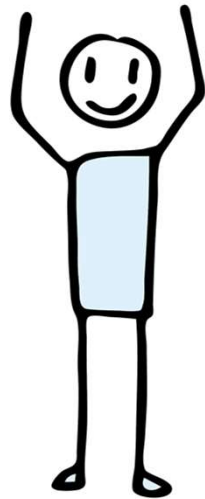


**Certified Scrum Developer**



Hola me llamo Ezequiel.  
¡Bienvenidos al curso  
CSD!



# Sobre vosotros...



## OBJETIVO

Aprender técnicas y buenas prácticas ágiles de Ingeniería Software para implementar con éxito Scrum a nivel de desarrollo técnico (TDD, Refactorización, Integración continua) y aplicarlas en un entorno Java profesional.



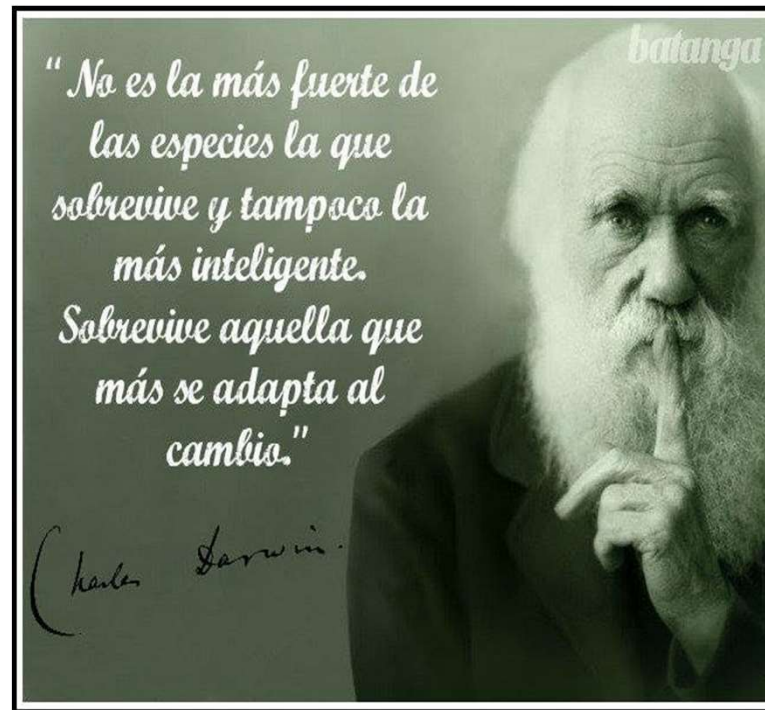
## Contenidos Formativos

1. Arquitectura en un entorno ágil con Scrum.
2. Técnicas de construcción de un equipo ágil.
3. Comportamientos colaborativos de apoyo.
4. Diseño de productos software con Test Driven Development (TDD).
5. Ciclo de desarrollo TDD.
6. Principio y buenas prácticas en pruebas unitarias.
7. Medidas de calidad en pruebas.
8. Pruebas de aceptación con ATDD.
9. Prácticas fundamentales de integración continua (IC).
10. Técnicas de automatización de pruebas de software.
11. Configuración de la IC. Repositorio de código.
12. Refactorización del código. Código limpio. Patrones.

## Valores y Comportamientos (Scrum)

1. Comunicación
2. Colaboración
3. Feedback
4. Coraje
5. Respeto
6. Empatía

## Algunas pistas antes de seguir



¿**Agile** tiene que ver sólo con rapidez?

# Atributos comunes a equipos de alto rendimiento

- Autoorganizados
- Toma de decisiones efectivas
- Comunicación abierta y clara
- Valoración de la diversidad
- Confianza mutua
- Conflicto sano
- Objetivos claros y propósitos
- Concentración y focalización
- Propiedad y responsabilidad
- Roles definidos y responsabilidades
- Coordinados y colaborativos
- Atmósfera positiva
- Retroalimentación (feedback) efectiva y oportuna
- Suficientes recursos para el control local
- El éxito sobre prevención de fallos
- Habilidades equilibradas con retos
- **Diversión**



## ¿Por qué un equipo?

### Empatía:

Percibir, compartir y entender lo que el otro puede sentir. "Ponerse en la piel del otro".

### Responsabilidad compartida:

Responsabilidad completa del éxito o del fracaso compartida por todos los miembros de un equipo "Nadie gana si no ganamos todos"

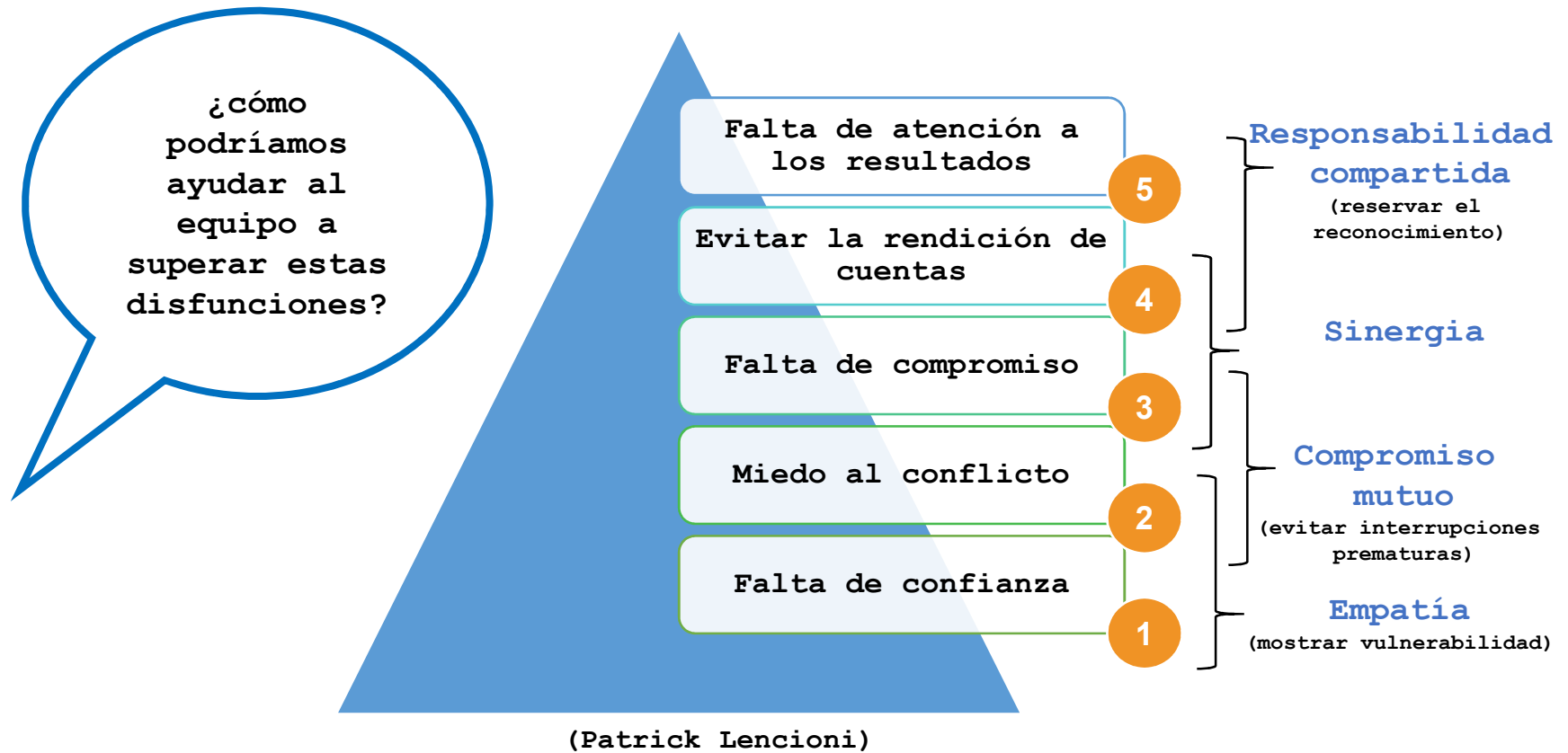
### Sinergia:

