



Metodologías de Desarrollo de Software

Auditoría, Calidad y Gestión de Sistemas
Curso 2023-24



Las metodologías de desarrollo de software definen marcos de trabajo para que los equipos de desarrollo puedan llevar a cabo sus tareas





procesos

notaciones

herramientas

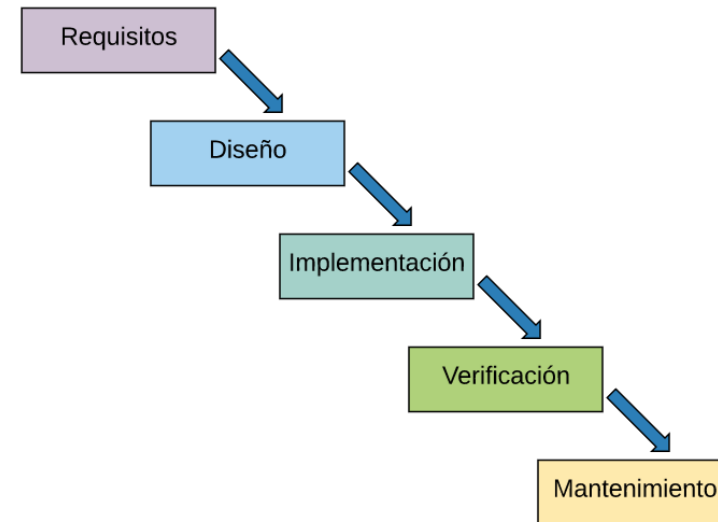
lenguajes



- Cascada (waterfall model)
- Prototipado
- Incremental
- Espiral



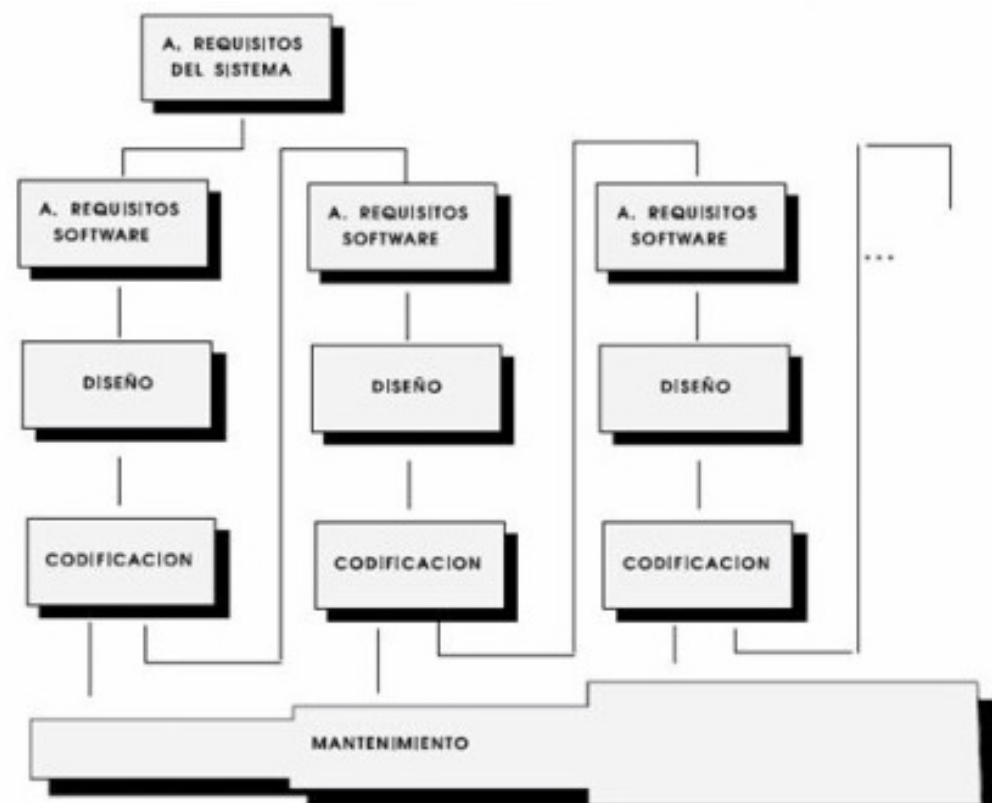
- Cascada
- Prototipado
- Incremental
- Espiral



