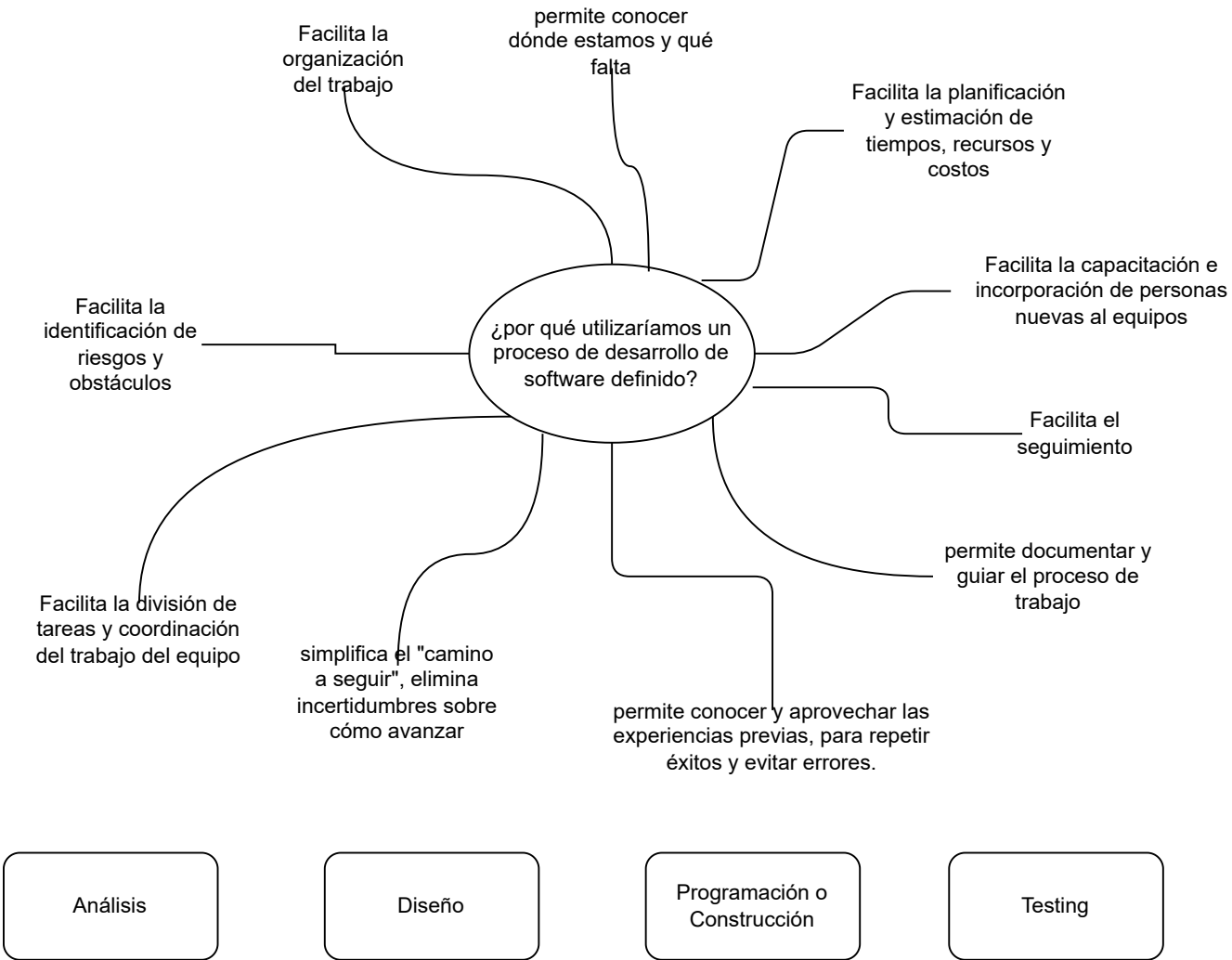
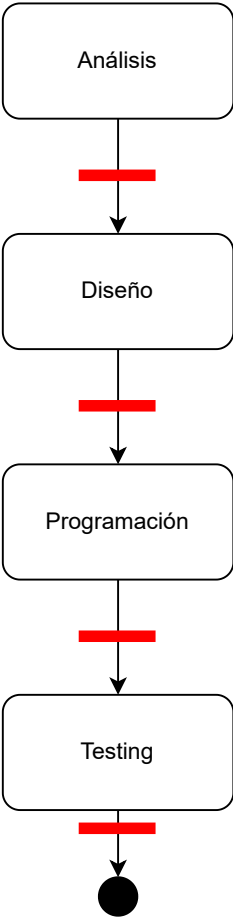


