

Gestión de Proyectos de Software

4.3. Metodologías de desarrollo

Contenidos

En este módulo recordaremos las metodologías para el desarrollo de software en dos categorías: clásicas y ágiles.

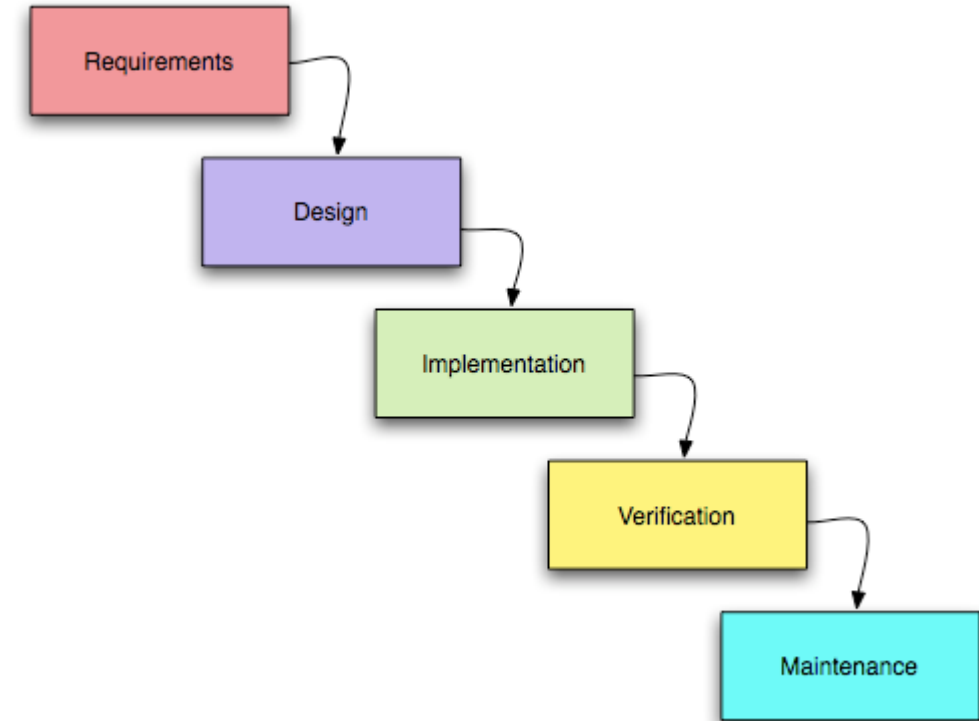
El ícono  te llevará a más recursos interesantes

- Cascada
- Prototipado
- Incremental
- RAD
- Agile
- Scrum
- eXtreme Programming
- FDD
- Kanban
- Crystal
- ASD
- Lean
- RUP
- OpenUp

Metodologías clásicas

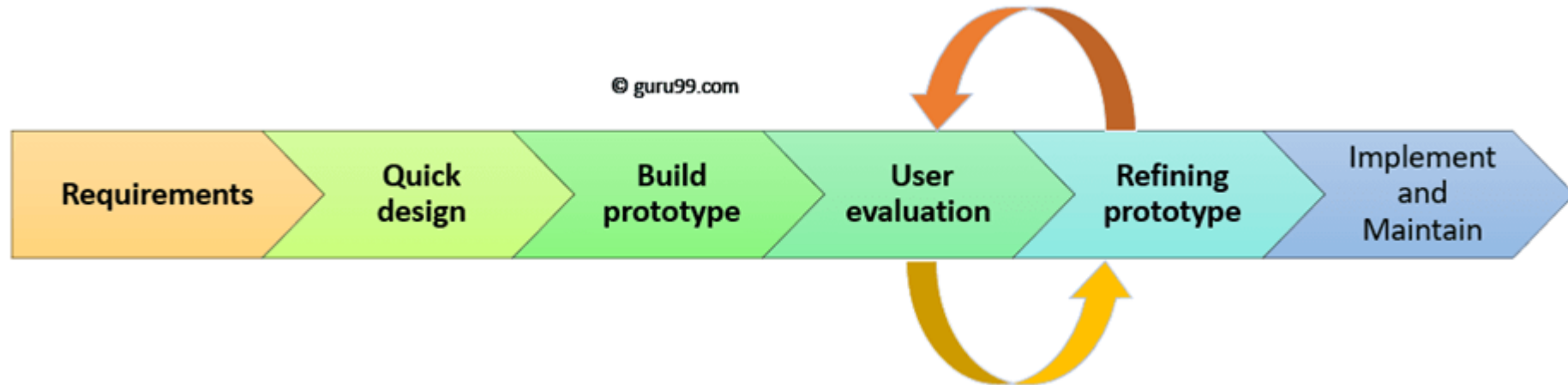
Cascada

- La metodología desde cascada para el análisis y diseño de sistemas fue el primer enfoque moderno establecido para construir un sistema. Este método fue definido originalmente por Winston W. Royce en 1970. Rápidamente se ganó el apoyo de los directivos porque todo fluye de forma lógica desde el principio del proyecto hasta el final



