GESTIÓN DE PROYECTOS SOFTWARE

Unidad 2: Enfoques, Metodologías y Técnicas en la Gestión de proyectos software

Presentación 2.2: Enfoques adaptativos (SCRUM)

AGENDA DE HOY

Unidad 2 - 2.2 Enfoques Adaptativos - Ágiles

Resultados de aprendizaje

Comprender la gestión de proyectos desde un enfoque adaptativo.

Comprender lo fundamental del framework Scrum

¿El manifiesto ágil?

CRYSTAL - ALISTAIR COCKBURN
FDD - JEFF DE LUCA
DSDM -The DSDM Consortium
XP - KENT BECK
SCRUM - KEN SCHWABER
TDD - KENT BECK

Pregunta….

¿Por qué se llaman metodologías ágiles?

A: Actively

G: Gain

I: Improvements

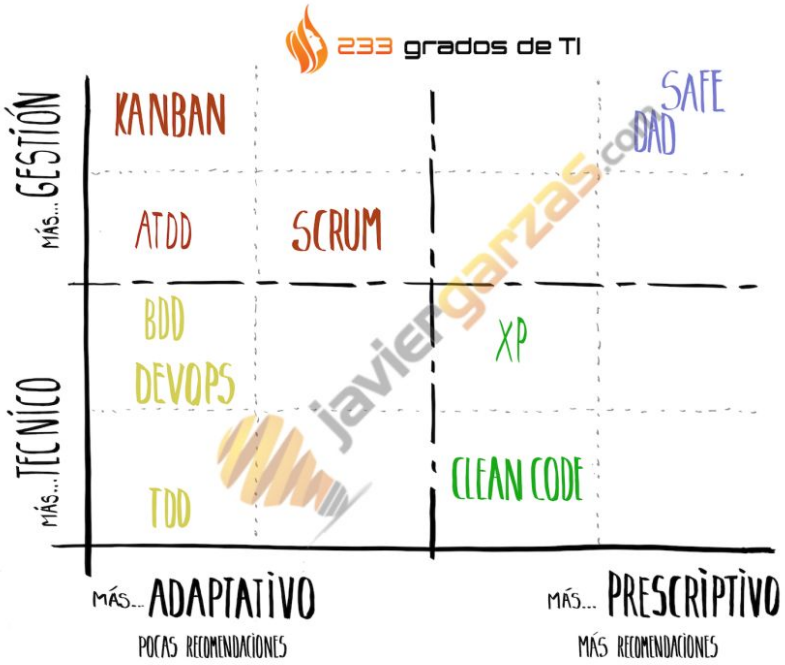
L: Learning

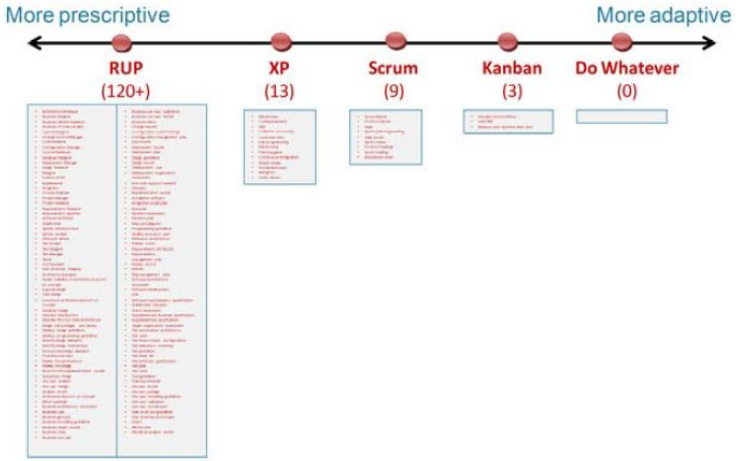
E: Everyday

La agilidad implica una concentración intensa, disciplina y acción decidida y espontánea.

Basadas en un conjunto de valores (los del manifiesto ágil) o una cultura, pero siempre sobre la implementación de un conjunto de prácticas (reales e implantadas en tu proyecto).

Equipos pequeños, multifuncionales, auto-organizados, iterativo e incremental, etc





Prescriptivo vs. adaptativo

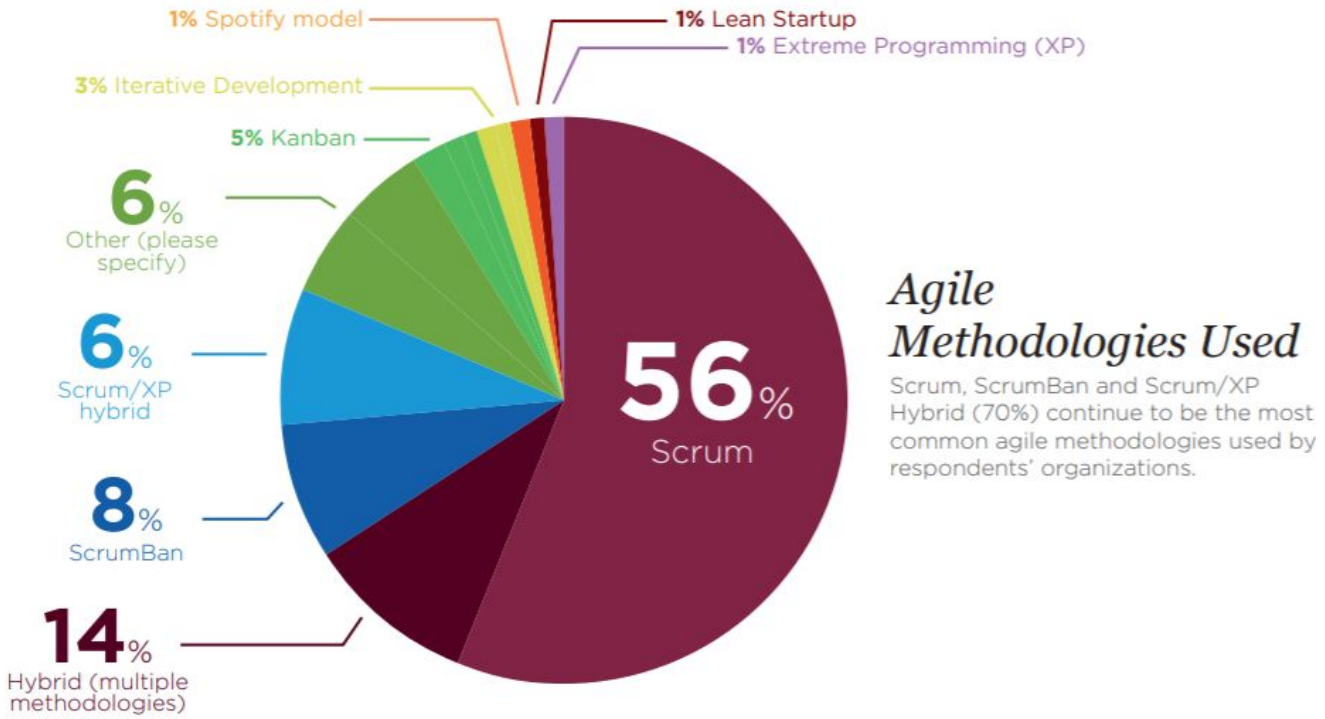
Agile es un framework adaptativo

RUP	XP	Scrum	Kanban	Mano de Dios
>120	13	9	3	0

¿Más de 120 procesos, herramientas y roles?

