



Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

# Clase 3

## Desarrollo ágil

IIC2143 - Ingeniería de Software  
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

14 de marzo de 2018

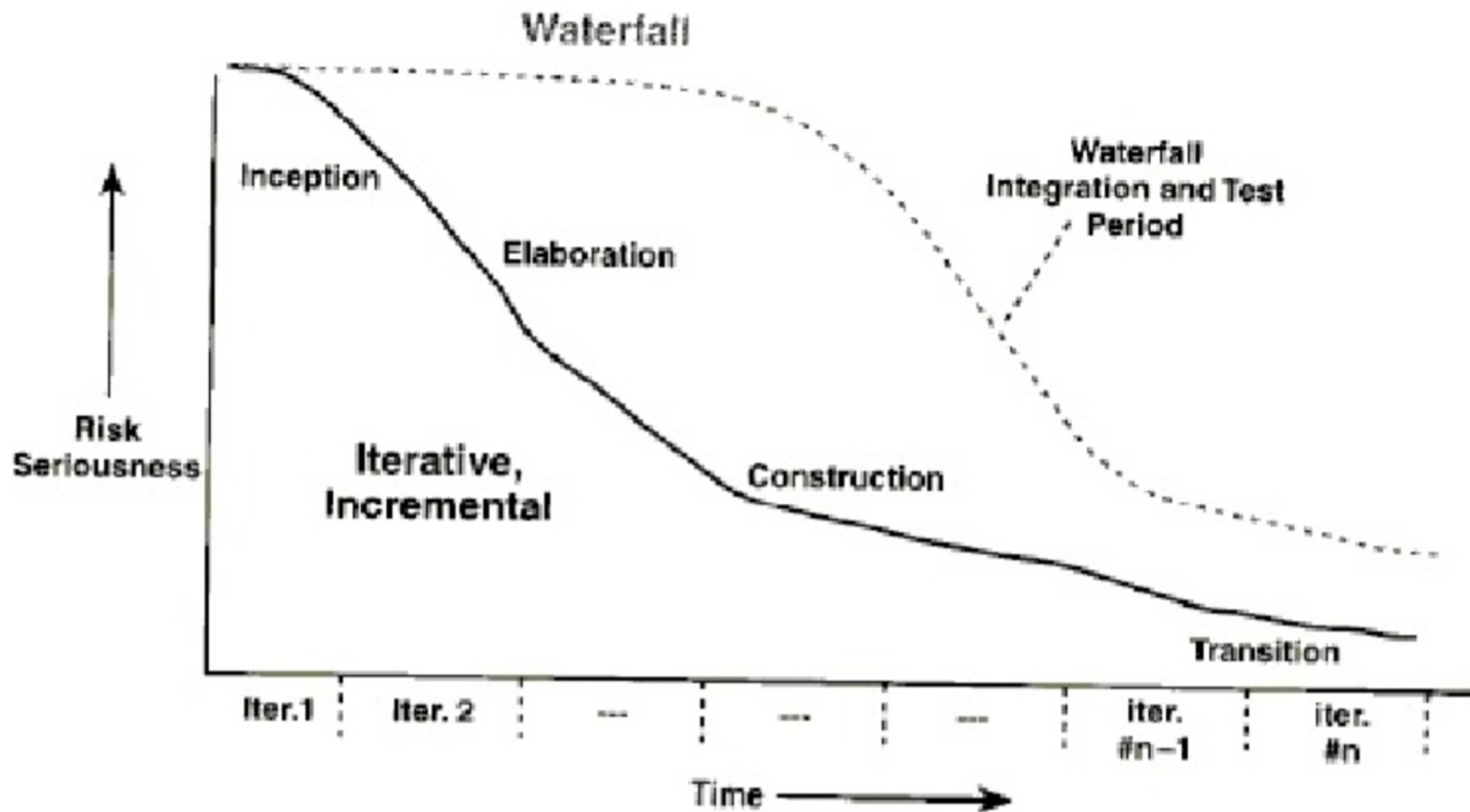
# ¿Por qué se utilizan un proceso de desarrollo?

