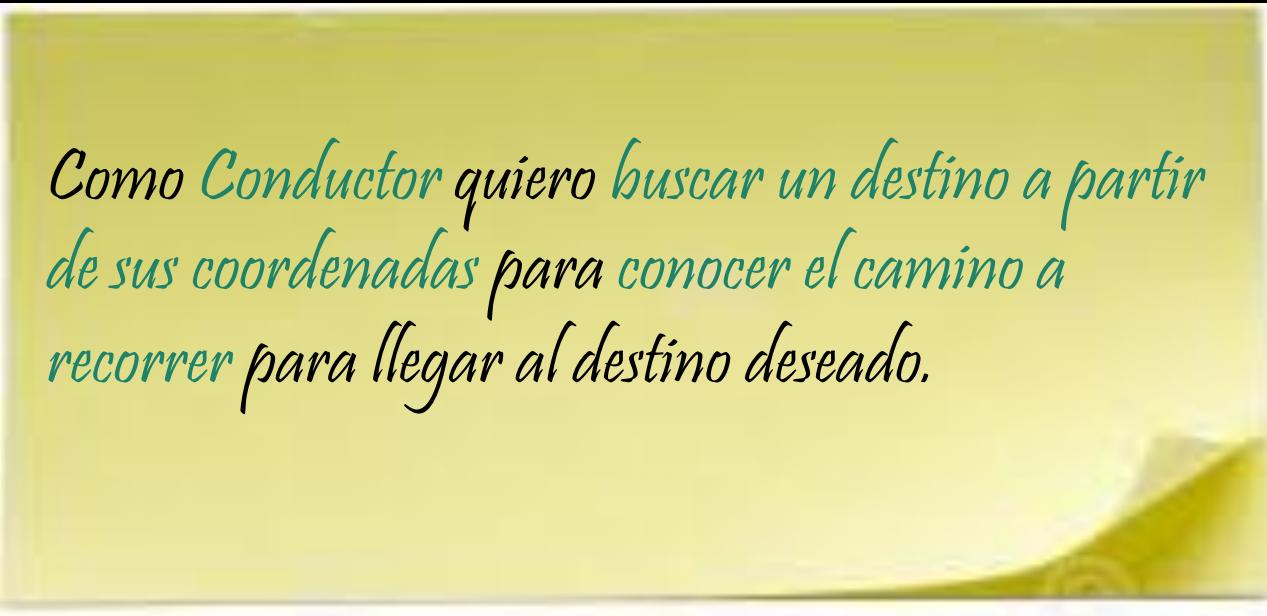
- La altura de la calle es un número.
- La búsqueda no puede demorar más de 30 segundos.

si yo espero ↗
PASE
FALLE

Pruebas de Usuario

- Probar buscar un destino en un país y ciudad existentes, de una calle existente y la altura existente (pasa). *no esperado*
- 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).

Ejemplo: para un software de un GPS



Como Conductor quiero buscar un destino a partir de sus coordenadas para conocer el camino a recorrer para llegar al destino deseado.

Ejemplo:

Como *Conductor* quiero buscar un destino a partir de sus coordenadas para conocer el camino a recorrer para llegar al destino deseado.

Criterios de Aceptación: Las coordenadas se representan con tres números que indican longitud y tres números que indican latitud. Cada número representa los grados, minutos y segundos respectivamente. Además se debe indicar la orientación (norte, sur, este, oeste).

- Probar buscar un destino en un país y ciudad existentes, de dos coordenadas existentes (pasa).
- Probar buscar un destino en un país y ciudad existentes, de una coordenada inexistente (falla).
- Probar buscar un destino en un país y ciudad existentes, de dos coordenadas existentes sin indicar la orientación (falla).
- Probar ingresar coordenadas de latitud y longitud válidas (pasa).
- Probar ingresar coordenadas de latitud y longitud inválidas (falla).

Ejemplo: User Stories / Casos de Prueba

Como compañía quiero pagar por una búsqueda de puestos con una tarjeta de crédito, así resuelvo mi necesidad en forma más eficiente.

