



TALLER SCRUM DEVELOPER

SMC, SPOC, AMC, ESMC, PMP Harvey Aquino Mas

Exam Format

- Multiple choice
- 75 questions per exam
- No negative marks for wrong answers
- 90 minutes duration
- Proctored online exam
- Current pass rate: 98%

Agile como una Cultura



Experiencias en Agile



Fuente:

<http://dendritasoft.com/logos/>

El Círculo Dorado del Cambio

basado en Simon Sinek



QUÉ

Iniciativas en todos los niveles
Opciones y experimentos

QUIÉNES

Coalición del Cambio
Agentes de Cambio como un equipo ágil

CÓMO

Lean Change Management
Ciclos de experimentación y aprendizaje

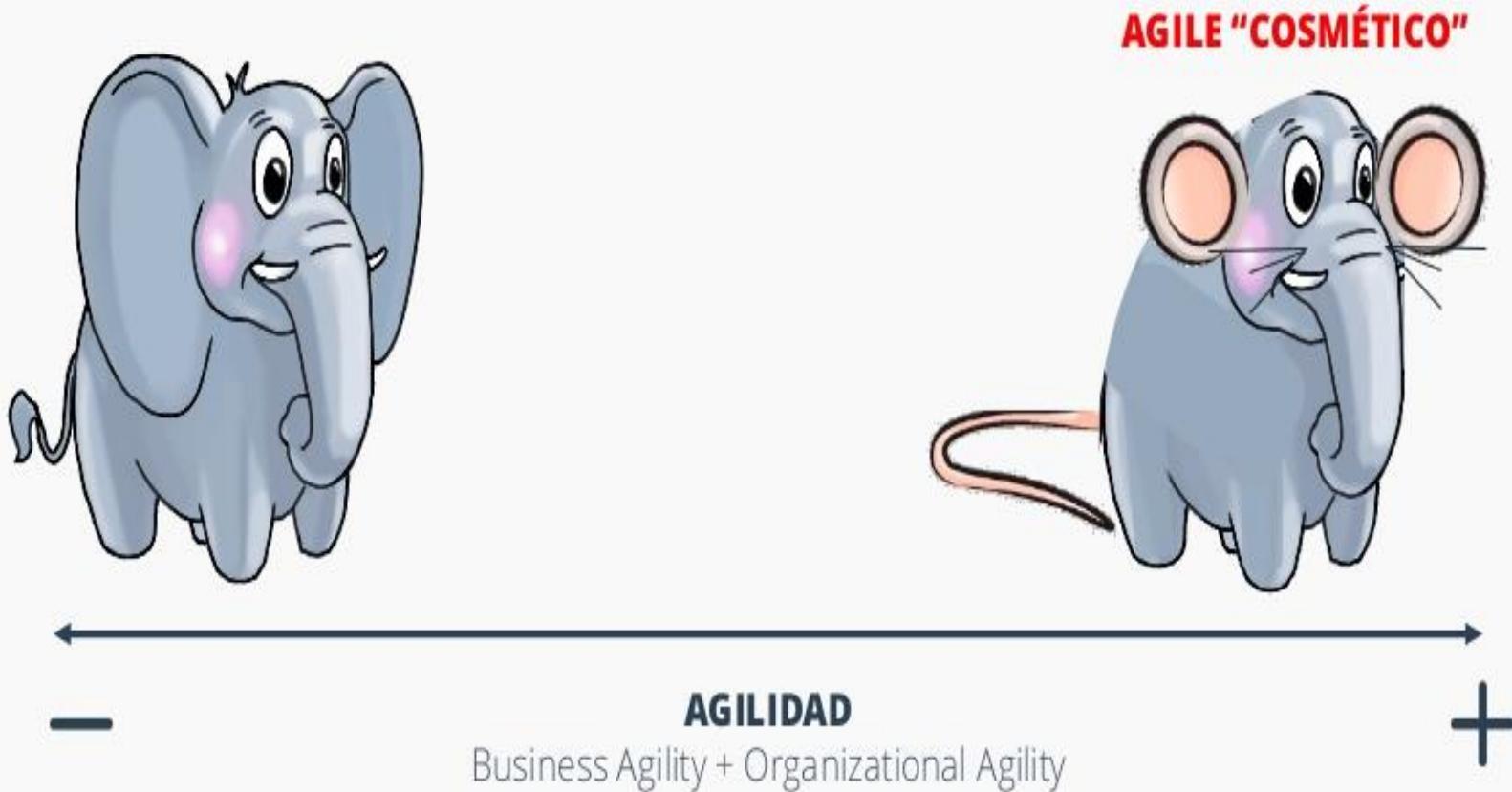
POR QUÉ

Enfocarse en Transformación, no sólo adopción
Evitar el “Agile cosmético”
Crear resultados de negocio sostenibles
Agile es sobre personas

¿Y qué tan Ágiles somos ahora?



¿Y qué tan Ágiles somos ahora?



POR QUÉ

Enfocarse en Transformación, no sólo adopción

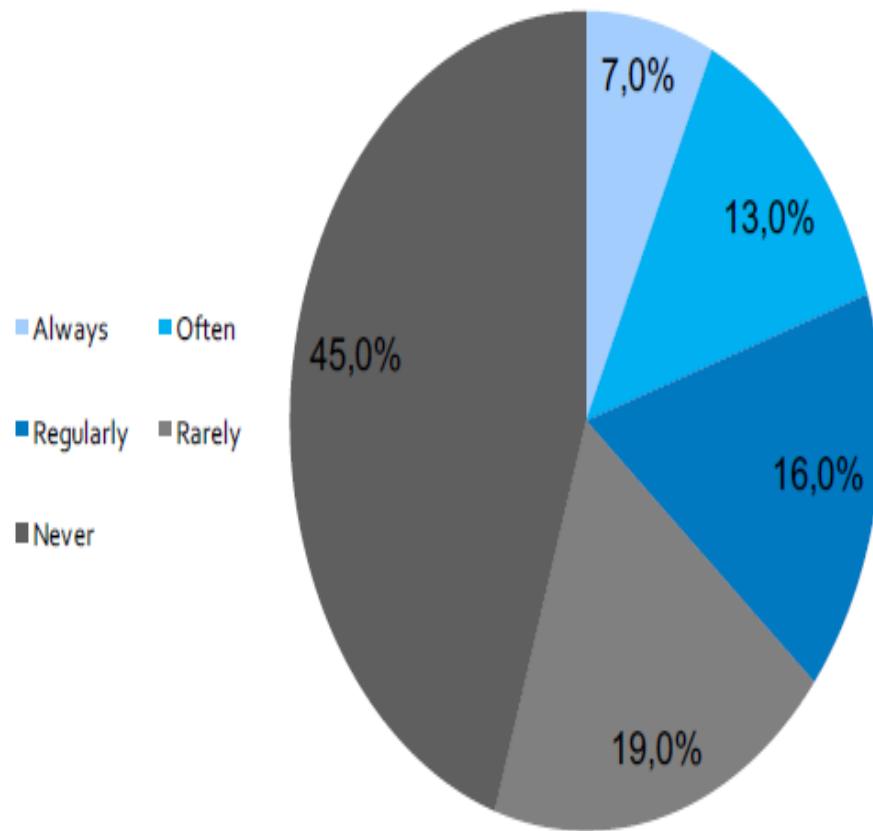
Evitar el “Agile cosmético”

Crear resultados de negocio sostenibles

Agile es sobre personas

Uso de Características en un Sistema de Software

Chaos Report 2011, Standish Group



A menudo /
Siempre usado **20%**

Raramente /
Nunca usado **64%**



80% del Valor de Negocio está ubicado en el 20% de las Features (*Pareto*).