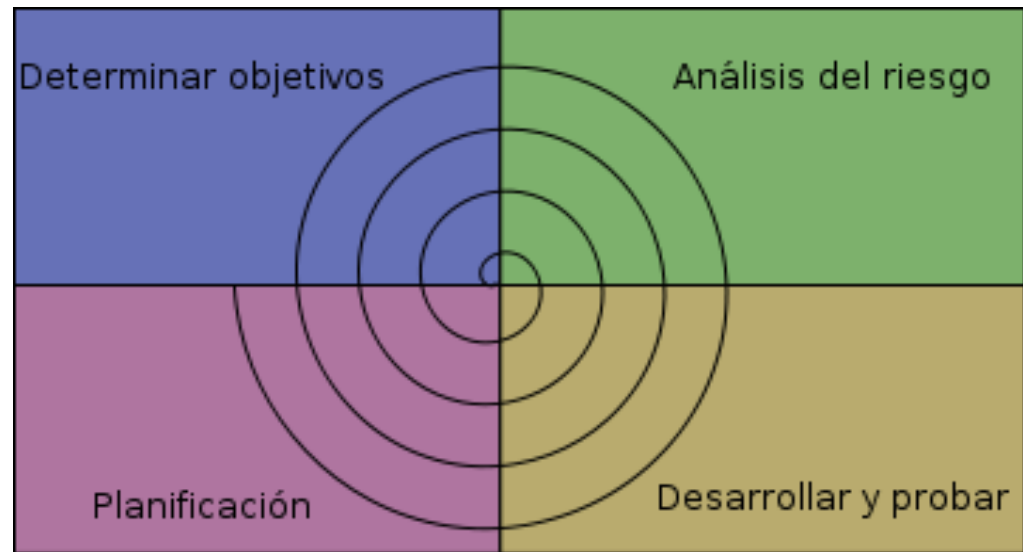
- Cascada
- **Prototipado**
- Incremental
- Espiral



- Cascada
- Prototipo
- **Incremental**
- Espiral



- Cascada
- Prototipado
- Incremental
- **Espiral**

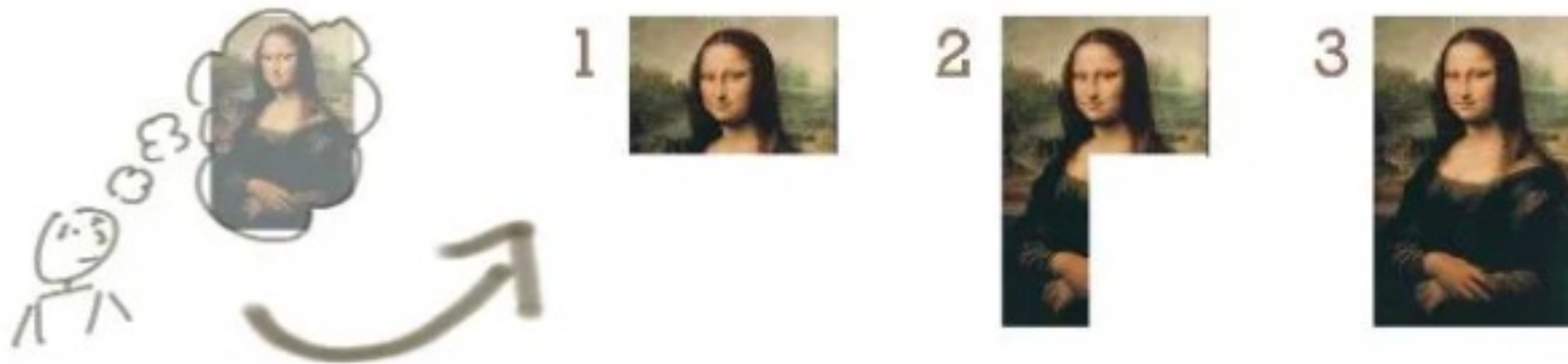


¿Incremental e iterativo?



- Un **desarrollo incremental** es aquel en el que, con cada entrega, tenemos acabada una pieza más del sistema.
- Por ejemplo, primero la base de datos, luego la lógica de negocio y finalmente «las pantallas».
 - Problema!! no hemos pensado en el usuario a la hora de planificar nuestras entregas. Con cada incremento, el usuario no puede hacer nada nuevo. No hay nuevas funcionalidades que utilizar por lo que, desde su punto de vista, no habría mucha diferencia entre un desarrollo incremental y un «waterfall» (con una única entrega al final del proyecto).





Incremental development» by Jeff Patton (oil and bits)



- Un **desarrollo iterativo** es aquel en el que, con cada entrega, esperamos el feedback del usuario para decidir los siguientes pasos a seguir.
- Por ejemplo, podríamos entregar un prototipo en papel para validar las primeras hipótesis sobre cómo quiere el usuario que le mostremos la información, luego, a partir del feedback recibido, podemos desarrollar un prototipo con el que empezar a validar hipótesis técnicas (integración con fuentes de datos, etc.) y así hasta tener la aplicación completada.