Criterio de Aceptación:

- Se acepta Visa, MasterCard y American Express
- En compras mayores de \$100 se piden el número del dorso de la tarjeta

Probar con Visa (pasa)

Probar con MasterCard (pasa)

Probar con American Express (pasa)

Probar con Dinner's Club (falla)

Probar con números de tarjeta buenos

Probar con números de tarjeta malos

Probar con números de tarjeta faltantes

Probar con tarjetas vencidas

Probar con montos menores de \$100

Probar con montos mayores de \$100

CONCEPTO

Definición de listo – Definition of Ready



Acuerdo
de equipo → tamb p/ done

checklist según equipo,
+ alto de pila que cumpla
criterios y lo

incorporado en iteración

puedo trabajar
o no

y si trabajo
puede o no estar
done.

SGDO:
Mi US está definida
de tal manera que
tiene el nivel de det
suficiente p/ ser

Incorporado en una
Iteración

✓ puede según criterio
tiene nivel p/ equipo
+ INVEST
pueda trabajar
Establecimiento b/
Acuerdo equipo según
caract.

Definición de Hecho – Definition of Done

Equipo 10 define



rgdo: cuando user está lista para ser presentada al PO y ejecutar Pruebas de usuario



criterios definidos por equipo



cumplimiento.

2º Done:

Debe poder presentarse (release)

1º Ready:

Debe poder producirse

3º Acept. por PO



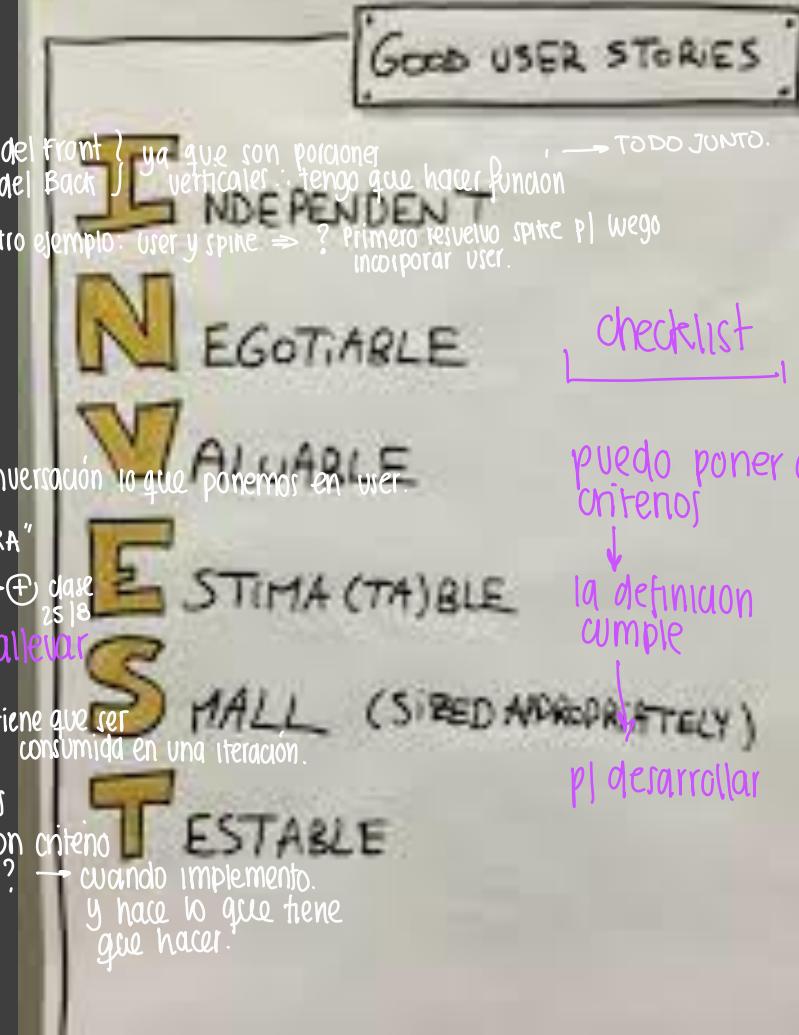
esta en condiciones pero no
incluida efectivamente

usamos def. p/
incluir en prod.

