

IIC 2143 – Ingeniería de Software

Procesos del Desarrollo de Software

M. Trinidad Vargas
mtvargas1@uc.cl

Anuncios

En canvas está publicado las **cápsulas de Ruby on Rails**

Ruby y Proyecto			✓	▼	+	⋮
⋮	🔗	Guía de instalación Ruby y Ruby on Rails 📄	✓			⋮
⋮	🔗	Capsulas Ruby on Rails 📄	✓			⋮
⋮	🔗	Git cheat sheet 📄	✓			⋮
⋮	📎	Clase práctica 1.pdf	✓			⋮

Junto con las slides de las clases está subido los **capítulos del libro**

Procesos de desarrollo			🔒	▼	+	⋮
⋮	📎	C2_ProcesosDeDesarrollo.pdf	✓			⋮

¿Qué es un proceso de desarrollo de Software?

“Un proceso de desarrollo de software es un conjunto estructurado de actividades requeridas para desarrollar un sistema de software.”

“Ingeniería de software”
Ian Sommerville, Novena edición

¿Por qué necesitamos procesos?

¿Vibe coding?

https://en.wikipedia.org/wiki/Vibe_coding

¿Por qué necesitamos procesos?

Se necesitan procesos de desarrollo **bien definidos, entendibles y repetibles** para:

- Encontrar y mantener buenas prácticas
- Administrar recursos, estimar y planear
 - ¿Cómo saber **qué debemos hacer después?**
 - ¿Cómo saber **en qué tarea vamos** y cuál haremos después?
¿Cómo sabemos si **estamos tarde?**
 - ¿Cómo medimos el **progreso?**
 - ¿Cómo estimamos el **tiempo o costo?**
 - ¿Cómo nuevos miembros del equipo sabrán que hacer?

¿Qué es un proceso de desarrollo de Software?

Es crear y traducir necesidades de las personas en

- Especificación del software
- Diseño e implementación de software:
- Validación del software
- Evolución del software

Cada actividad puede tener sus entregables, especificar roles y puede requerir herramientas.

Tipos de procesos de desarrollo

1. Proceso en cascada
2. Proceso Iterativo
3. Proceso Incremental
4. Proceso Iterativo e Incremental

Proceso en Cascada

- Cada etapa se ejecuta una sola vez
- Los resultados de la etapa N son necesarios para empezar la etapa N + 1



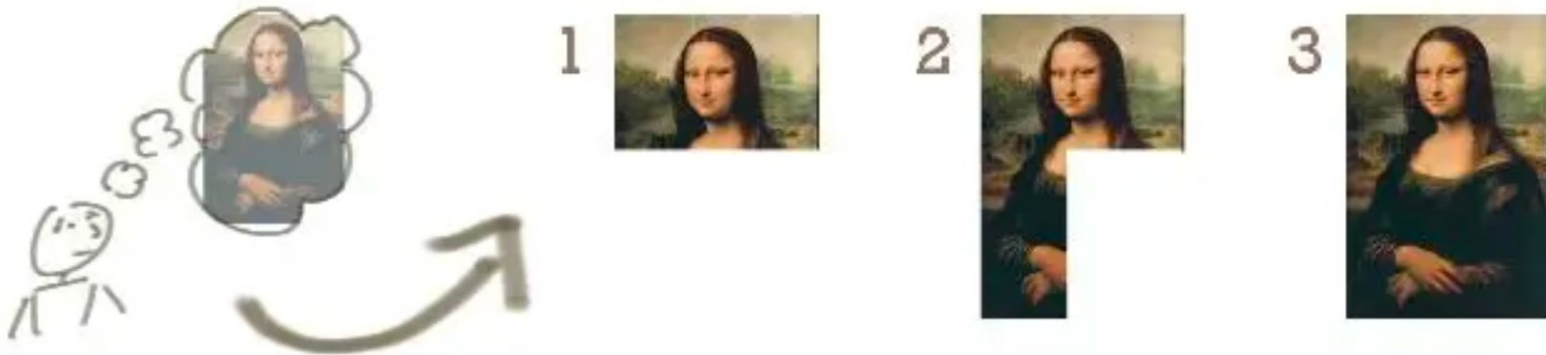
Proceso iterativo

- Con cada entrega, esperamos feedback del cliente para decidir los siguientes pasos.
- Podemos modificar nuestro desarrollo constantemente.