El resultado del trabajo en equipo es mucho más que la suma de la contribución individual de sus miembros.

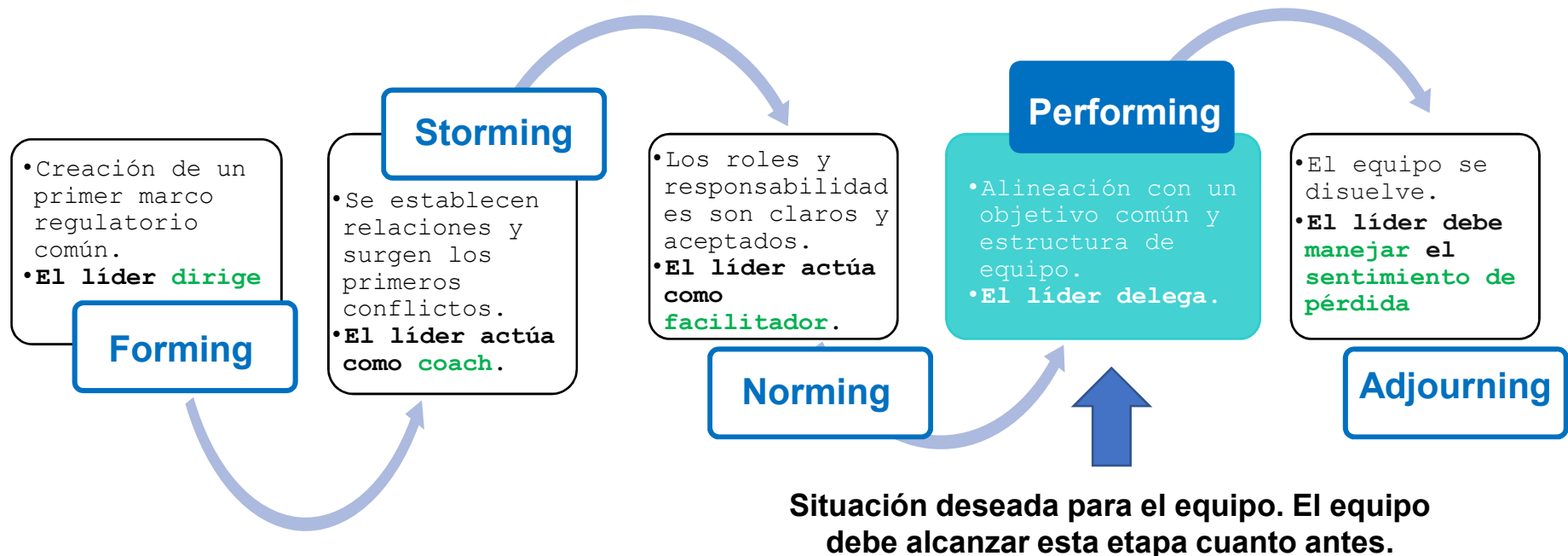
### Compromiso mutuo:

Relación interpersonal basada en un compromiso mutuo acordado con la transparencia, la colaboración, la confianza, el respeto, la apertura y el coraje.

# Disfunciones del equipo



# Etapas para el desarrollo del equipo



Las fases del desarrollo del equipo no son discretas – la progresión no es inevitable y los equipos pueden dar pasos hacia atrás, oscilar entre las etapas o encallarse.

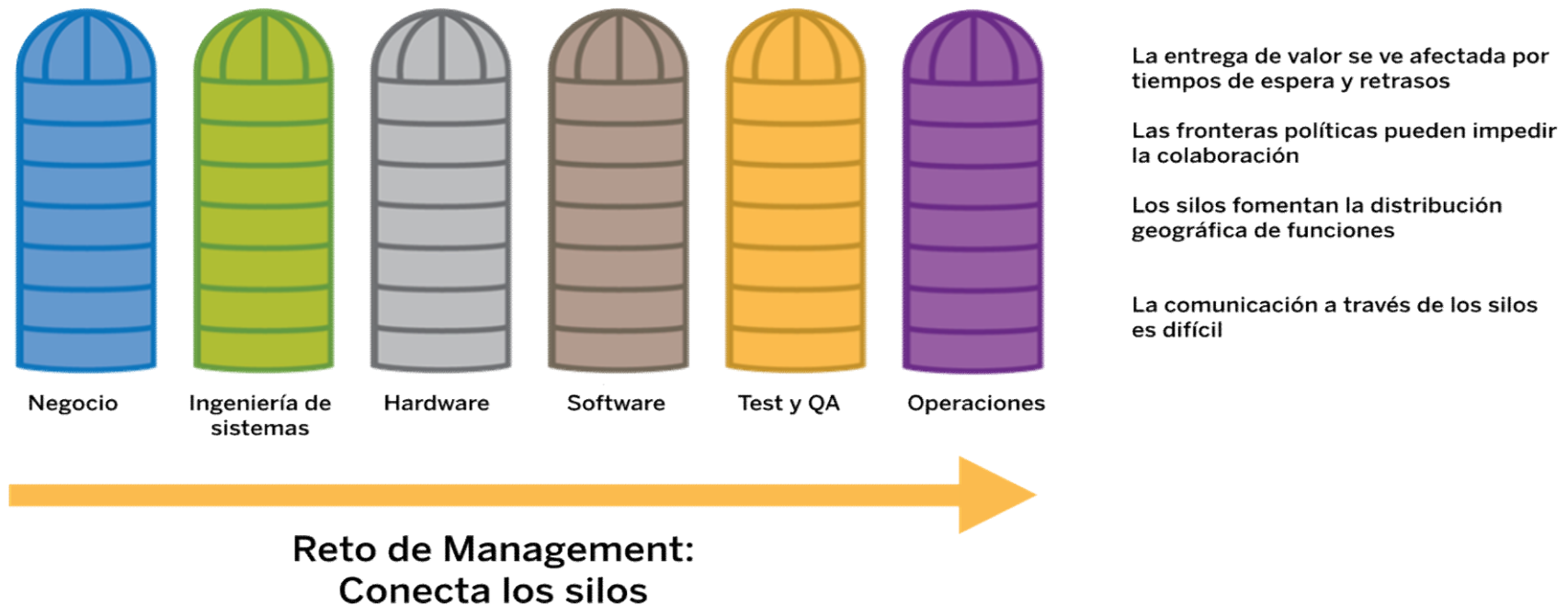
(Tuckman)