¿Cuales son las metodologías ágiles más usadas? III

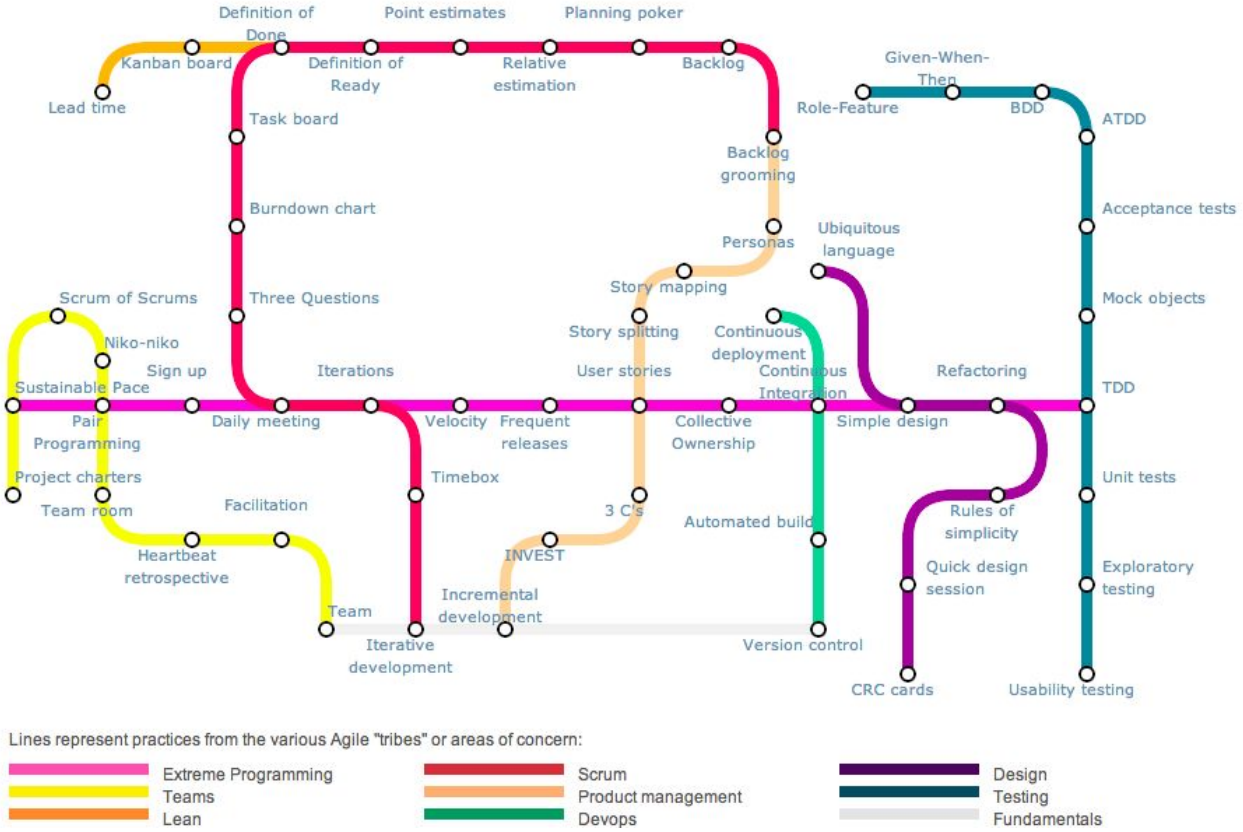
Pregunta...



file:///C:/Users/andari/Downloads/versionone-12th-annual-state-of-agile-report.pdf

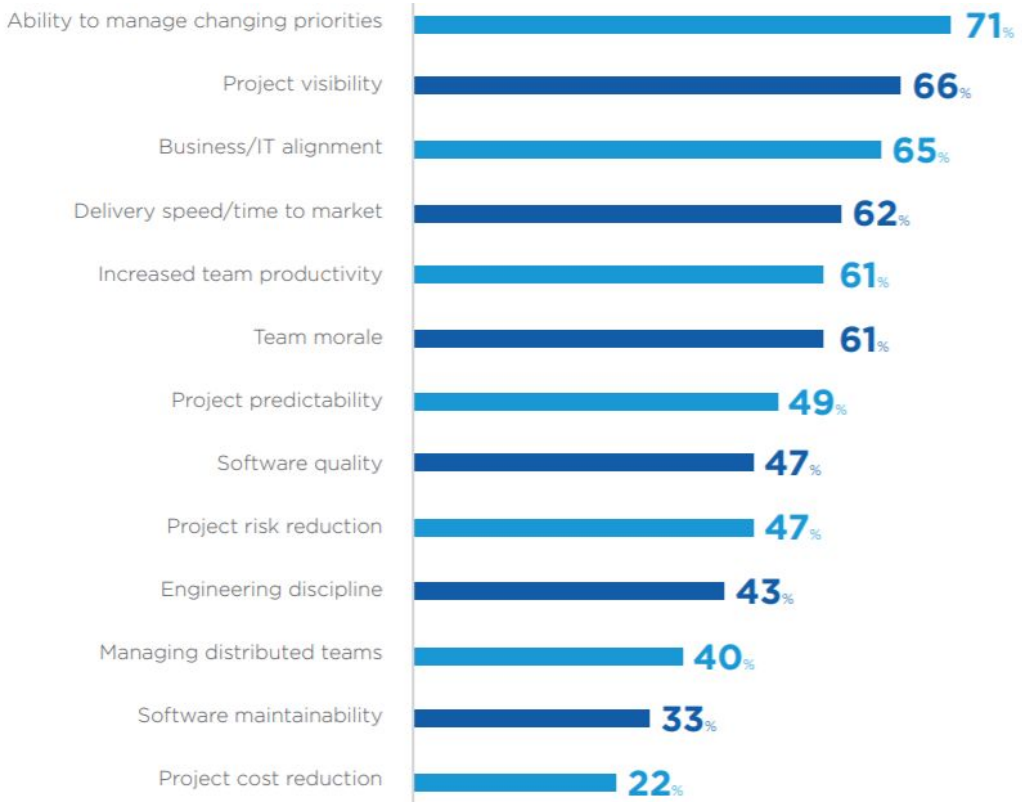
Pregunta...

¿Cuales metodologías ágiles existen, además de Scrum?
III



Pregunta...

¿Por qué usar las metodologías ágiles? Ventajas y desventajas, importancia de usarlas, diferencias al usarlas?



Pregunta…¿Cómo decidir la mejor metodología ágil para usar?