- Disminuir el riesgo de fracaso
- Disminuir costos
- Visibilizar el avance del desarrollo

# Modelos de procesos

- Cascada
- Iterativo
  - Prototipado
  - Espiral
  - RUP
- Incremental
- Ágil
  - XP
  - Kanban
  - Scrum

# Riesgo vs Tiempo



# Iterativo vs Incremental

- Iterativo: pequeños avances sobre todas las funcionalidades
- Incremental: desarrollo de algunas funcionalidades completas

**Incremental**

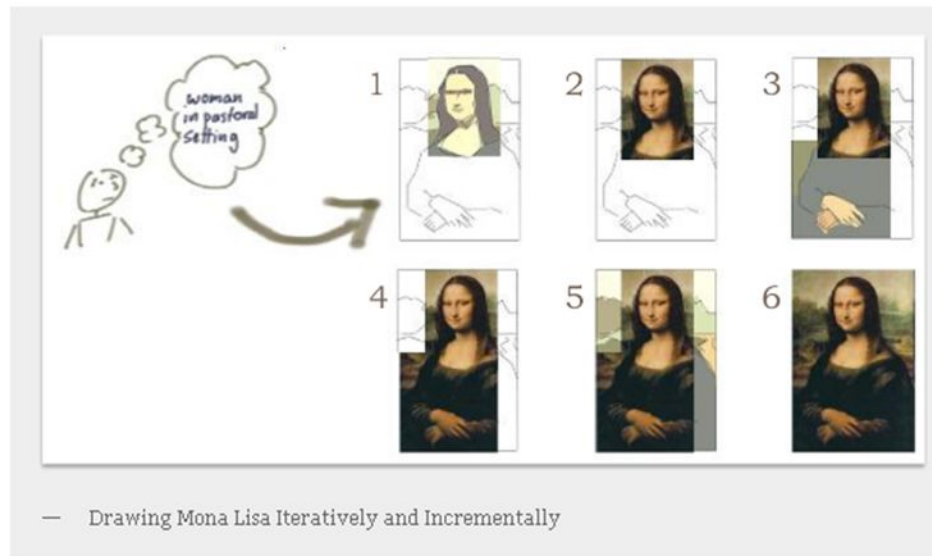


**Iterative**



# Ágil

- Iterativo + Incremental
- Desarrollar pequeñas porciones de algunas funcionalidades en cada iteración, siempre con un producto funcional



# Ágil

- Formalizado en 2001 por 17 desarrolladores de software
- Juntos publican el “*Manifesto for Agile Software Development*”

href: <http://agilemanifesto.org/>

# Ágil

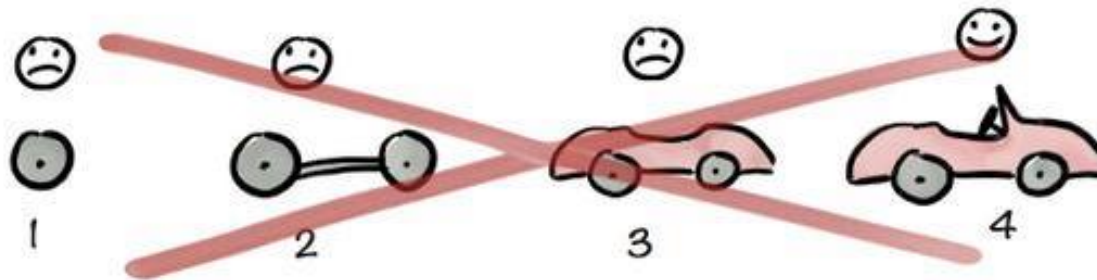
- Conjunto de 12 principios que fomentan:
  - **Individuos e interacciones** sobre procesos y herramientas
  - **Software funcionando** sobre documentación extensiva
  - **Colaboración con el cliente** sobre negociación contractual
  - **Respuesta ante el cambio** sobre seguir un plan