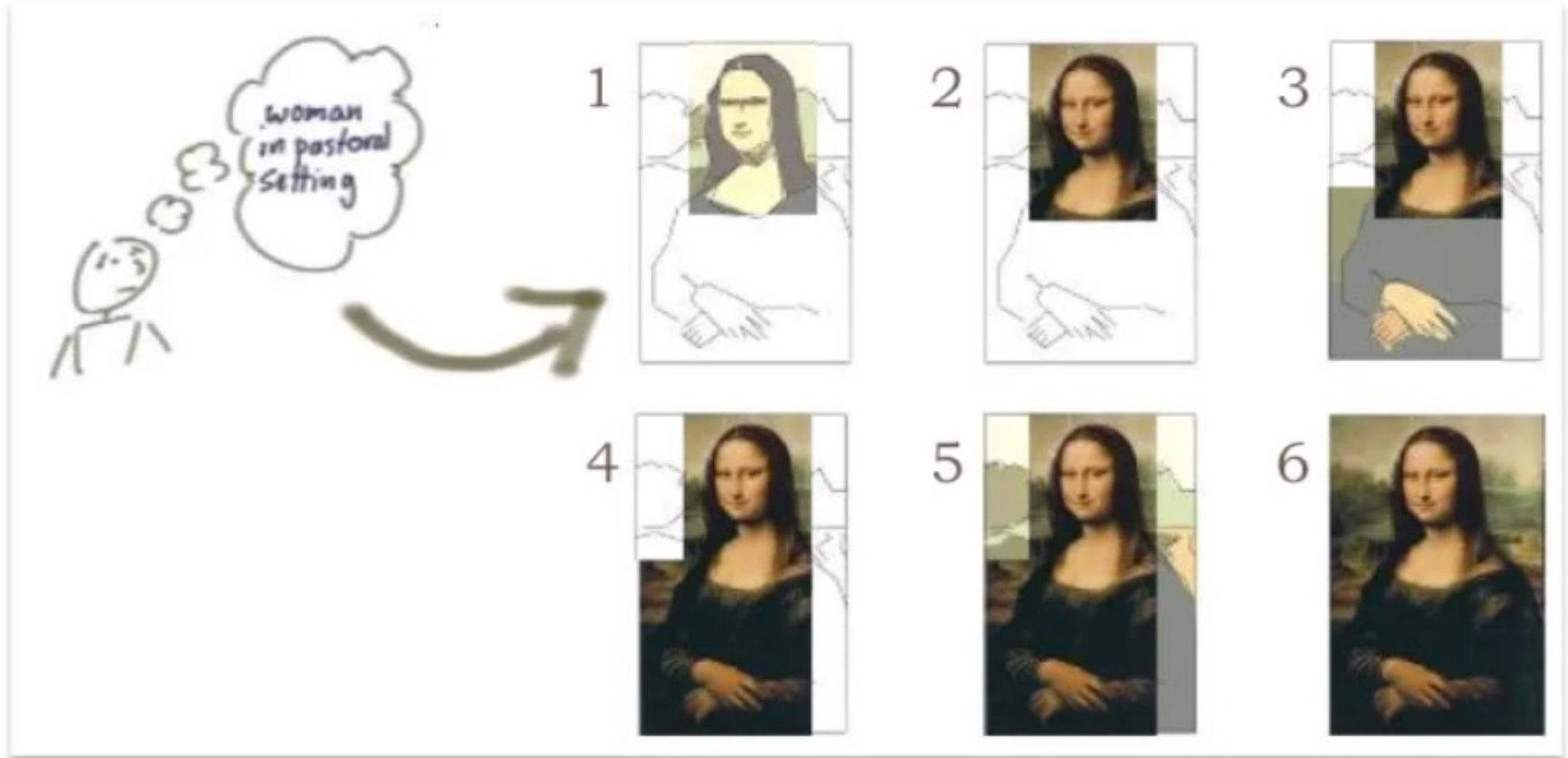


«Iterative development» by Jeff Patton (oil and bits)



- Un desarrollo **iterativo e incremental** es aquel en el que, con cada entrega, añadimos funcionalidades completamente nuevas (**incremental**) pero cada incremento también incluye mejoras sobre funcionalidades que ya existían (**iterativo**).
- Ventajas:
 - adaptar nuestros planes según lo que vayamos aprendiendo, evitando así hacer cosas que no habría que hacer, y
 - entregar valor al usuario antes de acabar todo el desarrollo.





«Iterative Incremental Development» by Steven Thomas (oil and bits)



Not like this....



1



2



3



4

Like this!



1



2



3



4



5

«Iterative & incremental development, horizontal vs vertical» by Henrik Kniberg



Metodologías Tradicionales

- Enfoque lineal
- Proceso de documentación y revisión formal
- Orientada al proyecto

Metodologías/Frameworks Ágiles

- Enfoque iterativo e incremental
- Orientada al equipo
- Se basa en el **cambio**



Metodologías Tradicionales

- Enfoque lineal
- Proceso de documentación y revisión formal
- Orientada al proyecto

