



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

Clase 4

Scrum + Requisitos

IIC2143 - Ingeniería de Software
Sección 1

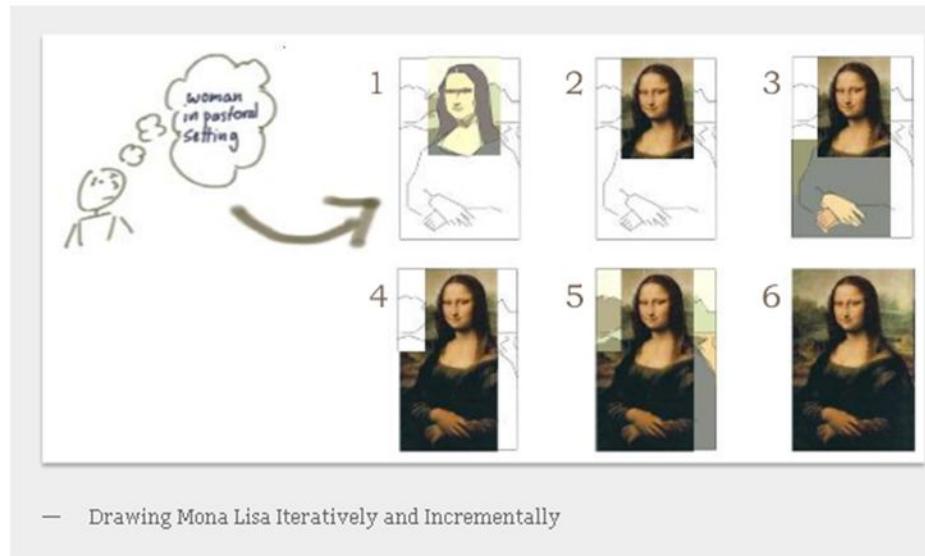
Rodrigo Saffie

rasaffie@uc.cl

19 de marzo de 2018

Ágil

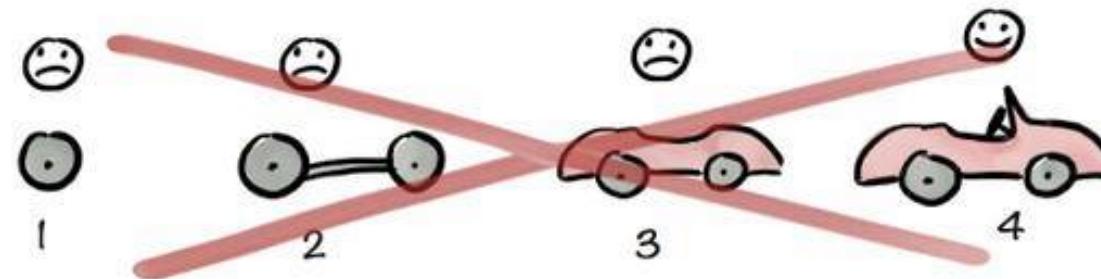
- Iterativo + Incremental
- Desarrollar pequeñas porciones de algunas funcionalidades en cada iteración, siempre con un producto funcional



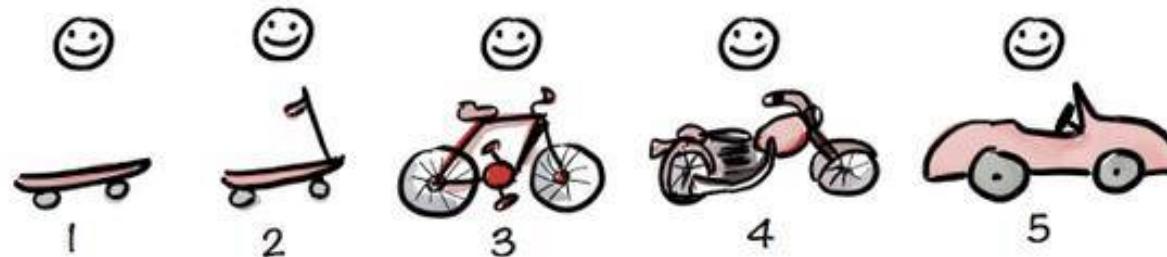
Ágil

Minimum Viable Product (MVP):

Not like this....