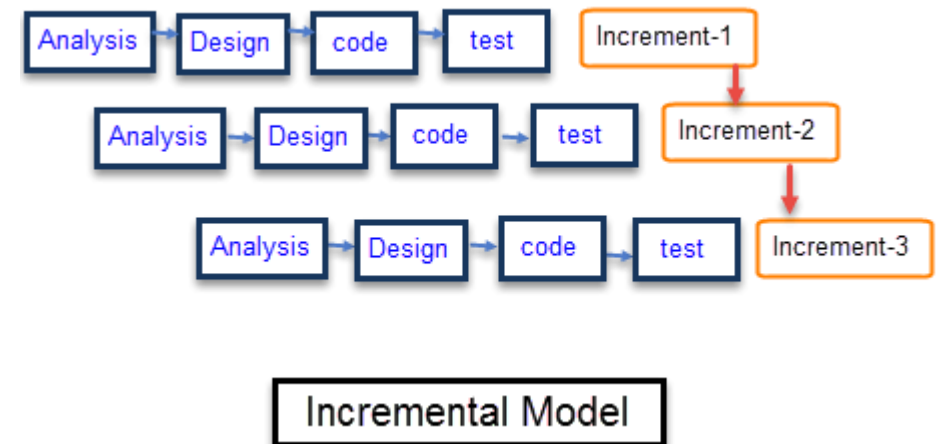
Metodología de prototipado

- Es un modelo de desarrollo de software en el que el prototipo se construye, se prueba y se reelabora hasta conseguir un prototipo aceptable.



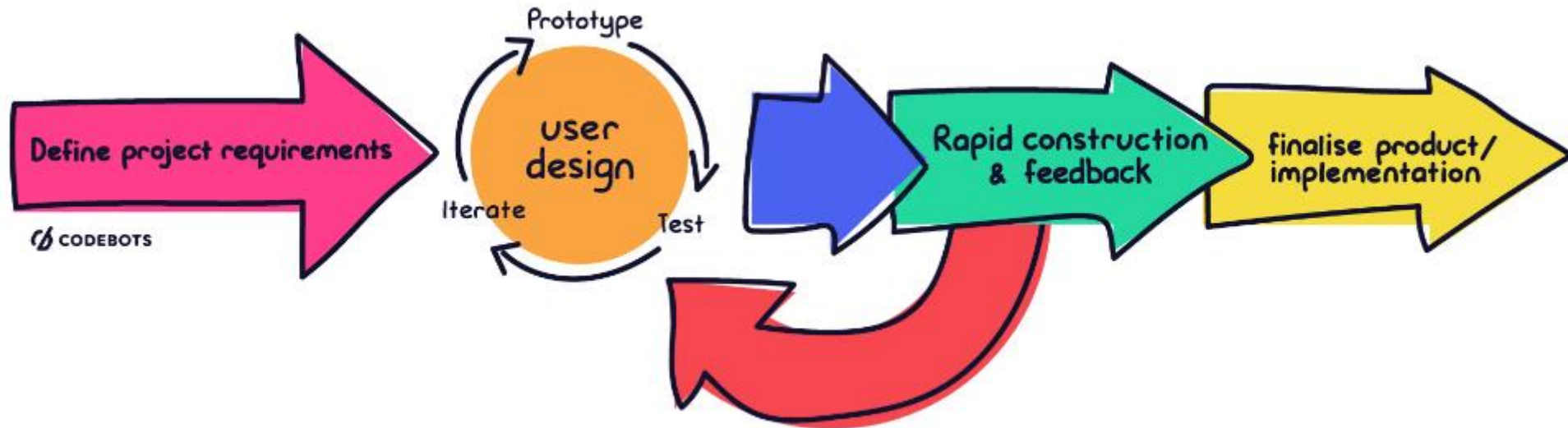
Modelo incremental

- El desarrollo del sistema se divide en muchos miniproyectos de desarrollo
- Los sistemas parciales se construyen sucesivamente para producir un sistema total final
- Los requisitos más prioritarios se abordan en primer lugar.
- Una vez desarrollado el requisito, se congelan los requisitos de ese incremento



Rapid Application Development (RAD)

- Es una metodología que se centra en el desarrollo rápido de aplicaciones mediante iteraciones frecuentes y una retroalimentación continua