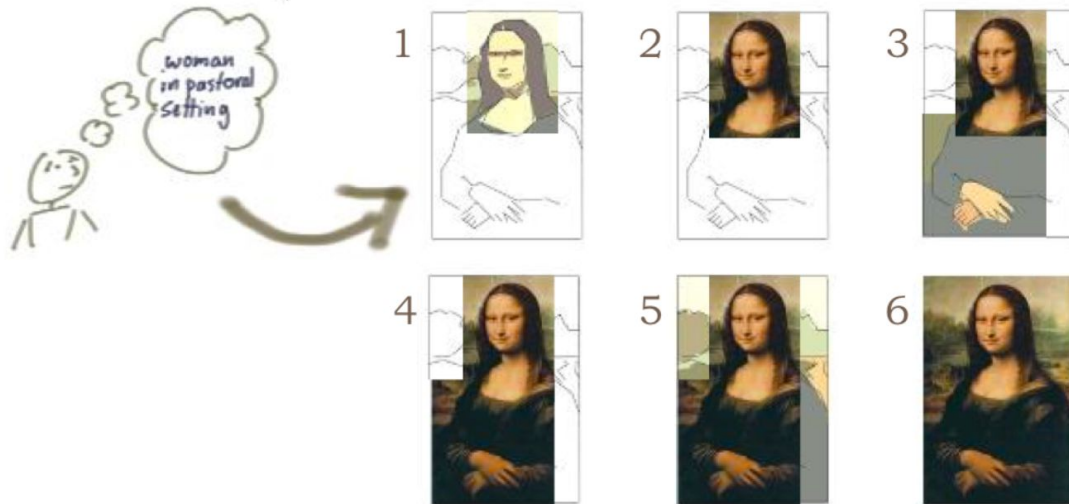
Proceso incremental

- Con cada entrega, tenemos una pieza más del software, pero no terminaremos hasta la última pieza.
- No tenemos el sistema final ni un feedback completo hasta terminar.



Proceso iterativo e incremental

- Con cada entrega incluimos funcionalidades nuevas (incremental),
- Con cada entrega incluimos mejoras en funcionalidades existentes (iterativo)



¿Cómo elegir un proceso?

1. Factores organizacionales y humanos

- El equipo está trabajando de manera remota, híbrida, o presencial.
- Cómo está estructurado el equipo: Project Manager, senior devs, equipo pequeño o grande, Product Manager.
- Cultura de la empresa.

2. Factores tecnológicos

- Ciclo de vida del producto.
- Tamaño del proyecto y requisitos técnicos.

¿Cómo elegir un proceso?

3. Factores del negocio

- Qué tan familiarizado estás con el producto.
- Cada cuánto te piden cambios o mejoras.
- Qué tan rápido cambia el mercado al que estás apuntando.

4. Factores regulatorios

- Qué aprobaciones son necesarias para operar.
- Necesidad de documentación específica.

¿Cómo elegir un proceso?

Ningún proceso es perfecto ni calzará al 100% para lo que necesitamos dado que depende de muchos factores

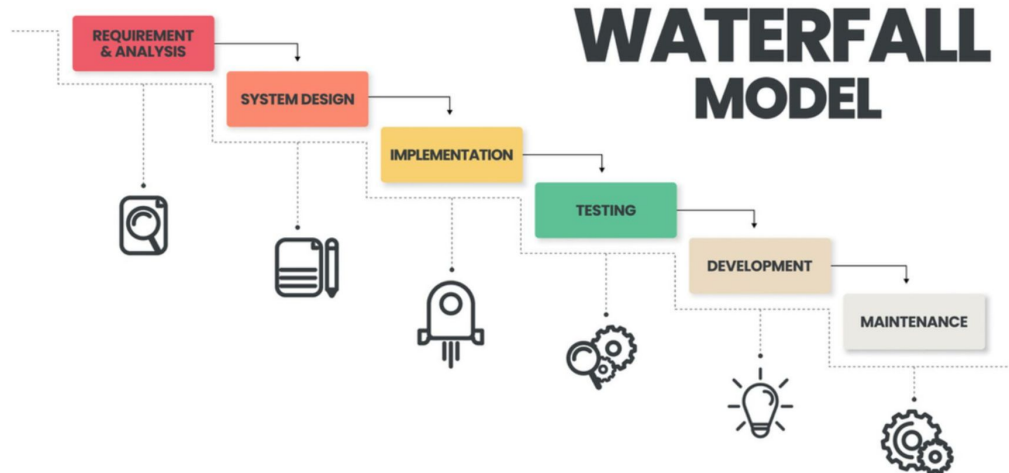
La mayoría de las empresas no utilizan un proceso de forma estricta sino que lo personalizan de acuerdo a sus necesidades.

