



Ingeniería de Software

3 - Procesos de Desarrollo de Software

IIC2143-3

Josefa España

jpespana@uc.cl



¿Por qué un proceso de desarrollo de software?

- Escribir código es relativamente simple, pero desarrollar software de calidad es difícil.
- Trabajamos con personas que tienen diferentes visiones, ¿Cómo generamos un modelo que nos permita **efectivamente** obtener los requerimientos para nuestro software?
- Vamos a necesitar **código complejo**, y para que sea **efectivo y eficiente** necesitamos planear cómo lo vamos a desarrollar.



Procesos definidos, entendibles y repetibles. ¿Por qué?

- Encontrar y mantener buenas prácticas.
- Administración:
 - ¿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 sabrán qué tienen que hacer?



Definiciones

Es crear y traducir necesidades de las personas, en:

- Requerimientos.
- Diseño de código.
- Pruebas
- Instalación.
- ...



Proceso de desarrollo: Conjunto de actividades que toman lugar en secuencia con el fin de crear un software.





Tipos de procesos de desarrollo

1. Proceso en cascada.
2. Proceso iterativo.
3. Proceso incremental.
4. Proceso iterativo e incremental.



1. Proceso en Cascada



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



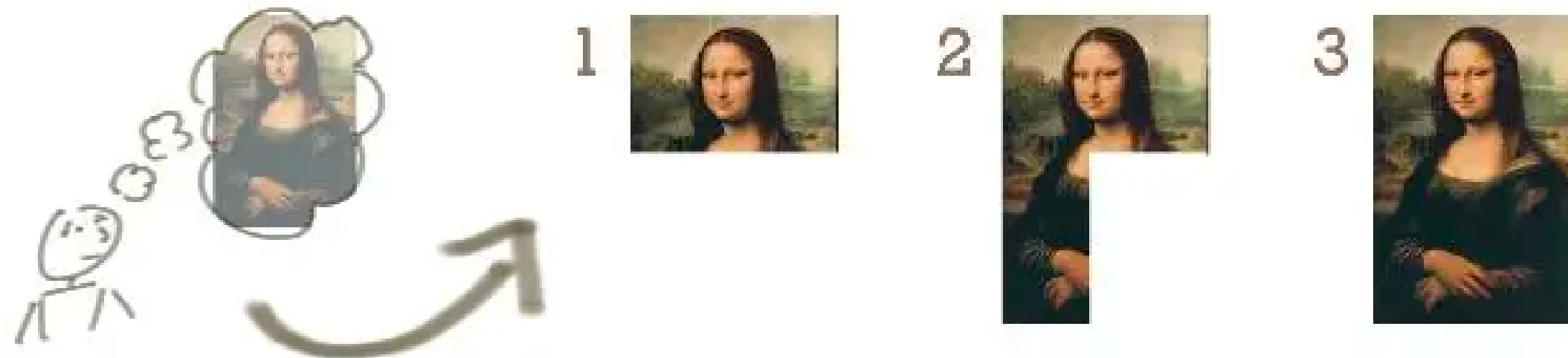
2. Proceso iterativo



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



3. Proceso incremental



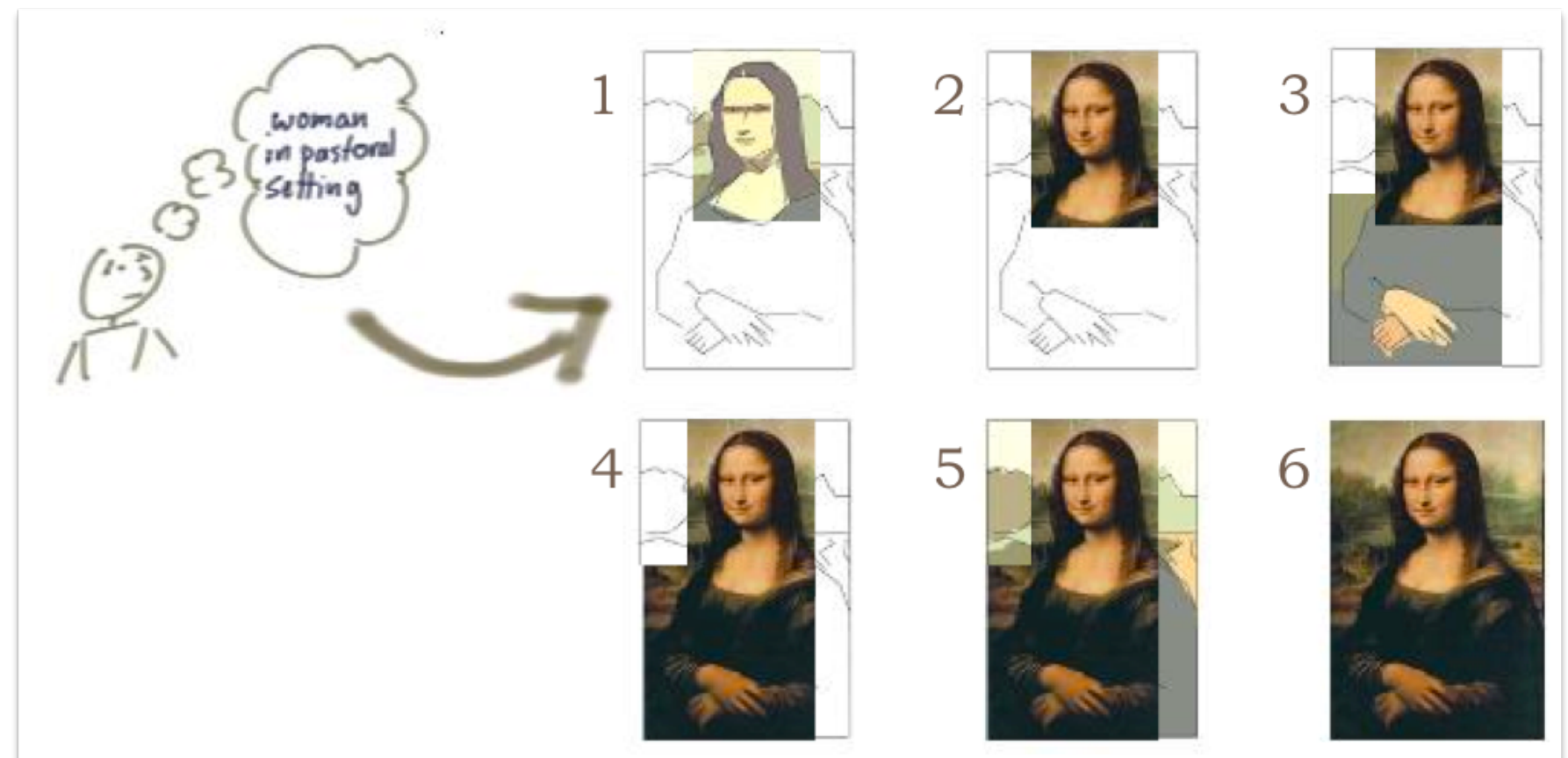
- Con cada entrega, tenemos una pieza más del software, pero no terminaremos hasta la última pieza.
- No tenemos el sistema final ni un feedback completo hasta terminar.



4. Proceso iterativo e incremental

Con cada entrega:

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





¿Cómo elegir el proceso?

Dependerá de:

1. Factores organizacionales / 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 el proceso?

Dependerá de:

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 el proceso?

Conclusiones

Ningún proceso es perfecto, ni calzará al 100% para lo que necesitamos. Dado que depende de muchos factores, en la práctica:

- La mayoría de las empresas siguen un proceso pero lo personalizan según el proyecto y equipo.



Procesos

Ejemplos prácticos

- Modelo en Cascada
- Modelo en Espiral
- Proceso Unificado.
- Programación Extrema.
- Scrum.