## Dando forma al equipo

- Equipos multifuncionales
- Habilidades en forma de T
- Matriz de competencias
- Desarrollar competencias



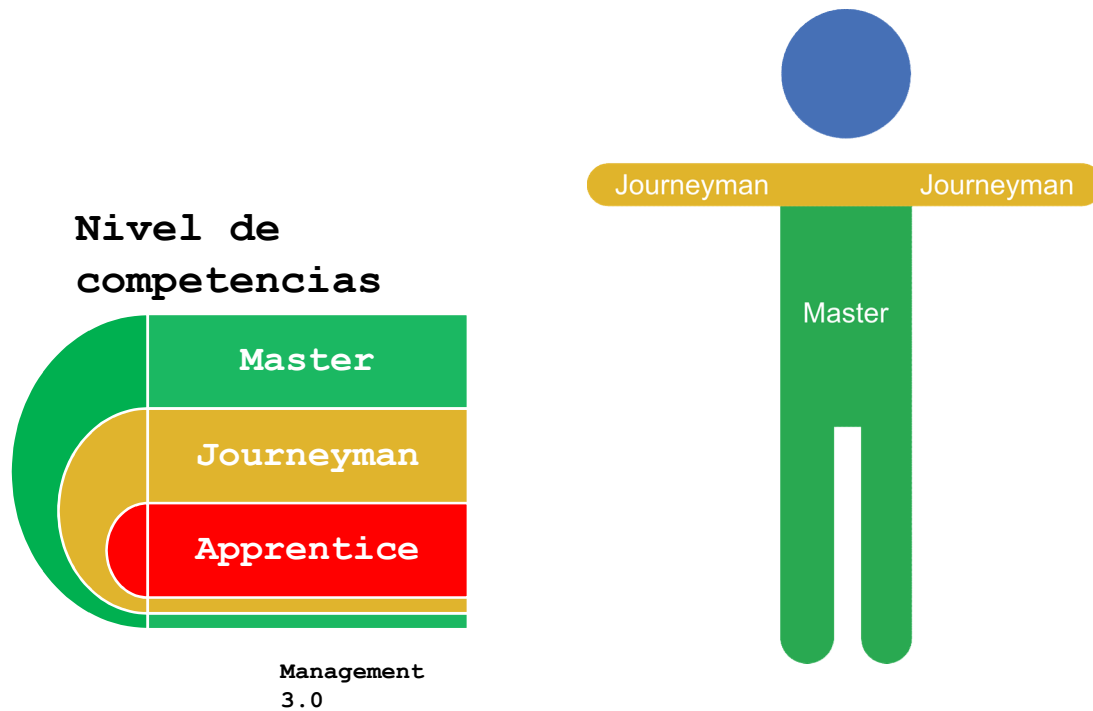
## Equipos multifuncionales



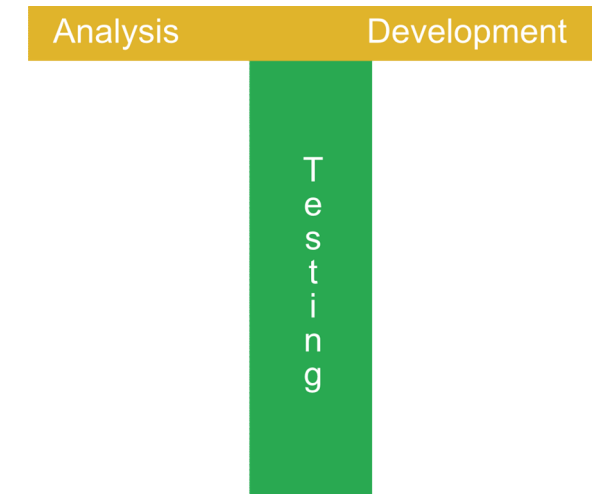
- El equipo debe incluir todas las competencias necesarias para hacer el trabajo.

# Perfil de los miembros del equipo

Habilidades en forma de T (T-skilled)