Ejemplos de procesos

- Modelo en cascada
- Modelo en Espiral
- Proceso Unificado
- Programación Extrema
- Scrum

Modelo Cascada (Waterfall)

- Creado en 1970
- Tiene un enfoque secuencial y lineal
- Sus etapas son interdependientes



Modelo Cascada (Waterfall)

Ventajas

- Simplicidad conceptual

Desventajas

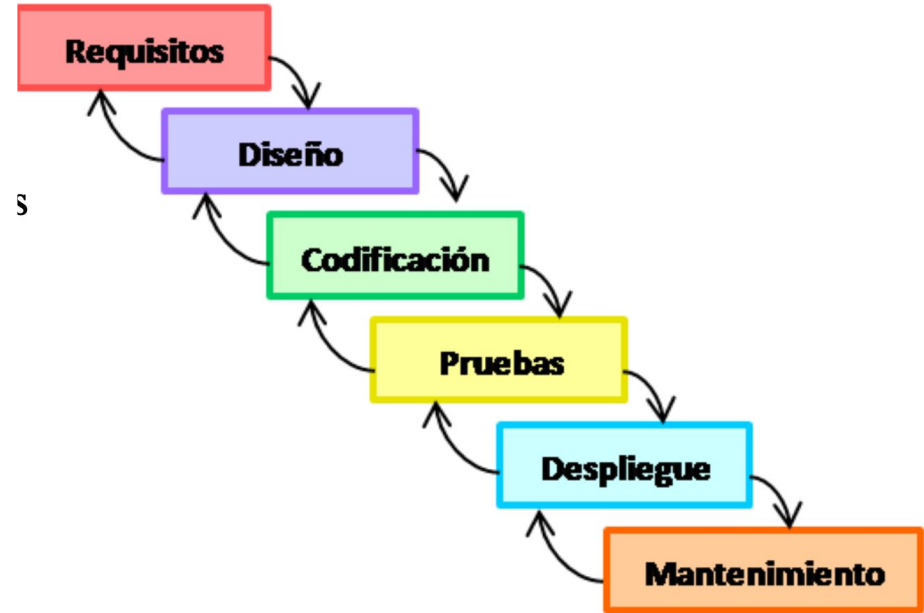
- No hay retroalimentación
- Dificultad para incorporar cambios
- Rigidez: en la práctica los resultados de cada etapa no son definitivos
- El riesgo se retrasa para las últimas etapas

Aplicable a pocos tipos de proyectos (muy cortos)

Modelo Cascada (Waterfall)

Para solucionar las desventajas, en la práctica es posible devolverse de etapas permitiendo:

- Retroalimentación
- Incorporar cambios
- Poder cambiar los resultados de cada etapa
- Mitigar el riesgo