Criterios…

- Tamaño del proyecto
- Criticidad
- Grado de conocimiento del equipo
- Grado de adaptabilidad

FAMILIA CRYSTAL

Las metodologías Crystal son una familia de metodologías ágiles, donde cada una de ellas está adecuada para un tipo de proyecto. Su creador **Alistair Cockburn.**

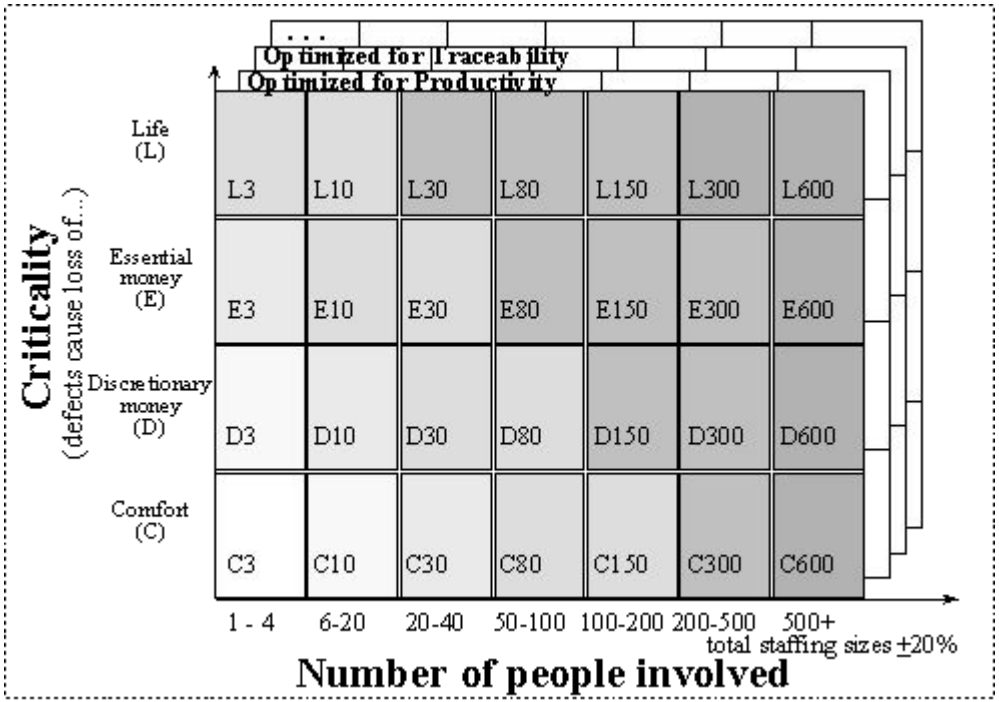
Proyectos y empresas grandes.

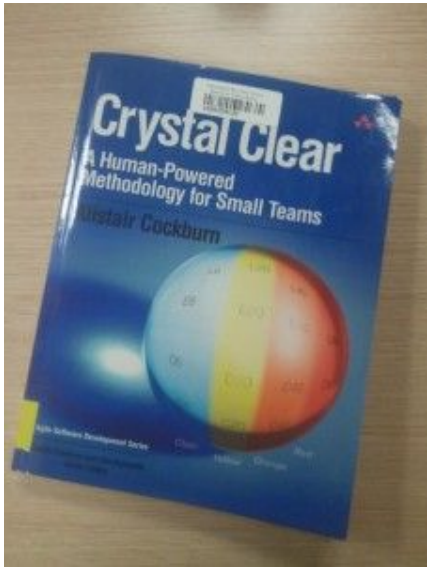
Minerales: Color y dureza

Proyecto software: Tamaño y criticidad

http://www.javiergarzas.com/2012/09/metodologias-crystal.html

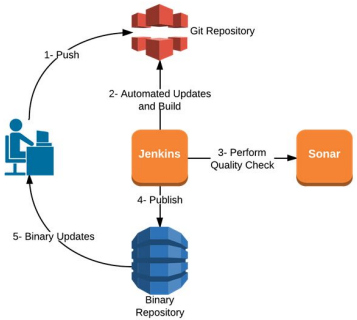
FAMILIA CRYSTAL

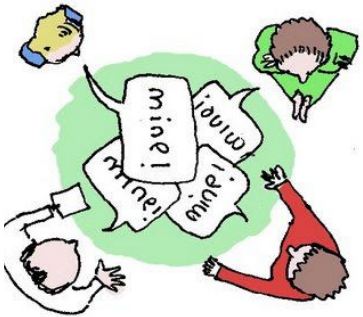




FAMILIA CRYSTAL

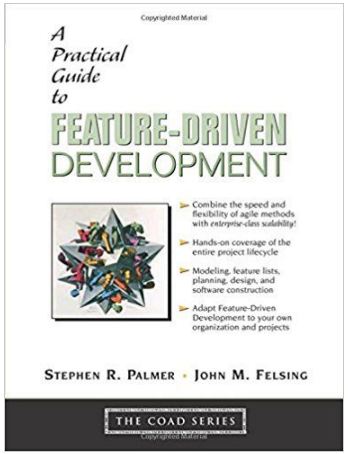
1 – Entregas frecuentes, en base a un ciclo de vida iterativo e incremental.
2 – Mejora reflexiva.
3 – Comunicación osmótica o cerrada.
4 – Seguridad personal.
5- Enfoque.
6 – Fácil acceso a usuarios expertos.
7 – Entorno técnico con pruebas automatizadas, gestión de la configuración e integración continua.





FDD (Feature Driven Development)

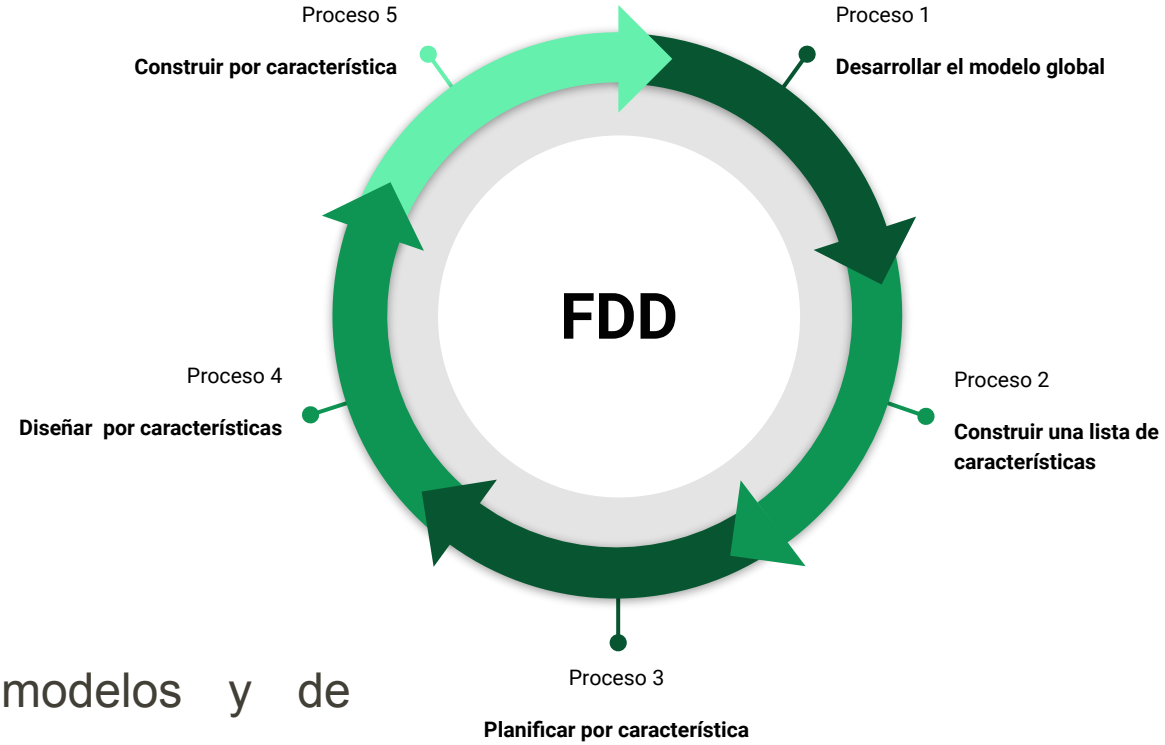
Si... **tu equipo es grande**, utilizas **personal externo**, no tienes claro que tu equipo **sea auto-organizado**, sigues pensando que **debe haber un jefe de proyecto y arquitecto**, que hay que tener un **diseño / arquitectura antes de empezar a desarrollar**… puede que **en FDD encuentres la respuesta a tus problemas.**



