**Ágil Scrum**

|  |
| --- |
| **Enlaces** |
| Enlace1  Enlace1 |
| **Videos** |
| Video1  Video1  Video1 |
| **Apuntes** |
| [Resumen Manifiesto Ágil chatGPT](#_Resumen_chatGPT)  [Resumen Scrum chatGPT](#_Resumen_CAP1_chatGPT) |

## Resumen Manifiesto Ágil chatGPT

El Manifiesto Ágil es un conjunto de valores y principios que guían el desarrollo ágil de software. Aquí tienes un resumen:

### Valores del Manifiesto Ágil:

1. **Individuos e interacciones sobre procesos y herramientas**: Enfatiza la importancia de las personas y su colaboración directa en el desarrollo de software sobre la dependencia de procesos y herramientas. Es crucial que los equipos trabajen juntos de manera efectiva y se comuniquen abierta y frecuentemente.
2. **Software funcionando sobre documentación extensiva**: Pone énfasis en la entrega de software de calidad que funcione correctamente sobre la elaboración excesiva de documentación. La idea es que el software es el producto final más valioso y tangible.
3. **Colaboración con el cliente sobre negociación contractual**: Destaca la importancia de la colaboración continua con el cliente para comprender y adaptarse a sus necesidades cambiantes, en lugar de depender únicamente de contratos y acuerdos rígidos.
4. **Respuesta ante el cambio sobre seguir un plan**: Reconoce que en el desarrollo de software, los requisitos y las circunstancias pueden cambiar, y que los equipos deben ser capaces de adaptarse y responder rápidamente a esos cambios en lugar de aferrarse a un plan original.

### Principios del Manifiesto Ágil:

1. **Satisfacer al cliente mediante la entrega temprana y continua de software con valor**: Entregar software funcional en intervalos cortos para obtener retroalimentación rápida y ajustar el desarrollo según las necesidades del cliente.
2. **Aceptar cambios de requisitos, incluso en etapas tardías del desarrollo**: Adaptarse a los cambios del proyecto para aprovechar nuevas oportunidades o resolver problemas emergentes.
3. **Entregar software funcional frecuentemente, en semanas en lugar de meses**: Priorizar la entrega continua de partes operativas del sistema para obtener retroalimentación y evaluar el progreso.
4. **Colaboración constante entre los desarrolladores y los interesados**: Fomentar la comunicación y la colaboración entre los miembros del equipo y los interesados en el proyecto para garantizar una comprensión compartida y una visión clara.
5. **Construir proyectos en torno a individuos motivados, dándoles el entorno y el apoyo que necesitan, y confiando en que harán el trabajo**: Empoderar a los equipos para que tomen decisiones y resuelvan problemas, confiando en su capacidad y motivación.
6. **El método más eficiente y efectivo de transmitir información al equipo de desarrollo y entre sus miembros es la conversación cara a cara**: Favorecer la comunicación directa y efectiva sobre la documentación extensa y las reuniones formales.
7. **Software funcionando es la medida principal de progreso**: Evaluar el progreso del proyecto por la cantidad de software funcional y útil que se ha entregado, en lugar de por métricas basadas en el tiempo o en la cantidad de trabajo completado.
8. **Mantener un ritmo constante de desarrollo de forma sostenible indefinidamente**: Evitar el agotamiento del equipo y mantener un ritmo de desarrollo sostenible para garantizar la calidad y la eficiencia a largo plazo.
9. **Atención continua a la excelencia técnica y al buen diseño para mejorar la agilidad**: Priorizar la calidad del software y la excelencia técnica para garantizar que el producto final sea sostenible y fácil de mantener y mejorar.
10. **Simplicidad, es decir, el arte de maximizar la cantidad de trabajo no realizado, es esencial**: Enfocarse en lo esencial y evitar el exceso de ingeniería o funcionalidades innecesarias para minimizar el desperdicio y maximizar el valor entregado.
11. **Las mejores arquitecturas, requisitos y diseños emergen de equipos autoorganizados**: Permitir que los equipos tomen decisiones sobre la mejor manera de abordar problemas y diseñar soluciones, fomentando la autoorganización y la autonomía.
12. **A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo y ajusta su comportamiento en consecuencia**: Fomentar la mejora continua mediante la reflexión y la adaptación de los procesos y prácticas para maximizar el valor entregado y la eficiencia del equipo.