*INVEST Model

- Independent – no necesito desarrollar una pl para hacer otra. calendarizables e implementables en cualquier orden
 - Negotiable – el "qué" no el "cómo" que tengo que hacer necesita prioridad / valor pl negocio "PARA"
 - Valuable – debe tener valor para el cliente
 - Estimatable – para ayudar al cliente a armar un ranking basado en costos → debo poder decir cuanto esfuerzo / tipo / plata me va a llevar. si no puedo es xq grande (?)
 - Small – deben ser "consumidas" en una iteración estimada y entrar en 1 iteración → parible / tiene que ser consumida en una iteración.
 - Testable – demostrar que fueron implementadas funciona como me dijeron
- ⌚ no tengo detalle suficiente

* "INVEST in Stories" – Bill Wake



Algo más sobre las User Stories...

- 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.

Diferentes niveles de abstracción

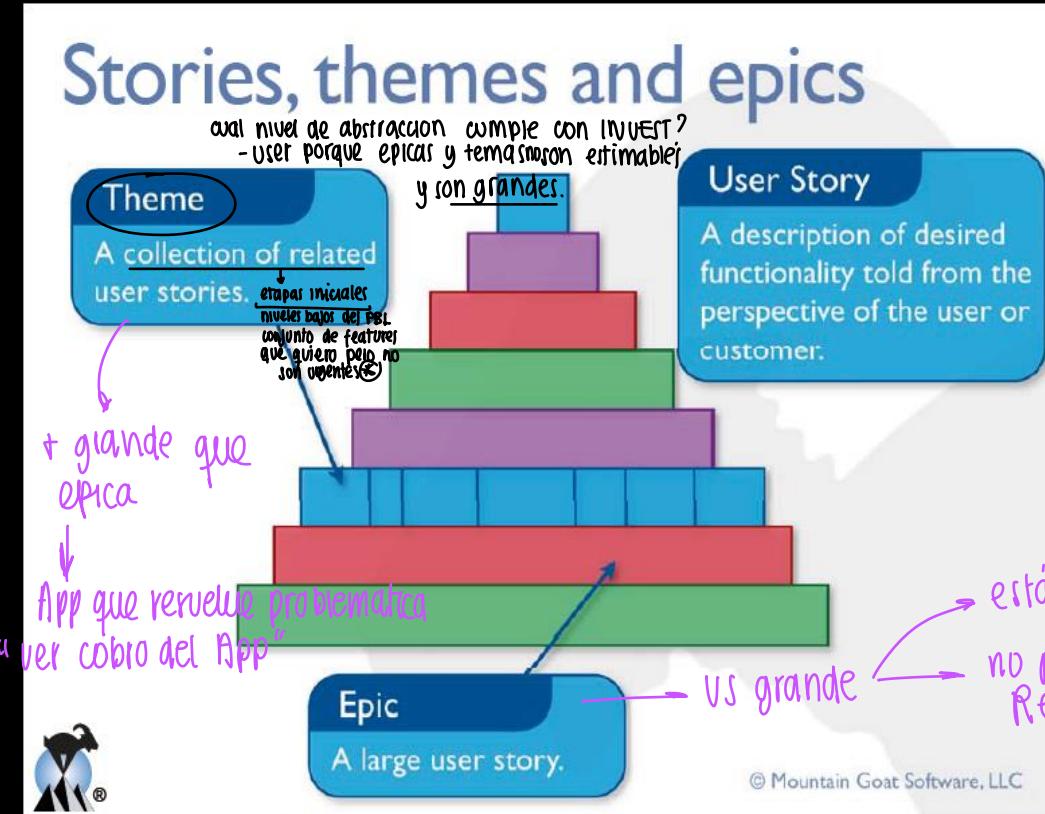
Tema > Epic

PBL → muchas cosas

↓
todo lo del producto

↗
No tengo ranking.

circunscripto al contexto.