Waterfall Model

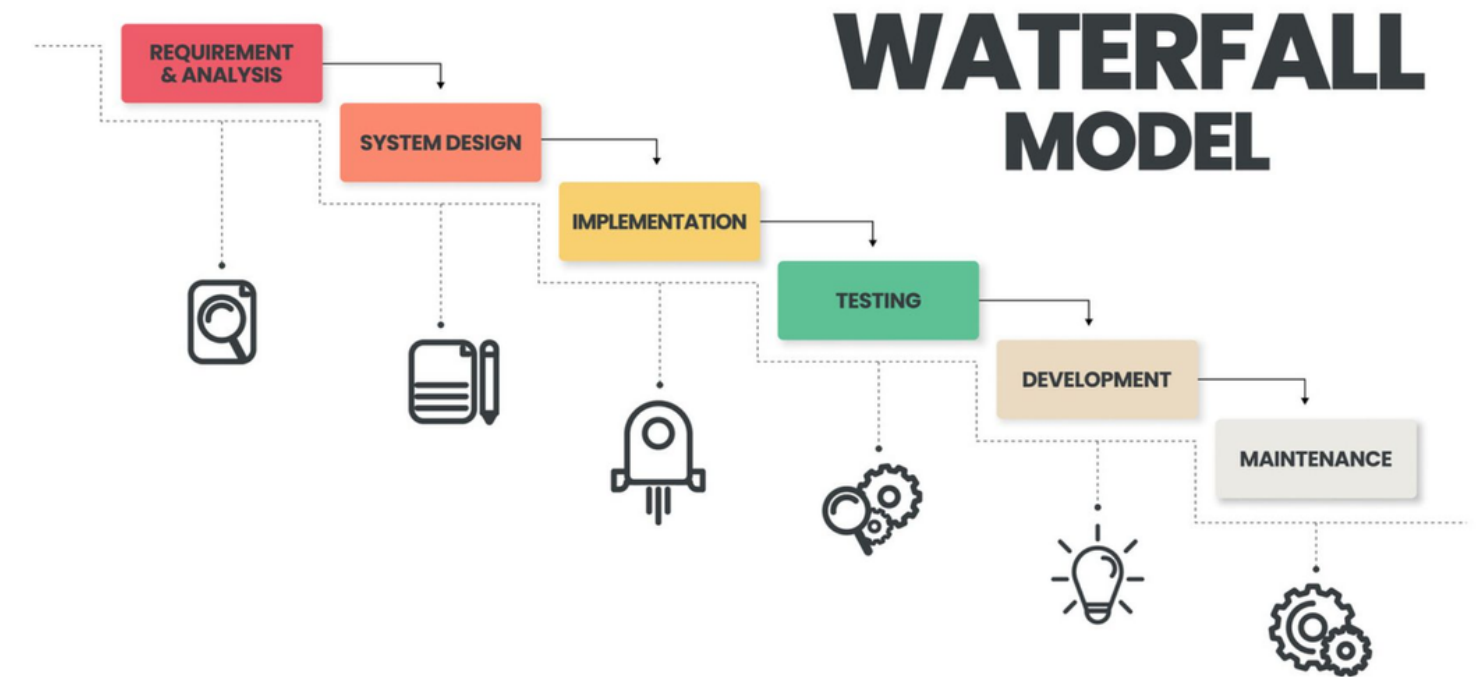
1970

Ventajas

- Fácil de entender.

Desventajas

- No hay retroalimentación.
- Difícil de incorporar cambios.
- Rígido, en la práctica los resultados de cada etapa no son definitivos.
- El riesgo se retrasa para las últimas etapas.