Like this!



Extreme Programming (XP)

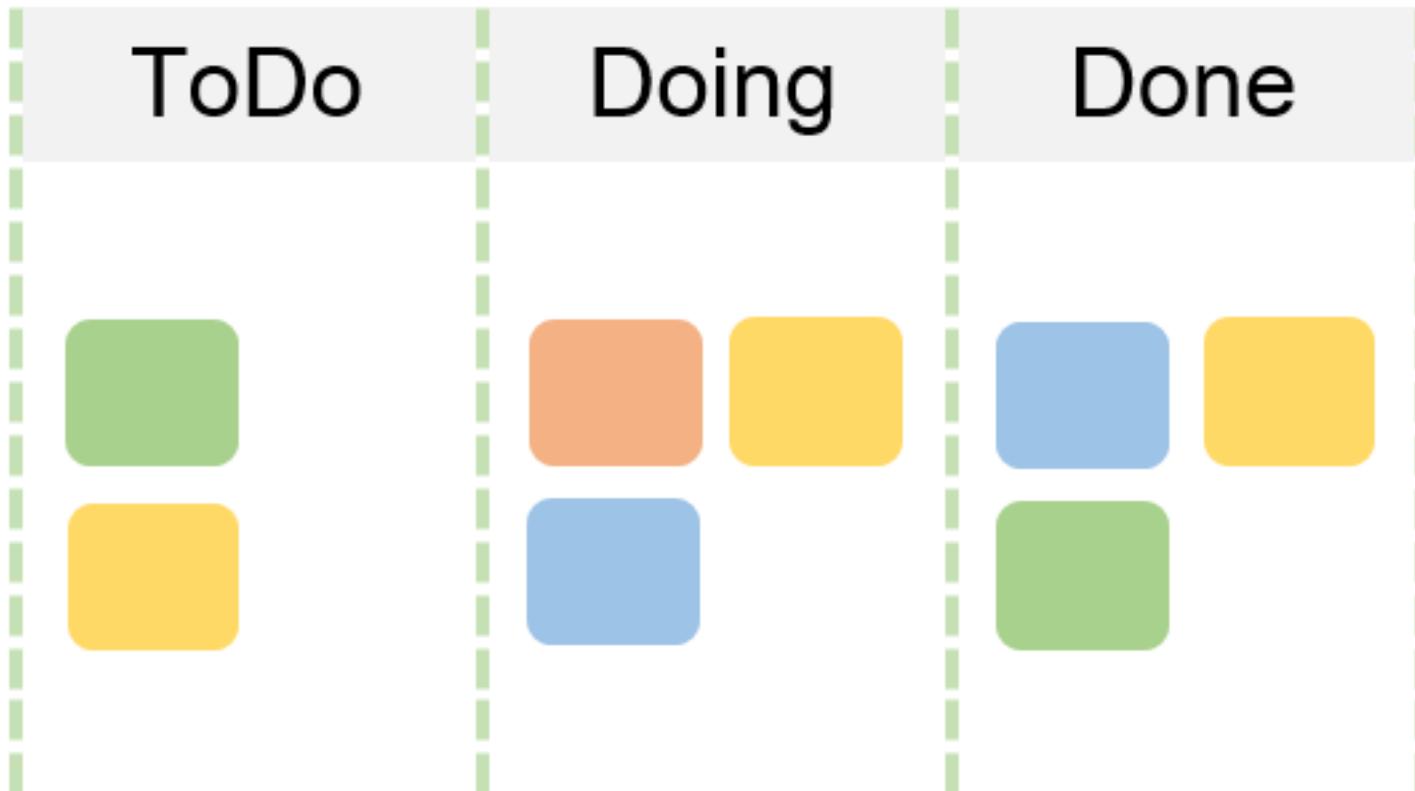
- Algunas prácticas que fomentan los valores:
 - Pair programming
 - Planning game – Planning poker
 - Test-driven development (TDD)
 - Continuous integration - Small releases
 - Coding standards - Collective code ownership