Idea

Problema

Necesidad

Cambios en el Negocio

Ideas / Cambios en el Software

Detalle de Implementación

Impacto en el Cliente

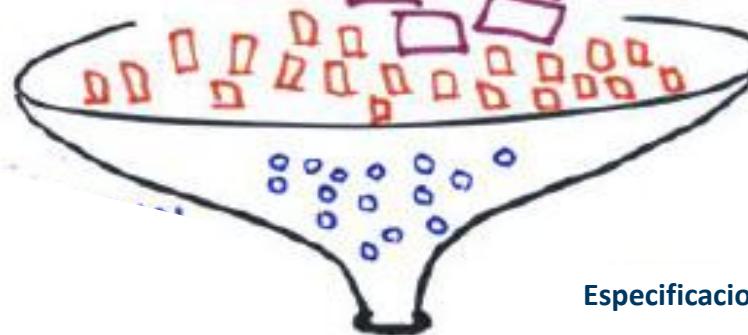
"EPICs"

"STORIES"

Especificaciones

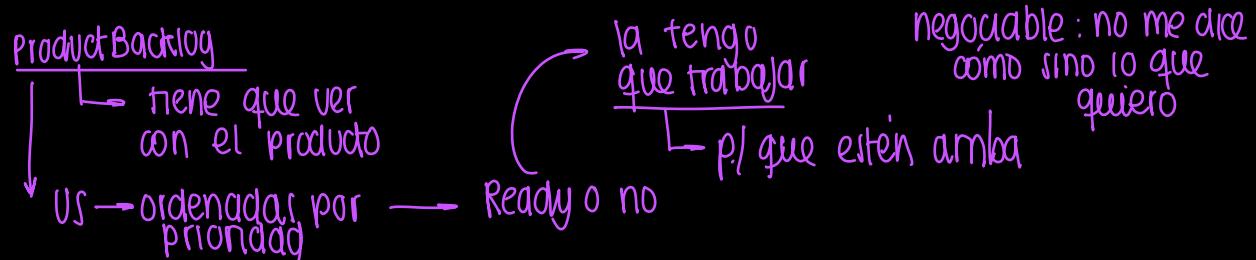
Objetivo

Objetivo



Spikes

Req Agil



- Tipo especial de historia, utilizado para quitar riesgo e incertidumbre de una User Story
u otra faceta del proyecto. → No feature, sino hace referencia a que me permite visualizar funciones o algo que tengo que aprender
- Se clasifican en : técnicas y funcionales. → "necesito investigar como consumir servicio de GM" ↗ incertidumbre
↓ investigaciones ↘ ver si la funcionalidad → usabilidad
- Pueden utilizarse para:
 - Inversión básica para familiarizar al equipo con una nueva tecnología o dominio.
 - Analizar un comportamiento de una historia compleja y poder así dividirla en piezas manejables.
 - Ganar confianza frente a riesgos tecnológicos, investigando o prototipando para ganar confianza.
 - Frente a riesgos funcionales, donde no está claro como el sistema debe resolverla interacción con el usuario para alcanzar el beneficio esperado.

Spikes y US → no estén en la misma iteración
en PBLog

→ resolver incertidumbre

1º resuelve spike
2º luego US

→ no todo spike

si me condiciona

si

↓

Técnicas

Funcionales

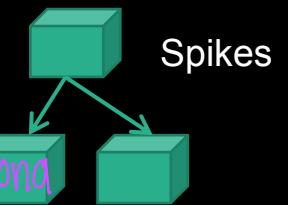
necesito poder estimar

Técnicas

- Utilizadas para investigar enfoques técnicos en el dominio de la solución.
 - Evaluar performance potencial
 - Decisión hacer o comprar
 - Evaluar la implementación de cierta tecnología.
- Cualquier situación en la que el equipo necesite una comprensión más fiable antes de comprometerse a una nueva funcionalidad en un tiempo fijo.

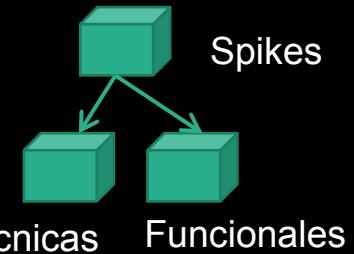
Funcionales

- Utilizadas cuando hay cierta incertidumbre respecto de cómo el usuario interactuará con el sistema.
- Usualmente son mejor evaluadas con prototipos para obtener realimentación de los usuarios o involucrados.