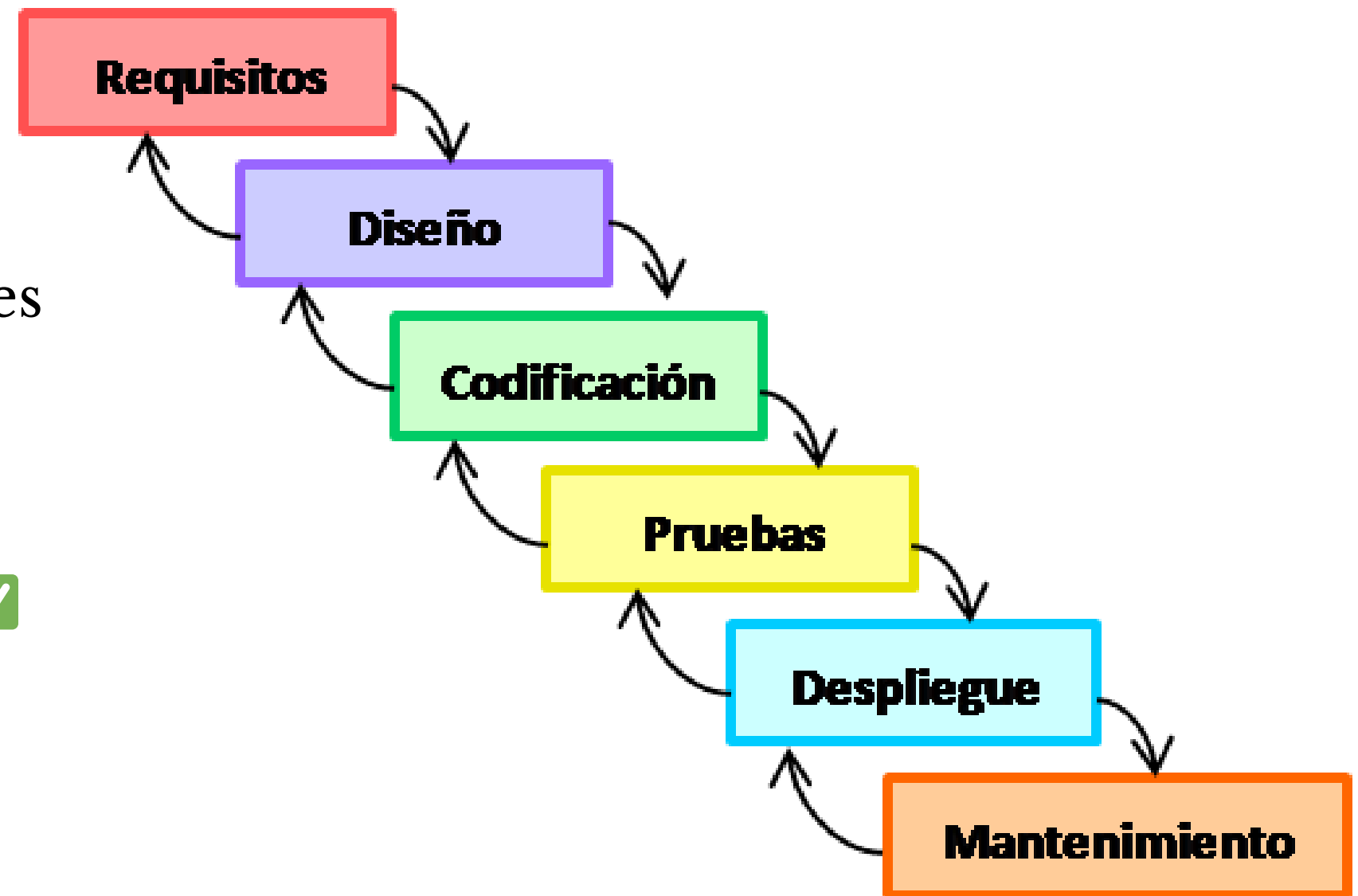


Waterfall Model

En la práctica

Para solucionar estas desventajas, en la práctica es posible devolver de etapa permitiendo:

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





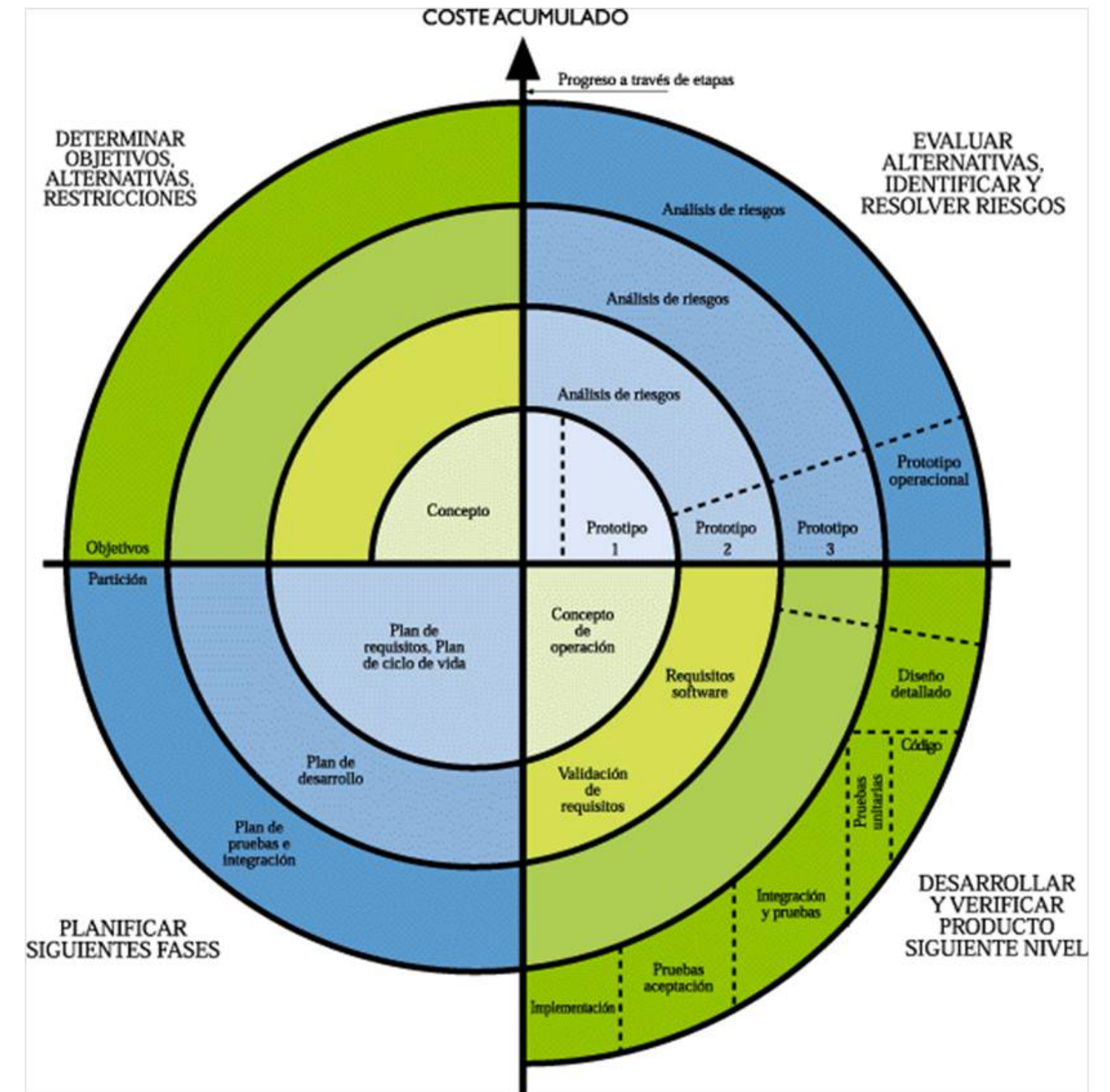
Spiral Model

1986

Mejora del modelo en cascada.
Ahora se puede iterar en ciclos y fases.

Ventajas:

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

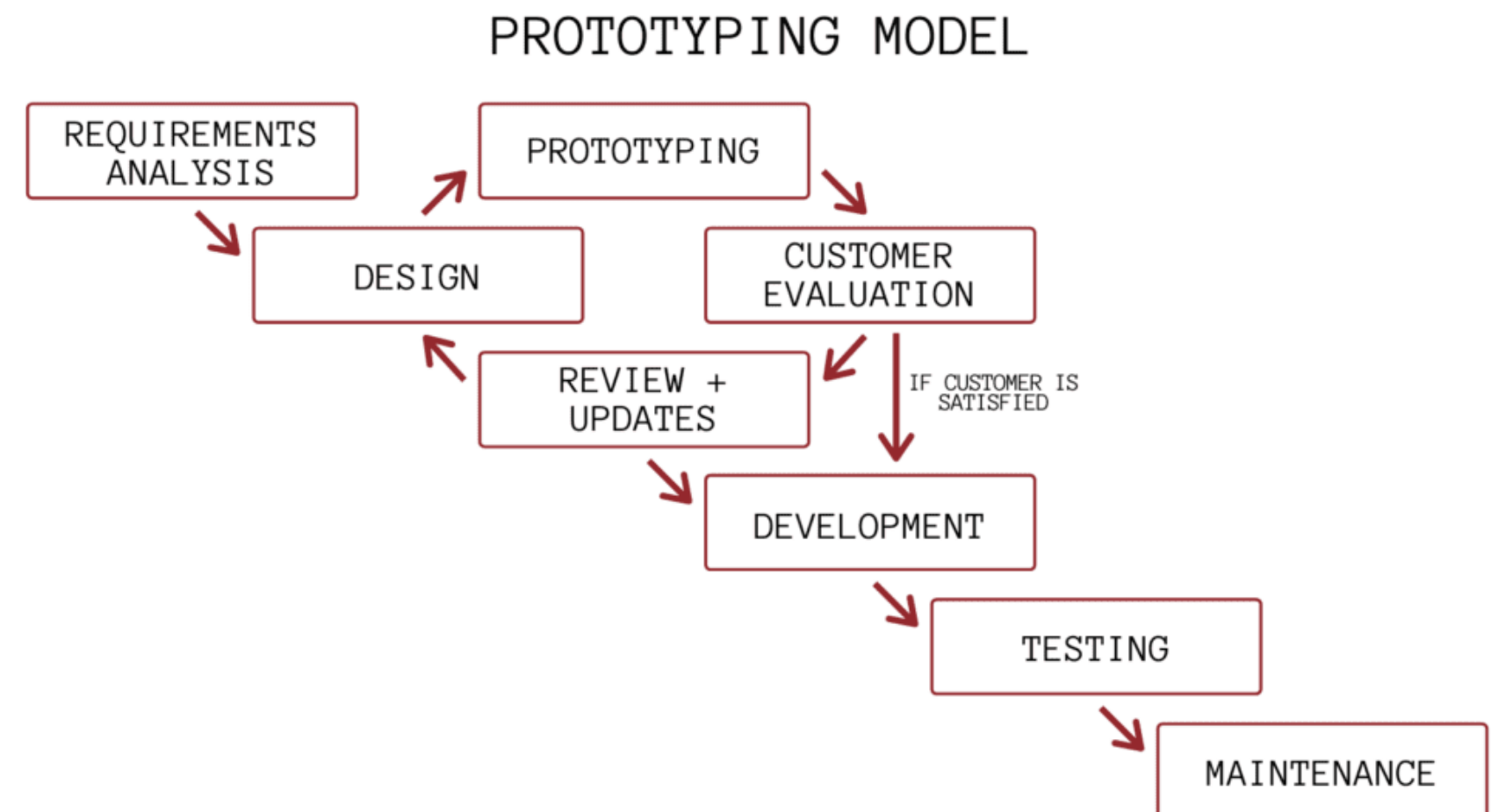