FRAMEWORK NO METODOLOGÍA

http://www.nebulon.com/

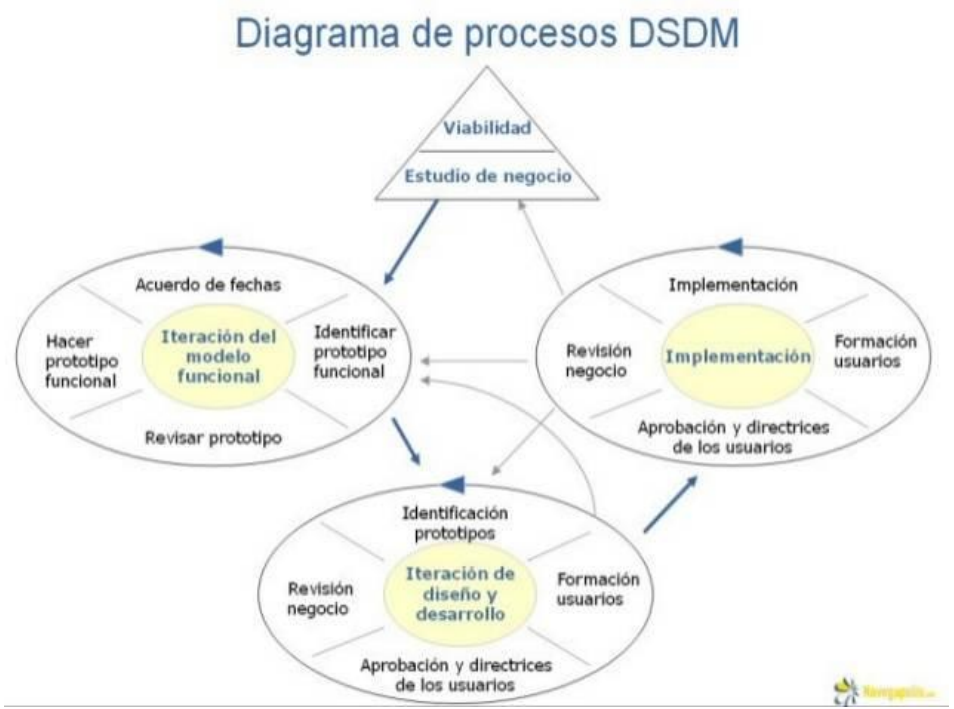
FDD (Feature Driven Development)



Dirigida por modelos y de iteraciones cortas.

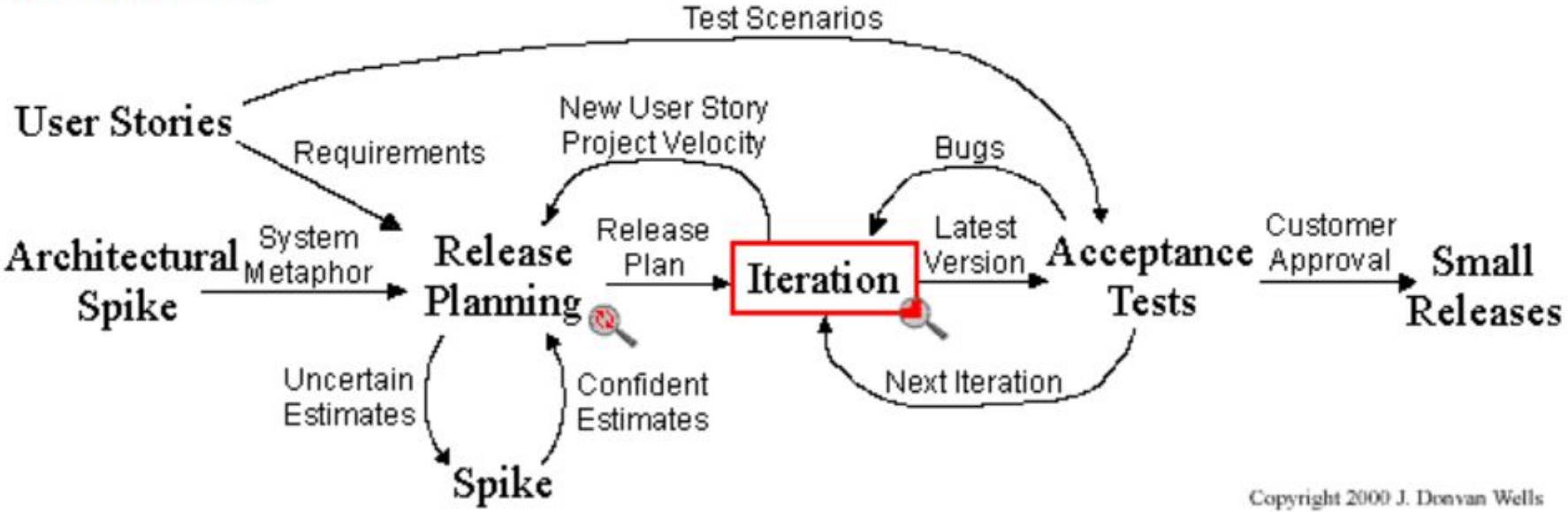
DSDM Dynamic Systems Development Method

Objetivo: diseñar y promover un marco industrial común para la entrega rápida de software. Desde 1994, la metodología DSDM ha evolucionado y madurado para proporcionar una base integral para planificar, gestionar, ejecutar y escalar proyectos ágiles y proyectos de desarrollo de software iterativo.

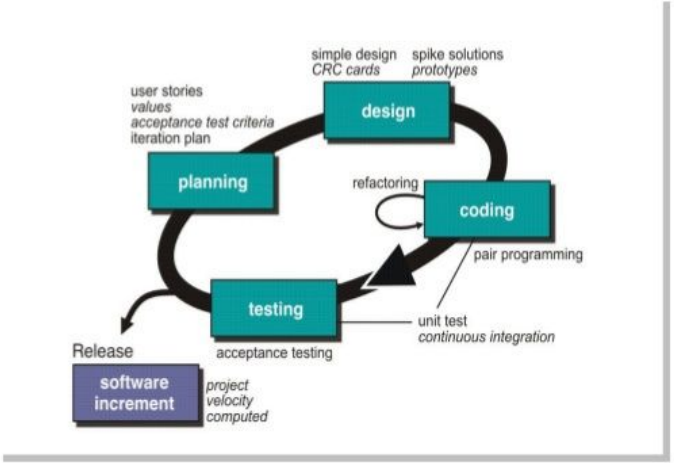


Extreme Programming

Extreme Programming Project



Extreme Programming (XP)



Juego de planificación
Pequeñas versiones
Pruebas de aceptación del cliente
Diseño simple
Programación por pares
Desarrollo guiado por pruebas
Refactoring
Integración continua
Propiedad colectiva del código
Estándares de codificación
Metáfora
Marcha sostenible
Don Wells ha representado el proceso de XP en un diagrama popular.

En XP, el "Cliente" trabaja muy de cerca con el equipo de desarrollo para definir y priorizar unidades de funcionalidad granulares conocidas como "Historias de usuarios". El equipo de desarrollo estima, planifica y entrega las historias de usuarios de mayor prioridad en forma de software probado y en funcionamiento iterativo a iterativo. Con el fin de maximizar la productividad, las prácticas proporcionan un marco de apoyo y ligero para guiar a un equipo y garantizar un software de alta calidad.