## Ejemplo de IT Tester



## Competencias que mi equipo necesita

### Materia

Seguridad

Management

Desarrollo

Negocio

Operaciones

Producto

### Herramientas y tecnologías

Plataforma de herramientas

TDD

BDD

Jira

Micro-servicios

Google Apps

### Procesos y prácticas

Kanban

XP

ITIL

Scrum

Lean

### Habilidades

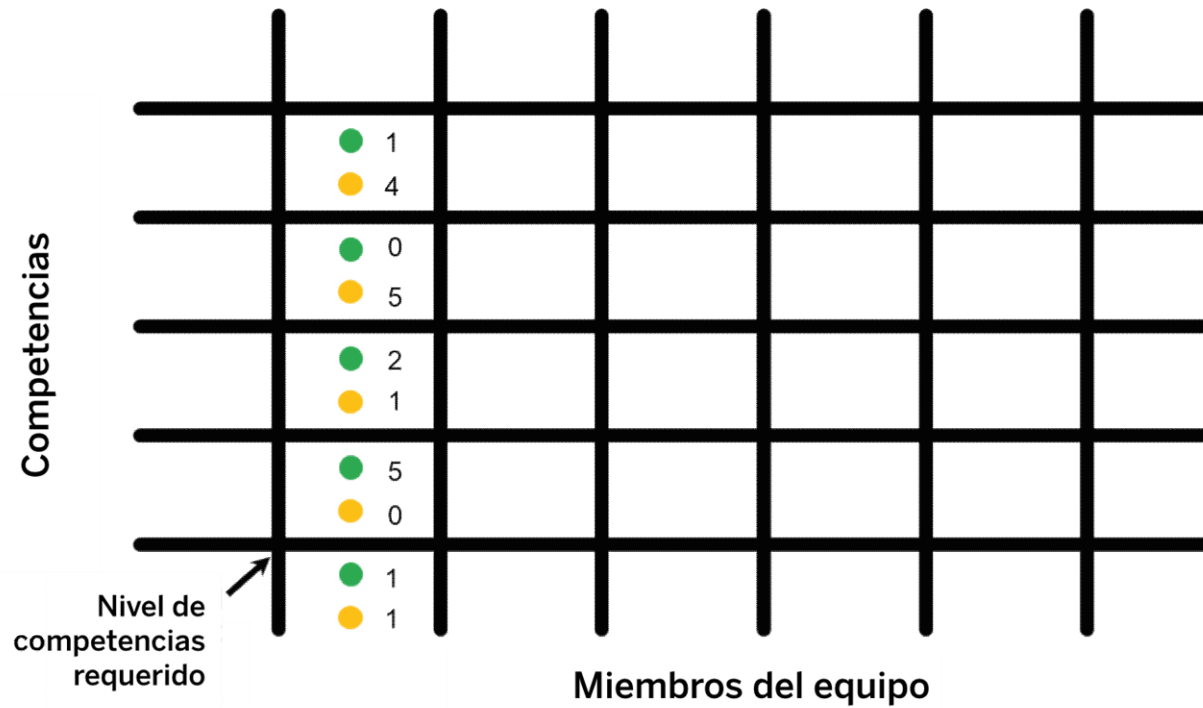
Visual thinking

Inteligencia emocional

Comunicación no violenta

Liderazgo

## Matriz de competencias del equipo



Master



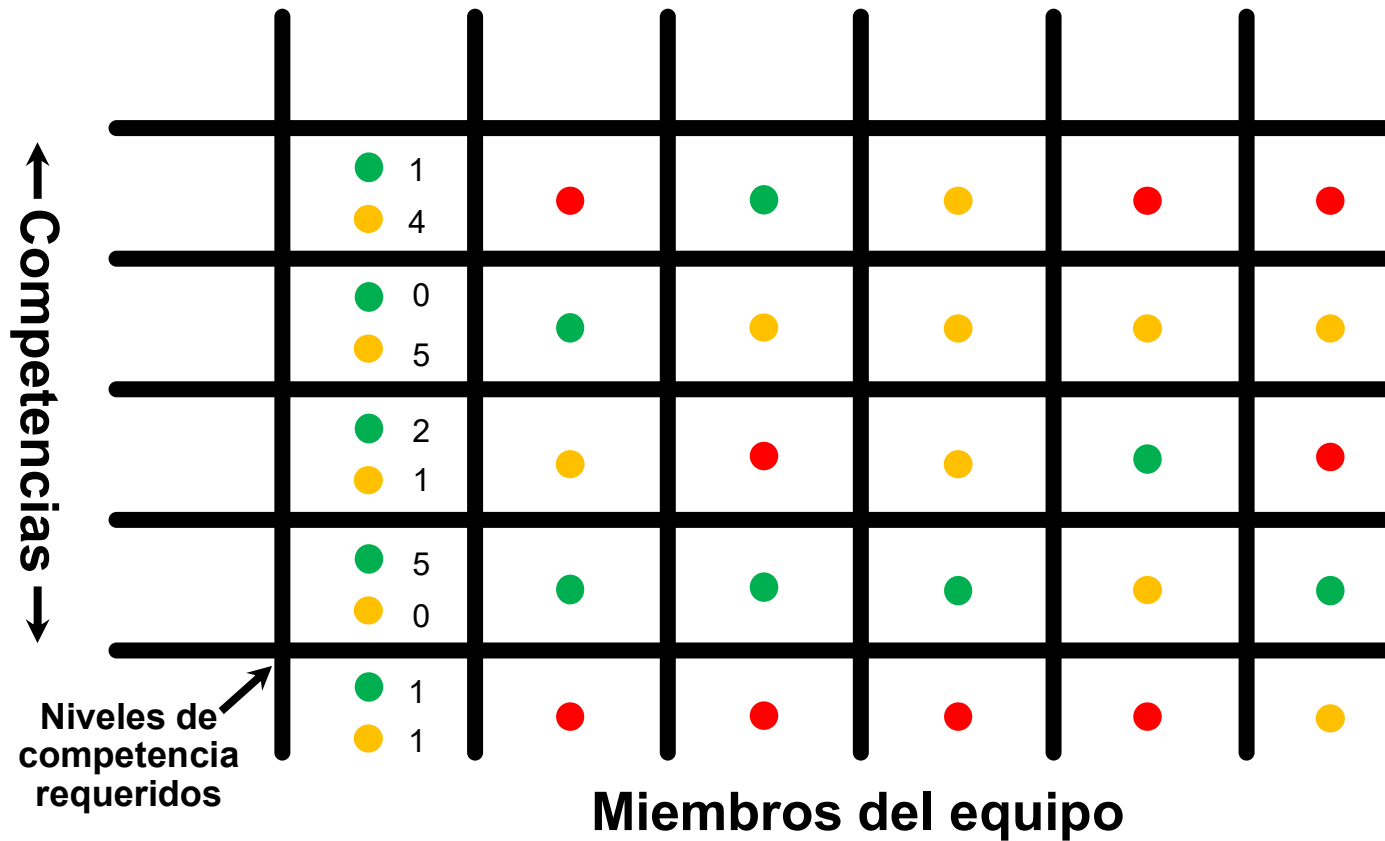
Journeyman



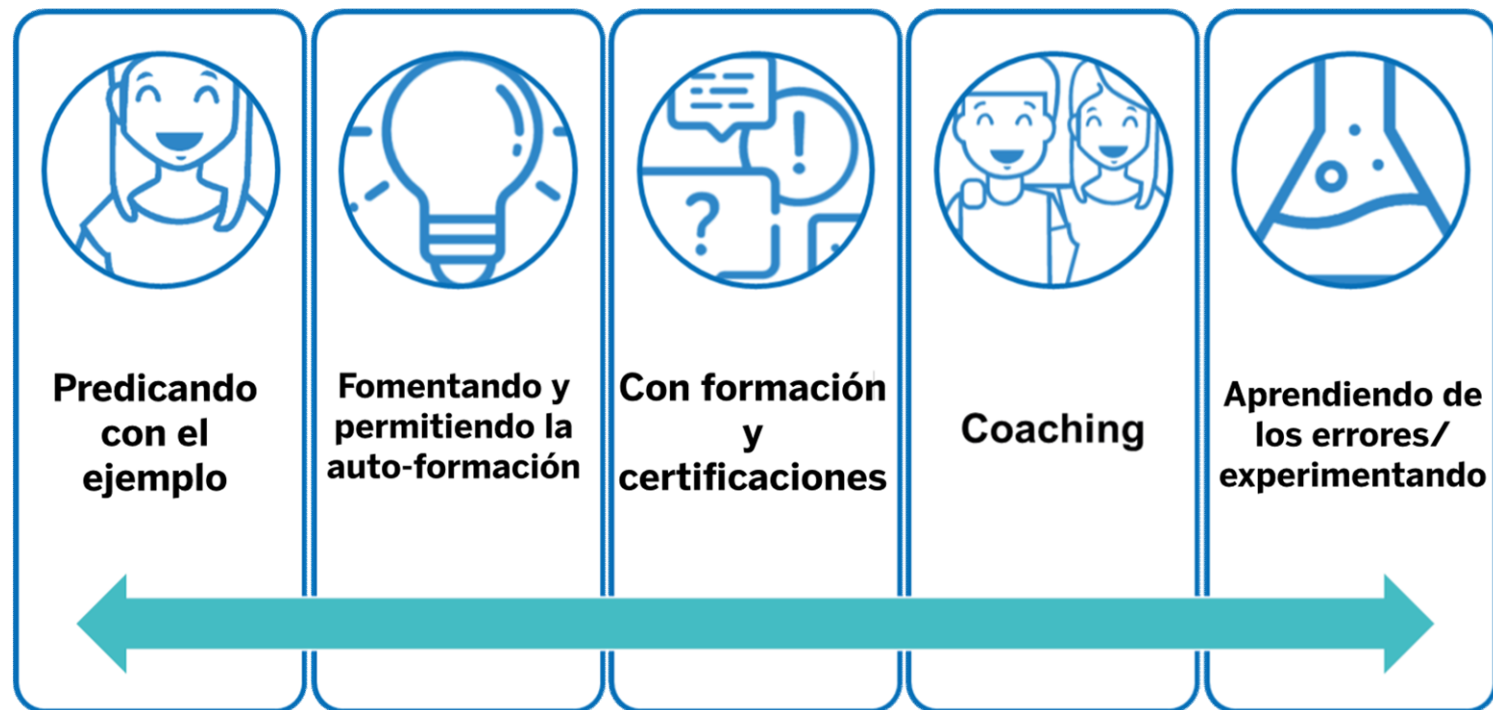
Apprentice



## Matriz de competencias del equipo



¿Cómo puedes ayudar a desarrollar competencias individuales en tus equipos?