Metodologías/Frameworks Ágiles

- Enfoque iterativo
- Orientada al equipo
- Se basa en el cambio

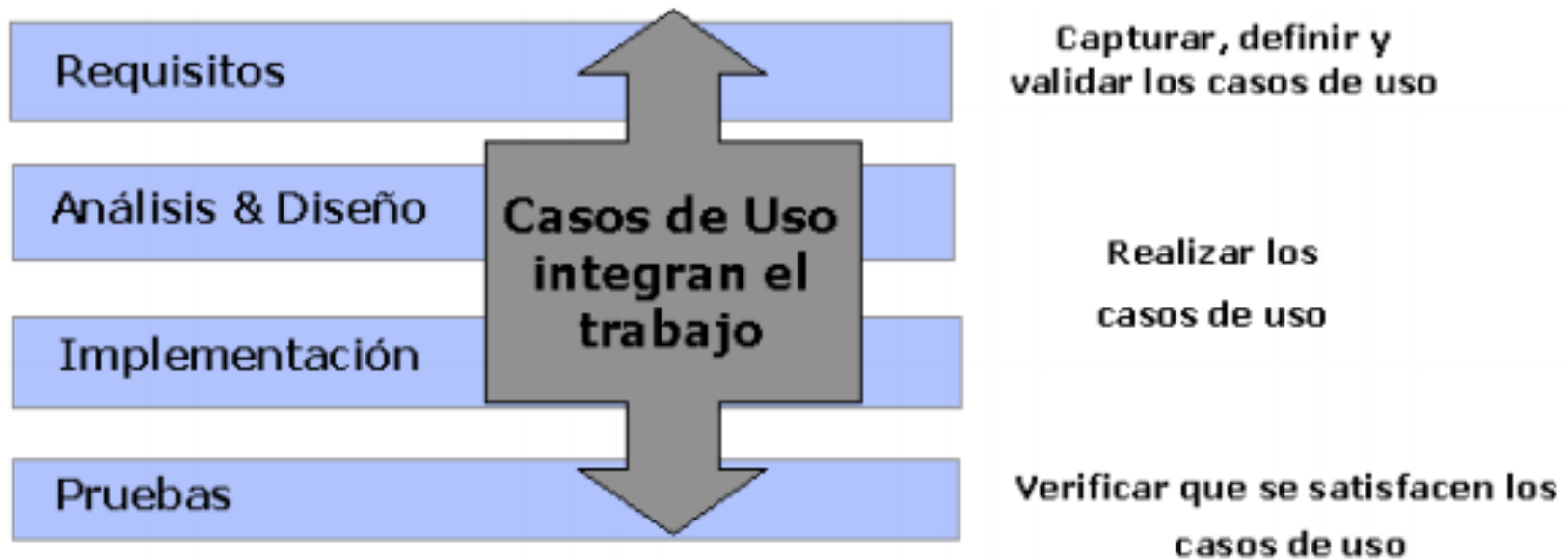


RUP. Rational Unified Process

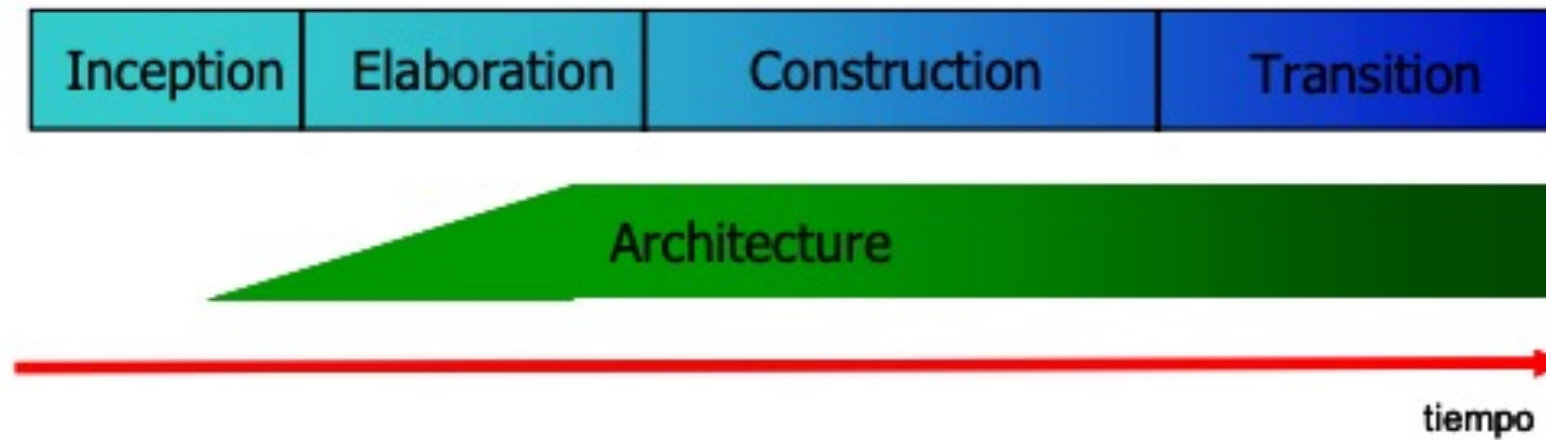
- Autores: Grady Booch, Ivar Jacobson, James Rumbaugh
- Modelo en espiral
- Modela visualmente el software
- Proceso dirigido por los casos de uso
- Proceso centrado en la arquitectura