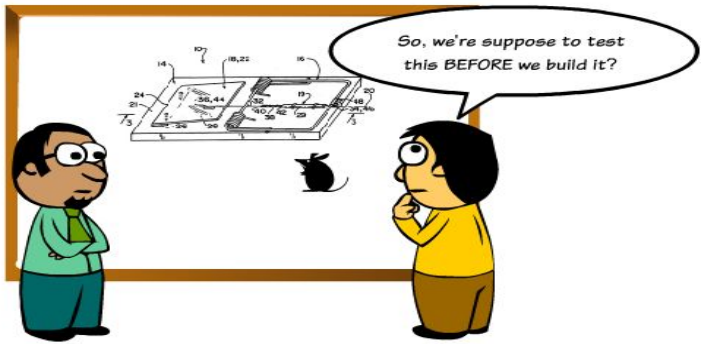
https://www.youtube.com/watch?v=hbFOwqYIOcU

Desarrollo guiado por pruebas de software
Test-driven development (TDD)

Desarrollo guiado por pruebas de software
Test-driven development (TDD)

Se propone los siguientes pasos:

- **Escribir** las pruebas primero (Test First Development) usando las pruebas unitarias (unit test).
- **Verificar** que las pruebas fallan.
- **Implementar** el código que hace que la prueba pase satisfactoriamente
- **Refactorizar** el código escrito.

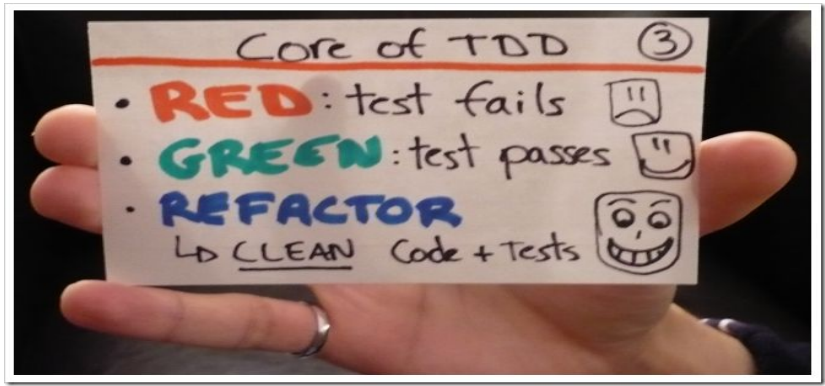


Desarrollo guiado por pruebas de software
Test-driven development (TDD)

Propósito:

Lograr un **código limpio** que funcione.

Requisitos traducidos a pruebas, de este modo, garantizar que el software cumple con los requisitos que se han establecido.

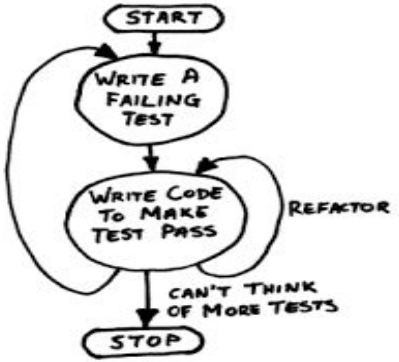


Desarrollo guiado por pruebas de software
Test-driven development (TDD)

Requisitos

- Desarrollo del sistema de forma **flexible** para permitir que sea probado automáticamente.
- Cada prueba será suficientemente **pequeña** como para que permita determinar unívocamente si el código probado pasa o no la verificación que ésta le impone.
- El diseño se ve favorecido ya que se evita el indeseado "**sobre diseño**" de las aplicaciones y se logran **interfaces más claras y un código más cohesivo.**

Ciclo de desarrollo para TDD



- **Elegir un requisito**
- **Escribir una prueba**
- **Verificar que la prueba falla:** Si la prueba no falla es porque el requisito ya estaba implementado o porque la prueba es errónea.
- **Escribir la implementación:** Escribir el código más sencillo que haga que la prueba funcione. Se usa la metáfora "Déjelo simple" ("Keep It Simple, Stupid" **KISS**).

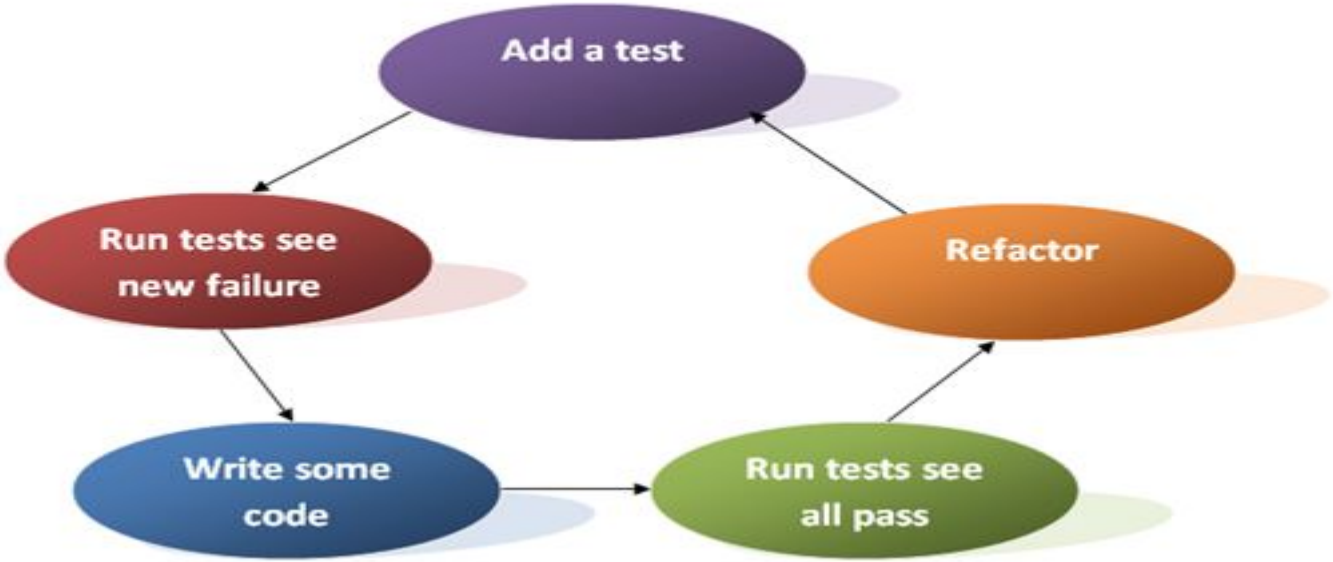
Ciclo de desarrollo para TDD

- **Ejecutar las pruebas automatizadas:** Verificar si todo el conjunto de pruebas funciona correctamente.
- **Eliminación de duplicación:** Se usa la refactorización, se elimina el código duplicado y se ejecuta la prueba hasta que funcione
- **Actualización de la lista de r**

Características

- Evita escribir código innecesario ("You Ain't Gonna Need It" (**YAGNI**)).
- Confianza en el código escrito. Después de pasar todas las pruebas tendremos un código que funcione correctamente.
- Hacer fallar los casos de prueba, asegurarse de que los casos de prueba realmente funcionen y puedan recoger un error.

The TDD Process



Ventajas

- Calidad en el código generado
- Centrarse en una tarea
- El código escrito debe estar cubierto por las pruebas

Desventajas

- Interfaz gráfica de usuario (GUIs).
- Cuando se trabaja con objetos distribuidos.
- Cuando no se tiene conocimiento total de la base de datos.

Lectura recomendada

Profundizar más sobre los marcos ágiles vistos en esta sesión.
Consultar sobre otros marcos ágiles.