Kanban

3 principios:

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

Kanban



Scrum



Scrum

- Desarrollado a principios de los 90' por Ken Schwaber
- Uno de los modelos más populares en la actualidad

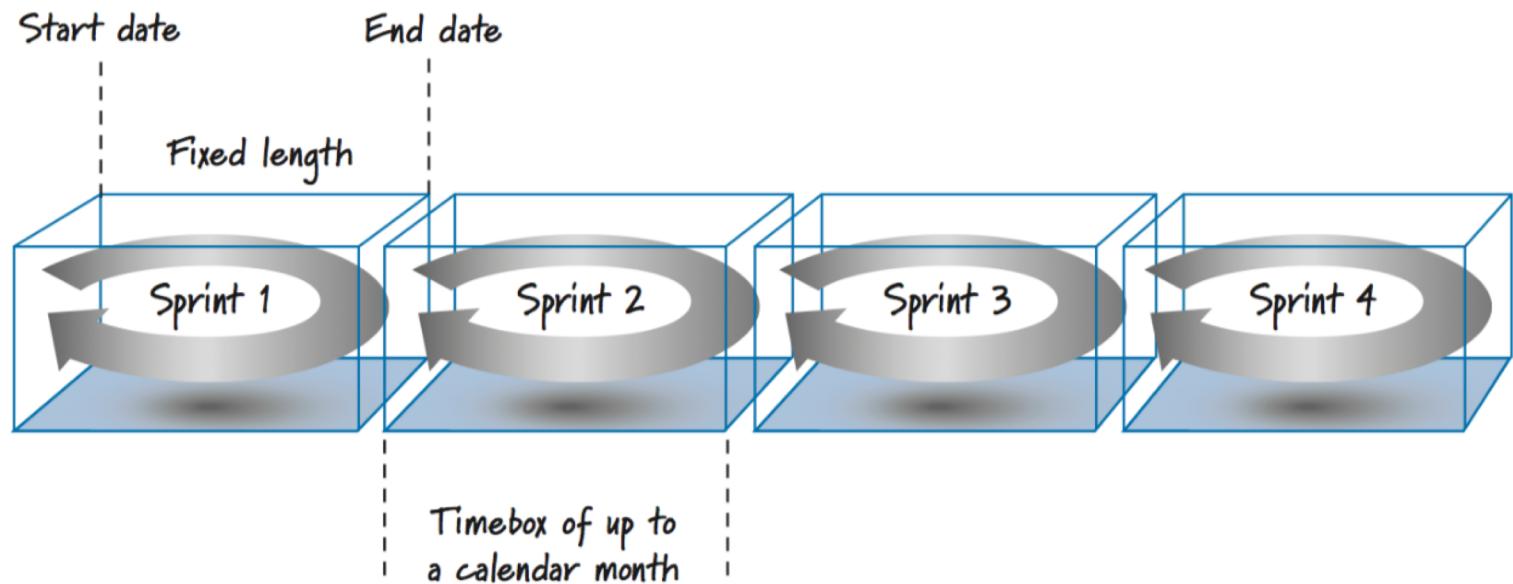
Scrum

Roles en el equipo:

- Product Owner:
 - responsable del producto, representa al negocio
 - decide y prioriza funcionalidades
- Scrum Master:
 - facilitador del desarrollo, resuelve problemas del equipo
 - mediador de los procesos
- Development Team:
 - 3 a 8 personas
 - independiente, multifuncional

Scrum

- Software se desarrolla en iteraciones llamadas *sprints* (cada una de 2 a 3 semanas)