Busque el 20%

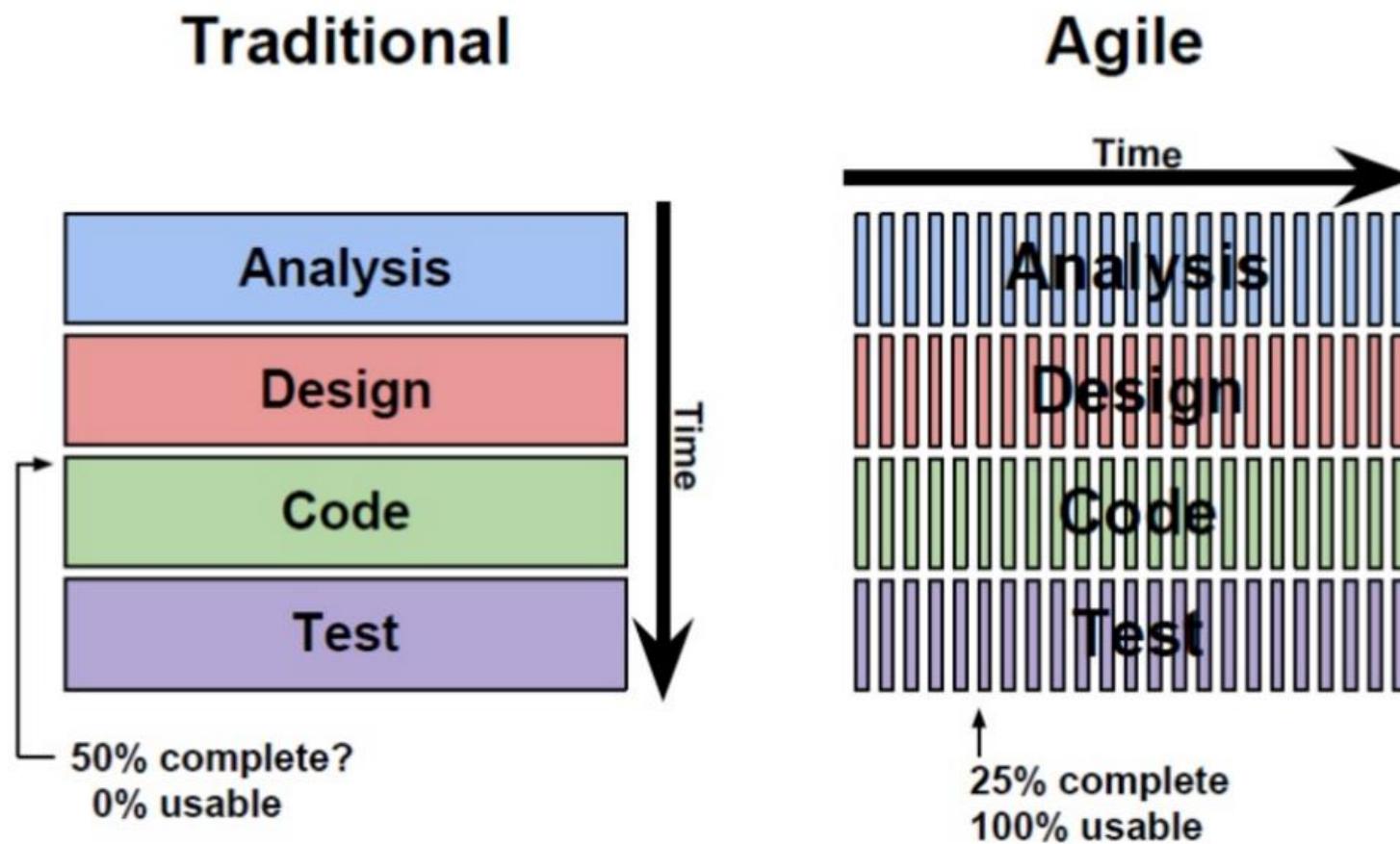
Desarrollo iterativo e incremental

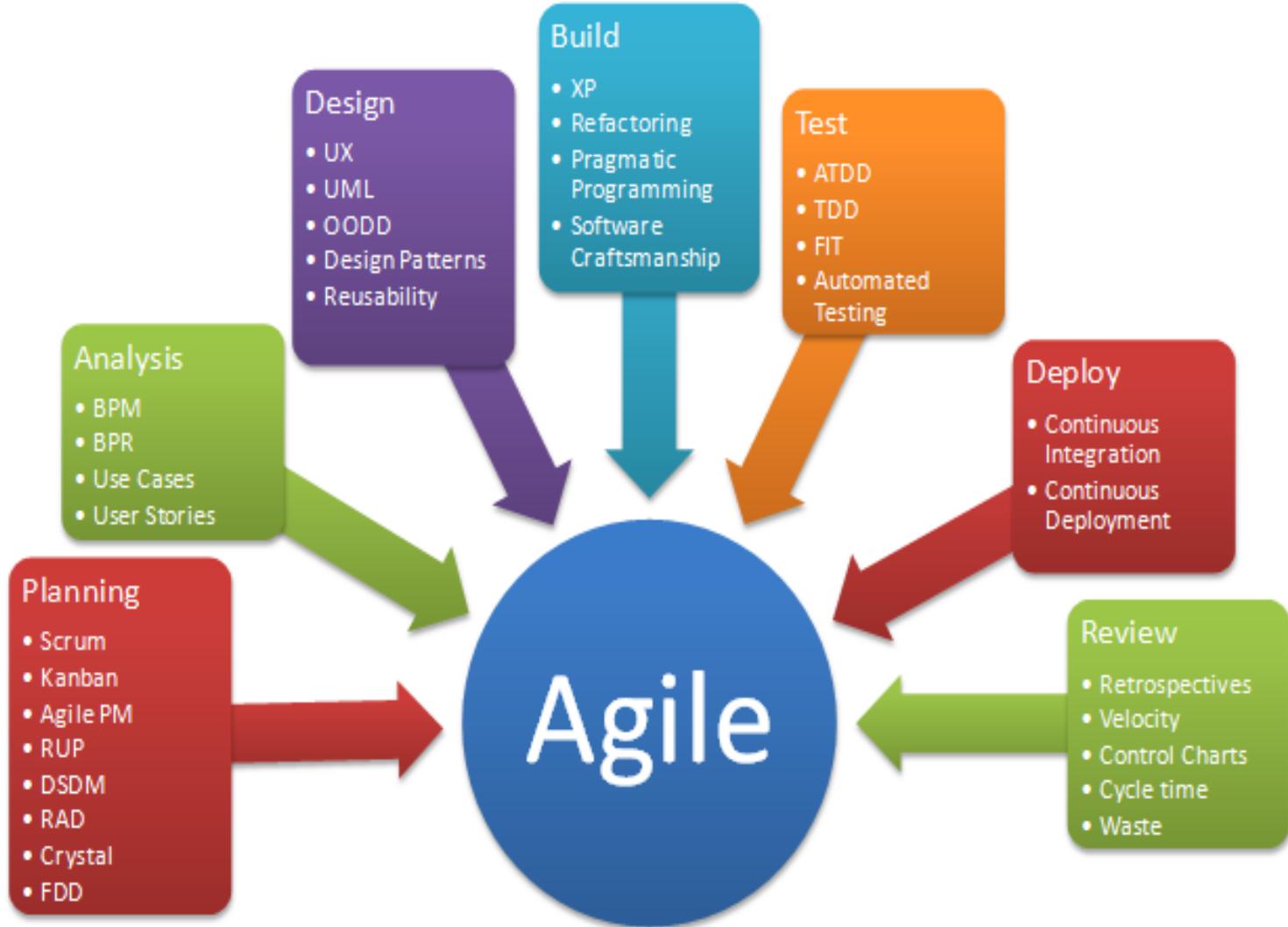
En un desarrollo iterativo e incremental el proyecto se planifica en diversos **bloques temporales** (en el caso de Scrum de un mes natural o hasta de dos semanas, si así se necesita) llamados iteraciones.

Las iteraciones se pueden entender como miniproyectos: **en todas las iteraciones se repite un proceso de trabajo similar** (de ahí el nombre “iterativo”) **para proporcionar un resultado completo sobre producto final**, de manera que el cliente pueda obtener los beneficios del proyecto de forma incremental.