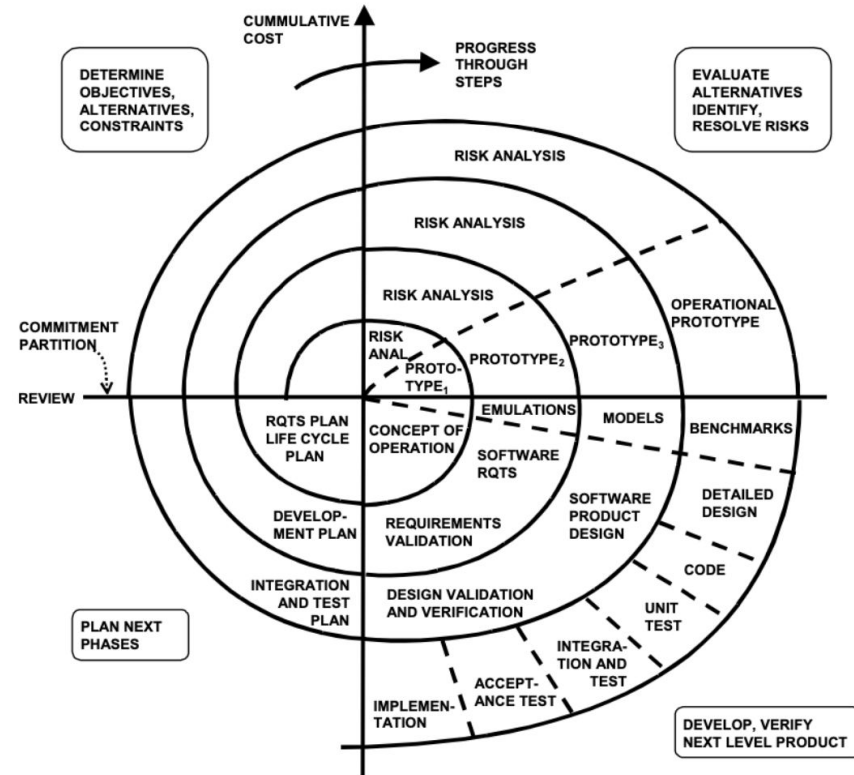
Modelo Espiral

Creado en 1986

Se puede iterar en ciclos y fases iterativas

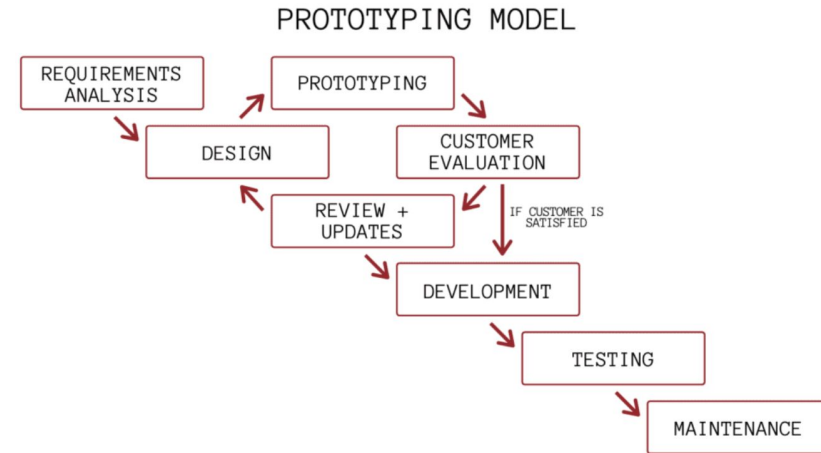
Ventajas

- Análisis de riesgo en cada etapa
- Se introduce el proceso iterativo y prototipo
- Repetición explícita de actividades