Practica concreta

El Pato de LEGO

La tarea es simple: armar equipos de 3 o 4 personas, seleccionar varias fichas, hacer varios prototipos, solo se pueden construir patos de 6 piezas, al final de la iteracción presentar el pato que sea seleccionado por el equipo para que los otros equipos lo vean.

https://www.online-stopwatch.com/spanish/

Practica concreta

Reflexionar

Qué opinamos de los patos de los otros equipos?.

Qué opinamos del trabajo en equipo?

Qué pienso de las opiniones de los demás?

Enfoque de la complejidad de Cynefin

Acuñado en 2000 por Dave Snowden1.

Framework que muestra diferentes tipologías, dominios, modelos en los que te mueves, por ejemplo, cuando trabajas.

Por qué unas maneras de trabajar funcionan en un entorno, otras no, por qué algo te aburre, por qué algo te motiva…

Explica el contexto en el cual Scrum, como marco de control empírico, es más eficiente.

https://www.youtube.com/watch?v=lLWWwKI6qjg&t=42s

¿Qué es SCRUM?

Un **framework** para la creación de **productos** en contextos **complejos**. Los pilares de Scrum son la **transparencia,** la **inspección y** la **adaptación.**

Dinámica by Kleer

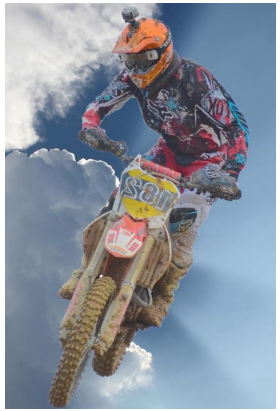
Con base en la afirmación anterior crea una lista de 10 palabras, las que vengan a tu mente, junto a tu equipo de trabajo.

Espera...

Luego entrevista a otros dos equipos y selecciona las 5 palabras que más se repiten.
Vuelve a tu equipo y seleccionen 3 palabras que consideren más relevantes.

Valores de Scrum

CORAJE



FOCO

RESPETO

COMPROMISO

APERTURA

Principios de Scrum

Empirismo: Ensayo y Error

Elementos Emergentes: Simplicidad

Auto-organización: Autonomía para crear productos de calidad y manejar su propio proceso.

Mecanismos

Priorización: Primero lo primero

Timeboxing: Poner límites de tiempo a una actividad. Compromiso.

Jugar con el alcance.

Roles de SCRUM

Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Roles de SCRUM

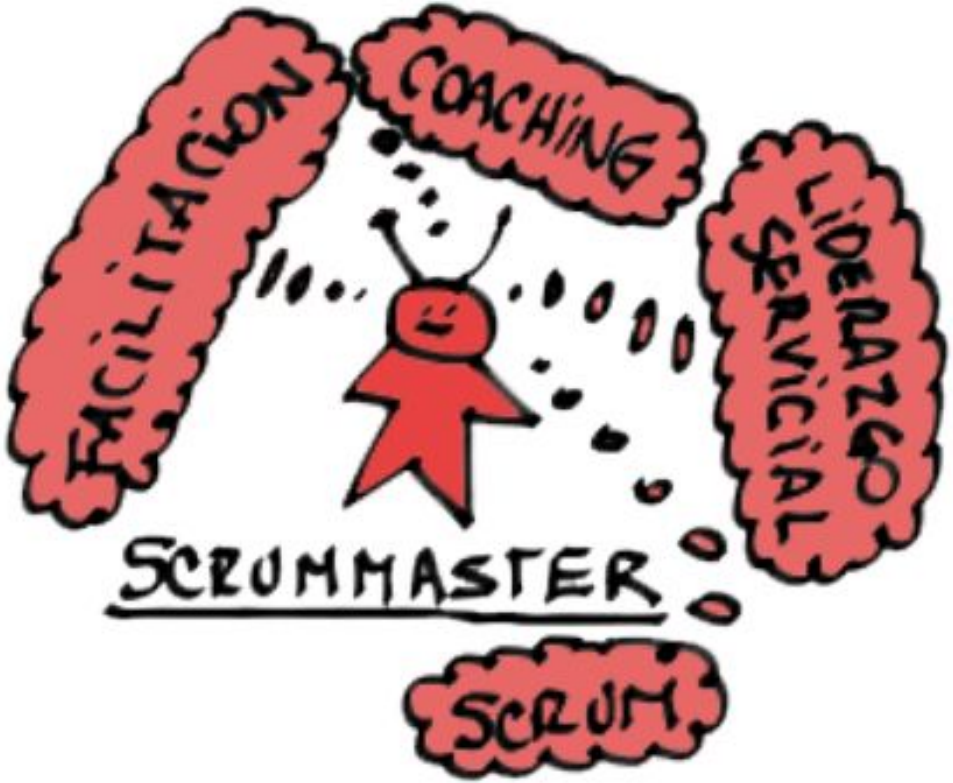
Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Multitasking

1. Dividir las hojas en 3 sectores en forma vertical.
2. Empiece a escribir en el sector I las letras de la A a la Z. En el sector 2 números desde el I. En el sector 3 números romanos.
3. Se debe escribir de esta manera, primero la letra A luego el número I, luego el número I en romanos. Luego se pasa a la siguiente línea.
4. Se cronometra el tiempo dando 2 minutos.
5. Luego en otra hoja se pide que escriban lo mismo pero esta vez, primero solo letras, luego los números, finalmente los números romanos.

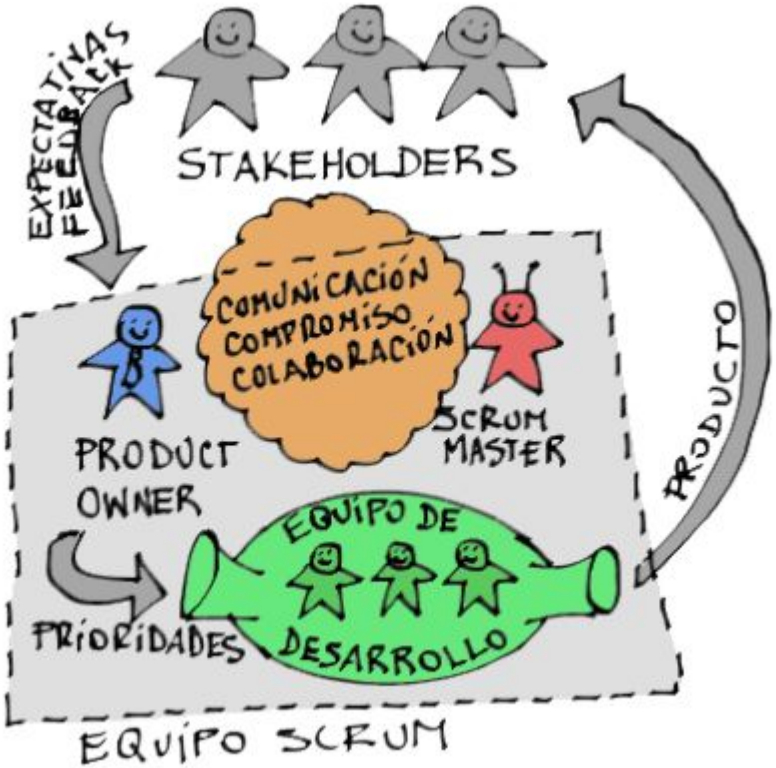
CONCLUSIONES....

Roles de SCRUM



Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Roles de SCRUM



Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

¿Verdad o Mentira?