Metodologías ágiles



Agile

- Agile no es una metodología. Es un conjunto de valores y principios recogidos en un manifiesto
- Ideas para contrarrestar los métodos más complicados se usaban
- Los métodos y técnicas ágiles, promueven valores similares a los de Lean, se centran principalmente en los siguientes **principios** clave
 - Satisfacer a los clientes es lo más importante
 - Desarrollar proyectos con colaboradores inspirados
 - Las interacciones son mejores cuando se realizan en persona
 - El software que funciona es una medida de progreso
 - Reflexionar y adaptar de forma continua



Agile

- Además, los cuatro **valores** fundamentales de Agile son los siguientes
 1. Las personas y las interacciones por encima de los procesos y las herramientas
 2. El software que funciona por encima de la documentación exhaustiva
 3. La colaboración con el cliente por encima de la negociación de contratos
 4. Responder al cambio en lugar de seguir un plan

Scrum

- Scrum es un marco de trabajo ágil que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos.
- En pocas palabras, Scrum requiere un *Scrum Master* para fomentar un entorno donde:
 1. Un *Product Owner* ordena el trabajo de un problema complejo en un **Product Backlog**.
 2. El *Scrum Team* convierte una selección del trabajo en un **Increment** de valor durante un **Sprint**.
 3. El *Scrum Team* y sus interesados inspeccionan los resultados y se adaptan para el próximo **Sprint**.
 4. Repita

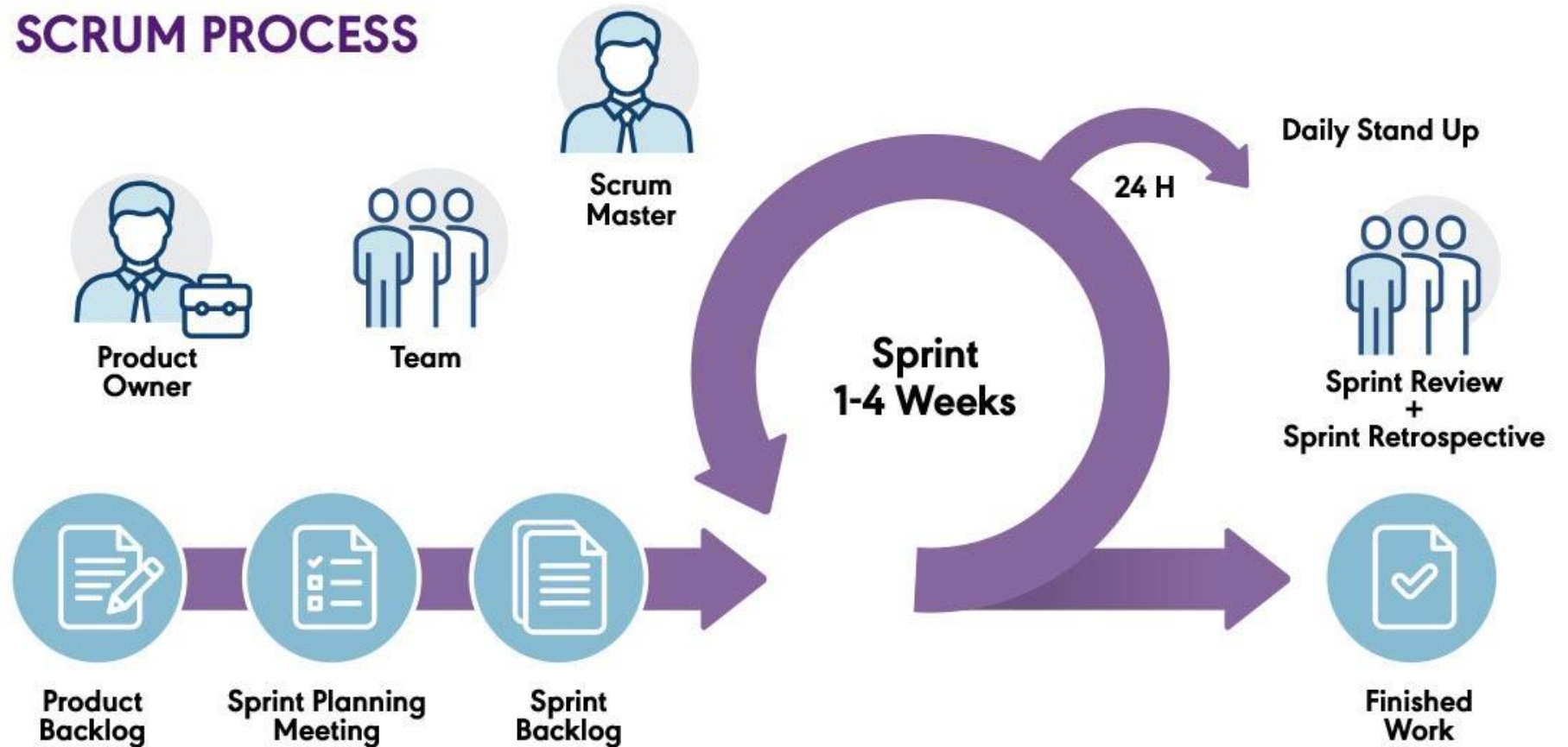


Scrum

- Valores de Scrum
 - Compromiso, Foco, Franqueza, Coraje
- Scrum emplea un enfoque iterativo e Incremental para optimizar la previsibilidad y controlar el riesgo.
- Scrum involucra a grupos de personas que colectivamente tienen todas las habilidades y experiencia para hacer el trabajo y compartir o adquirir dichas habilidades según sea necesario.