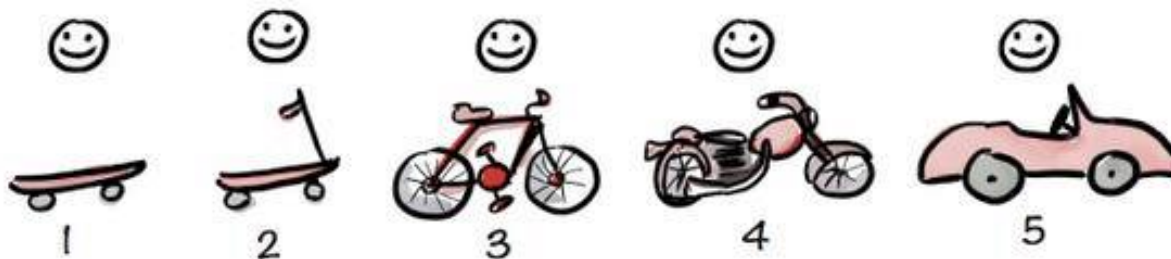
# Ágil

## Minimum Viable Product (MVP):

Not like this....



Like this!



# Extreme Programming (XP)

- Propuesto en 1999 por Kent Beck
- Se basa en 5 valores:
  - Simplicidad (YAGNI o KISS)
  - Comunicación
  - Retroalimentación
  - Coraje
  - Respeto

# Extreme Programming (XP)

- Algunas prácticas que fomentan los valores:
  - Pair programming

# Extreme Programming (XP)

- Pair programming:



By Lisamarie Babik - Ted & IanUploaded by Edward, CC BY 2.0, <https://commons.wikimedia.org/w/index.php?curid=9546406>

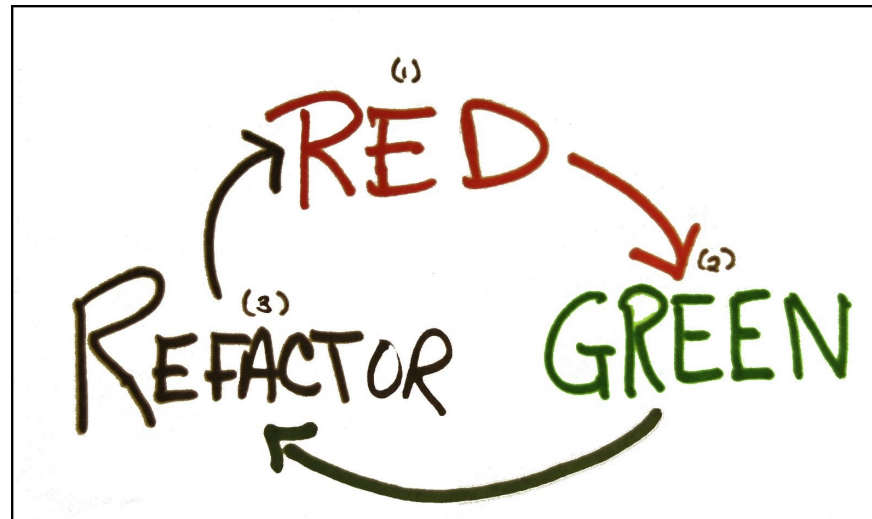
# Extreme Programming (XP)

- Algunas prácticas que fomentan los valores:
  - Pair programming
  - Planning game – Planning poker
  - Test-driven development (TDD)

# Extreme Programming (XP)

## Test-driven development (TDD):

1. Agregar tests hasta que fallen
2. Escribir código hasta que pasen
3. Refactorizar código sin romper tests



# Extreme Programming (XP)

- Algunas prácticas que fomentan los valores:
  - Pair programming
  - Planning game – Planning poker
  - Test-driven development (TDD)
  - Continuous integration - Small releases
  - Coding standards - Collective code ownership

# Kanban

- Del japonés Kan (visual) – Ban (tarjeta)
- Implementado en fábricas de Toyota en los 50
- Proceso para creación de productos con énfasis en entrega continua