Scrum

- Software se desarrolla en iteraciones llamadas *sprint* (cada una de 2 a 3 semanas)
- Al comenzar un *sprint*: ***Sprint planning***
 - entre todo el equipo se seleccionan tareas a realizar y se discuten
- Durante un *sprint*: **Daily scrum**
 - al principio de cada día cada persona dice **qué hizo ayer, qué hará hoy y qué problemas podría tener**
 - no debe durar más de 15-20 minutos

Scrum



Scrum

- Al finalizar un *sprint*:
 - **Sprint review:**
 - se revisa el trabajo realizado (completado y no completado)
 - se realiza una demo al cliente
 - **Sprint retrospective:**
 - se reflexiona sobre el trabajo realizado
 - se discute cómo mejorar el proceso para el siguiente *sprint*

Requisitos

2 tipos:

- **Funcionales**
 - ¿qué debe hacer el sistema?
- **No funcionales**
 - ¿cómo debe hacerlo? / ¿bajo qué condiciones?
 - rendimiento, usabilidad, disponibilidad, plataforma

Requisitos funcionales

¿Todos entienden lo mismo?

Customer: Hello, I'd like to order a cake.

Employee: Sure, what would you like written on it?

Customer: Could you write "So long, Alicia" in purple?

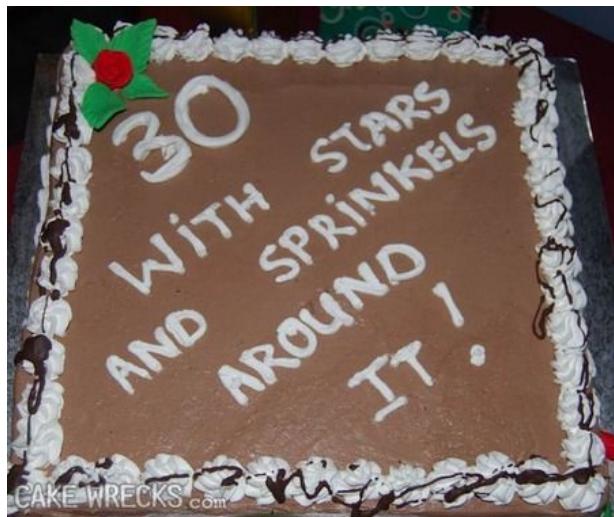
Employee: Sure.

Customer: And put stars around it?

Employee: No problem. I've written this up, and will get a cake decorator right away. We'll have it for you in



Requisitos funcionales



Requisitos funcionales

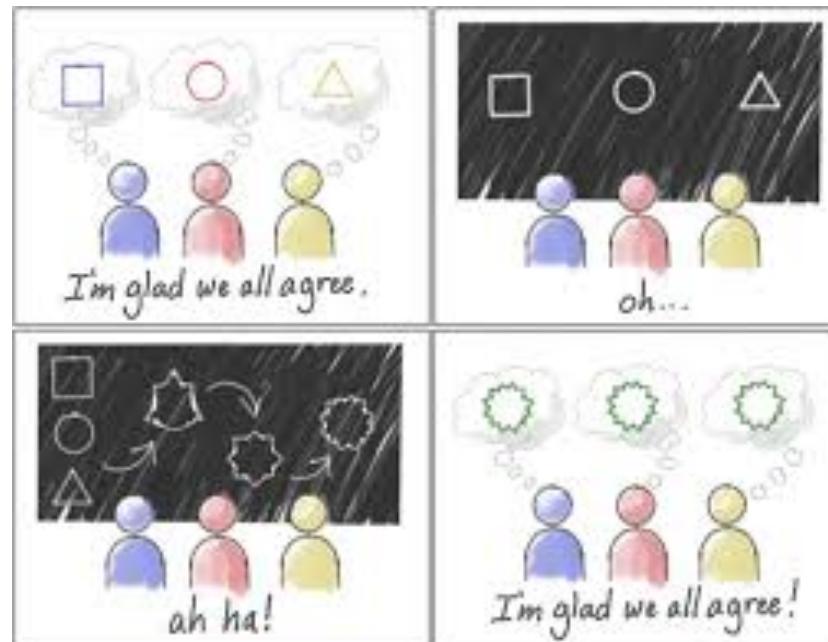
¿Cómo se pueden especificar los requisitos?