Scrum: Equipo

- Scrum Team
- Developers
- Product Owner
- Scrum Master



Scrum: Equipo

- **Scrum Team**

- Consta de un Scrum Master, un Product Owner y Developers

- **Developers**

- Crear un plan para el Sprint, el Sprint Backlog;
- Inculcar calidad al adherirse a una Definición de Terminado;
- Adaptar su plan cada día hacia el Objetivo del Sprint; y,

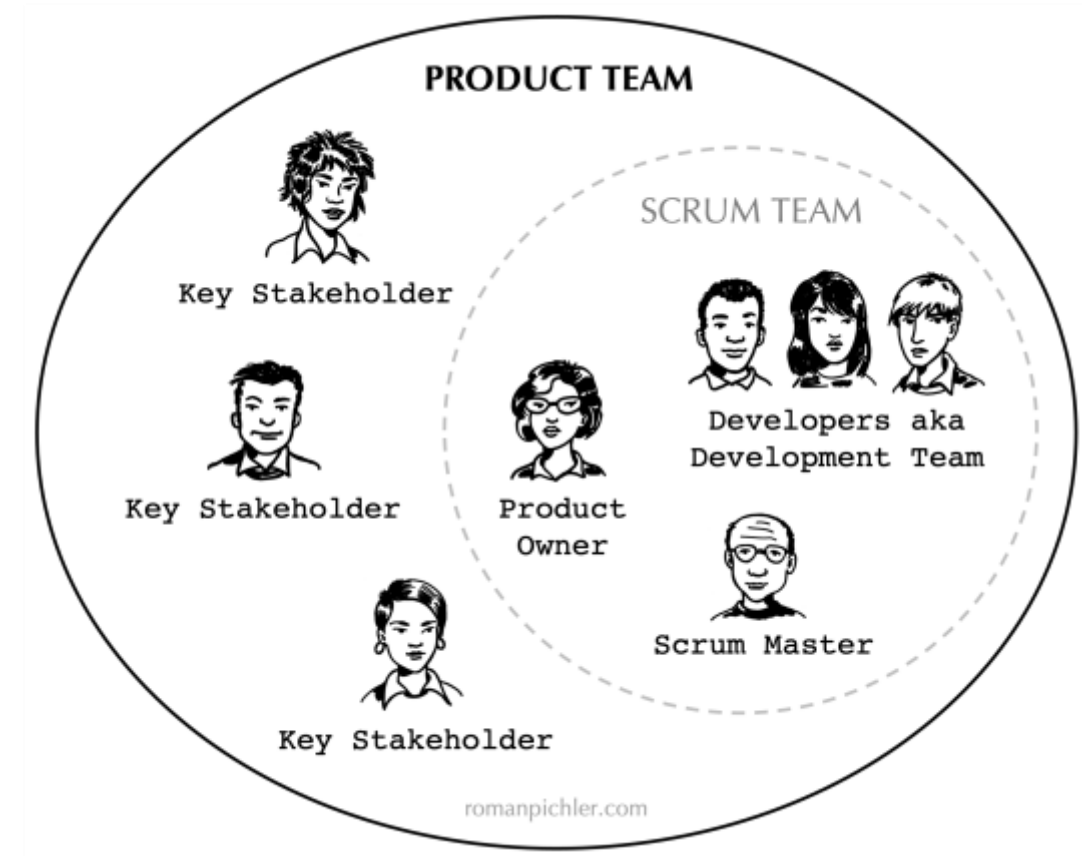
- **Product Owner**

- Desarrollar y comunicar explícitamente el Objetivo del Producto;
- Crear y comunicar claramente los elementos del Product Backlog;
- Ordenar los elementos del Product Backlog; y,
- Asegurarse de que el Product Backlog sea transparente, visible y se entienda.

Scrum: Equipo

- **Scrum Master**

- Guiar a los miembros del equipo en ser autogestionados y multifuncionales;
- Ayudar al Scrum Team a enfocarse en crear Increments de alto valor que cumplan con la Definición de Terminado;
- Elimina impedimentos para el progreso del Scrum Team
- Asegurarse de que todos los eventos de Scrum se lleven a cabo y sean positivos, productivos



Scrum: Eventos

- Sprint
- Sprint planning
- Daily Scrum
- Sprint Review
- Sprint Retrospective



Scrum: Eventos

- **Sprint**

- Eventos de duración fija. Todo sucede dentro de un sprint.