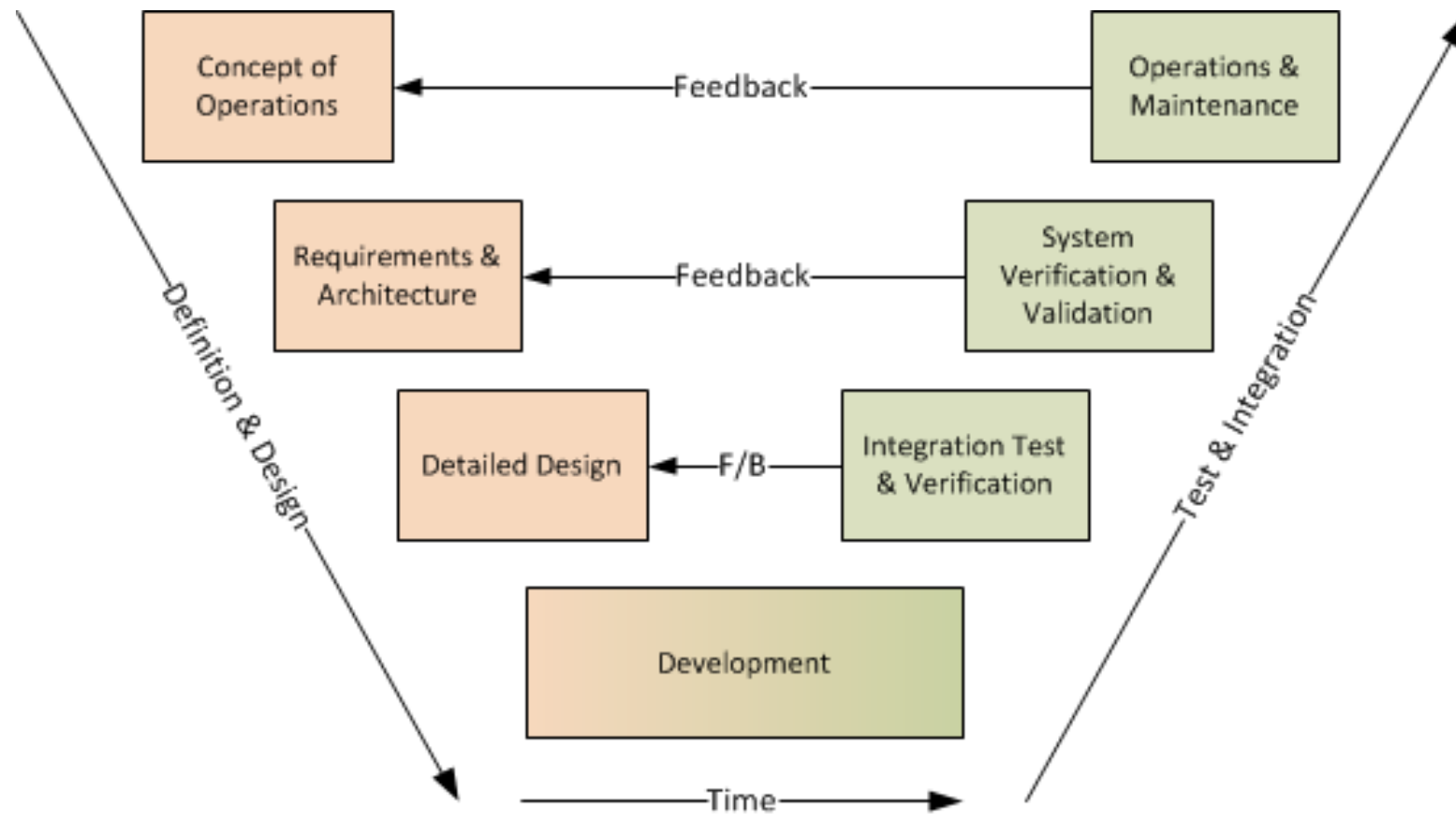
RUP. Rational Unified Process



RUP. Rational Unified Process



V-Model



V-Model

- Nivel 1 orientado al “cliente”. El inicio del proyecto y el fin del proyecto constituyen los dos extremos del ciclo. Análisis de requisitos y especificaciones → documento de requisitos y especificaciones.
- Nivel 2 características funcionales del sistema. Funciones que son directa o indirectamente visibles por el usuario final → documento de análisis funcional.
- Nivel 3 componentes hardware y software → arquitectura del sistema.
- Nivel 4 fase de implementación, en la que se desarrollan los elementos unitarios o módulos del programa.