# Team canvas

## TEAM CANVAS

Nombre del equipo

Fecha

PERSONAS Y  
ROLES

OBJETIVOS COMUNES

VALORES

REGLAS Y  
ACTIVIDADES

PROPÓSITO

OBJETIVOS  
PERSONALES

NECESIDAD Y  
EXPECTATIVAS

RELACIONES ENTRE EL EQUIPO

RELACIONES CON OTROS EQUIPOS

Adaptado del Team canvas creado por [www.theteamcanvas.com](http://www.theteamcanvas.com)

## Más factores de un equipo efectivo

- Composición del equipo (multifuncional y autoorganizado)
- Aprendizaje continuo
- Estabilidad y coubicación
- Liderazgo emergente
- Alinear los objetivos del equipo a la estrategia de la empresa
- Diseñar tareas vinculadas a los objetivos del equipo (OKRs)

## Agile significa Pull

### “PUSH”

En los modelos tradicionales la organización del trabajo es “**push**”: Un conjunto de tareas se definen y asignan a un individuo para que las ejecute o complete.

**El equipo elige cuándo y cuánto trabajo acometer.** Tanto Scrum como Kanban son sistemas de planificación tipo “Pull”



### “PULL”

En los modelos “**pull**” las tareas están en una cola (backlog, “to do”, etc.) a menudo priorizadas. Los individuos o equipos van tomando tareas y empujando la cola de trabajo (pull) fuera de la lista entregando resultados y permitiendo el flujo.

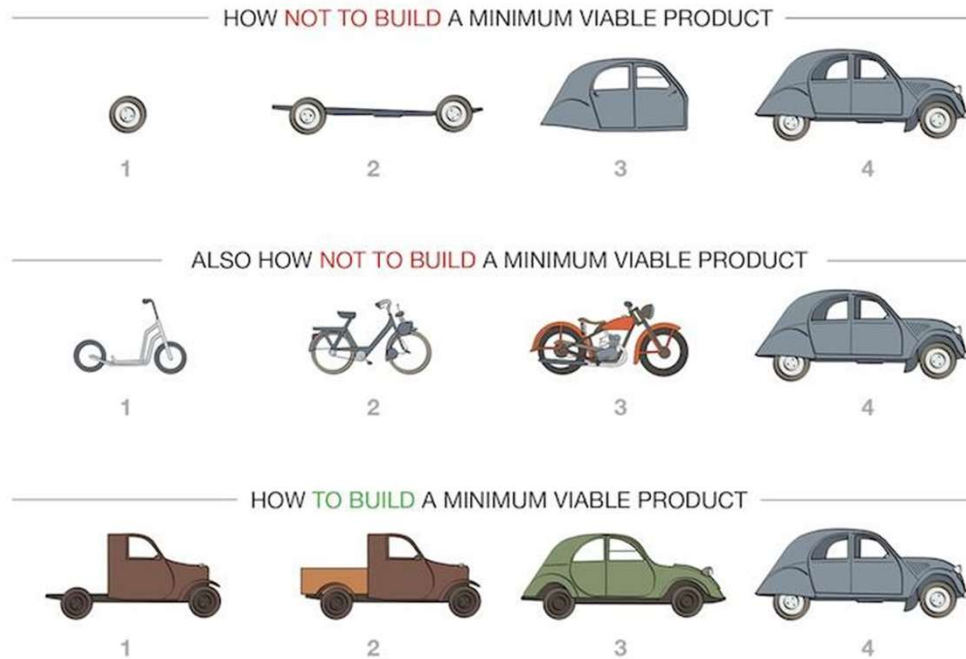
## Contexto VUCA



# Mindset Agile: Ser vs. Hacer

**Ágil = gestión de la incertidumbre. Descubrir valor lo antes posible**

**El Mínimo Producto Viable no se planifica ni se anticipa...se descubre**



## Principios del Manifiesto Ágil

<https://agilemanifesto.org>



## Principios del Manifiesto Ágil

1/ Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor

## Principios del Manifiesto Ágil

**2/ Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente**

## Principios del Manifiesto Ágil

**3/ Entregamos software funcional  
frecuentemente, entre dos semanas  
y dos meses, con preferencia al  
periodo de tiempo más corto  
posible**