## Resumen Scrum chatGPT

Resumen de la metodología Scrum:

**1. Roles:**

* **Product Owner:** Representa las necesidades del cliente y es responsable de gestionar el backlog del producto.
* **Scrum Master:** Facilita el proceso Scrum, elimina los obstáculos del equipo y ayuda a mantener un entorno de trabajo productivo.
* **Equipo de Desarrollo:** Son los encargados de convertir los elementos del backlog en incrementos de producto.

**2. Artefactos:**

* **Backlog del producto:** Lista prioritizada de todas las funcionalidades deseadas para el producto.
* **Backlog del sprint:** Subset del backlog del producto seleccionado para el sprint actual.
* **Incremento:** La suma de todas las funcionalidades completadas durante un sprint, listas para ser entregadas al cliente.

**3. Eventos:**

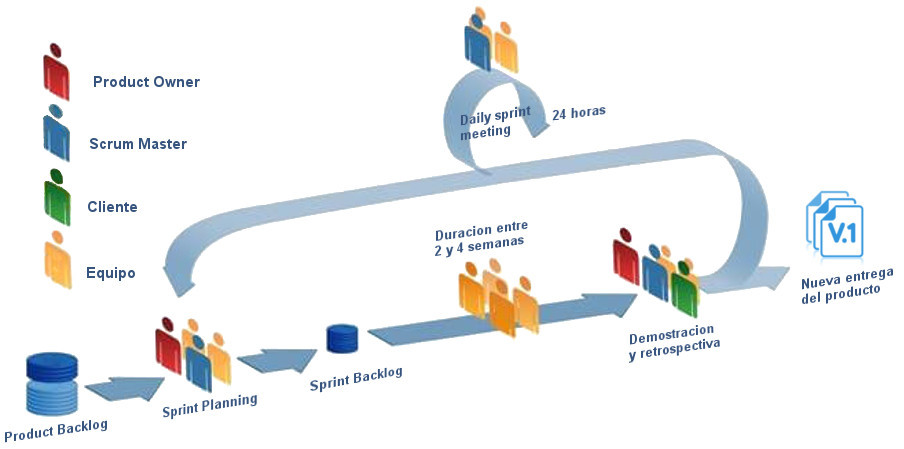
* **Sprint Planning (Planificación del Sprint):** Reunión al comienzo de cada sprint para determinar qué se entregará y cómo se llevará a cabo el trabajo.
* **Daily Scrum (Scrum Diario):** Reunión corta diaria donde el equipo actualiza el progreso y coordina las tareas para el día.
* **Sprint Review (Revisión del Sprint):** Reunión al final de cada sprint donde se presenta el trabajo completado al Product Owner y a los interesados.
* **Sprint Retrospective (Retrospectiva del Sprint):** Reunión al final de cada sprint donde el equipo reflexiona sobre qué salió bien, qué salió mal y cómo mejorar.

**4. Principios clave:**

* **Transparencia:** Todos los aspectos del proceso deben ser visibles para todos los implicados.
* **Inspección:** El progreso debe ser inspeccionado regularmente para detectar problemas y adaptar el enfoque.
* **Adaptación:** El equipo debe ser capaz de adaptarse continuamente para mejorar su efectividad.

**5. Ciclo de vida:**

* El trabajo se organiza en sprints (iteraciones cortas) que generalmente duran de 2 a 4 semanas.
* Al comienzo de cada sprint, se seleccionan elementos del backlog del producto para ser completados durante ese sprint.
* Durante el sprint, el equipo trabaja en la implementación de estas funcionalidades.
* Al final del sprint, se presenta el trabajo completado en la revisión del sprint y se reflexiona sobre el proceso en la retrospectiva.



**Beneficios de Scrum:**

* Flexibilidad para adaptarse a los cambios.
* Mayor transparencia y visibilidad del progreso del proyecto.
* Entrega regular de incrementos de producto funcionales.
* Mejora continua a través de la retroalimentación y la reflexión.

En resumen, Scrum es un marco de trabajo ágil que promueve la entrega temprana y regular de valor, la adaptabilidad a los cambios y la mejora continua del proceso de desarrollo.