- **Sprint planning**

- Inicia el Sprint al establecer el trabajo que se realizará durante el Sprint. El Scrum Team crea este plan resultante mediante trabajo colaborativo.

- **Daily Scrum**

- Es un evento de 15 minutos para los Developers
- El propósito es inspeccionar el progreso hacia el Objetivo del Sprint y adaptar el Sprint Backlog según sea necesario.

Scrum: Eventos

- **Sprint Review**

- Inspecciona el resultado del Sprint y determinar futuras adaptaciones.
- Revisan lo que se logró en el Sprint y lo que ha cambiado en su entorno.

- **Sprint Retrospective**

- Planificar formas de aumentar la calidad y la efectividad
- Inspecciona cómo fue el último Sprint con respecto a las personas, las interacciones, los procesos, las herramientas y su Definición de Terminado.

Scrum: Artefactos

- Product Backlog
 - Objetivo del Producto
- Sprint Backlog
 - Objetivo del Sprint
- Increment
 - Definición de terminado



Scrum: Artefactos

- **Product Backlog**

- El Product Backlog es una lista emergente y ordenada de lo que se necesita para mejorar el producto. Es la única fuente del trabajo realizado por el Scrum Team.
- El Objetivo del Producto es el objetivo a largo plazo del Scrum Team. Ellos deben cumplir (o abandonar) un objetivo antes de asumir el siguiente.

- **Sprint Backlog**

- El Sprint Backlog se compone del Objetivo del Sprint (por qué), el conjunto de elementos del Product Backlog seleccionados para el Sprint (qué), así como un plan de acción para entregar el Increment (cómo).