Prototyping Model

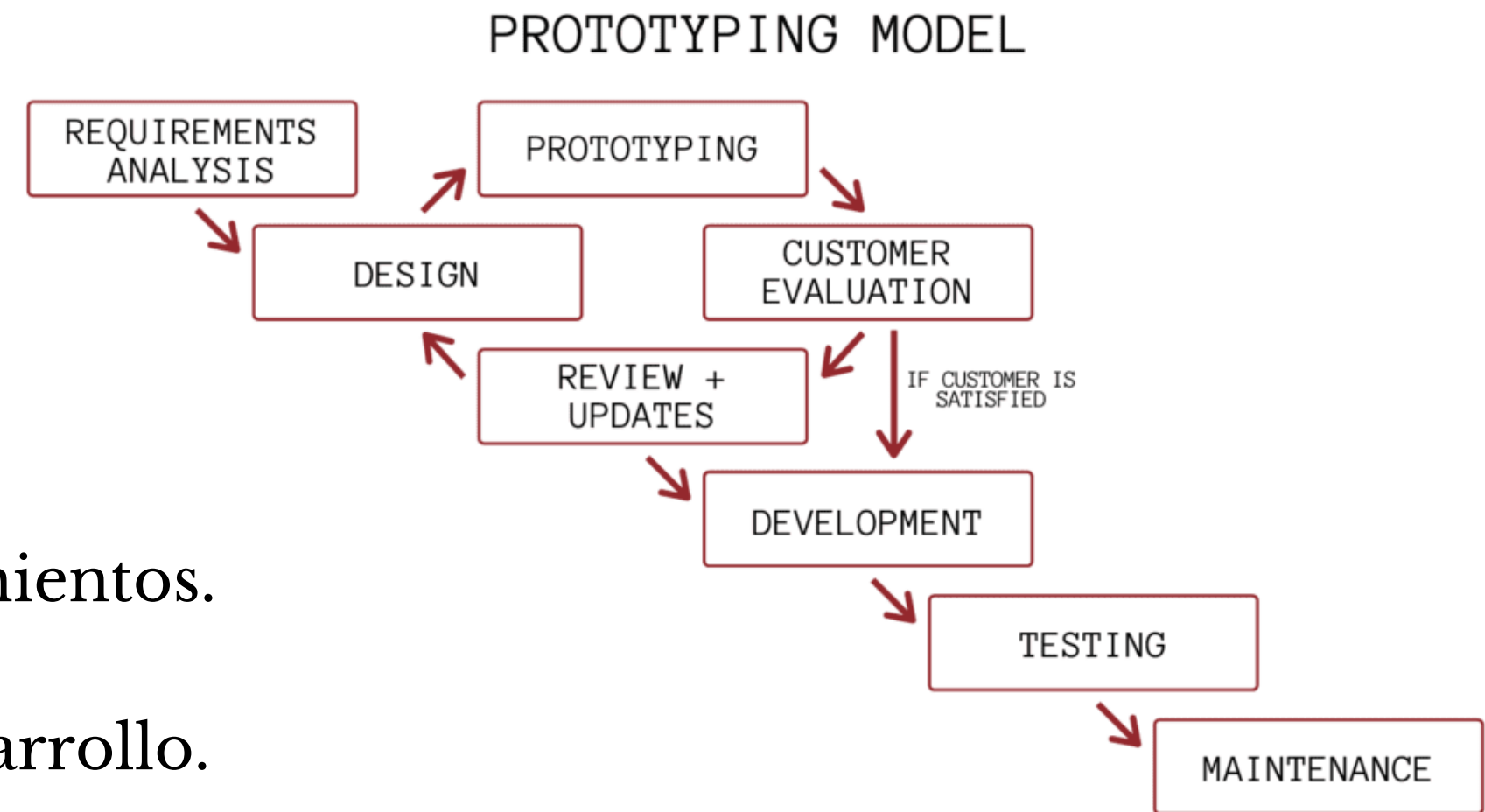
Ahora el usuario evalúa el producto de manera temprana.