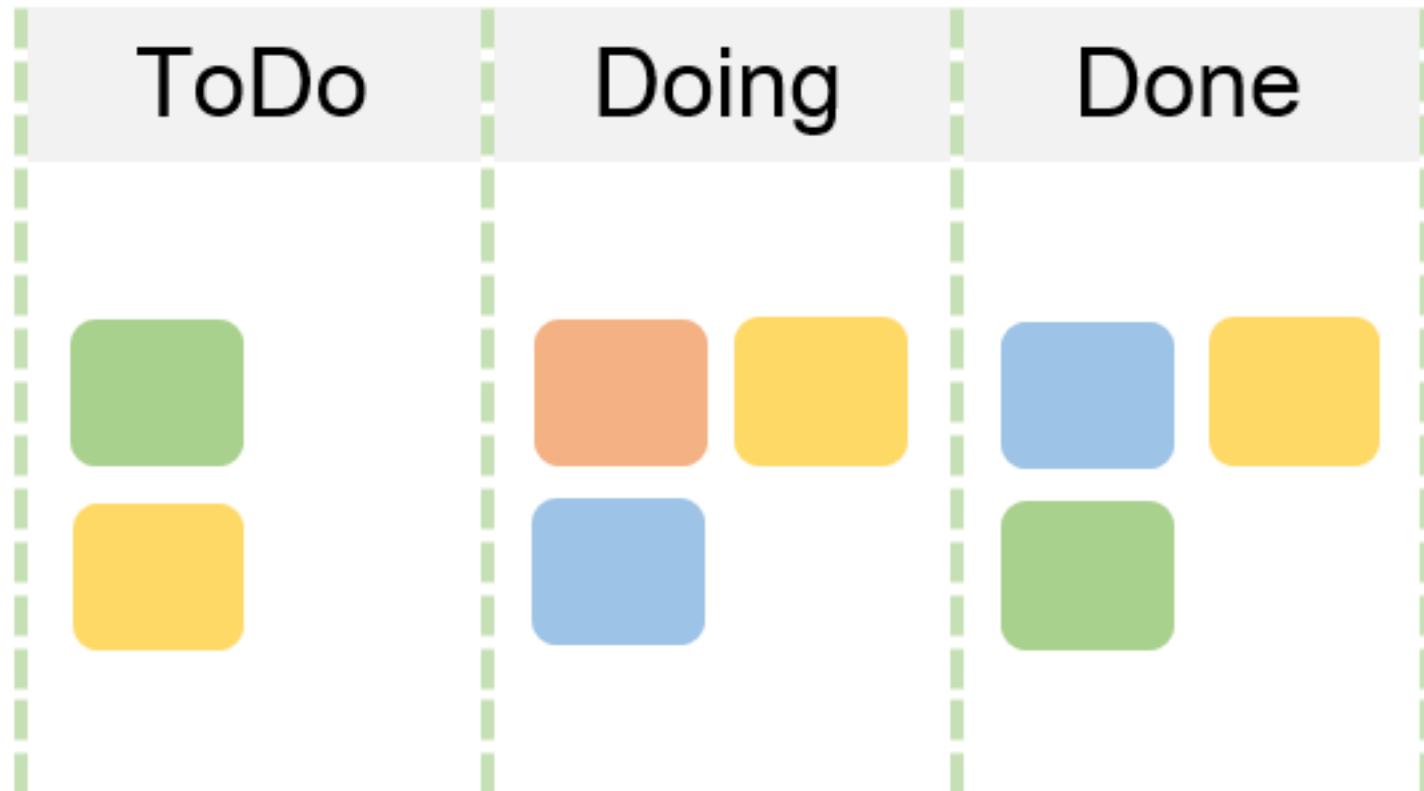


# Kanban

## 3 principios:

- Visualizar el trabajo
- Limitar el trabajo simultáneo
- Optimizar el flujo de tareas finalizadas

# Kanban





Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ciencia de la Computación

# Clase 3

## Desarrollo ágil

IIC2143 - Ingeniería de Software  
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

14 de marzo de 2018