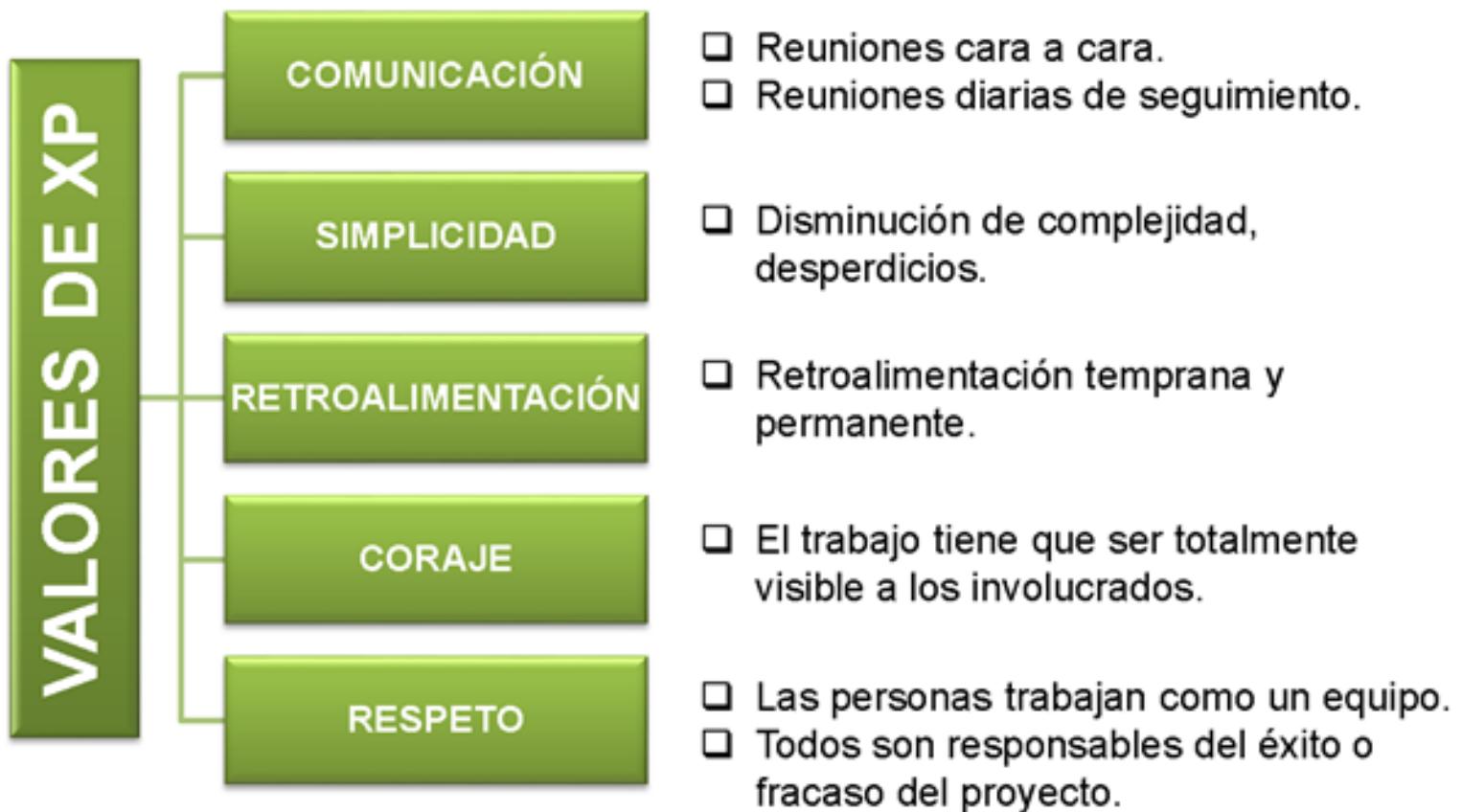
Para ello, cada requisito se debe completar en una única iteración: el equipo debe realizar todas las tareas necesarias para completarlo

Tradicional vs Agile





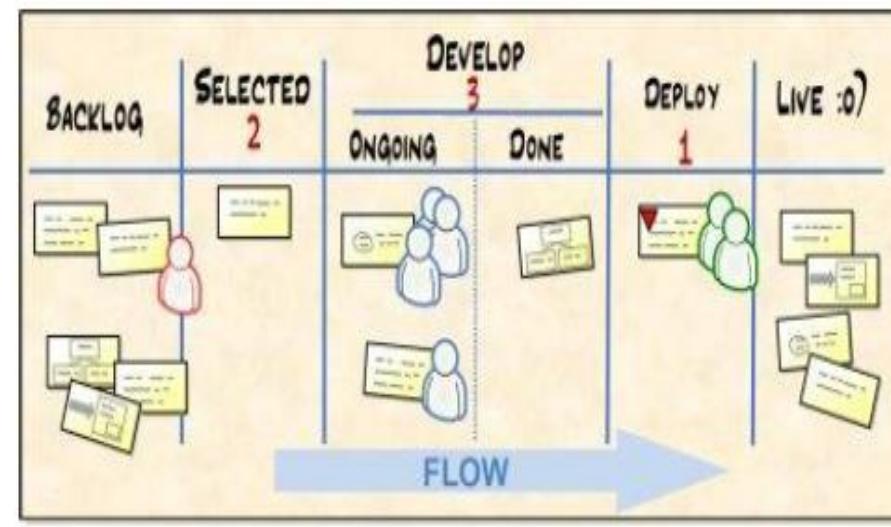
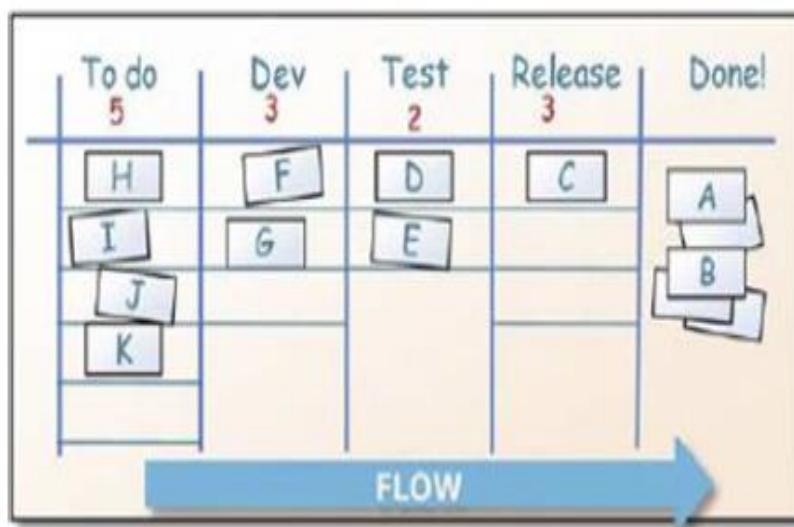
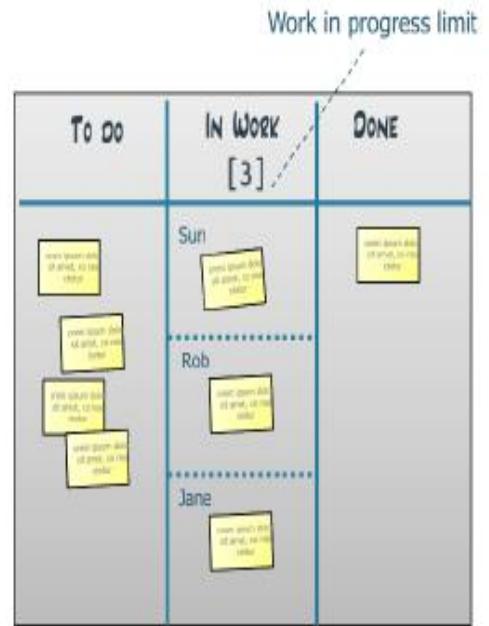
Extreme Programming (XP)



KANBAN

Del japonés kanban: *kan* "visual" y *ban* "tarjeta".

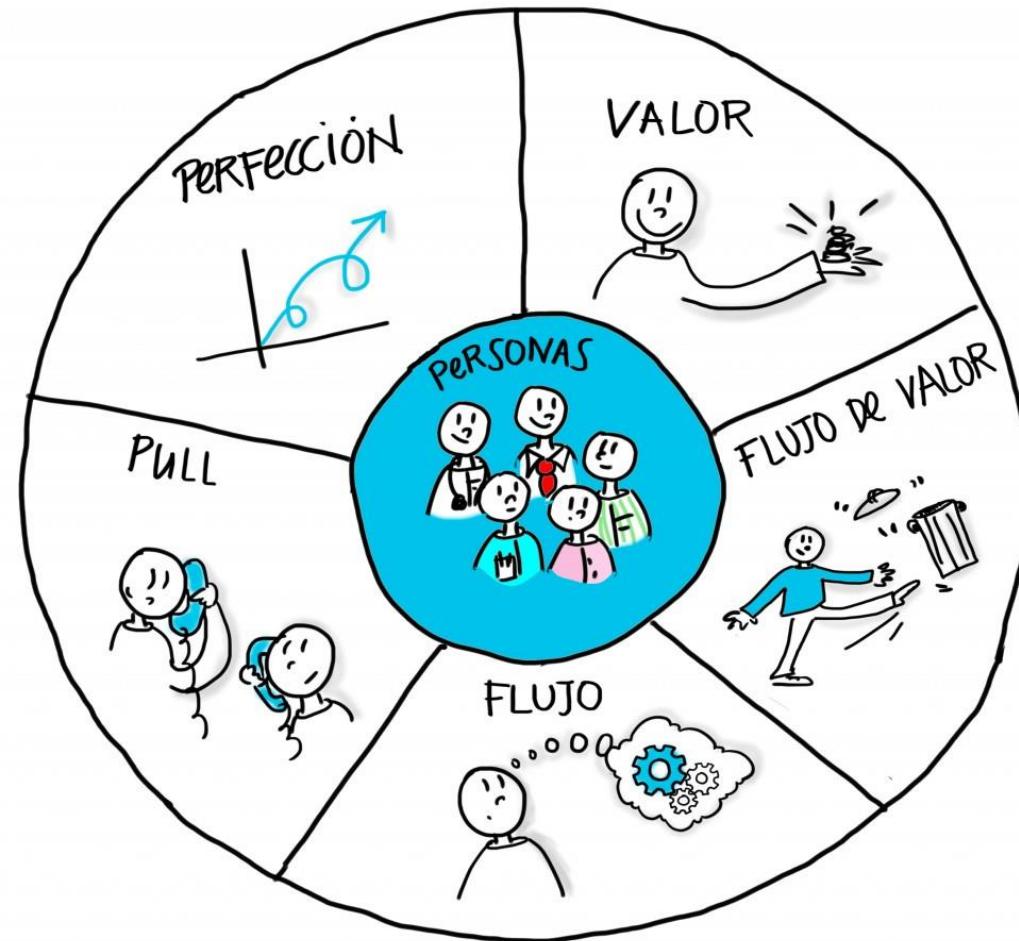
Utilizados en las empresas como el "sistema de datos".