2 metodologías ampliamente usadas:

- **Casos de uso (Use cases)**
- **Historias de usuarios (User stories)**

Historias de usuarios

- un documento no garantiza que todos entiendan lo mismo
- el objetivo de las historias de usuarios es generar ***shared understanding***



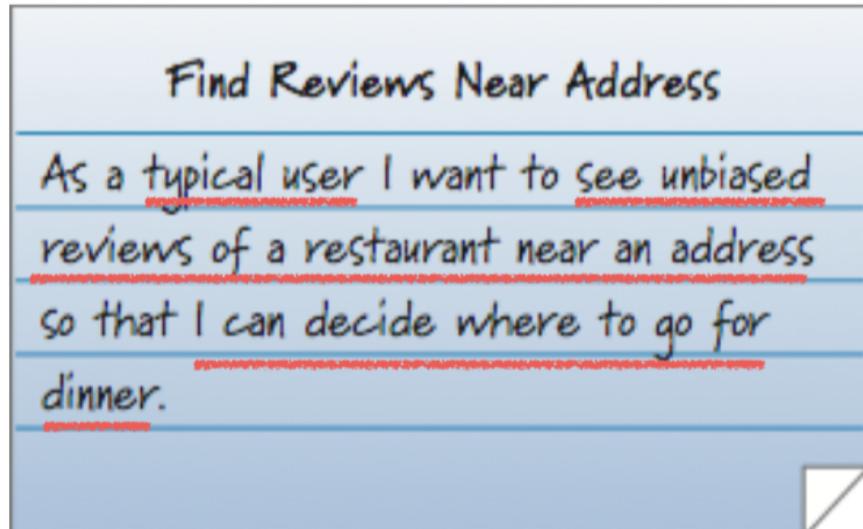
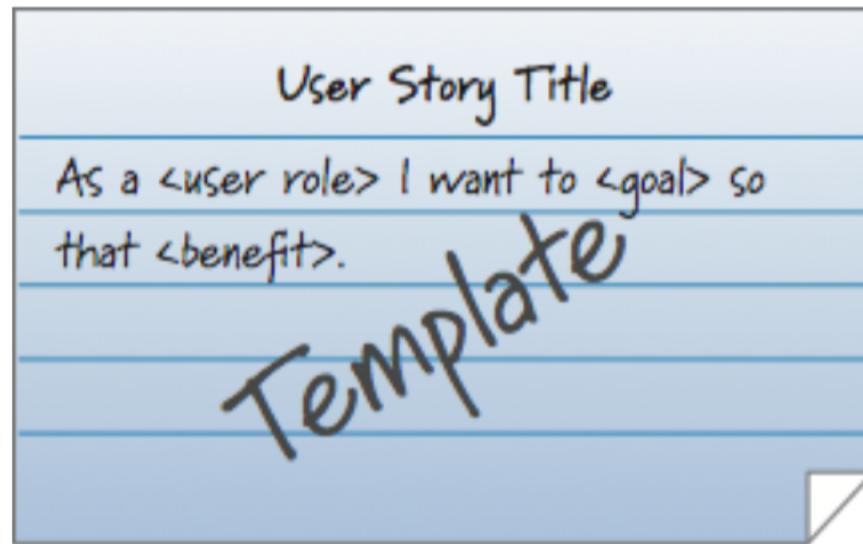
Historias de usuarios

- muy utilizadas en procesos ágiles
- es una descripción informal de las funcionalidades
- se centran en **el valor** que agrega una funcionalidad para **un usuario**
- se definen a través de la colaboración y comunicación de todos los actores del proyecto
- lo importante es el proceso de creación, no el relato mismo

Historias de usuarios

Componentes:

- un usuario
- una meta
- un beneficio



Historias de usuarios

- deben ir acompañadas de condiciones de satisfacción

Upload File

As a wiki user I want to upload a file to the wiki so that I can share it with my colleagues.