Metodologías Tradicionales

- Enfoque lineal
- Proceso de documentación y revisión formal
- Orientada al proyecto

Metodologías/Frameworks Ágiles

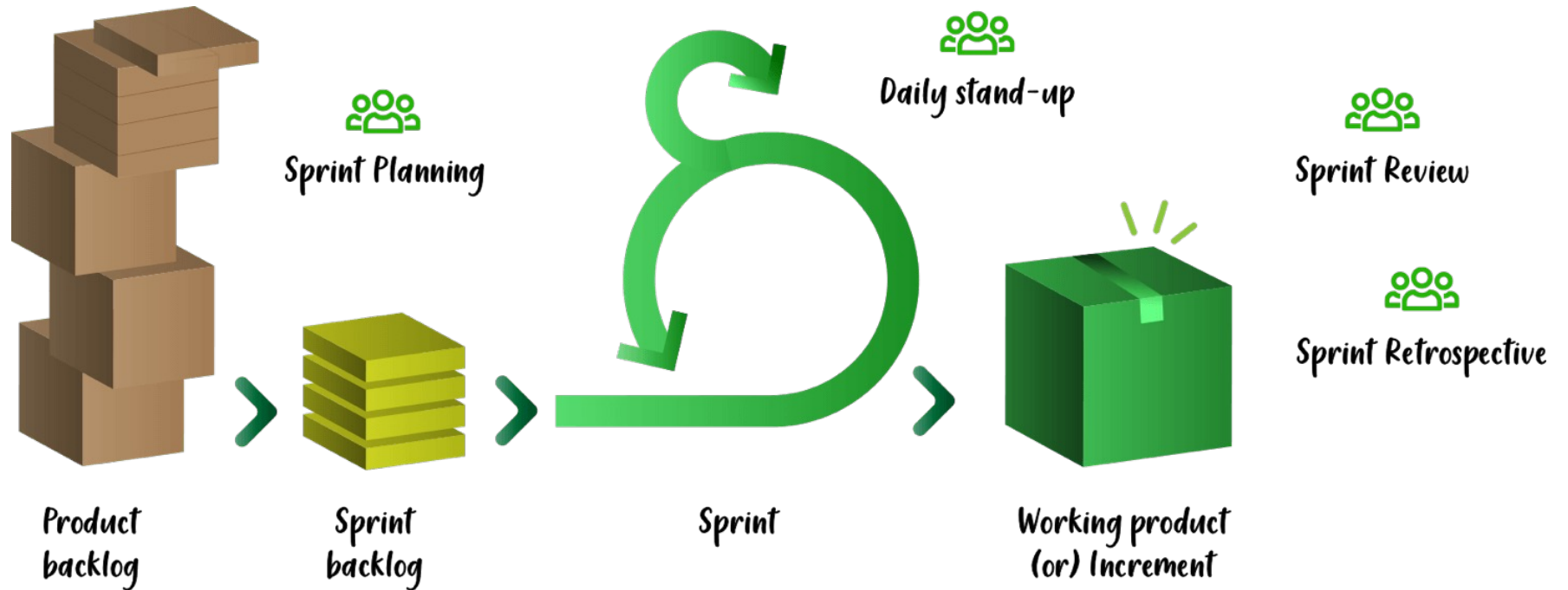
- Enfoque iterativo e incremental
- Orientada al equipo
- Se basa en el cambio



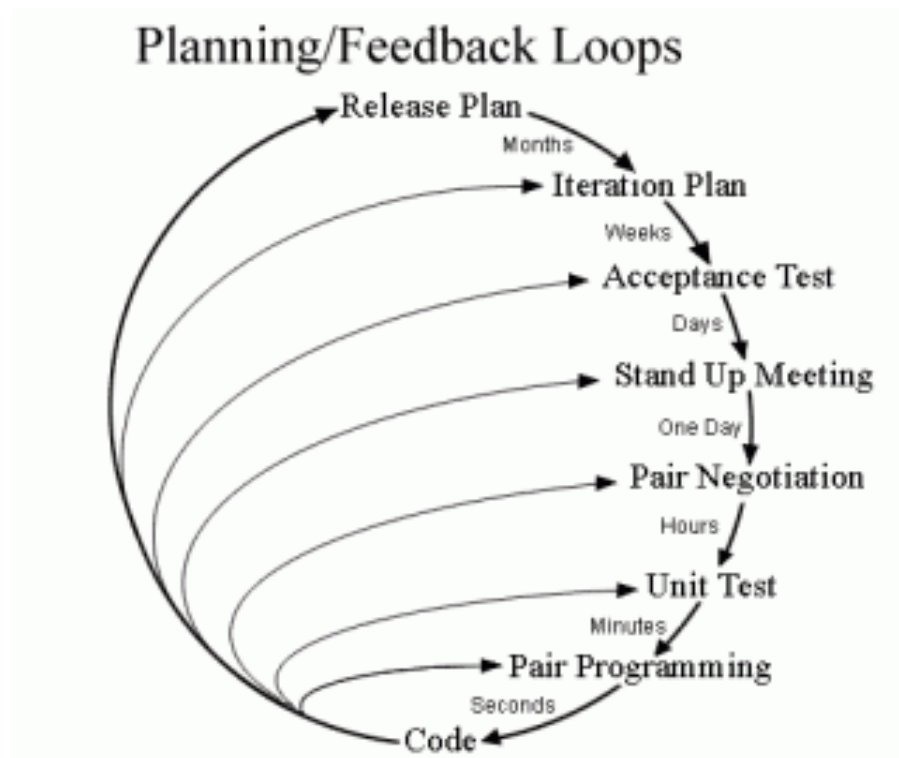
- Iteraciones cortas: 2-4 semanas
- Planificación al finalizar cada iteración
- Valores del manifiesto ágil:
 - Individuos e iteraciones
 - Software funcionando
 - Colaboración con el cliente
 - Respuesta al cambio