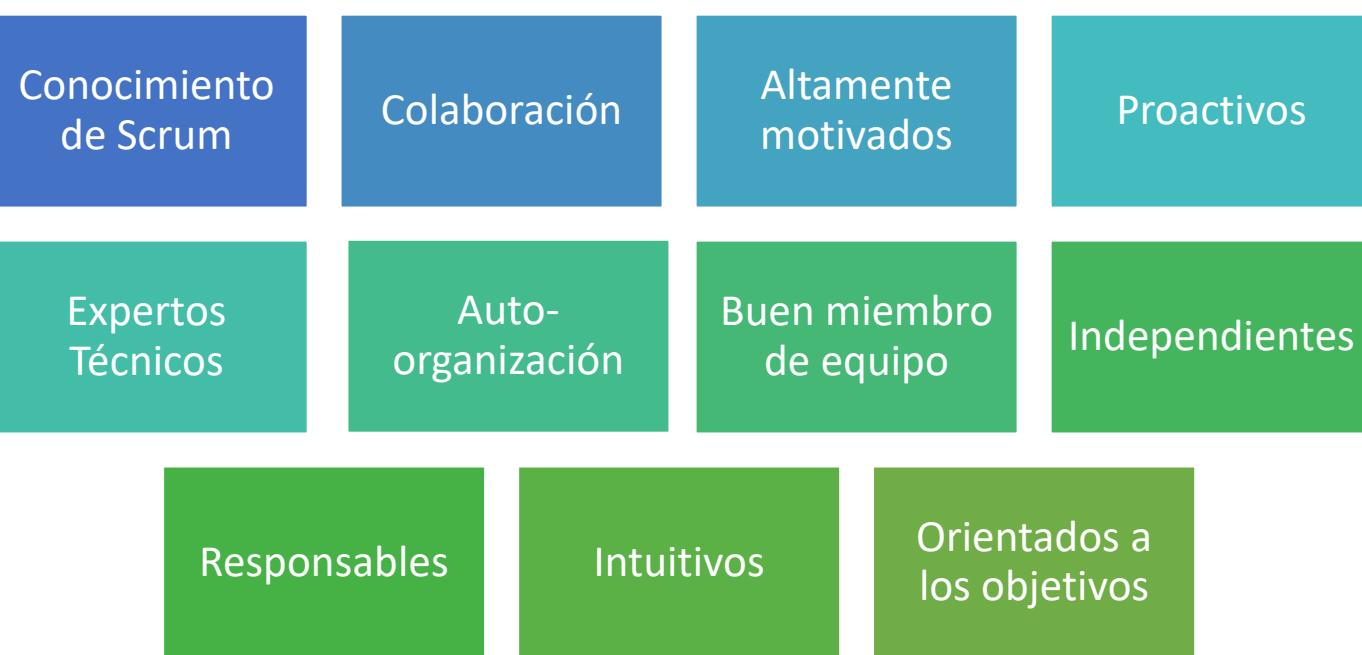
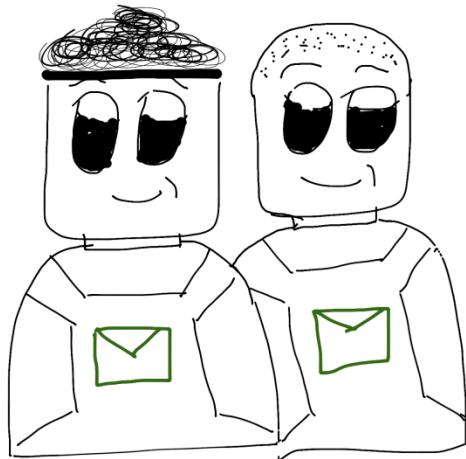
¿Que es Lean?

«Filosofía que permite mejorar la eficiencia y la productividad de cara al futuro»

Principios Lean



Development Team - Características



Development Team - Responsabilidades

Asegurar una
comprensión clara
de los
requerimientos

Estimar los User
Stories aprobados
por el PO

Crear entregables
de alta calidad

Desarrollar la lista
de tareas basada
en las User Stories
aprobadas

Calcular el
esfuerzo para las
tareas
identificadas

Development Team - Responsabilidades

Desarrollar el
Sprint Backlog y el
Sprint Burdown
Chart

Crear los
entregables

Identificar el riesgo
y ejecutar acciones
para su mitigación

Identificar
oportunidades de
mejora

Participar en la
retrospectiva del
proyecto y Sprint

Tamaño del Development Team



- 3 a 10 personas
- Tamaño adecuado para poder manejar las tareas.

Planificación Ágil

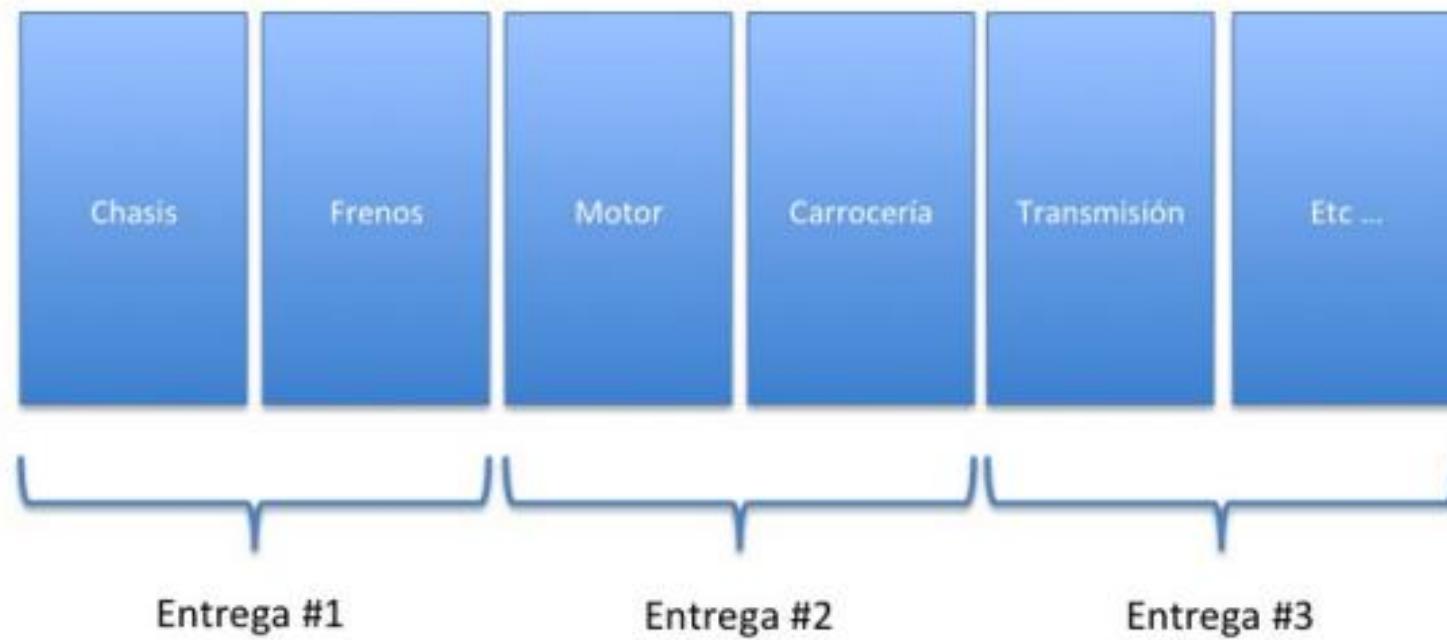
Supongamos que nos han contratado de una empresa de transporte para construir autobuses para el traslado de niños desde su casa a la escuela y desde la escuela a su casa.



Luego de analizar las características o funcionalidades que el autobús debe tener, hemos dividido la problemática en:



Una alternativa:



De forma iterativa:



Partiendo de esta base, vamos a introducir dos conceptos complementarios entre sí: Minimum Marketable Feature y Visual Story Mapping.