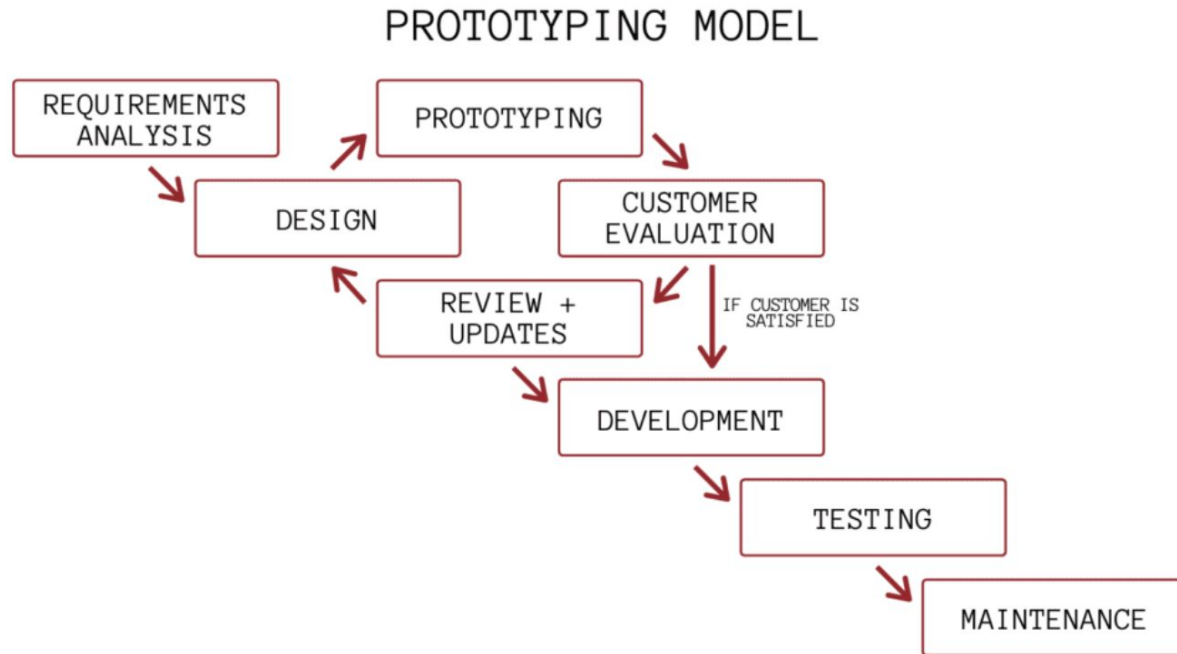


Modelo Prototipado (Prototyping)

- Ahora el usuario evalúa el producto en etapa temprana
- Se resuelve el problema “el usuario solo sabe lo que quiere cuando le muestras el resultado”
- Prototipos desechables
- Prototipos evolutivos



Modelo Prototipado (Prototyping)



Modelo Prototipado (Prototyping)

Ventajas

- Soluciona la falta de claridad en los requerimientos
- No hay grandes sorpresas al final
- Menor frustración y ansiedad durante el desarrollo

Desventajas

- Expectativas de que el desarrollo está “casi listo” siempre
- Producto final no queda igual que prototipo desechado
- Usuario que nunca está satisfecho
- Análisis incompleto de requisitos

