

5 Metodologías

Metodología está compuesta por modos sistemáticos de realizar, gestionar y administrar un proyecto a través de etapas y acciones, partiendo desde las necesidades del producto hasta cumplir con el objetivo para el que fue creado.

Metodologías Tradicionales

- RUP (Rational Unified Process)

Metodologías Ágiles

- XP
- SCRUM
- Kanban

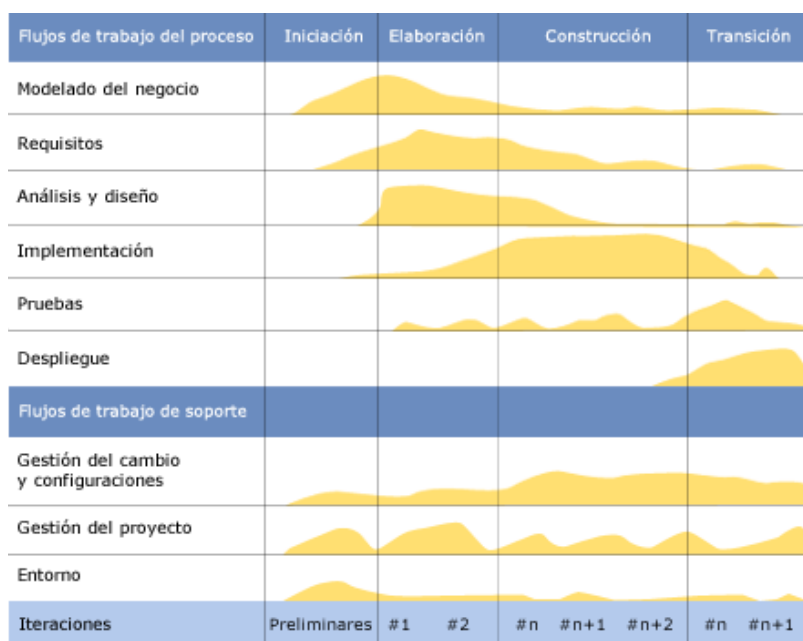
5.1 Metodología tradicional: RUP

Dirigido por casos de uso: los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

Centrado en la arquitectura: Se definen los Casos de Usos principales, los cimientos de la aplicación y sobre ello se va incrementando.

Iterativo e Incremental: Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

Uso extensivo de documentación.