Minimum marketable feature

Todas las metodologías ágiles coinciden en que un producto debe construirse de forma evolutiva en pequeñas entregas.

De todas formas no es suficiente, como vimos anteriormente, dividir el producto en tres o cuatro entregas sucesivas, sino que debemos hacerlo de forma criteriosa para que cada entrega pueda aportar valor suficiente a los usuarios finales.

Esos grupos de características se denominan MMF: Minimum Marketable Features, y pueden definirse como “el conjunto más pequeño posible de funcionalidad que, por si misma, tiene valor en el mercado”



¿Cuál es la versión
mínima de una
hamburguesa?

Versionemos



- R0: Dos panes y una carne
- R1: Más salsas
- R2: Más tomate, lechuga y cebolla
- R3: Más queso y tocineta
- R4: Más otra carne
- R5: Más papas y gaseosa
- R6: Más postre
- Etc. etc. etc



Malas Prácticas

SerVIP / User Story Map



Releases por módulos que no generan valor de negocio



Trate mejor esto...

**Generar valor de punta a punta
(end to end), y logrando un
Release útil para el negocio**



Crear Historias de Usuarios

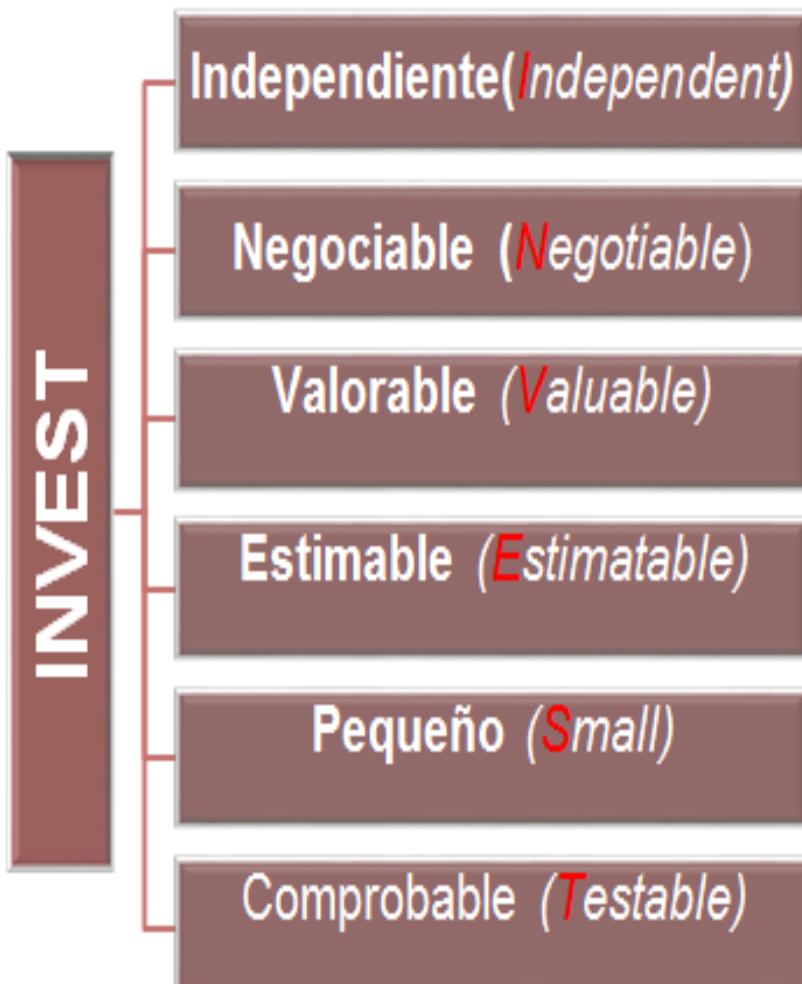
Las 3Cs para escribir Historias de Usuario

(Ron Jeffries):

- Card:** se escriben en tarjetas anotaciones.
- Conversation:** se conversan, detallan y validan con el cliente.
- Confirmation:** se escriben criterios de aceptación para confirmar que la historia se desarrolle correctamente.



Características de Historia de Usuarios Efectivas



¿ Cómo se redactan ?



Historias de usuario

Como [cliente habitual], **quiero** [ver productos relacionados] **para** [ver si hay otros productos que me puedan interesar]

Condiciones de completitud

- Los productos estarán ordenados por valoración y margen de beneficio.
- Cuando el usuario haga clic en un producto, se desplegará el detalle.
- Etc.

Prioridad

70

Coste

5

Iluminar la salida de dinero

Como Juana “la señora mayor”

Quiero que el cajero me avise de por dónde va a salir el dinero con una señal luminosa

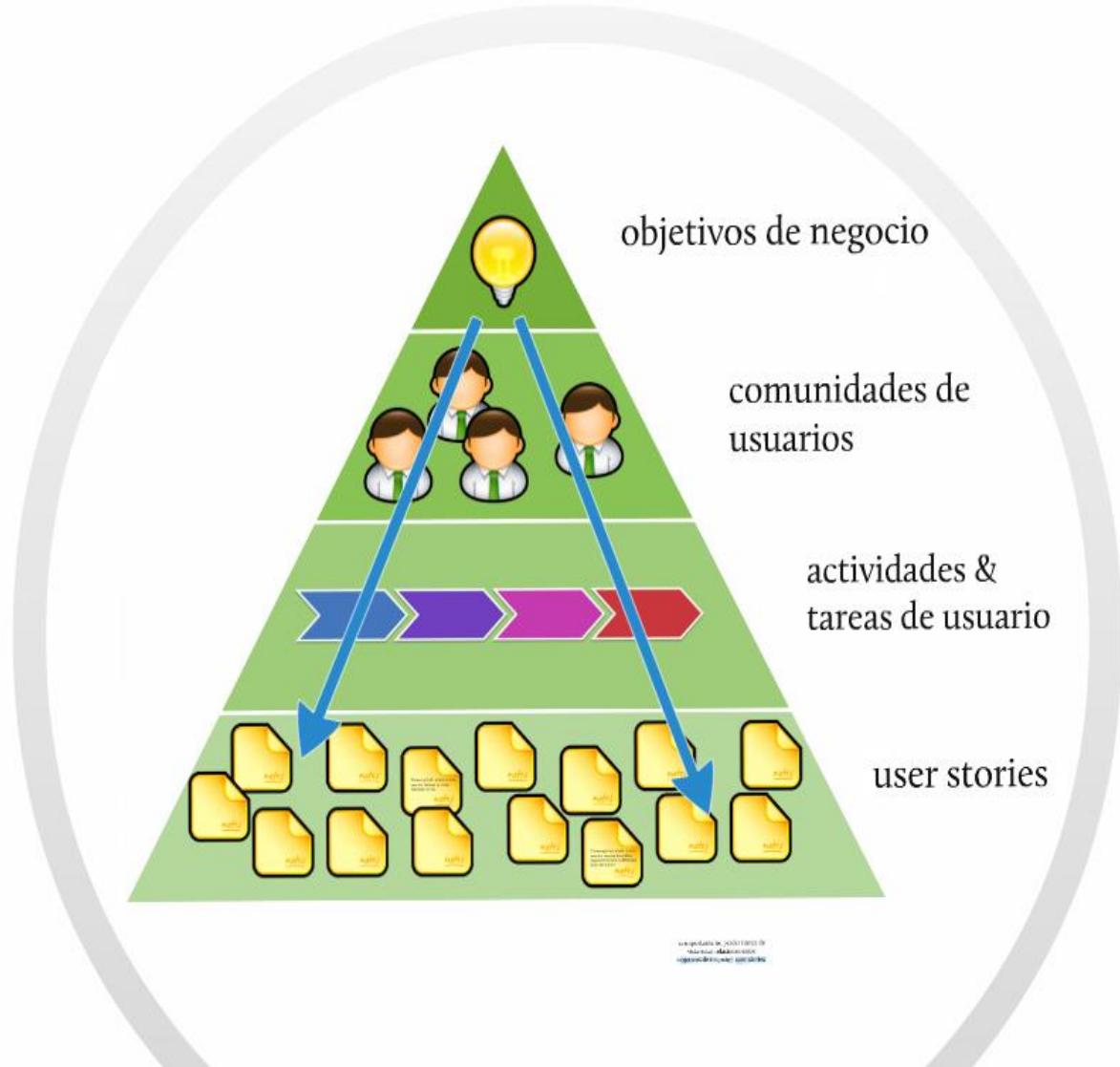
Para no desorientarme al terminar la operación

Partida multijugador

Como Ricardo “el jugador competitivo”

Quiero jugar partidas con otros
jugadores

Para divertirme más que jugando solo



Visual Story Mapping

Conjugando el Desarrollo Evolutivo, la Priorización del Backlog y el concepto de Minimum Marketable Feature, Jeff Patton plantea una técnica de Análisis Ágil llamada Mapeo Visual de Historias o Visual Story Mapping.