SCRUM Roles

"Stop by Stuart Heath, on Flickr"

One-Minute CONCEPT review

Scrum Master

1.El ScrumMaster necesita tener coraje Verdadero Falso	6. El ScrumMaster es responsable porque se siga Scrum. Verdadero Falso
2.El ScrumMaster es el rol más importante en un equipo de trabajo. Verdadero Falso	7.Un buen ScrumMaster hace que su equipo de desarrollo haga lo que él necesita Verdadero Falso
3.Los mejores ScrumMaster fueron gerentes de proyecto en el pasado Verdadero Falso	8.El ScrumMaster es responsable por la planeación del proyecto Verdadero Falso
4. La presencia del ScrumMaster no es obligatoria en la reunión diaria. Verdadero Falso	
5. El Scrummaster debe gestionar al equipo de trabajo. Verdadero Falso	

Product Owner

El Product Owner es el único que puede modificar el Backlog del producto Verdadero Falso	6. El PO debe ser el cliente. Verdadero Falso
2.El Product Owner no es necesario durante la planificación del Sprint. Verdadero Falso	7.El PO determina cómo trabaja el equipo de desarrollo Verdadero Falso
3.Parte del trabajo del Product Owner es colaborar con el Scrum Master y el equipo de desarrollo Verdadero Falso	8.El Product Owner no puede proveer estimaciones Verdadero Falso
4. EL PO es el “vocero del usuario” Verdadero Falso	
5. El PO determina el mejor producto a construir. Verdadero Falso	

Equipo de desarrollo

1. Trabajar en múltiples proyectos (multitasking) es una práctica a ser evitada Verdadero Falso	6. El equipo de desarrollo debe evitar trabajar de cerca del Product Owner. Verdadero Falso
2.El equipo de desarrollo es responsable de la calidad del producto Verdadero Falso	7.El equipo de desarrollo comunica impedimentos al scrummaster solo en la daily scrum Verdadero Falso
3.El equipo de desarrollo tiene entre 5 y 9 miembros Verdadero Falso	
4. La forma mas efectiva de comunicación con el equipo son los emails detallados Verdadero Falso	
5. El equipo de desarrollo puede cancelar un sprint antes de su finalización Verdadero Falso	

Dinámica (Flujo de Trabajo)

SPRINT

Timebox

Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Sprint planning

Reunión Planeación Parte I

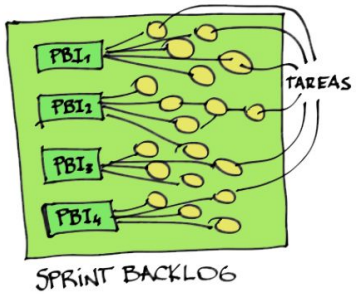
Presentación de PBIs
Preguntas sobre PBIs
Estimar PBIs
Compromiso para el sprint

Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Sprint planning

Reunión Planeación Parte 2

Identificación de tareas
RESULTADO: sprint backlog



Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Reunión Diaria

Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Which of the following are explicitly defined questions in the Daily Scrum Meeting?

A) What did I do yesterday (or since the last Scrum meeting)?

B) What will I do today (or before the next Scrum meeting)?

C) What impedes me (blocks my progress, reduces my effectiveness, etc.)?

D) What are my actuals compared to my estimates (in hours or days)?

E) What time is the next Daily Scrum Meeting?

Test Driven Development (TDD) involves creating tests and code nearly simultaneously, while constantly improving the design. Many Agile developers believe TDD helps ensure correct implementation while reducing the cost of change. Is TDD part of Scrum?

A) Yes. Scrum is a complete methodology containing everything you need to succeed.

B) No. Scrum is only a management framework. It does not specify particular technical practices.