Scrum



Extreme Programming (XP)



¿Qué ocurre con la agilidad y el testing?

- Las metodologías ágiles no ven al software testing como una fase separada, sino como **parte integral del desarrollo de software** al igual que la programación.
- Los equipos ágiles utilizan un enfoque de “**todo el equipo**” **al testing**, con la finalidad de integrar la calidad al desarrollo del producto, al contrario de un enfoque de primero fabricar el producto y luego inspeccionar para determinar su nivel de calidad.



Agile testing

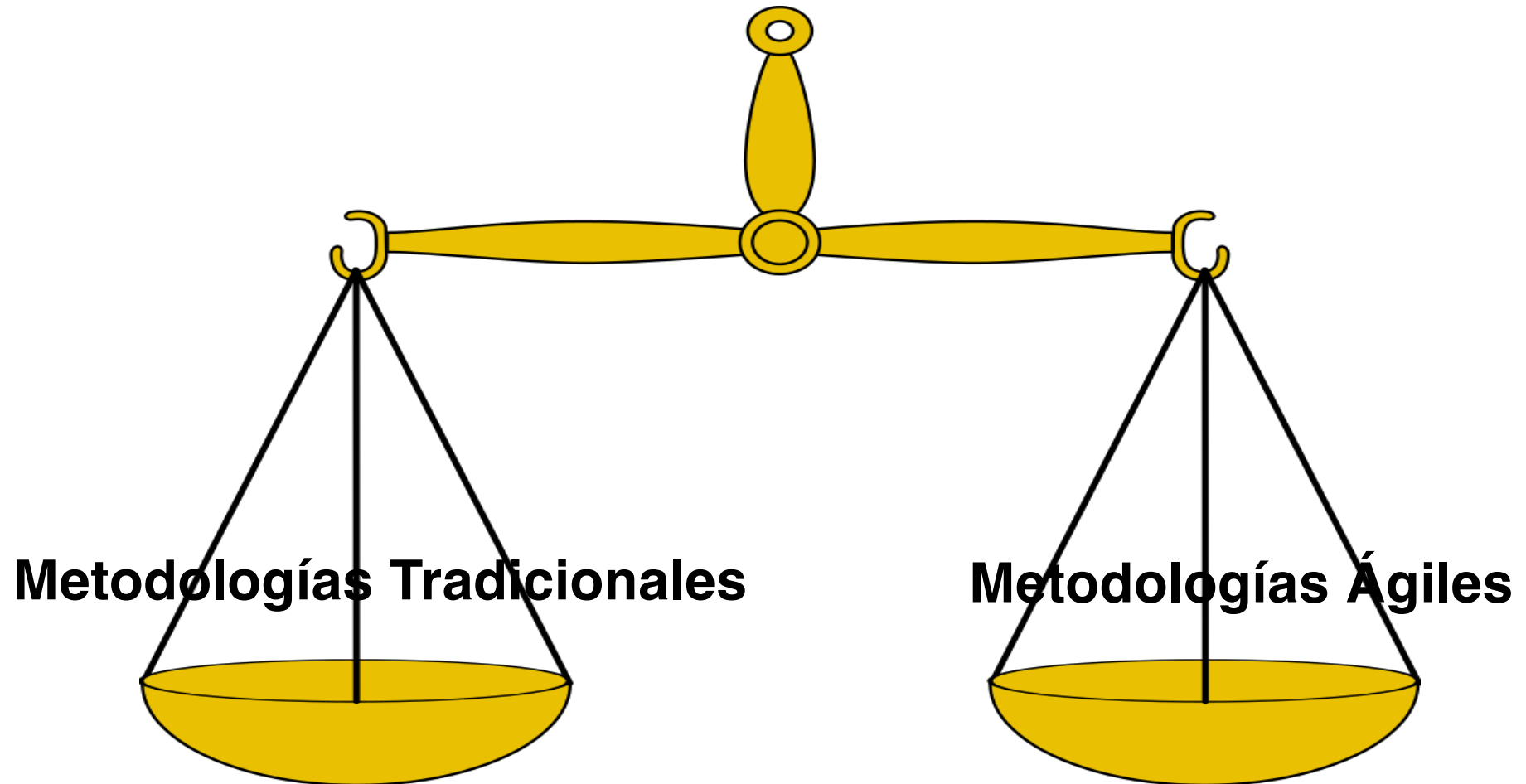
- **Agile Testing** es una práctica de pruebas de software que sigue los principios del desarrollo ágil de software.
- Involucra a todos los miembros de un equipo ágil multifuncional, en el cual el rol del tester es el de un experto multifuncional, garante que se entregue el valor de negocio deseado por el cliente a un ritmo sostenible y continuo.