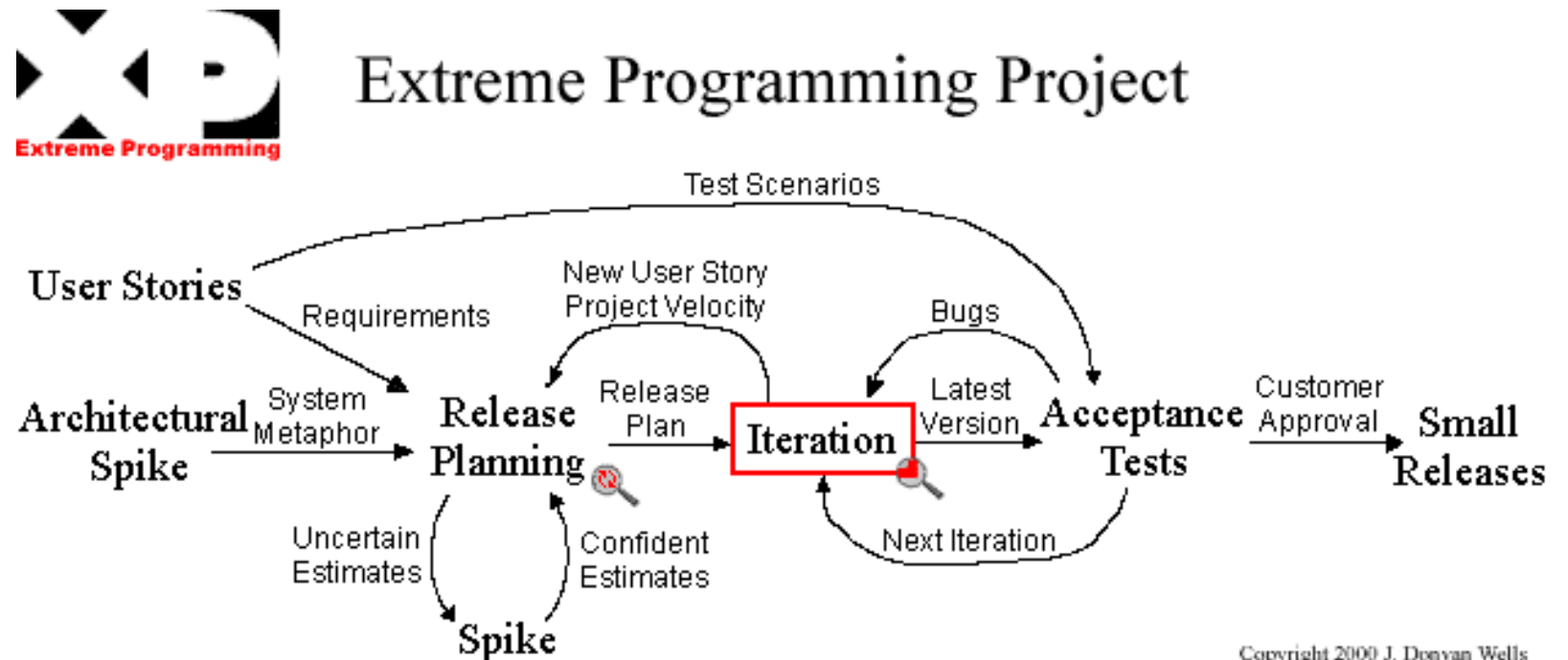
Scrum: Artefactos

- **Increment**

- Es un peldaño concreto hacia el Objetivo del Producto. Cada Increment se suma a todos los Increments anteriores y se verifica minuciosamente, lo que garantiza que todos los Increments funcionen juntos. Para proporcionar valor, el Increment debe ser utilizable.
- En el momento en que un elemento del Product Backlog cumple con la Definición de Terminado, nace un Increment

eXtreme Programming

- Metodología ágil
- Lo que diferencia a XP de las demás metodologías ágiles es que hace hincapié en los aspectos técnicos del desarrollo de software.

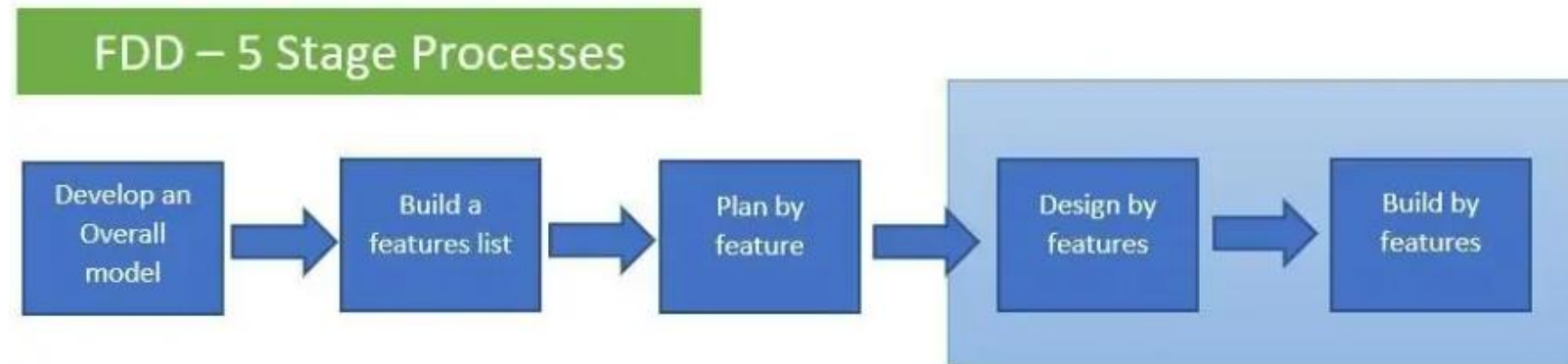


eXtreme Programming

- Implica un alto grado de participación los clientes y los desarrolladores. Los primeros inspiran el desarrollo posterior destacando las características más útiles de un determinado producto de software a través de testimonios.
- Los desarrolladores, por su parte, basan cada serie sucesiva de actualizaciones de software en estos comentarios, al tiempo que continúan probando nuevas innovaciones cada pocas semanas.

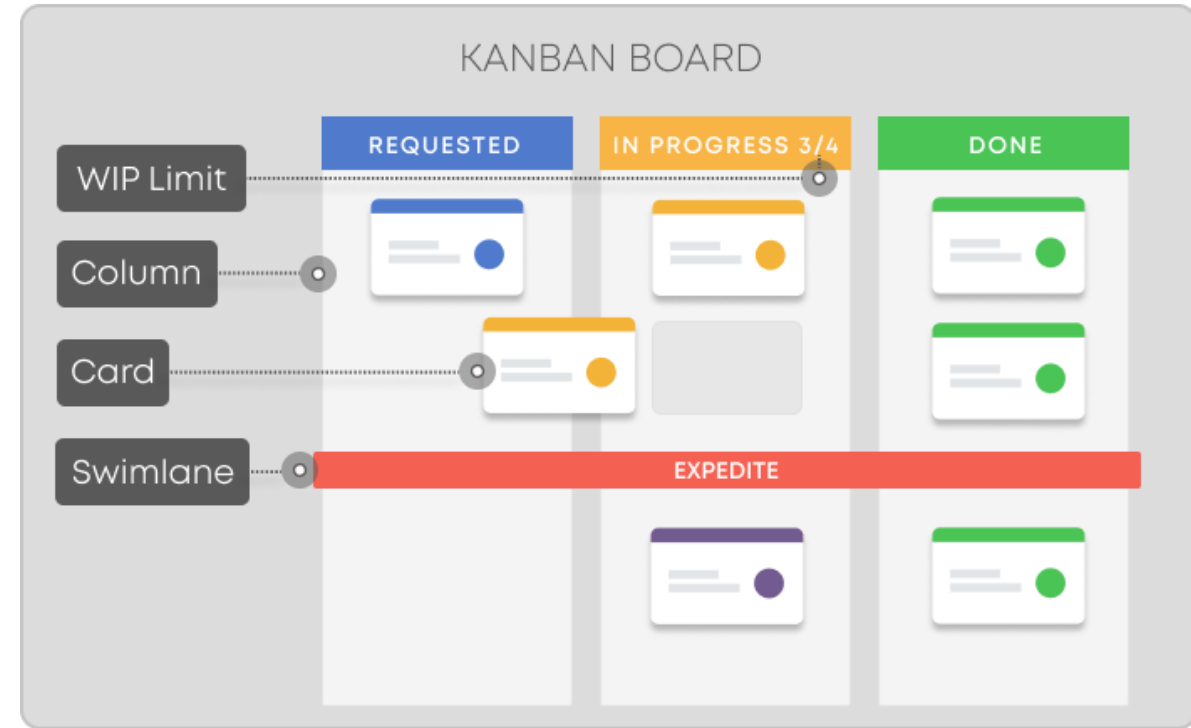
Desarrollo Dirigido por Características (FDD)