## Principios del Manifiesto Ágil

4/ Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto

## Principios del Manifiesto Ágil

**5/** Los proyectos se desarrollan en torno a individuos motivados.

Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo

## Principios del Manifiesto Ágil

**6/** El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara

## Principios del Manifiesto Ágil

**7/** El software funcionando es la medida principal de progreso

## Principios del Manifiesto Ágil

**8/ Los procesos Ágiles promueven el desarrollo sostenible.**

Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida



## Principios del Manifiesto Ágil

**9/** La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad

## Principios del Manifiesto Ágil

**10/** La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial

## Principios del Manifiesto Ágil

**11/** Las mejores arquitecturas,  
requisitos y diseños emergen de  
equipos auto-organizados

## Principios del Manifiesto Ágil

**12/** A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia

## Valores del manifiesto ágil

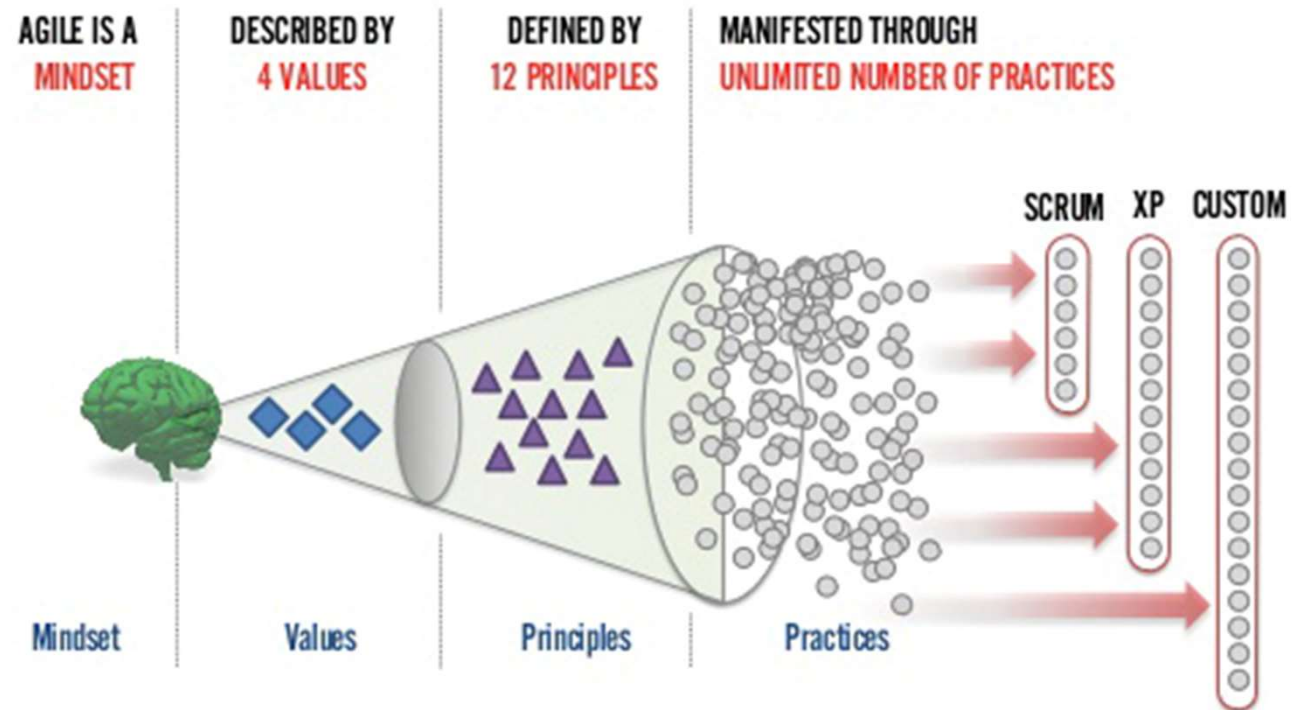
1. **Individuos e interacciones** sobre Procesos y Herramientas

2. **Software funcionando** sobre Documentación extensiva

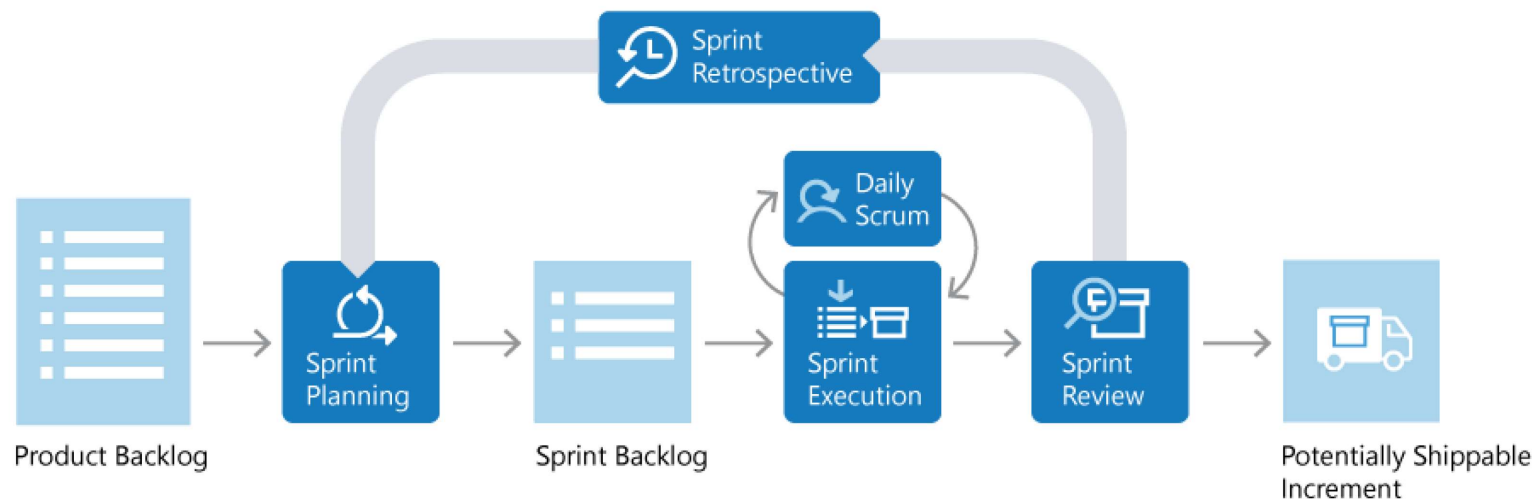
3. **Respuesta al cambio** sobre Seguir un plan

4. **Colaboración con el cliente** sobre Negociación contractual

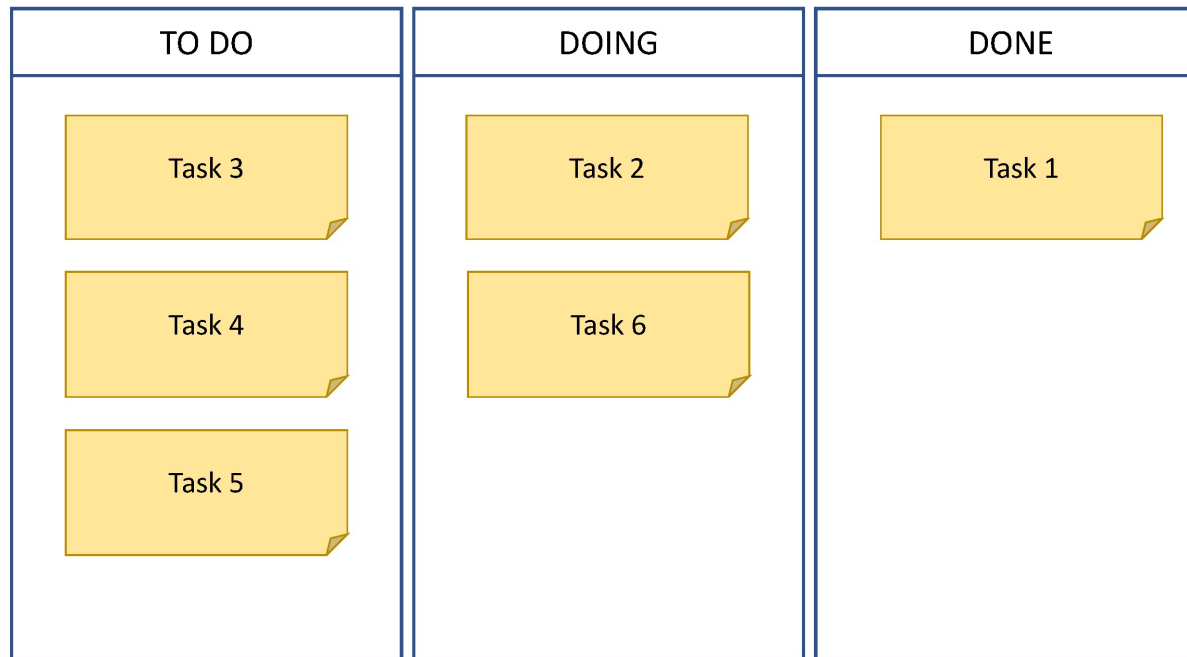
# Resumiendo...