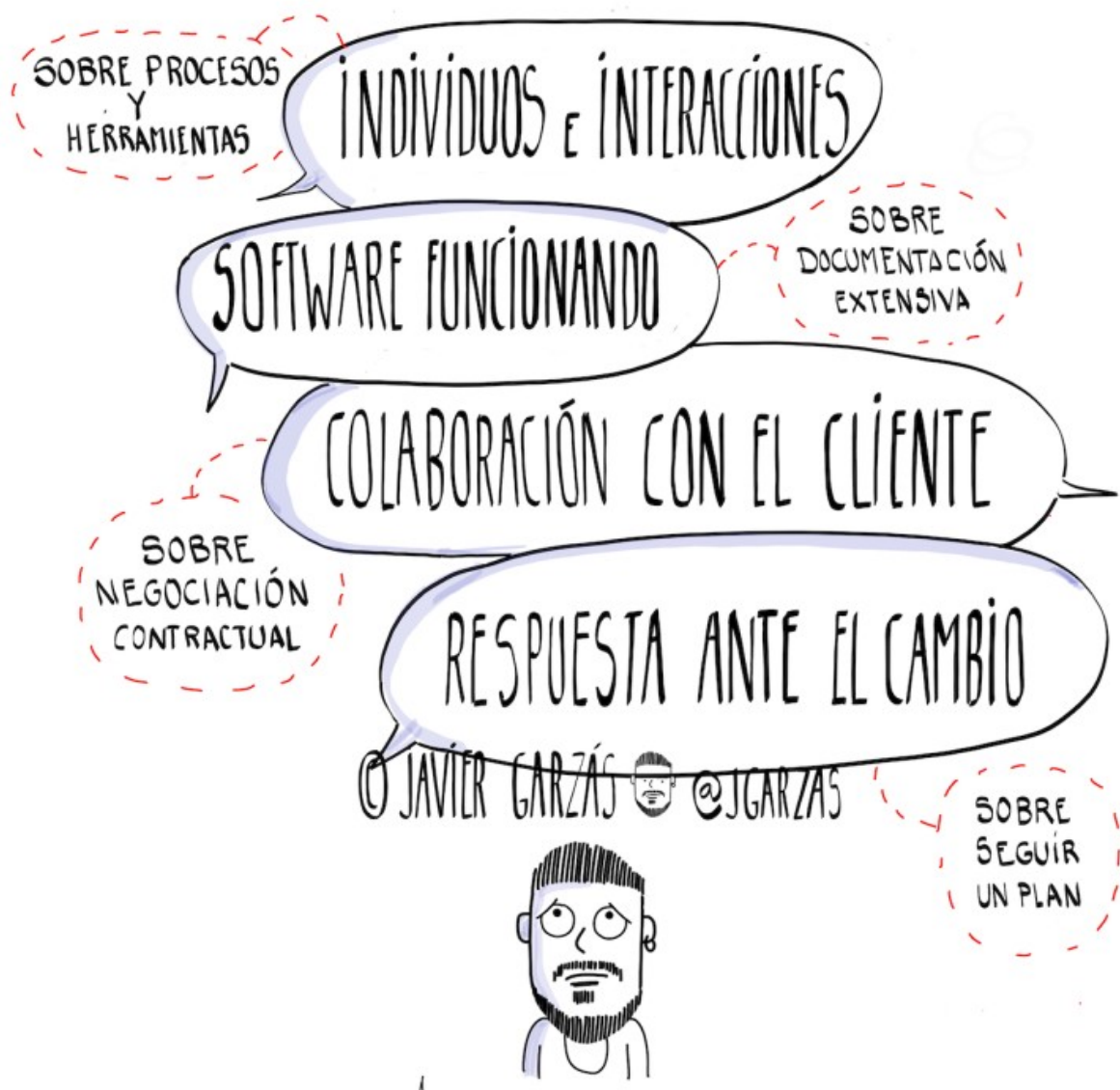


5.2 Metodologías ágiles

El 12 de febrero de 2001 diecisiete críticos de los modelos de mejora del desarrollo de software basados en procesos, convocados por Kent Beck, se reunieron en Snowbird, Utah para tratar sobre técnicas y procesos para desarrollar software.

En la reunión se acuñó el término “Métodos Ágiles” para definir a los métodos que estaban surgiendo como alternativa a las metodologías formales (CMMI, SPICE) a las que consideraban excesivamente “pesadas” y rígidas por su carácter normativo y fuerte dependencia de planificaciones detalladas previas al desarrollo.

Los integrantes de la reunión resumieron los principios sobre los que se basan los métodos alternativos en cuatro postulados, lo que ha quedado denominado como Manifiesto Ágil.

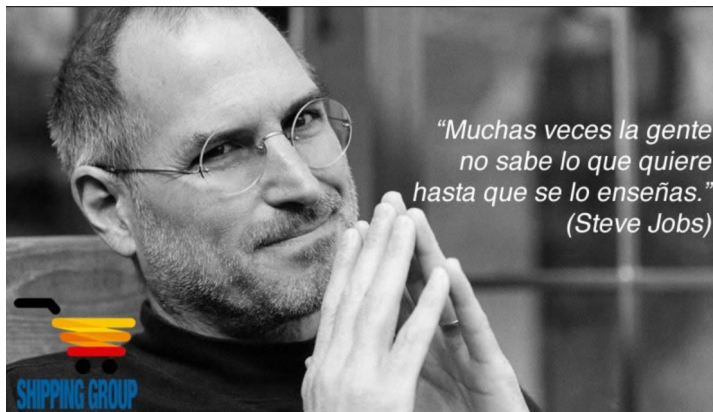


5.2.1 XP. eXtreme Programming

Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos.

El cliente no sabe lo que quiere hasta que ve lo que no quiere.