La teoría del Visual Story Mapping comienza en un nivel “humano” identificando los Objetivos que toda persona persigue y dividiéndolos en Actividades para las cuales deben utilizarse Herramientas



Fig. X: Jerarquía Objetivo → Actividad → Herramienta

El último nivel denominado “herramientas”, puede desagregarse a su vez en diferentes niveles de confort.

Por este mismo principio una persona puede viajar de una ciudad a otra en un automóvil del año 1965 o en un último modelo siendo que la actividad “llegar de una ciudad a otra” seguirá cumpliéndose.

Este nivel de confort está dado por

- 1) la necesidad de negocio (necesito que el viaje se haga en menos de 30 minutos)
- 2) cuánto estemos dispuestos a invertir (precio del auto).

User Story Mapping

User Story Map es una herramienta que permite generar una representación visual de la sistema completo. Ofrece una vista general de todas las funcionalidades que lo componen (*the big picture*) de punta a punta. Permite identificar *User Stories* faltantes en el *Backlog*, planificar *Releases* partiendo en rebanadas (*Slicing*), visualizar cómo se distribuyen las funcionalidades de acuerdo a las diferentes áreas del sistema.

Es una forma de reorganizar el *Product Backlog* en dos dimensiones, una dimensión para el tiempo (medido en *Releases*) y otra dimensión para las funcionalidades.

¿Cómo se construye?

Para la construcción resulta fundamental pensar todo el proceso desde el punto de vista del usuario y sus objetivos con nuestro sistema; nunca desde el lugar de Product Owner o de quienes construimos el producto u ofrecemos el servicio.

Backbone: Identificar las grandes etapas en las que se divide nuestro sistema. Por ejemplo, en un sistema de venta online, podría ser: llegar al website, búsqueda de producto, compra/pago y entrega.

Activities: Identificar todas las **actividades** que debe o puede realizar el usuario en nuestro sistema, desde que llega al mismo hasta que finaliza su proceso. Ordenarlas de manera secuencial de manera horizontal. Esta secuencia conforma la cadena de valor de nuestro sistema (Value Stream).

User Stories: Identificar las **distintas maneras** que nuestro sistema puede ofrecer para que el usuario realice cada una de las actividades identificadas. Por ejemplo, para la actividad "encontrar producto" podría ser: navegar el catálogo, usar el cuadro de búsqueda, sugerencias de productos, etc.

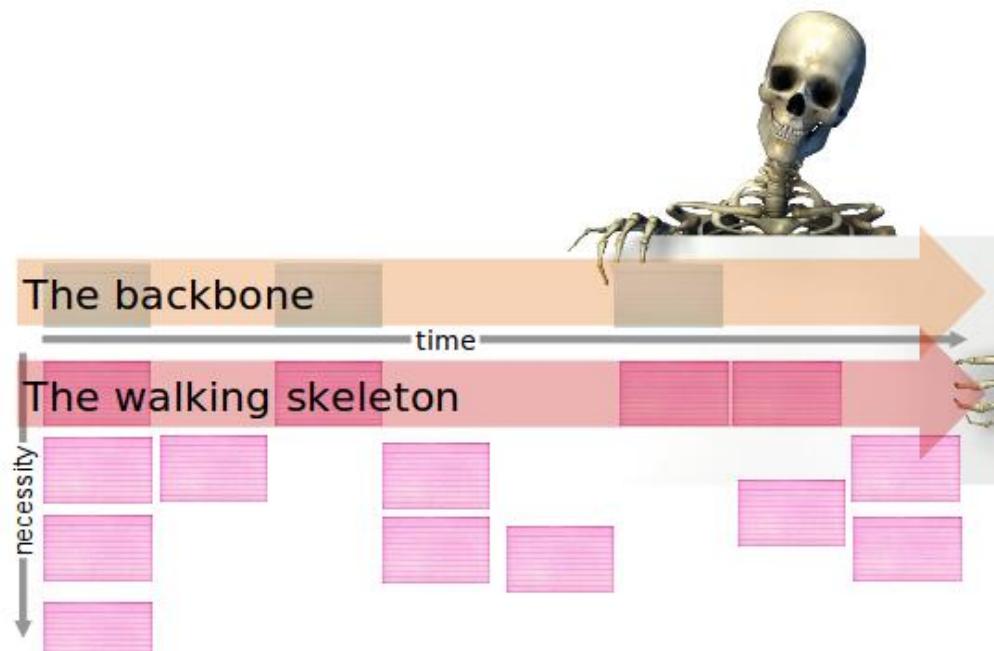
Priorización: Para la priorización debemos utilizar el criterio de MoSCoW (Must Have, Should Have, Could Have).

De manera que la funcionalidad más elemental, que no puede faltar debe estar primera.

The Walking Skeleton

Con las actividades y la primera fila de User Stories obtenemos lo que Jeff Patton denomina "*The Walking Skeleton*" (el esqueleto que camina) porque está completo y funciona, aunque sea en su forma más elemental.

Nuestro producto en este estado puede ser identificado como la versión MVP (Minimum Viable Product). O como él lo llama, MMF (*Minimum Marketable Feature*).



Release Mapping

El User Story Mapping permite planificar las próximas versiones o releases del producto representando de manera visual en qué partes de nuestro sistemas estamos dedicando mayor esfuerzo para generar Incremento de Producto.

Esto puede ser muy poderoso, como estrategia de negocio si se lo asocia al Customer Life Cycle para comprender qué partes de nuestro sistema debe ser mejoradas.



El Poder Visual

El User Story Map permite a cualquier stakeholder comprender qué funcionalidades ya fueron realizadas, cuáles se están trabajando en este momento y cuales son las próximas funcionalidades para todo el producto y la organización.

User Story Map y la Estructura Organizacional

El User Story Map, está centrado en el usuario y representa el flujo del producto de punta a punta, por lo tanto integra a muchos (sino todos) los equipos de la organización, promoviendo alineamiento de prioridades una visión más integral.

Formas de Implementación

El User Story Map suele requerir mucho espacio físico, lo cual puede representar un desafío.

En la Pared

El Story Map se puede implementar directamente sobre la pared con Post-its y cinta de enmascarar.



Esquemas de Priorización – MoSCoW



- Must have:** El requisito debe ser satisfecho obligatoriamente.

- Should have:** El requisito es prioritario y **debería** ser satisfecho de ser posible.

- Could have:** El requisito es considerado **deseable pero no necesario**. Podría estar.

- Would like to have, but not this time:**
Representa un requisito se quisiera tener, sin embargo **no es necesario implementar** ahora.

Estimaciones Ágiles

Cono de la incertidumbre

En gestión de proyectos, el cono de la incertidumbre describe la evolución de la incertidumbre durante la ejecución de un proyecto. Al comienzo, poco es conocido sobre el producto y el resultado del trabajo, por tanto las estimaciones están sujetas a una gran incertidumbre. A medida que avanzamos en el proyecto obtenemos mayor conocimiento sobre el entorno, la necesidad de negocio, el producto y el proyecto mismo.

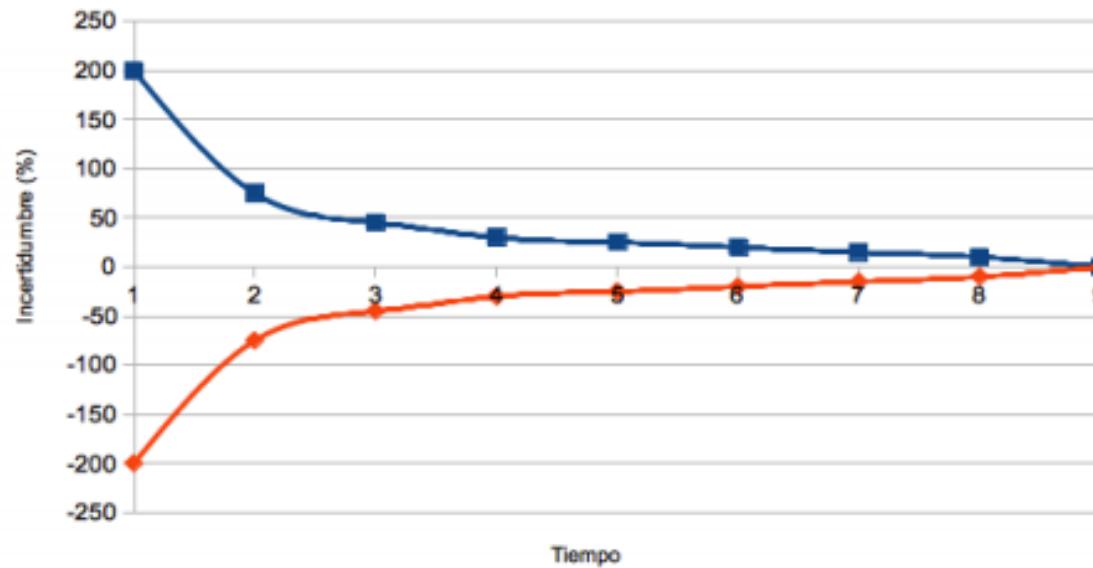


Fig. 2: Cono de la Incertidumbre en un contexto estable.