Spikes

US → no está ready hasta que resuelve spike
independiente en concepto de Ready
depende del desarrollador tiene clone y ready
↓
puede generar otras us o spikes



- Algunas User Stories requieren de ambos tipos de spikes. Por ejemplo:
US • Como un cliente, quiero ver mi uso diario de energía en un histograma, para poder comprender rápidamente mi consumo de energía pasado, presente y proyectado.
 └→ valor
- En este caso un equipo puede crear dos spikes:
 - Spike Técnico:
 - Investigar cuánto tiempo requiere actualizar un display de un cliente al uso actual, determinando requerimientos de comunicación, ancho de banda y si los datos se actualizan en formato push o pull. ⇒ salida: rta a preguntas, valor → documentado
 - Spike Funcional:
 - Crear un prototipo de histograma en el portal web y obtener la retroalimentación de algunos usuarios respecto del tamaño, el estilo de la presentación y los atributos gráficos.

└→ prototipo

Lineamientos para Spikes

Estimables, demostrables, y aceptables

resultado que obtengo

La excepción, no la regla → no lo hago para todo

- Toda historia tiene incertidumbre y riesgos.
- El objetivo del equipo es aprender a aceptar y resolver cierta incertidumbre en cada iteración.
- Los spikes deben dejarse para incógnitas mas críticas y grandes.
- Utilizar spikes como última opción.

Implementar la spike en una iteración separada de las historias resultantes

necesito otro certero para resolver spike y wedge user fronte

- Salvo que el spike sea pequeño y sencillo y sea probable encontrar una solución rápida en cuyo caso, spike e historia pueden incluirse en la misma iteración.

Algunas cosas para dejar en claro

wando no conozco
tecnología

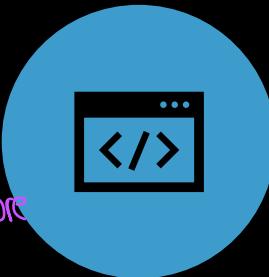


Riesgo a

reducir → estimación

→ siempre tengo incertidumbre
↓ tpo, esfuerzo → lo sé en base a mi
conocimiento, equipo, tu
de trabajo.

no hago spike por
las dudas.



Esfuerzo

④ DIFERIR EL ANÁLISIS
DETALLADO TAN TARDE
↓ COMO SEA POSIBLE, LO QUE
CERTEZA ES JUSTO ANTES DE QUE EL
en medida que avanza. TRABAJO COMIENCE.

Agil → gestión del cambio
pongo foco en lo que
trabaja ahora sino tengo
que detallar

HASTA ENTONCES, SE
CAPTURAN
REQUERIMIENTOS EN LA
FORMA DE "USER STORIES".

no son req de sw. No lo describen. Son req.
de usuario/negocio.

Temas: los pongo pero no los detallo

LAS USER STORIES NO SON
REQUERIMIENTOS DE
SOFTWARE, NO NECESITAN
SER DESCRIPCIONES
EXHAUSTIVAS DE LA
FUNCIONALIDAD DEL
SISTEMA.

Tips para que las user stories sean útiles para el equipo



Un paso a la vez (evitar la palabra "Y") evitar xq mudar user



Usar palabras claras en los criterios de aceptación



No olvides la parte invisible: la conversación → no se ve escrita lo que no entendamos se pregunta PD



Las user stories se escriben desde la perspectiva del usuario



No forzar todo para escribirlo como user stories

Nota:

P0 : representa al cliente.

Material Bibliográfico de Referencia

- Libro:

- Cohn Mike - USER STORIES APPLIED – Editorial Addison Wesley 2004-
Capítulos 1, 2 y 6

- Papers

- Dean Leffingwell and Pete Behrens – A user story primer (2009)

- Link

- <http://www.mountaingoatsoftware.com/>

↓
1170

empleo renom.

- Puntos de historia ↓
Estimación
↓
(desp)

Tarea :
leer MA

Estimable → Recursos
Tpo
Equipo

Dificultad de estimaciones: no son exactas
└ componente: incertidumbre