# Scrum Life Cycle

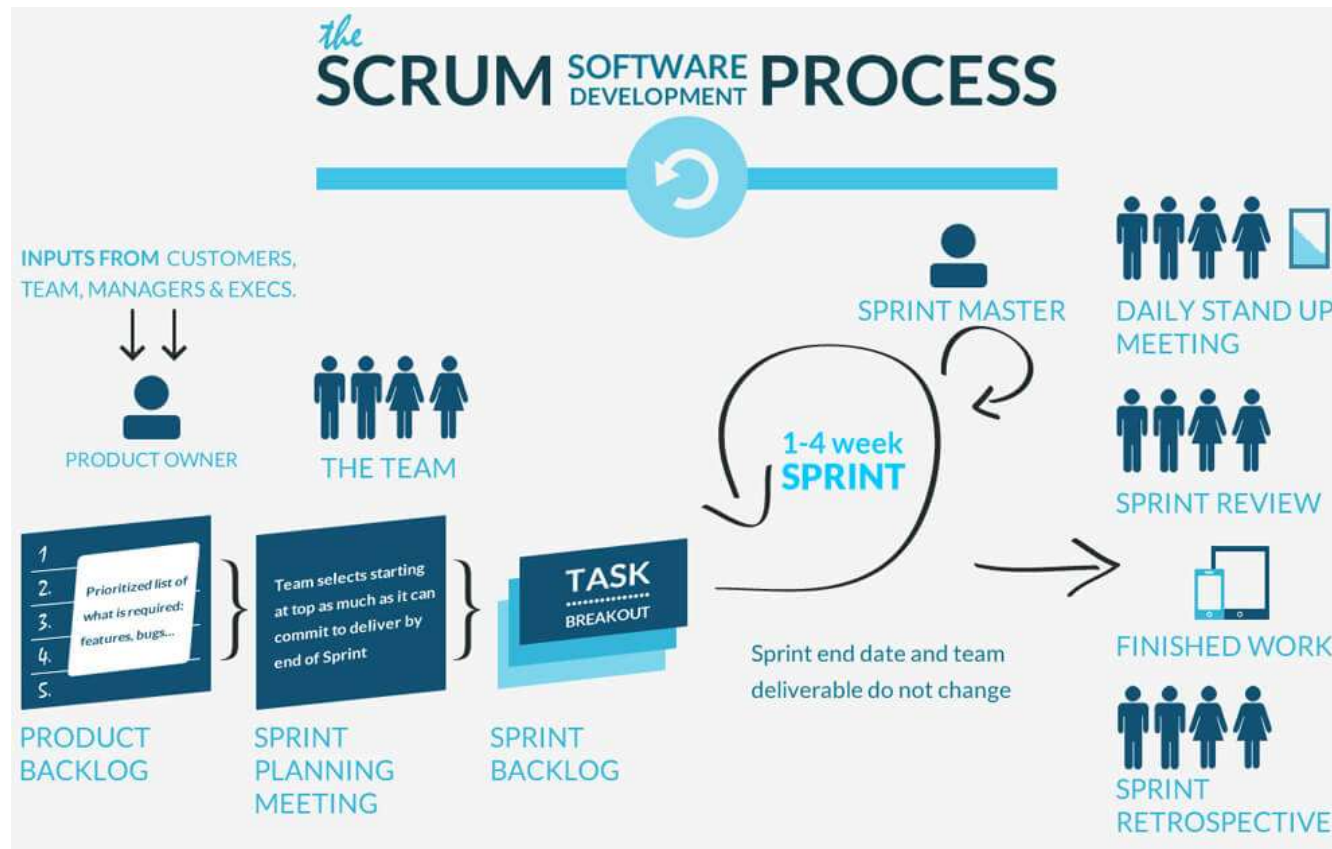


# Kanban board

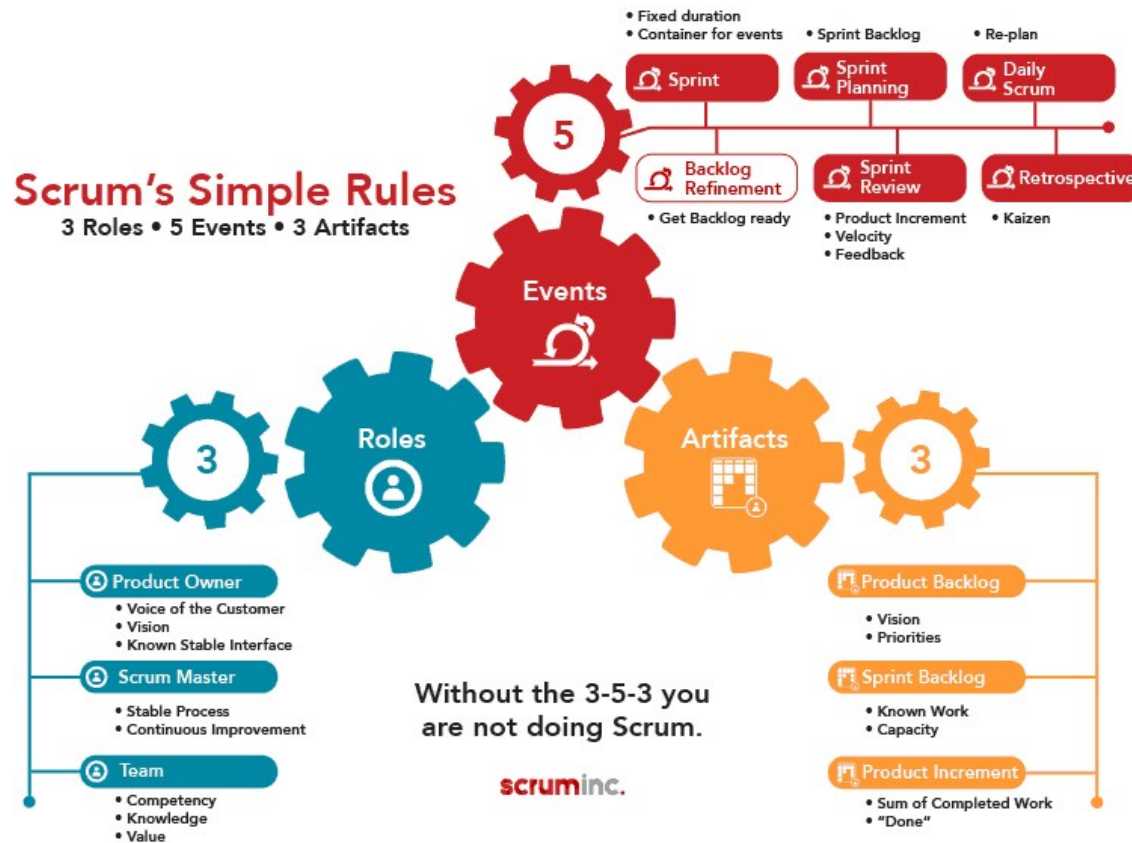




# The Scrum Process



# The Scrum Process



## Summarizing...

Roles	Artifacts	Events
<ul style="list-style-type: none"><li>• PO</li><li>• Scrum master</li><li>• Development team</li></ul>	<ul style="list-style-type: none"><li>• Product backlog</li><li>• Sprint backlog</li><li>• Product increment</li></ul>	<ul style="list-style-type: none"><li>• Sprint</li><li>• Sprint planning</li><li>• Sprint review</li><li>• Sprint retrospective</li><li>• Daily scrum</li></ul>