- Se resuelve el problema “el usuario solo sabe lo que quiere cuando le muestras el resultado”.
- Prototipos desechables.
- Prototipos evolutivos.





Prototyping Model



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

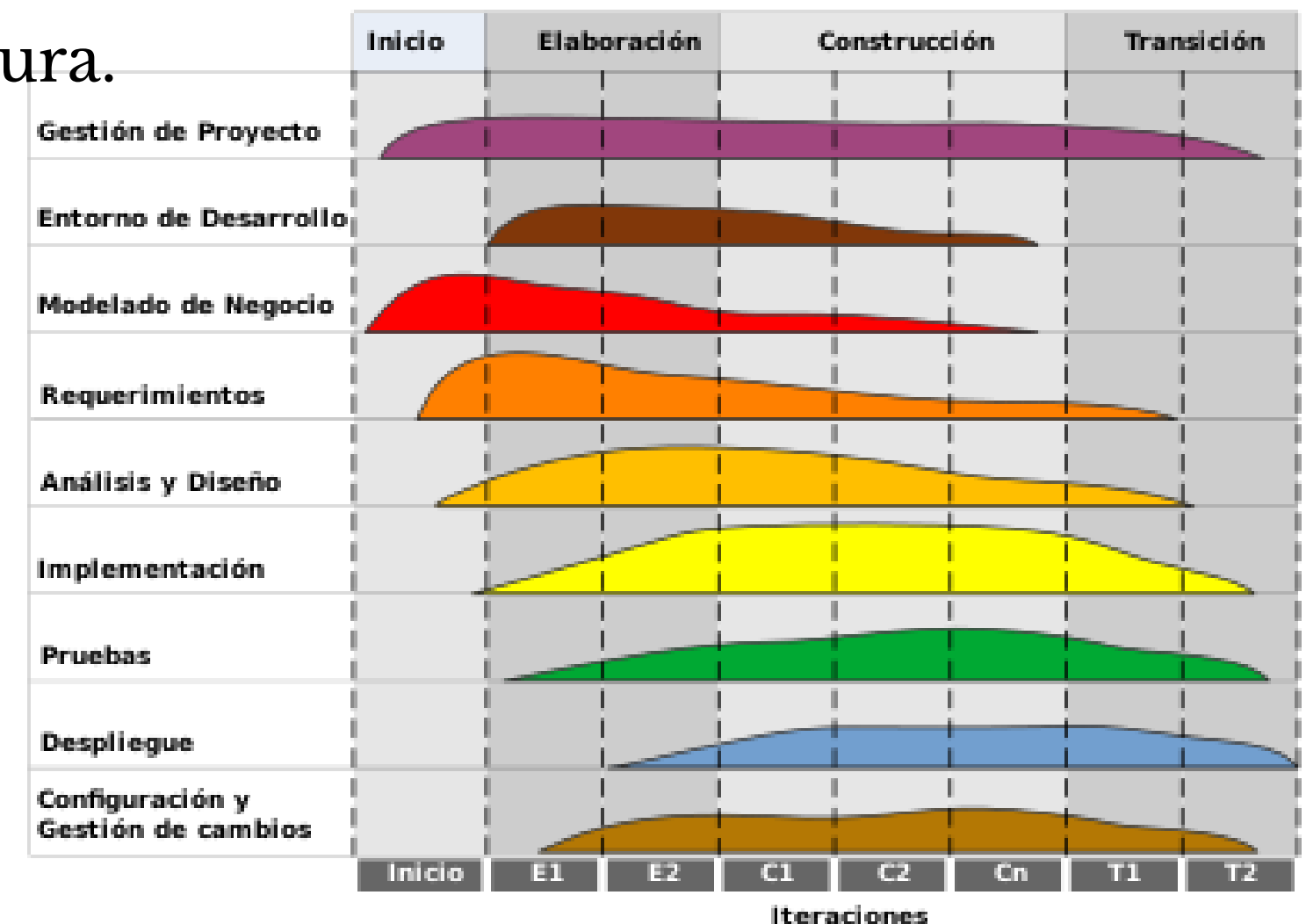
Tiene su propio lenguaje (UML) para documentar sistemas.