Metodologías

Describir el proceso que se va a seguir para construir el software, teniendo en cuenta los requerimientos (funcionales y no funcionales) identificados para el mismo.



Cascada (Waterfall)



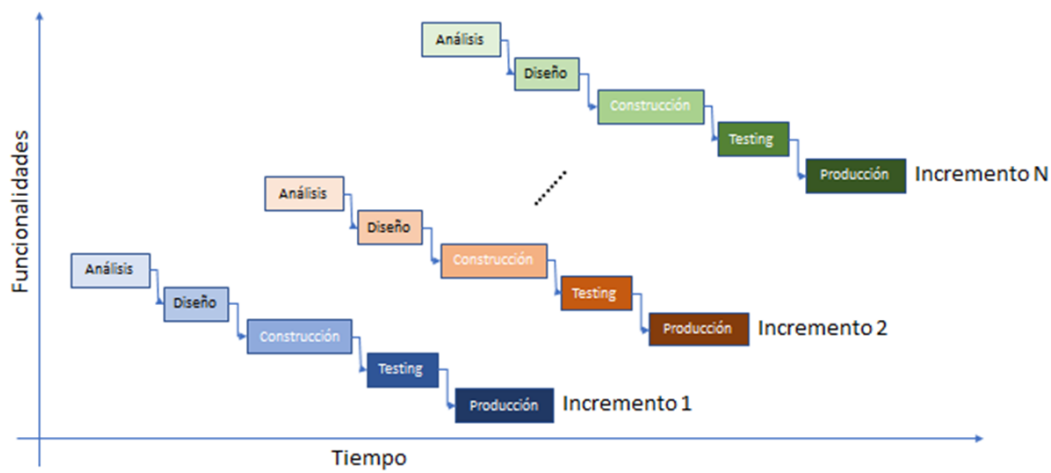
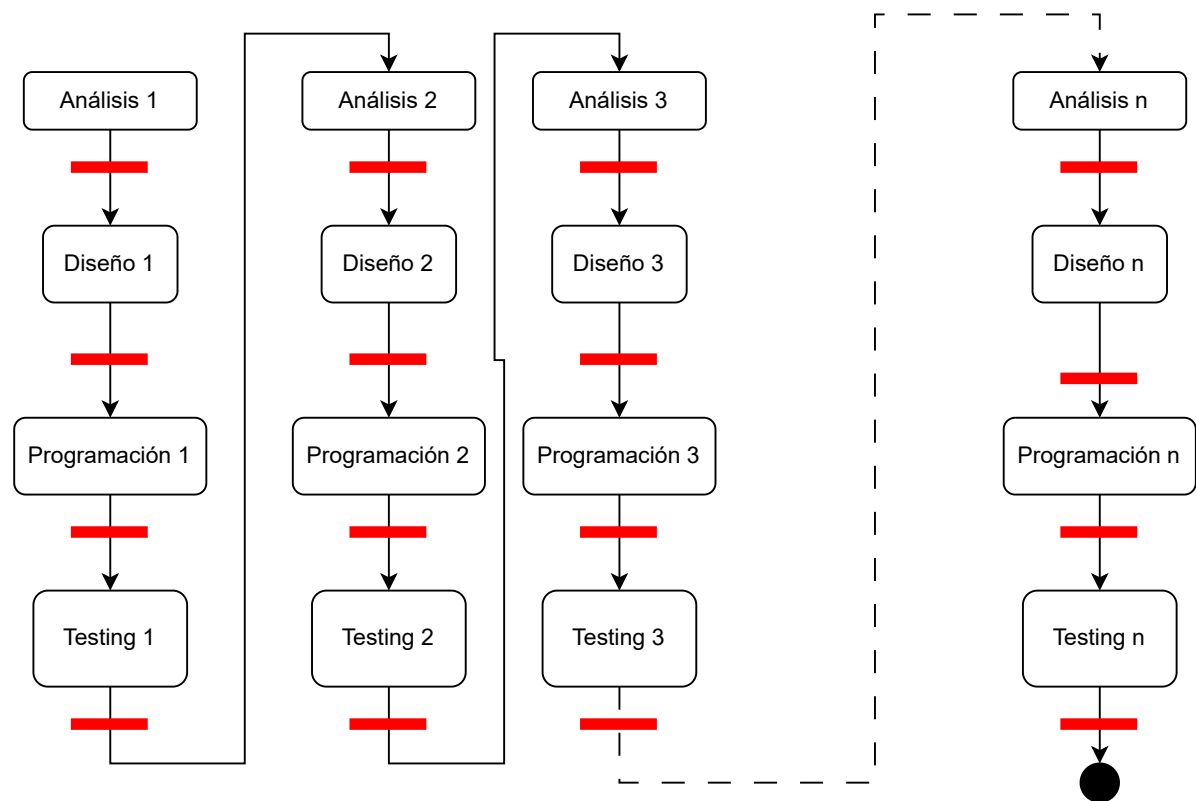
Gobernanza