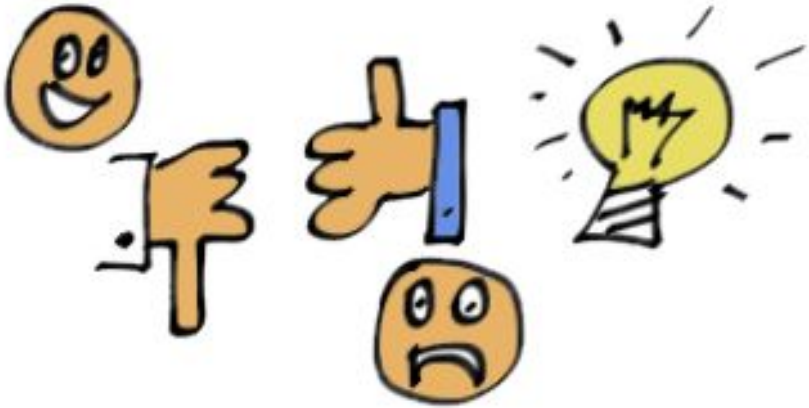
Sprint Review

Reunión Revisión del Sprint

Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

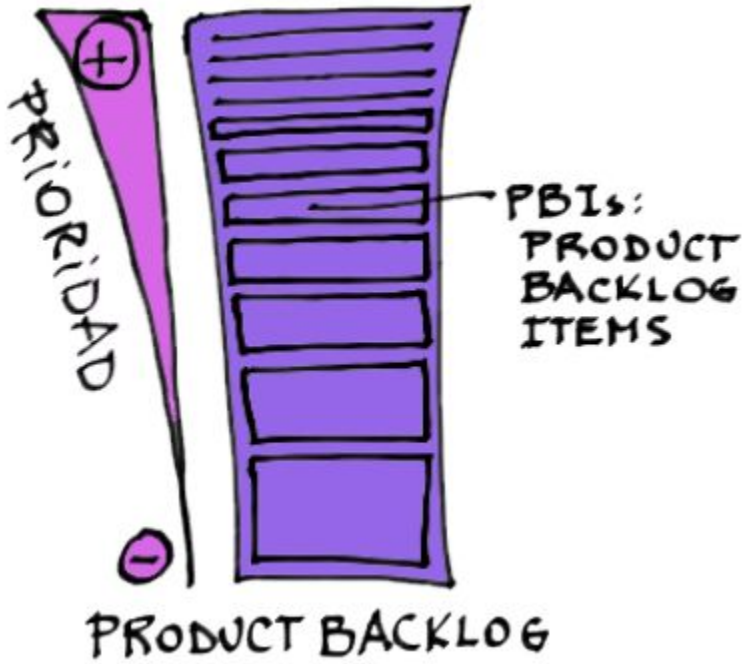
Retrospectivas

Espacio de reflexión y mejora del equipo de desarrollo o equipo completo

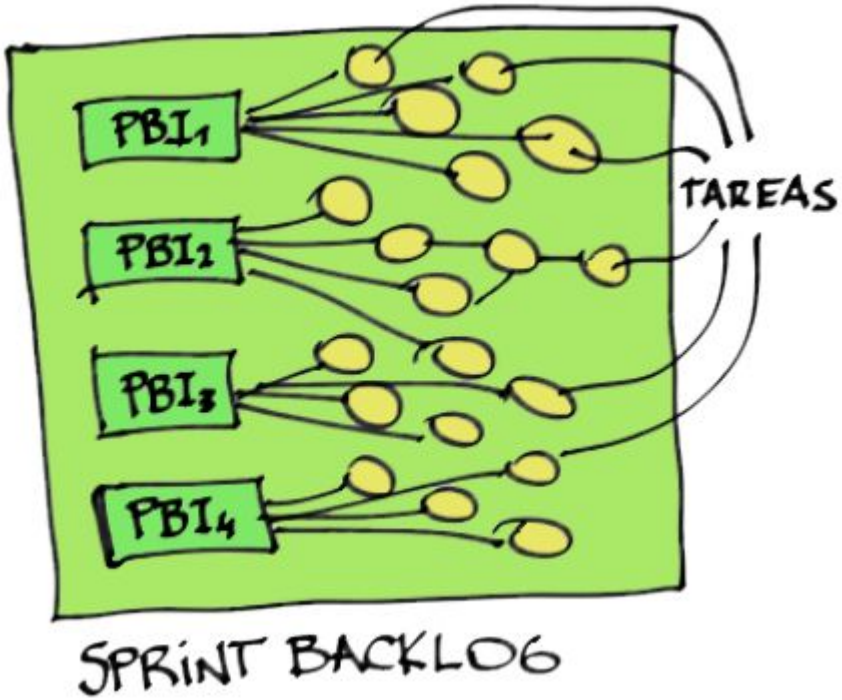


Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Artefactos de SCRUM



Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf



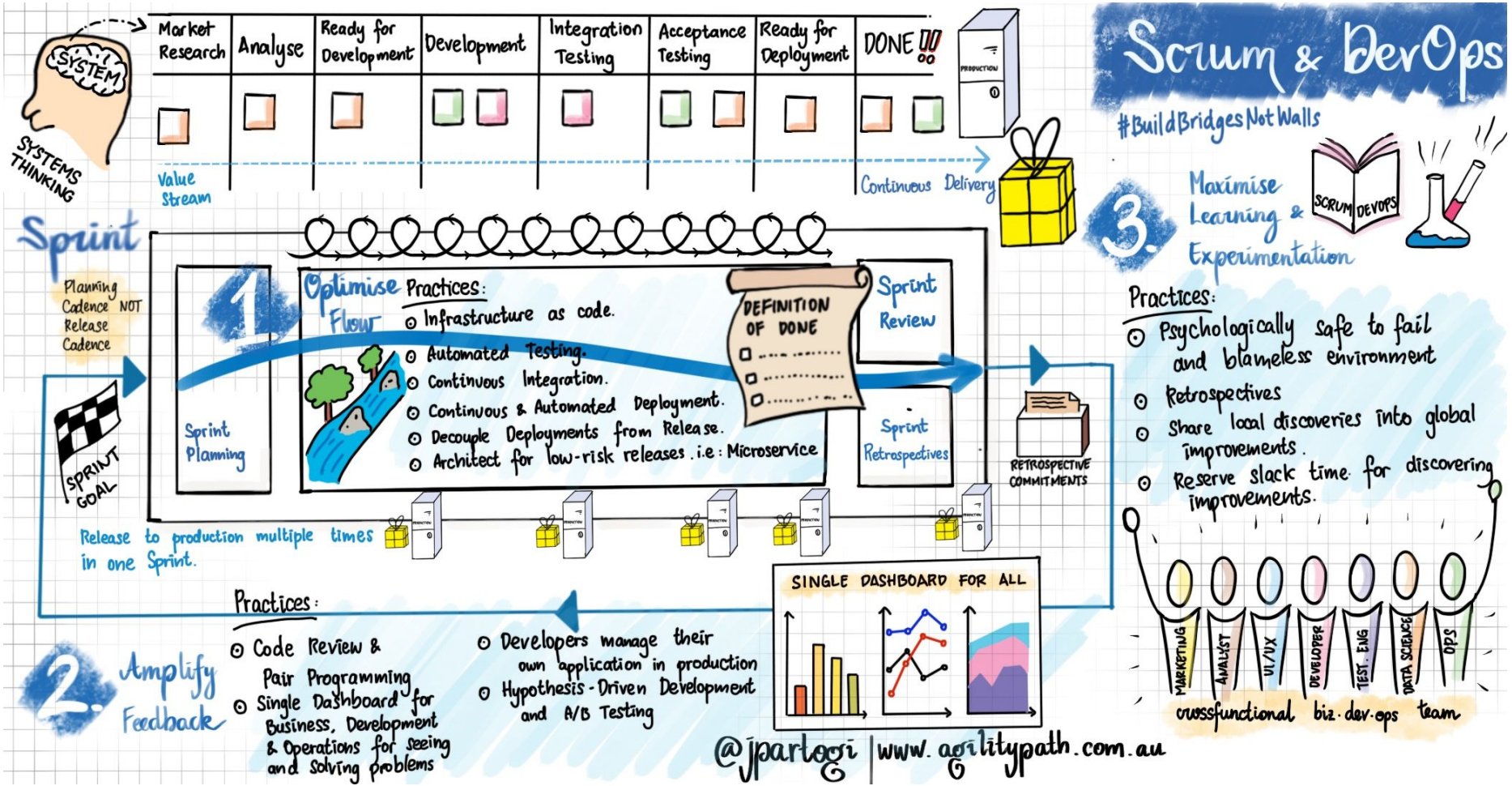
Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf

Incremento Funcional Potencialmente Entregable

Tomado de http://media.kleer.la/kleer-proyecto-agiles-con-scrum.pdf


DEFINICIÓN
DE LISTO
SPRINT
PLANNING


DEFINICIÓN
DE TERMINADO
SPRINT



Tomado de: https://dzone.com/articles/scrum-and-devops

Match

SCRUM

"Stop by Stuart Heath, on Flickr"

One-Minute CONCEPT Review

MATCH

Junto a tu equipo, determina cuál descripción corresponde a cada concepto.

a.Product Owner	I. Lista de ítems o características del producto a construir
b. Equipo de desarrollo	2. Lista de ítems del PB escogidas para ser construidas en el sprint actual
c. Stakeholders	3. Responsable por el éxito del producto desde el punto de vista de los interesados
d. Scrum Master	4.Características del producto construidas durante el sprint
e. Backlog del producto	5. Todos los individuos necesarios para construcción del producto.
f. Sprint Backlog	6. Coach del equipo, lo asiste para alcanzar su máximo nivel de productividad
g. Incremento de producto	7.Todas las partes interesadas y se ven afectados por el resultado o ejecución del proyecto

Práctica concreta

Qué modelo de ciclo de vida del software recomienda usar para gestionar el desarrollo de los siguientes sistemas.

- Un sistema móvil que recomiende a los usuarios de cable que programas deberían ver de la guía.
- Un sistema web de contabilidad que reemplace el existente. Tamaño Pequeño. Arquitectura base definida. Alta colaboración del cliente.

Trabajo extraclase

Consultar más sobre las propuestas aquí presentadas

TDD

Lean

Referencias

1. http://guide.agilealliance.org/subway.html
2. http://agileatlas.org/images/uploads/corescrum-es.pdf
3. http://marshmallowchallenge.com/Welcome.html
4. http://www.javiergarzas.com/2012/09/metodologias-crystal.html
5. http://www.lecciones-aprendidas.info/2015/03/una-actividadjuego-de-sensibilizacion.html
6. Agustín Villena https://www.youtube.com/watch?v=5waUjcqaFvY&feature=youtu.be
7. http://pmbok.certificacionpm.com/pmbok5
8. http://sce.uhcl.edu/helm/rationalunifiedprocess/process/workflow/manageme/wfd_pm.htm
9. CMMI *http://www.sei.cmu.edu/reports/10tr033.pdf*
10. https://www.youtube.com/watch?v=ZMFaUvJTW-4
11. http://agileatlas.org/atlas/scrum
12. http://www.cnde.es/cms_files/Resultados_aprendizaje.pdf
13. https://www.youtube.com/watch?v=1BtnT9tpKoE