- Está orientada principalmente a equipos más grandes y con más personas que aquellos a los que normalmente se aplican otras metodologías.
- FDD fue desarrollada por Jeff De Luca y Peter Coad en el año 1997.
- Se centra en iteraciones cortas, que permiten entregas tangibles del producto en un periodo corto de tiempo (2 semanas).
- El FDD es un proceso de 5 etapas, de las cuales las 3 primeras son secuenciales y las dos últimas son iterativas. Todas las metodologías ágiles siguen una serie de principios que hacen que se parezcan entre sí.



Kanban

- Sistema visual para gestionar el trabajo a medida que avanza en un proceso.
- Visualiza tanto el proceso como el trabajo real que pasa por ese proceso.
- Su objetivo es identificar posibles cuellos de botella en el proceso y solucionarlos, de modo que el trabajo pueda fluir a través de él de forma rentable y a una velocidad o rendimiento óptimos



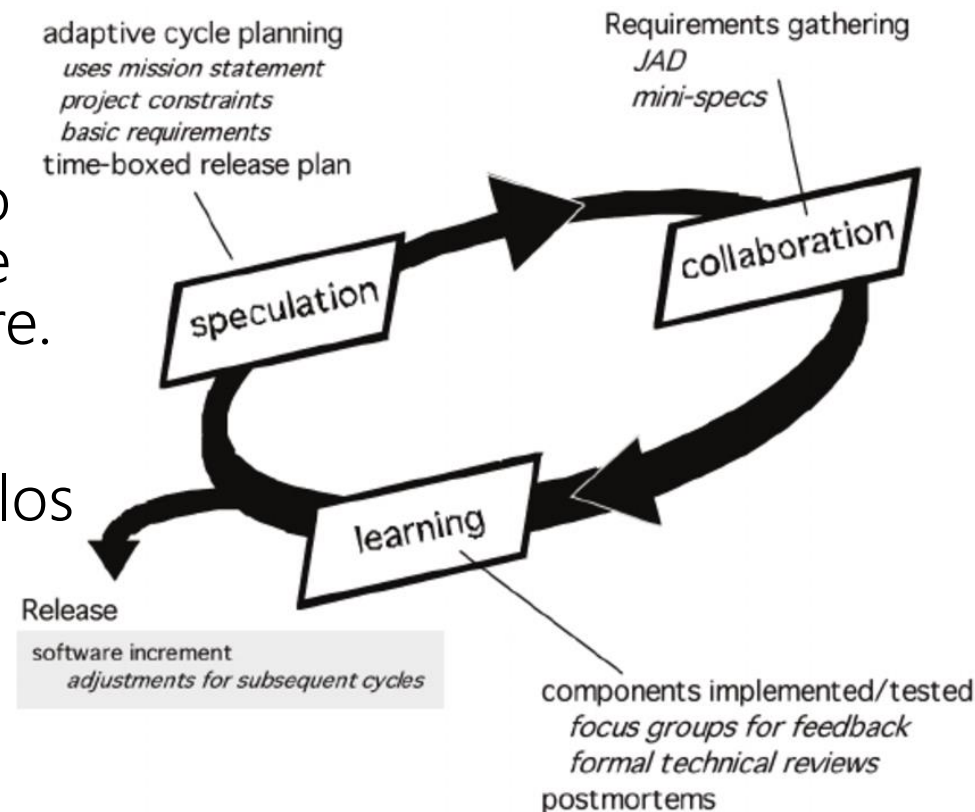
Crystal

- Es una familia de metodologías ágiles más pequeñas, como Crystal clear, Crystal yellow, Crystal red, etc.
- Hay tres factores principales que determinan las características de los distintos proyectos: el tamaño del equipo, la criticidad del sistema y las prioridades del proyecto.
- Los proyectos se clasifican en función de la criticidad del sistema, ya que existen cuatro niveles de criticidad: Comodidad (C), Dinero discrecional (D), Dinero esencial (E) y Vida (L).