Esta metodología lleva las buenas prácticas a niveles extremos. Por ejemplo, el TDD (test driven development) fue usado en la NASA en el proyecto mercurio en los 60.

Valores:

- Comunicación: dentro en un equipo de desarrollo es fundamental para que la información entre los miembros fluya con normalidad, esto ayudará a evitar errores y agilizar las tareas compartidas
- Retroalimentación (Feedback): Con el cliente es de gran ayuda entregar un software de calidad que cumpla las expectativas del mismo cliente. Para ello, el desarrollo debe realizarse en ciclos cortos que permitan mantener una retroalimentación constante y continua.
- Simplicidad: Cuando se trabaja en el diseño de una historia de usuario, es importante centrarse en esa tarea exclusivamente para mantener un código limpio y sencillo. Esto ayuda a evitar el over-engineering, es decir, no se debe preparar software “por si acaso”
- Coraje ser valiente para evitar over-engineering, para desechar un código fuente antiguo inservible, para refactorizar código que funcione o para sobreponerse de problemas que lleva tiempo sin resolverse.

5.2.2 Scrum

Scrum es un marco de trabajo que define un conjunto de prácticas y roles, y que puede tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto.

Los roles principales en Scrum son el Scrum Master, que procura facilitar la aplicación de scrum y gestionar cambios, el Product Owner, que representa a los clientes/usuarios y el Team Developer (equipo de desarrollo) que realiza el desarrollo y demás elementos relacionados con él.

Durante cada sprint, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo y debe ser lo más corta posible), el equipo crea un incremento de software potencialmente entregable (utilizable).

El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar.

La metodología Scrum se basa en:

- El desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos.
- Se da prioridad a lo que tiene más valor para el cliente.
- El equipo se sincroniza diariamente y se realizan las adaptaciones necesarias.
- Tras cada iteración (un mes o menos entre cada una) se muestra al cliente el resultado real obtenido, para que este tome las decisiones necesarias en relación a lo observado.
- Se le da la autoridad necesaria al equipo para poder cumplir los requisitos.
- Fijar tiempos máximos para lograr objetivos.
- Equipos pequeños (de 3 a 9 personas cada uno).

5.2.3 Kanban

La palabra Kanban viene del japonés y traducida literalmente quiere decir tarjeta con signos o señal visual.

El tablero más básico de Kanban está compuesto por tres columnas: “Por hacer”, “En proceso” y “Hecho”.

Si se aplica bien y funciona correctamente, serviría como una fuente de información, ya que demuestra dónde están los cuellos de botella en el proceso y qué es lo que impide que el flujo de trabajo sea continuo e ininterrumpido.

Las tarjetas Kanban visualizan las tareas de trabajo en un flujo de trabajo y le ayudan a:

- Registrar documentación clave de una tarea
- Revisar los detalles de un vistazo
- Minimizar la pérdida de tiempo
- Identificar oportunidades para colaboración

6 Roles que interactúan en el desarrollo

Diseñador del software: Nace como una evolución del analista y realiza, en función del análisis, el diseño de la solución que hay que desarrollar. Participa en la etapa de diseño.

Analista programador: Comúnmente llamado “desarrollador” domina una visión más amplia de la programación, aporta una visión general del proyecto más detallado, diseñando una solución más amigable para la codificación y participando activamente en ella. Participa en las etapas de diseño y codificación.

Programador: Se encarga de manera exclusiva de crear el resultado del diseño realizado por analistas y diseñadores. Escribe el código fuente del software. Participa en la etapa de codificación.

Arquitecto del software: Es la argamasa que cohesiona el proceso de desarrollo. Conoce e investiga:

- **Frameworks:** Estructura de soporte definida con módulos software concreto (soporte de programas, bibliotecas, lenguajes de scripting) que puede servir de base para la organización, desarrollo y unión de diferentes componentes del software.
- Tecnologías

Revisa que todo el procedimiento de desarrollo software se lleva a cabo de la mejor forma posible y con los recursos más apropiados. Participa en las etapas de análisis, diseño, documentación y explotación.