Fase	Entregables	Aprobadores	Plazos y Condiciones
Análisis	Especificación de Requerimientos	Gte/Director 'Area	
		Gte IT	
	MER	Gte IT	
Diseño	Diseño de Base de Datos	DBA	
Programación			
Testing			

Características

- Proceso ordenado, documentado, predictivo
- Se trabaja "sobre seguro"
- Facilita la planificación y la coordinación de recursos.
- No es flexible para incorporar cambios, hay que volver para atrás.
- Posibilita "mejores diseños técnicos"
- Dependencia de las aprobaciones para avanzar
- El cliente sólo ve resultados al final, todo junto, en una única entrega.
- No requiere participación continua del cliente durante el desarrollo.
- Requiere que se conozcan de entrada todos los requerimientos del sistema

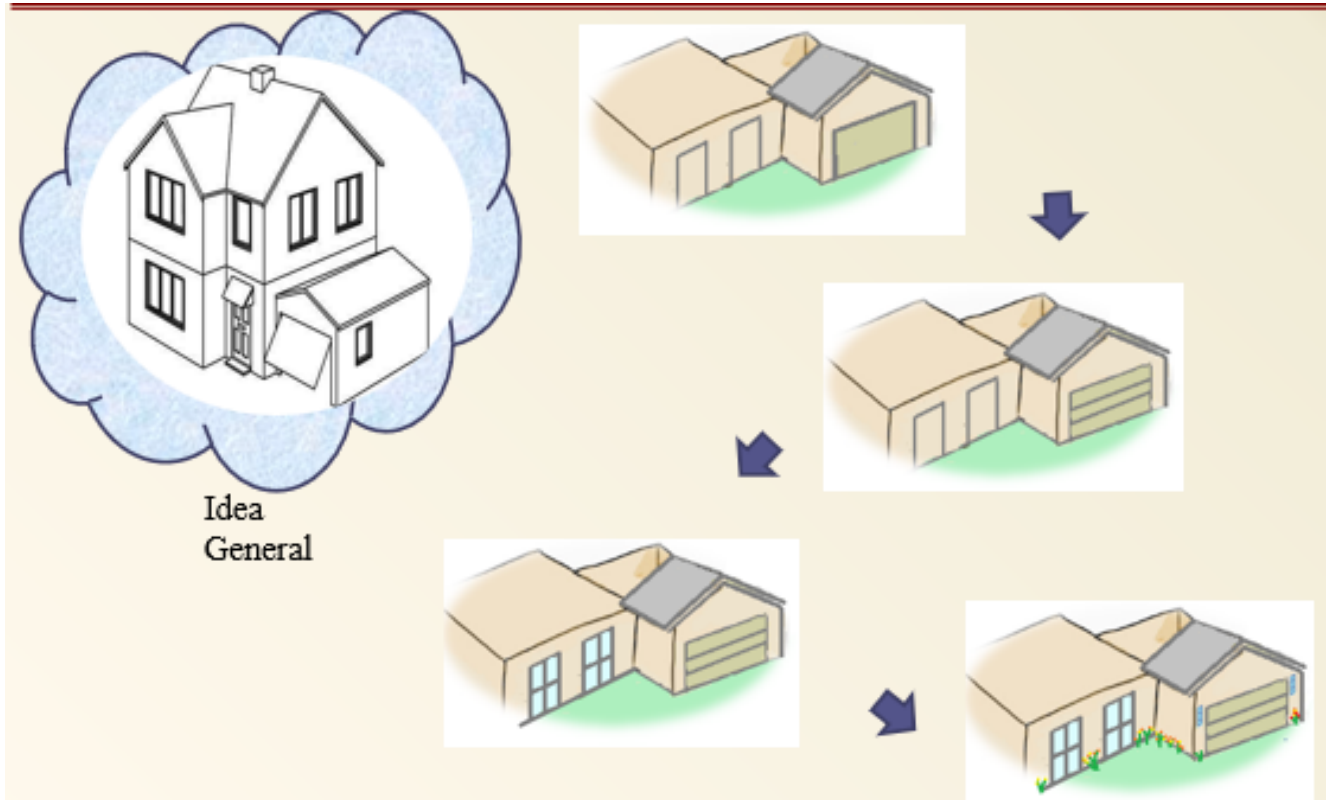
Incremental



Evolutivo.

Parte de una implementación inicial de la solución completa, que se refina en sucesivas versiones.

Se trabaja con el cliente explorando los requerimientos y entregando un sistema fina, que se va depurando según se vaya conociendo/comprendiendo.