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.





Modelo Agile

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.



Del manifiesto ágil:

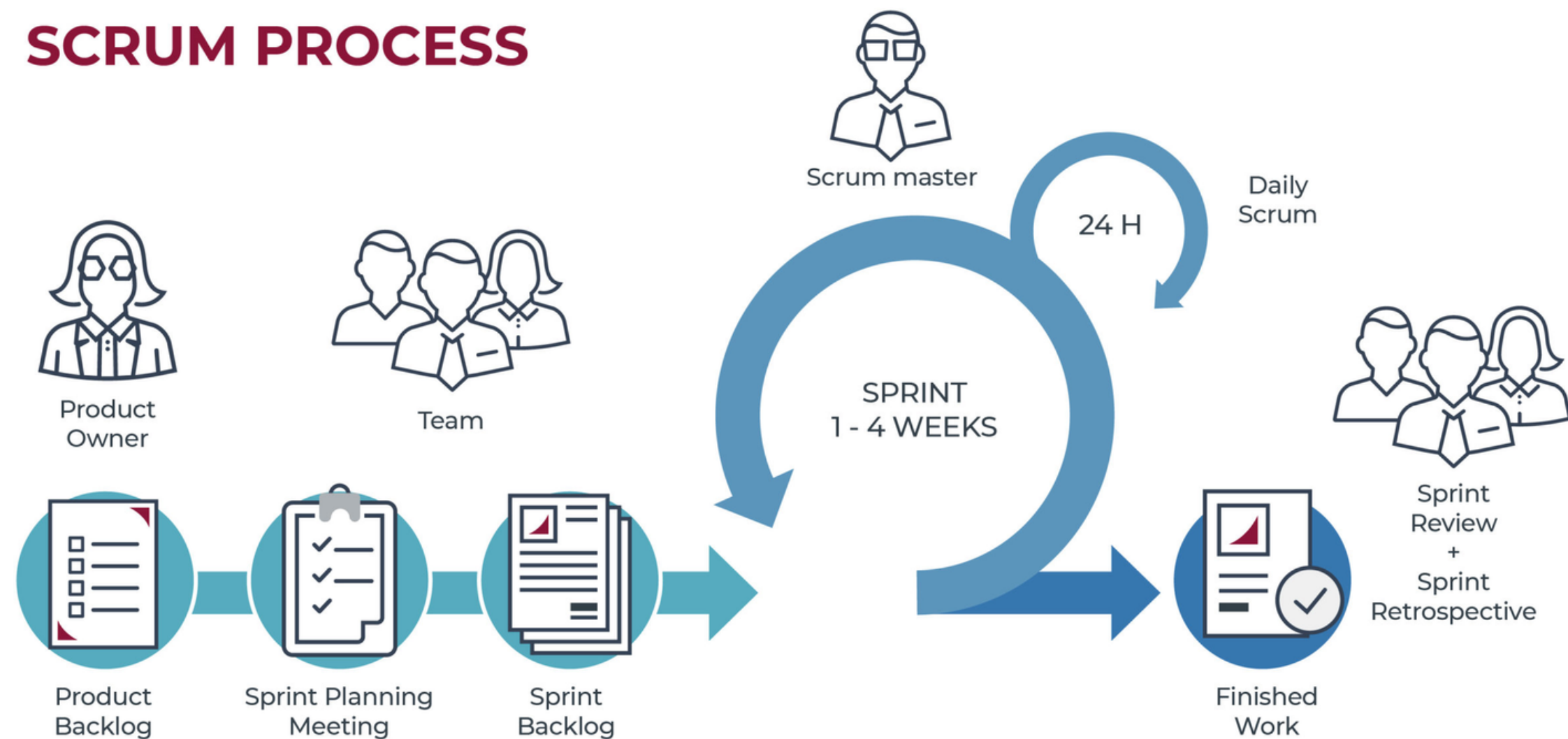
“Nuestra mayor prioridad es satisfacer al cliente a través de una entrega temprana y continua de software valioso para él”

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



Modelo SCRUM

SCRUM PROCESS



¡Próxima clase lo veremos!



¿Consultas?



Ingeniería de Software

3 - Procesos de Desarrollo de Software

IIC2143-3

Josefa España

jpespana@uc.cl