# XP: eXtreme Programming

## Marco de trabajo centrado en prácticas de ingeniería de software

Cambio de mentalidad y valores del equipo:

1. Comunicación

2. Simplicidad

3. Feedback

4. Coraje

5. Respeto

Conjunto de prácticas que aportan rigor técnico:

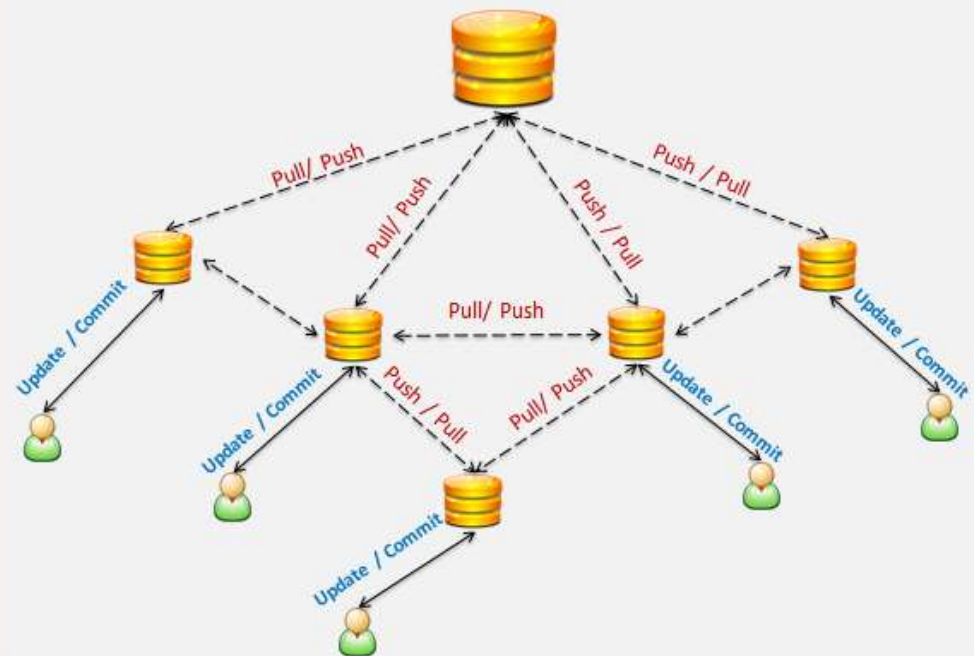
- **Refactoring**
- Simple Design
- **Pair Programming**
- **Test-Driven Development (TDD)**
- Coding Standard
- Sustainable Pace
- **Continuous Integration**
- Small Releases
- Metaphor
- Collective Ownership
- Whole Team
- Planning Game (Sprint Planning – Estimación del alcance)

# Integración Continua

Centralised Version Control System



Distributed Version Control System



## Tipos de Pruebas de Software

### FUNCIONALES

- Unitarias
- Aceptación
- Integración
- Regresión

### NO FUNCIONALES

- Carga
- Estrés
- Escalabilidad
- Portabilidad

## Herramientas de Testing

- **JMeter**: pruebas de Rendimiento
- **Selenium**: pruebas Automatizadas
- **JUnit**: pruebas Unitarias
- **TestNG**: Ejecutador pruebas Selenium
- ...

# TDD: Test Driven Development

TDD - Desarrollo dirigido por pruebas **Práctica de ingeniería de software** que involucra otras dos prácticas:

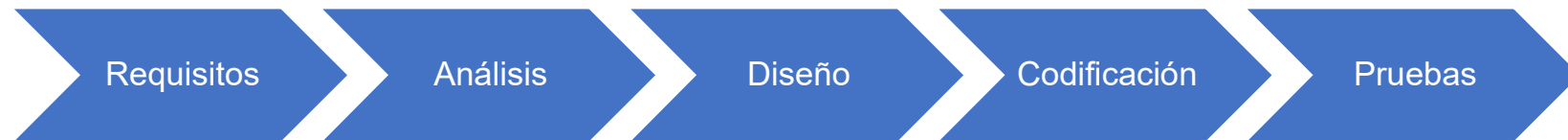
- **Escribir las pruebas primero** (Test First Development)
- **Refactorización** (Refactoring)

Objetivos:

1. **Minimizar** el número de bugs.
2. Implementar las **funcionalidades justas** que el cliente necesita.
3. Producir software **modular**, altamente **reutilizable** y preparado para el **cambio**.



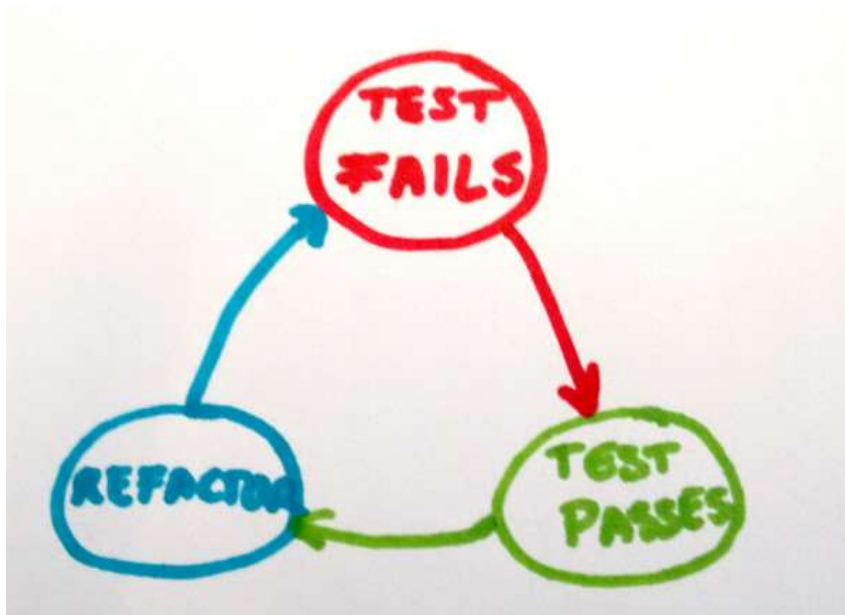
# Flujo de Trabajo Tradicional



## Flujo de Trabajo con TDD



## TDD: proceso recursivo basado en 3 pasos



1. **ROJO:** escribir un test que falle
2. **VERDE:** escribir el mínimo código fuente necesario para hacer pasar el test
3. **AZUL:** refactorizar el código, limpiar, evitar redundancia de código, etc.

# Referencias

**Guía oficial de Scrum en español:**

<http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf#zoom=100>

**Sitio de la Scrum Alliance:**

<http://scrumalliance.org>

**Sitio de Scrum.org:**

<http://scrum.org>

**Suggested Reading for PSD™:**

<https://www.scrum.org/resources/suggested-reading-professional-scrum-developer>