Ensamblaje de Componentes

Reutilización

Bibliotecas de Componentes

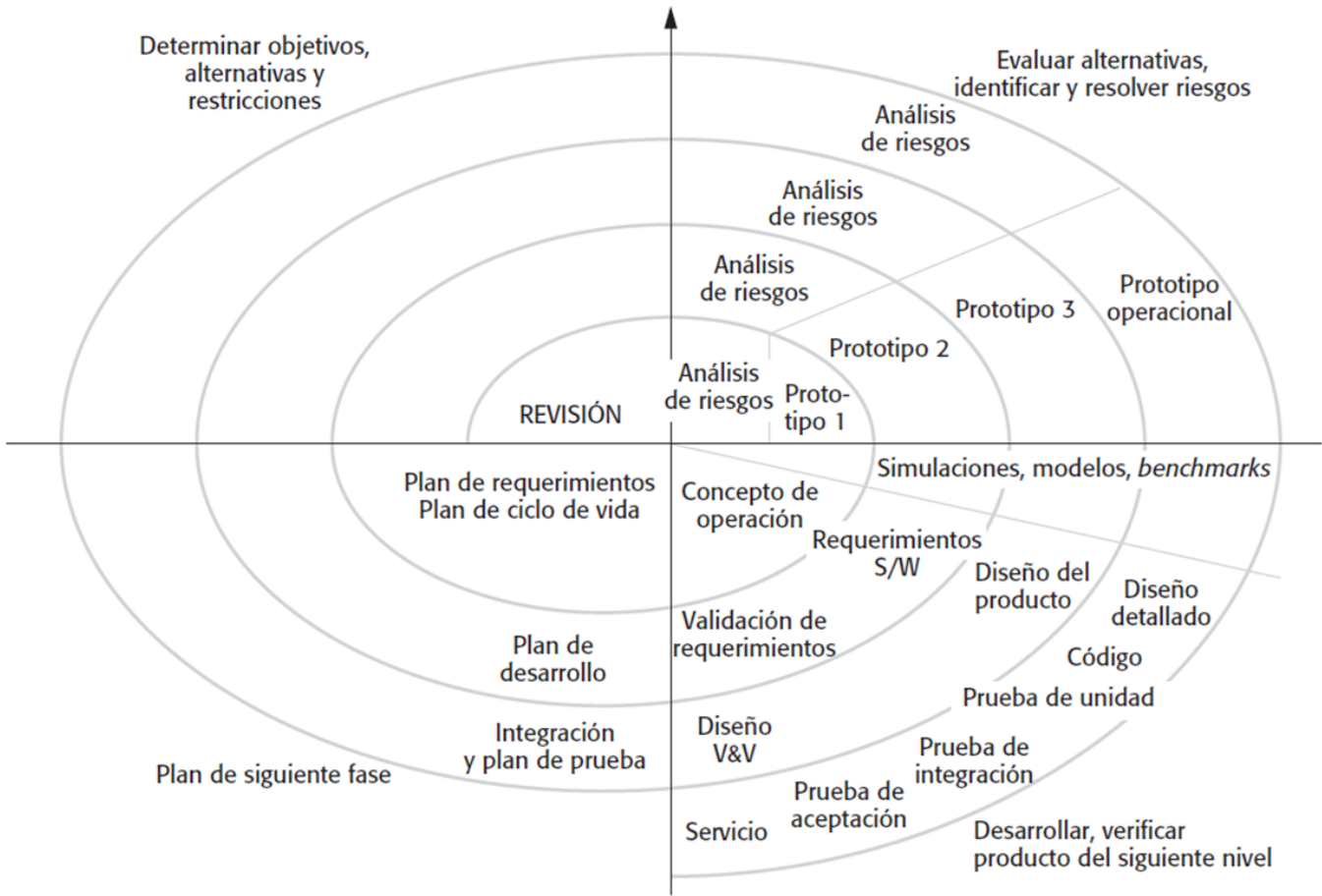
Identificación de componentes

"Encastrado" / Ajuste / Adaptación de
Componentes

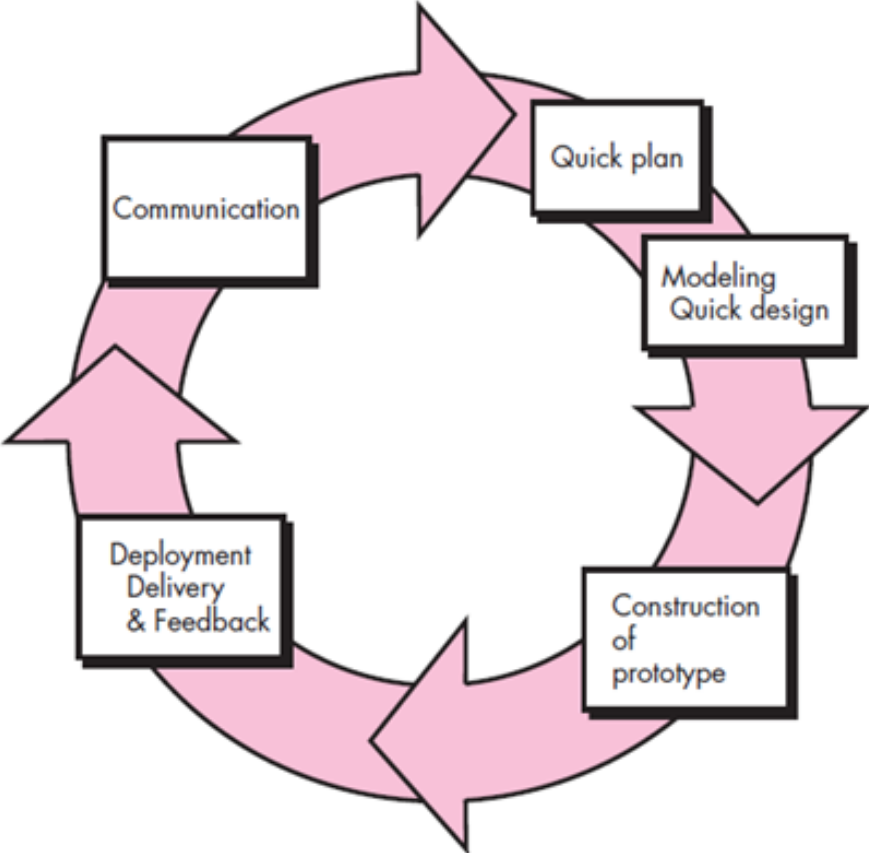


Modelo en Espiral

Combinación de Predictivo + Evolutivo, con FOCO en RIESGOS



Prototipación



Paradigma de Desarrollo Ágil

Manifiesto Ágil

Código real funcionando

Equipos colaborativos

Comunicaciones efectivas

Prioriza a las personas y las interacciones entre ellas por sobre el seguimiento de procesos y el uso de herramientas.

Prioriza software funcionando por sobre documentación exhaustiva

Colaboración con el cliente por sobre negociación de contratos.

Adaptación a los cambios por sobre seguimiento de plan.

Principios

Planificación dinámica

Feedback continuo

Adaptar e incorporar los cambios

Enfoque en trabajo en Equipo

Metodologías

Ágiles: XP

(eXtreme Programming)

Enfoque en primer entregable rápido, con feedback para lograr confianza del cliente.

Mantiene el diseño lo más simple posible.

