



# Arquitectura del Software

Agile Architecture



Universidad de Oviedo

|                           |   |          |
|---------------------------|---|----------|
| Mónica Villanueva Polledo | - | UO264427 |
| Lucas Suárez Cabal        | - | UO264423 |
| Luis Martínez Moreno      | - | UO258273 |

## Introducción

¿Qué es la arquitectura software?

Define la **estructura** que debe de tener un software, las piezas que debemos construir y el modo en el que se deben de juntar y trabajar entre ellas.

Permite a todo el conjunto de desarrolladores del software compartir una misma línea de trabajo

Uncle Bob (Coautor del Manifiesto Ágil) define como “**arquitecturas limpias**” aquellas que son independientes de los frameworks, de la UI, de las bases de datos y de cualquier componente externo.

El desarrollo de la arquitectura de software tiene las siguientes **etapas**: requerimientos, diseño, documentación y evaluación.

¿Qué es la arquitectura ágil?

Tiene su **origen** en febrero de 2001 cuando 17 expertos en desarrollo de software elaboran, durante un encuentro en Estados Unidos, un documento a favor de los nuevos métodos ágiles de actuación.

Permite adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo gestionarlos de forma flexible, autónoma y eficaz reduciendo los costes e incrementando su productividad.

Desde sus inicios, la metodología Agile reivindica 4 **valores**:

- Las interacciones de las personas sobre los procesos y las herramientas.
- Un software en funcionamiento frente a documentación exhaustiva.
- La participación activa del cliente durante todo el proceso de desarrollo.
- La capacidad de respuesta ante los cambios e imprevistos.

Algunas de las **ventajas** que nos brinda esta metodología son la mejora de la calidad del producto, una mayor satisfacción del cliente, mayor motivación de los trabajadores, mayor control y capacidad de predicción y muchas otras más.

En Agile existen diferentes marcos de trabajo como **Scrum** o **Kanban** todas ellas se guían a través de un patrón establecido por el **Manifiesto Ágil**.

## Kanban

En japonés “tarjeta con signos”, es un método para gestionar el trabajo, con énfasis en la entrega “just in time”.

El método Kanban consiste en organizar el flujo de trabajo, dividiendo el desarrollo de éste en distintas tareas a realizar y en el estado que pueden tomar.

Se compone de 4 pilares básicos

1. **Comenzar a implementarlo con lo que se va a hacer ahora:**
2. **Comprometerse a buscar e implementar cambios incrementales y evolutivos**
3. **Respetar los procesos, las responsabilidades y los cargos actuales**
4. **Animar el liderazgo en todos los niveles**

Tiene 6 prácticas

Kanban es susceptible de cambios, es decir, cada organización puede aplicar Kanban "a su manera" pero hay 6 pilares básicos que son troncales a cualquier alteración.

- 1. Visualizar el flujo de trabajo**
  - 1.1. Lo principal es entender qué se necesita para el transcurso del proyecto, desde su concepción hasta su entrega.
- 2. Eliminar interrupciones**
  - 2.1. Establecer un número máximo de tareas a realizar en una etapa del trabajo, puede evitar que el equipo intente abarcarse en un número excesivo de tareas
- 3. Gestionar el flujo**
  - 3.1. Por flujo, se entiende como el movimiento de tareas del trabajo sobre los distintos estados que pueden adoptar, se busca un flujo continuo y rápido.
- 4. Hacer las políticas explícitas**
  - 4.1. Se debe de definir bien el proceso que hay que realizar en cada tarea y establecer un objetivo común que se busca conseguir.
- 5. Circuitos de retroalimentación**
- 6. Mejorar colaborando**
  - 6.1. La forma de lograr la mejora continua y el cambio sostenible dentro de una organización se consigue a través de la visión compartida para un futuro mejor y la comprensión colectiva de los problemas que deben superarse.

## Scrum

Scrum es un framework de desarrollo de proyectos aplicable a Software, entre sus principales ventajas son la transparencia de la elaboración del producto con el cliente y la capacidad evolutiva de conocimiento de los integrantes con respeto al proyecto.

Para comprender Scrum, es necesario definir los distintos roles de todas las personas involucradas en el proyecto y las fases de desarrollo del mismo.

### Roles

Product Owner: Se encarga de convertir los intereses del cliente en funcionalidades del proyecto. Para ello redacta un documento llamado Product Backlog, donde se incluyen todas las funcionalidades ordenadas por orden de importancia.

Por cada funcionalidad se añade una estimación del tiempo requerido para elaborarla, esta estimación se puede actualizar conforme avanza el proyecto.

Scrum Master: Es el especialista de Scrum dentro del proyecto, se encarga de instruir al Equipo de Desarrollo para que siga las pautas del framework.

Developer Team: Se encarga de implementar las funcionalidades, suele ser de unos 8 miembros

### Fases

- El Product Owner se encarga de redactar el Product Backlog antes del primer Sprint.
- Se hace una reunión inicial llamada Sprint Planning Meeting, donde se crea un documento llamado Sprint BackLog, un documento creado a partir del Product BackLog en el que se

establecen una serie de requisitos que se tienen que elaborar entre 1 y 4 semanas de trabajo (a este periodo se le conoce como Sprint) .

- Cada requisito del Sprint BackLog consta de tres partes: resumen, participante y duración
- Cada día del Sprint se realiza una reunión entre el Scrum Master y el Developer Team llamada Daily Scrum de unos 15 min.
- Fin de Sprint y entrega del producto al cliente
- Se hace una reunión llamada Sprint Retrospective donde se evalúa el producto entregado y se redacta un nuevo Sprint Backlog y se inicia un nuevo Sprint

## Principios del manifiesto ágil

1. Lograr la satisfacción del cliente a través de la entrega continua de software.
2. No tener miedo de realizar cambios
3. Entregar software funcional en una escala de tiempo menor.
4. Desarrolladores y gerencia deben trabajar juntos.
5. Desarrollar proyectos en torno a personas motivadas.
6. Interactuar cara a cara es el modo de comunicación más eficiente y efectivo.
7. Un software que funciona es la medida principal del progreso
8. Los procesos ágiles promueven el desarrollo sostenible.
9. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
10. La simplicidad es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados.
12. Inspeccionar y adaptar.

## Referencias

- Scrum en 6 minutos - Cristian Hernao  
<https://www.youtube.com/watch?v=HhC75lonpOU>
- The Scrum Primer - <https://scrumprimer.org>
- <https://devexperto.com/arquitectura-del-software/>
- <https://sg.com.mx/revista/27/arquitectura-software>
- <https://blog.hubspot.es/marketing/metodologia-agile>
- <https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban>
- [https://es.wikipedia.org/wiki/Kanban\\_\(desarrollo\)](https://es.wikipedia.org/wiki/Kanban_(desarrollo))
- <https://kanbantool.com/es/metodologia-kanban>
- <https://blog.trello.com/es/metodologia-kanban>
- <https://www.youtube.com/watch?v=dA0MCq6ReWA>
- <https://www.youtube.com/watch?v=WP6Nt5XV980>