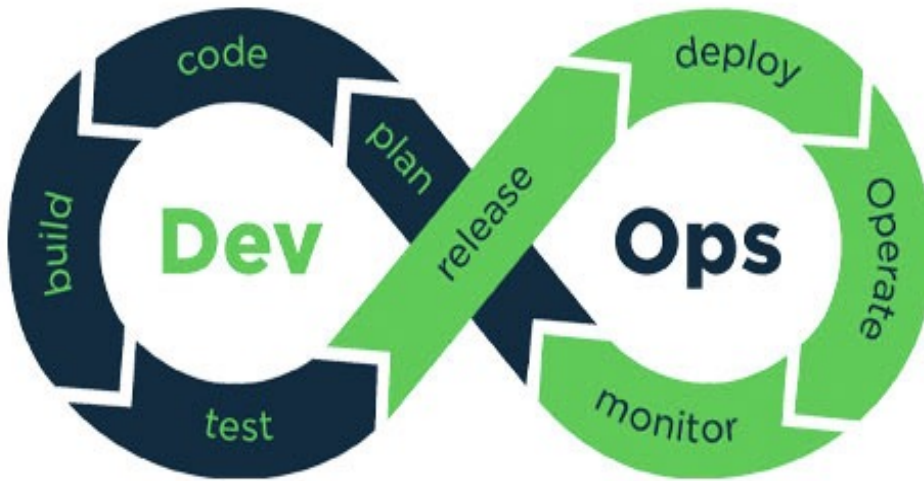
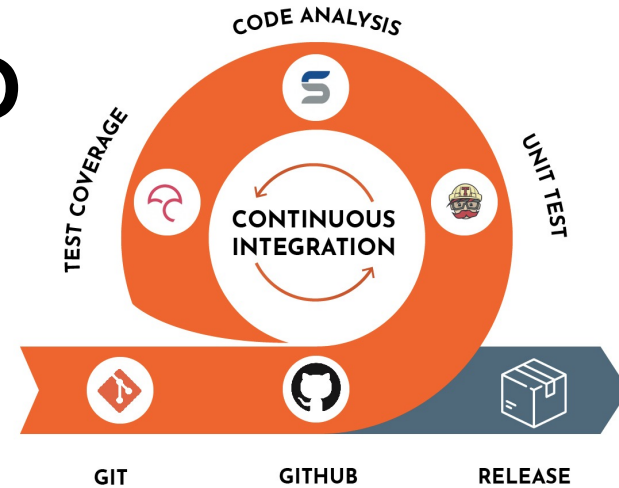
Agile testing

- Testing de “todo el equipo”
- Integración continua, T
- Desarrollo guiado por pruebas (Test Driven Development, TDD)
- Desarrollo guiado por pruebas de aceptación (Acceptance Test Driven Development – ATDD)
- ...





CI/CD



DevOps



Testing

- Una correcta estrategia de testing es lo que nos da la confianza necesaria para dar el paso para **implementar la entrega continua de valor (continuous delivery)**.
- Tras cualquier tarea, se aplica una validación que informa de que lo que se ha hecho es correcto → **feedback**.
- El testing es nuestra herramienta de feedback (se necesita saber qué es lo que ha pasado y rápido!).
- Características básicas de cualquier estrategia de testing:
 - Nos proporcionan feedback.
 - Son rápidas.



Testing

- Un elemento fundamental de DevOps cuando hablamos de testing es que **debe automatizarse**.
- Esto permite que el testing se ejecute sistemáticamente, lo que genera un **feedback continuo y rápido**.
- Esta automatización sirve también como test de regresión sobre el código antiguo.
- El testing proporciona confianza al proceso de despliegue, confiando en que nos proteja del peligro de introducir problemas indeseados en el entorno de producción.



“El testing es una actividad integrada en el desarrollo, no una fase posterior al desarrollo”



Aplicar distintas metodologías a la organización de un congreso educativo



Supongamos los ***congresos que se celebran en el ámbito educativo.***

- Un congreso es una *reunión sobre una temática* que tiene lugar en una fecha (normalmente puede durar desde 1 día hasta 5 días) y en un lugar.
- Tiene un *Conference chair*, uno o dos *General chairs*, y varios *local organizers*.
- En los congresos se exponen trabajos del área temática de distintos autores.
- Además de la exposición de estos trabajos, se organizan Key notes. Charlas impartidas por científicos de relevancia.
- El congreso ofrece coffee breaks y comidas para los asistentes.
- Además, una de las noches se celebra una cena de gala.