Aplica "refactoring" para depurar y mejorar la calidad del diseño continuamente.

Test Driven Development: se encara el desarrollo con un testing continuo, para detectar los defectos tan pronto se pueda y arreglarlos al menor costo.

Scrum

Marco de trabajo para la resolución de problemas complejos, entregando productos del mayor valor posible.

Transparencia

Inspección

Adaptación



3 roles:

Product Owner: representante del cliente/negocio en el equipo scrum. Dueño de la definición y priorización de los requerimientos. Responsable de maximizar el valor del producto.

Equipo de Desarrollo: construye el producto. Responsable de las estimaciones.

Equipos de entre 3 a 9 personas, auto-organizados, Cubren todos las habilidades/conocimientos necesarios para la construcción del producto. 100% dedicados al equipo scrum.

*Estimaciones: Puntos de User Stories
Fibonacci: 1, 1, 2, 3, 5, 8, 13 ...*

Scrum Master: responsable de "velar" (asegurar) la correcta implementación de scrum.

"Leader Servant" - ayuda a resolver obstáculos

3 Artefactos

Product Backlog: Conjunto de todos los requerimientos (User Stories) que definen las características del producto.

User Stories:
*Como <tipo de usuario>
necesito <X> para poder
<Y>*

Sprint Backlog: subconjunto de user stories del Product Backlog que se van a trabajar en una sprint.

Incremento: Producto que se construye en la sprint

Eventos

Cajas de Tiempo (time boxes)

Sprint (Iteración) - Bloques de tiempo en los cuales se construye el producto. 2 a 4 semanas

Planificación (2hs por semana de duración del sprint)

Define el Objetivo de la Sprint y el Sprint Backlog

Velocidad: cantidad de "Puntos de User Stories" que el equipo puede construir por iteración

Daily Stand Up Meeting: 15 minutos

¿qué hice ayer?
¿qué voy a hacer hoy?
¿qué problemas tengo?

Demo o Review: 1h por cada semana del sprint se presenta el resultado de la Sprint.

Retrospectiva: evaluación interna de cómo funcionó la sprint. 45 minutos por cada semana del sprint.

Definition of Done

Condiciones para entender que una "user Story" está completa

No "hard coding"
casos de pruebas ejecutados correctamente
código documentado

Burn-Down Chart