Cono de la incertidumbre

Investigaciones han demostrado que en la industria del software, el nivel de incertidumbre al comienzo de un proyecto es del +/- 400%¹¹, esta incertidumbre tiende a decrementarse durante la evolución del proyecto, pero sin garantías de ello

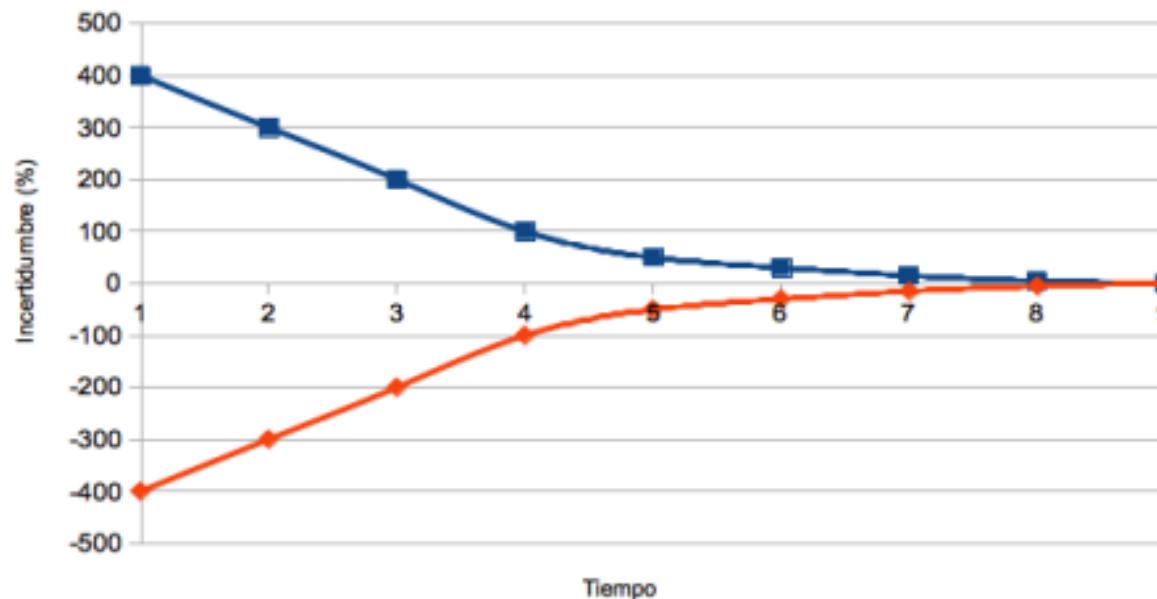


Fig. 3: Cono de la Incertidumbre en desarrollo de software.

Estimaciones en entorno complejos

Como es de esperar según los gráficos previos, proveer una estimación precisa en etapas tempranas de un proyecto tiene como consecuencia un compromiso poco probable de ser cumplido

A medida que adquiramos conocimiento, nuestras estimaciones se harán cada vez más precisas. El problema aparece a la hora de estimar cuando muchas de las decisiones se toman en base a supuestos que probablemente no sucedan o no sean los correctos.

“La propia palabra “estimación” deja en claro que no calculamos un valor exacto, determinístico.

De hecho, toda estimación tiene supuestos, y estos supuestos suman incertidumbres

Estimaciones en entorno complejos

Para mitigar el riesgo de proveer estimaciones incorrectas, en metodologías ágiles se opta por reducir la precisión de las estimaciones en función de cuánto conocimiento se tiene sobre el esfuerzo que se requiere estimar. De esta manera, los “requerimientos” y sus “estimaciones” se categorizan en diferentes niveles de precisión.

Estimaciones

Podemos enumerar la siguiente estimaciones:

1. Alto Nivel: EPIC (bloque funcional) estimada en Tamaño (XS, S, M, L, XL)
2. Nivel Medio: Historia de Usuario (funcionalidad) estimada en Puntos de Historia (Sucesión de Fibonacci13)
3. Bajo Nivel: tareas o actividades estimadas en horas, preferiblemente menos de un día.

Técnica de los Polos

S		
M	2	
L	3	
XL	5	
XXL	8	
XXXL	3	

Técnica: Tamaño Relativo / Story Points

También para estimar Historias de Usuarios se debe tener en cuenta:

- El equipo es el dueño de la definición del Story Point.
- La estimación en Story Points debe incluir todo.
- Los tamaños deben ser relativos.
- Se debe considerar complejidad, y riesgo.



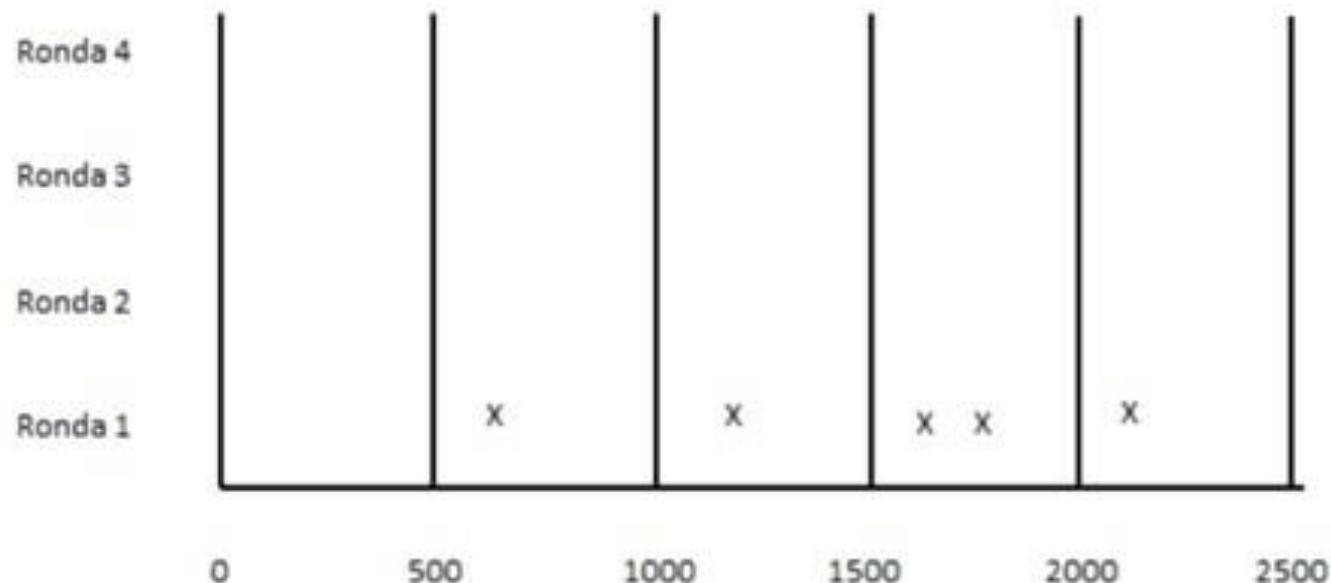
Técnica: Wideband Delphi

Grupo de expertos que estiman de forma anónima.

La repetición y retroalimentación controlada

La respuesta del grupo en forma estadística

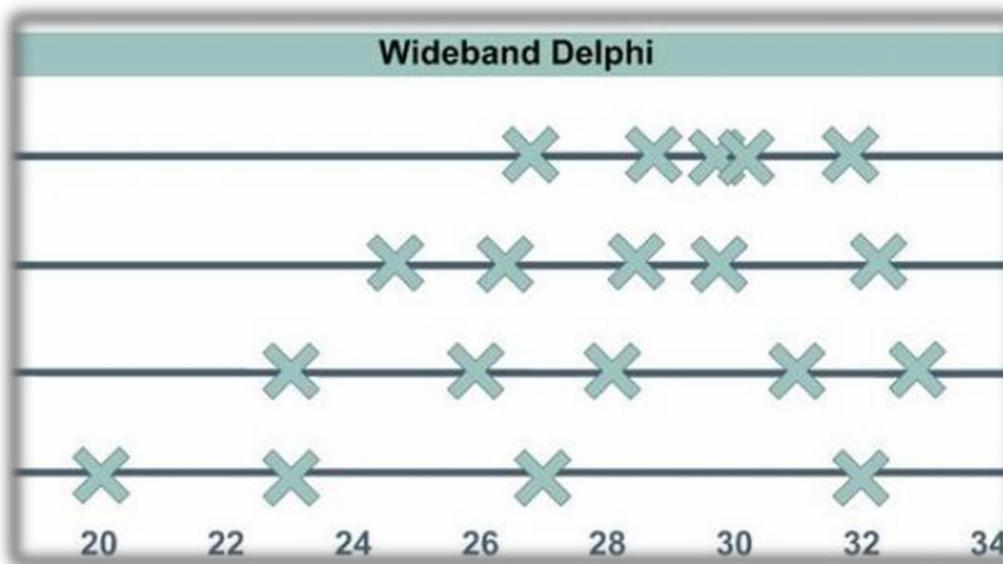
Ejemplo de primera ronda de estimación Wideband Delphi.