Conditions of Satisfaction

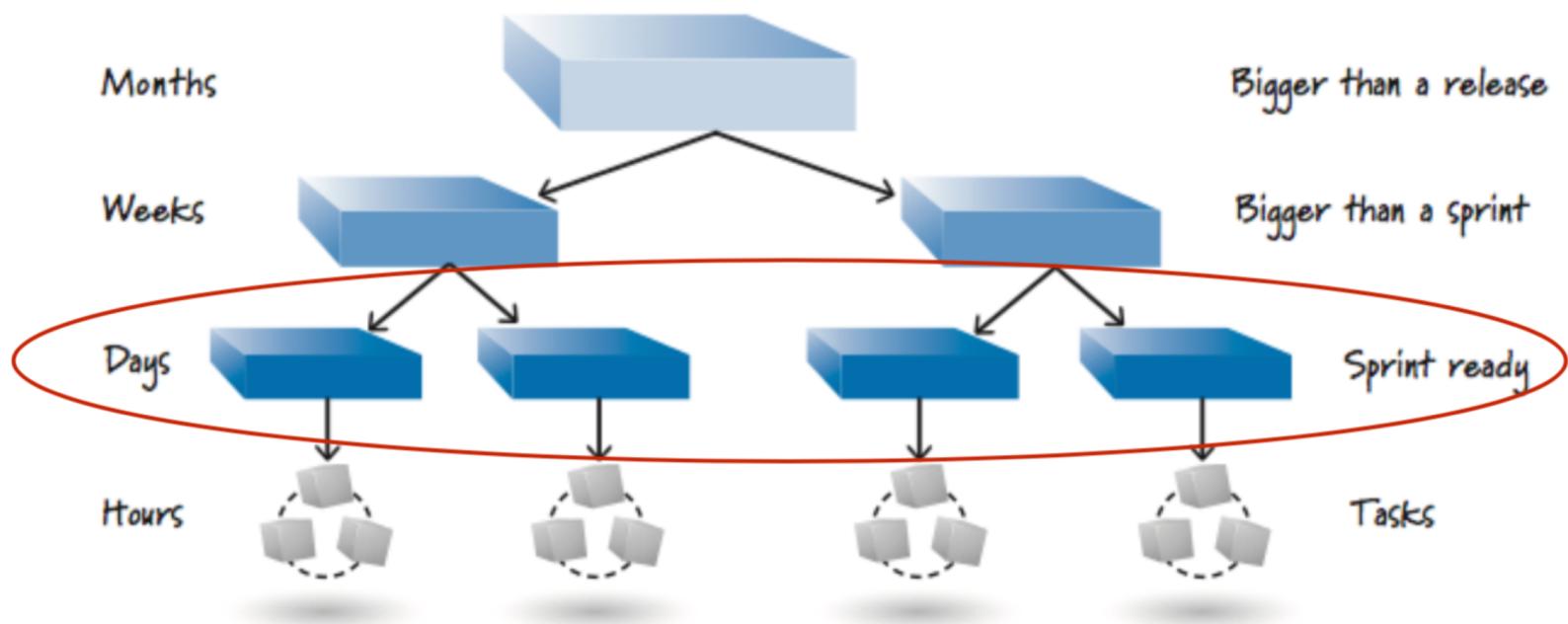
Verify with .txt and .doc files
Verify with jpg, gif, and png files
Verify with .mp4 files <= 1 GB
Verify no DRM-restricted files

Historias de usuarios

Deben ser:

- pequeñas descripciones
- independientes entre ellas
- estimables
- valoradas por un usuario

Nivel de detalle





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

Clase 4

Scrum + Requisitos

IIC2143 - Ingeniería de Software
Sección 1

Rodrigo Saffie

rasaffie@uc.cl

19 de marzo de 2018