	Clear	Yellow	Orange	Red	Maroon
Life (L)	L6	L20	L40	L80	L200
Essential Money (E)	E6	E20	E40	E80	E200
Discretionary Money (D)	D6	D20	D40	D80	D200
Comfort (C)	C6	C20	C40	C80	C200
	1-6	7-20	21-40	41-80	81-200

Adaptive Software Development (ASD)

- Especulación es lo que suele llamarse "planificación" en otras metodologías, ágiles o no. Reconoce explícitamente la paradoja de hacer planes en un escenario complejo y de rápida evolución.
- La fase de colaboración muestra la importancia de una colaboración más estrecha y constante, no sólo dentro del equipo de desarrollo, sino también entre los desarrolladores y los usuarios finales del software.
- Por último, la fase de aprendizaje indica que todos los implicados en el proceso de desarrollo de software estarán aprendiendo continuamente a lo largo del proyecto.



Lean

- Lean se centra en construir mejores procesos, mientras que el enfoque de Agile es construir mejores productos.
- Este enfoque tiene sus raíces en el movimiento de fabricación ajustada de los años 80.

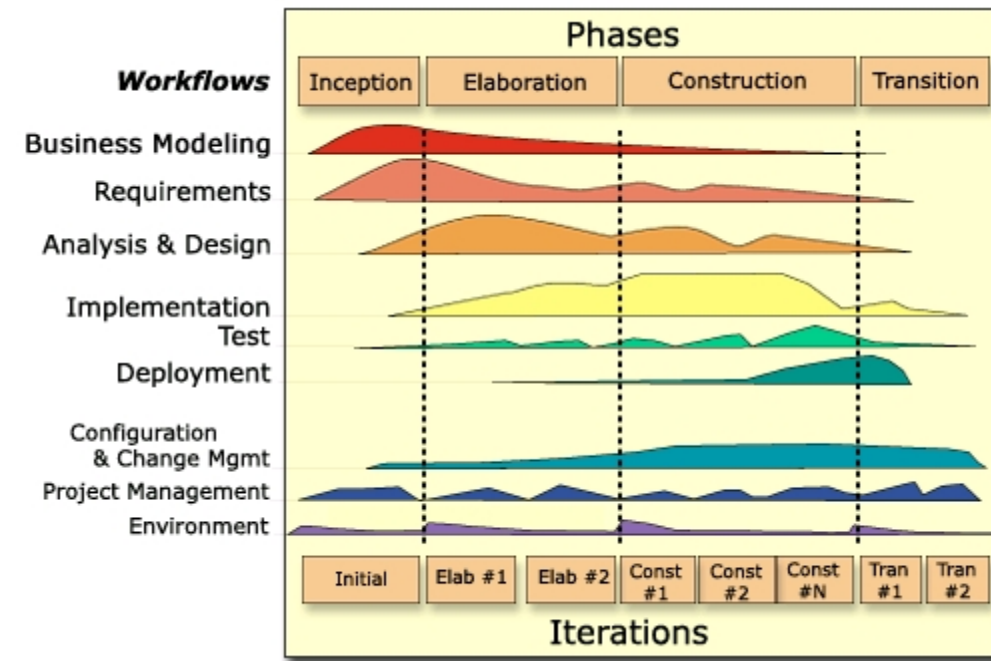
7 principles of lean software development

- Eliminate waste.
- Amplify learning.
- Decide as late as possible.
- Deliver as fast as possible.
- Empower the team.
- Build integrity in.
- See the whole.



IBM Rational Unified Process (RUP)

- Diseñado por IBM, con 4 fases
 - Inception. Validación del alcance del sistema y del presupuesto. Elaboración de un modelo básico de casos de uso, un plan de proyecto, una evaluación inicial de riesgos y una descripción del proyecto.
 - Fase de elaboración. Trabajo de análisis para mitigar los elementos de riesgo clave: análisis del dominio del problema y arquitectura [del sistema].
 - Fase de construcción. Construir el software/sistema.
 - Fase de transición. Pasar a la producción, incluidas las pruebas y la formación.



OpenUP

- Es un Proceso Unificado ajustado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado
- Adopta una filosofía pragmática y ágil que se centra en la naturaleza colaborativa del desarrollo de software. Se trata de un proceso agnóstico y poco ceremonioso que puede ampliarse para abordar una amplia variedad de tipos de proyectos.
- Desarrollado por Eclipse