Técnica: Wideband Delphi

Es útil por ser:

- Iterativa
- Adaptativa
- Colaborativa



La Sabiduría de las Multitudes (Wisdom of Crowds)

James Surowieki explica en su libro “La Sabiduría de las Multitudes” (2004):

“Normalmente solemos favorecer la opinión de los expertos, pues consideramos que sólo una persona con experiencia y conocimientos suficientes es capaz de emitir juicios correctos en un área o materia en particular.

Sin embargo, hay evidencias de que las decisiones tomadas colectivamente por un grupo de personas suelen ser más atinadas que las decisiones tomadas sobre la base del conocimiento de un experto”

Para que esto pueda suceder, **Surowieki** recomienda en su tesis las siguientes condiciones:

1. Diversidad de opiniones: cada persona debería tener información particular aún si es sólo una interpretación excéntrica de los hechos conocidos.

El grupo debe tener diversidad de perfiles.

2. Independencia: las opiniones de los participantes no deberían estar influenciadas por las opiniones de los que los rodean, con el objetivo de evitar el Pensamiento de Grupo.

3. Agregación: El grupo debería tener la capacidad de sumar las opiniones individuales y no simplemente votar por la mejor opción.

Planning Poker

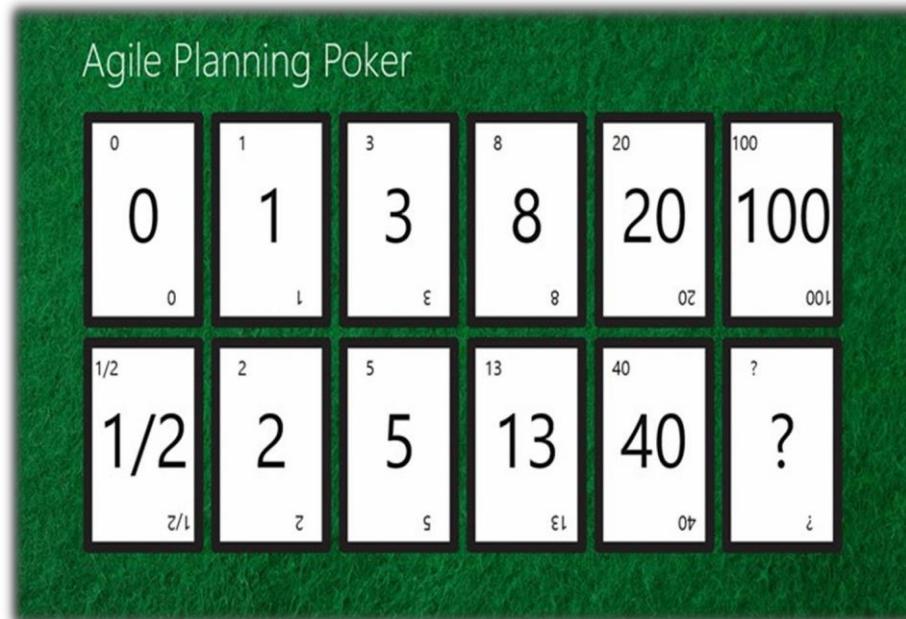


Planning Poker

"Es una técnica muy sencilla, divertida y eficaz, en la que todas las personas comprometidas en el desarrollo de un producto software opinan para estimar el esfuerzo de desarrollar cada una de las historias de usuario de un sprint"

Técnica: Planificación Poker

- La técnica Wideband Delphi se aplica normalmente como planificación Poker.
- Se utilizan cartas numeradas para estimar.
- La serie de Fibonacci



Conclusiones de las estimaciones Ágiles

Muchas teorías y enfoques convergen en las siguientes características sobre estimaciones en proyectos ágiles:

1. No tiene sentido presentar estimaciones certeras al comienzo de un proyecto ya que su probabilidad de ocurrencia es extremadamente baja por el alto nivel de incertidumbre.

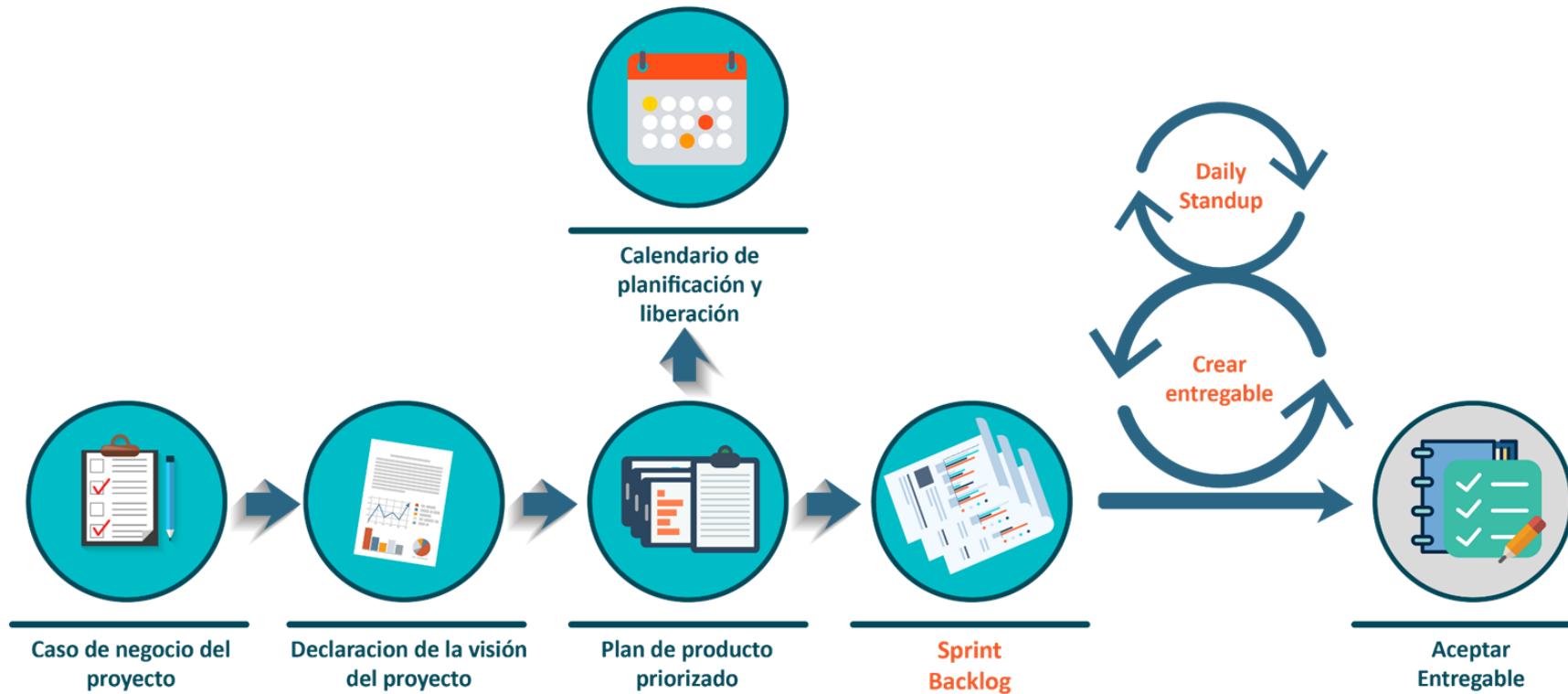
2. Intentar bajar dicha incertidumbre mediante el análisis puede llevarnos al “Análisis Parálisis”. Para evitar esto debemos estimar a alto nivel con un elevado grado de probabilidad, actuar rápidamente, aprender de nuestras acciones y refinar las estimaciones frecuentemente. Este enfoque se conoce también como “Rolling Wave Planning” o “Elaboración Progresiva”.

3. La mejor estimación es la que provee el Equipo de trabajo. Esta estimación será mucho más realista que la estimación provista por un experto ajeno al Equipo.

Pilares del Manifiesto Ágil



Ciclo de Vida de Scrum





2019
MUCHAS GRACIAS

Harvey Aquino Mas