Proceso unificado

Tiene su propio lenguaje (UML) para documentar sistemas

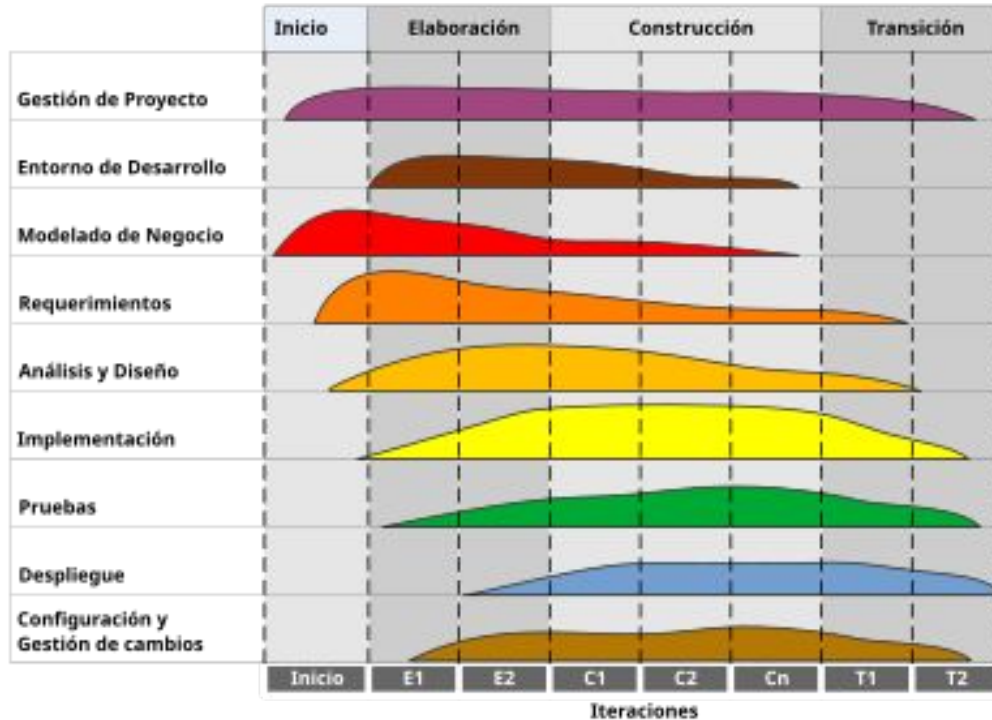
Ventajas

- Dirigido por casos de uso y centrado en la arquitectura
- Iterativo e incremental
- Enfocado en los riesgos críticos del proyecto

Desventajas

- Mucho énfasis en documentación
- Muy costoso de implementar
- Necesidad de gestión y supervisión alta

Proceso unificado



Modelo ágil

Ventajas

- Se valora más a los individuos e interacciones que procesos y herramientas
- Software funcional más que documentación exhaustiva
- Colaboración con el cliente
- Flexibilidad en vez de ajustarse a seguir un plan



Modelo ágil

Manifiesto ágil

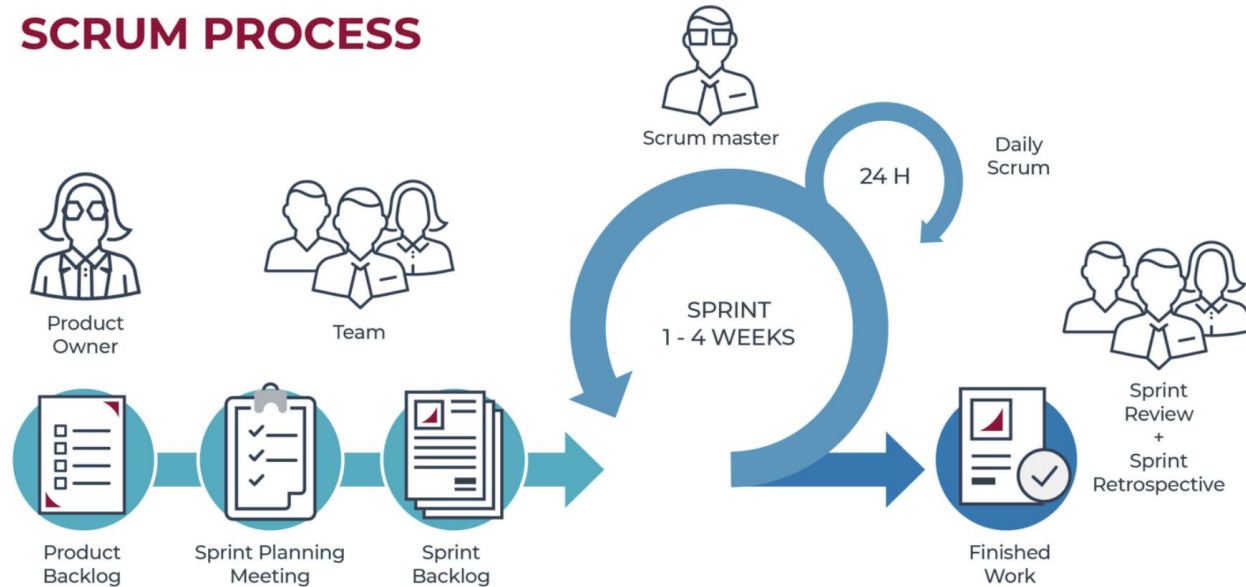
“Nuestra mayor prioridad es satisfacer al cliente a través de una entrega temprana y continua de software valioso para él”

“Abrazamos los cambios en los requerimientos, incluso si llegan tarde en el proceso”.

<https://agilemanifesto.org/iso/es/manifesto.html>

Modelo Scrum

SCRUM PROCESS



Lo veremos en profundidad la próxima clase

IIC 2143 – Ingeniería de Software

Procesos del Desarrollo de Software

M. Trinidad Vargas
mtvargas1@uc.cl