



El manifiesto ágil: cada persona y su equipo planifican en conjunto

Los métodos ágiles en el desarrollo de software surgieron a partir de principios de los años noventa, y empezaron a ganar popularidad en 1999 con el *Extreme Programming* (Kent Beck). Los métodos ágiles han ido adquiriendo mucha más importancia en los últimos años. Esto se aplica sobre todo – aunque ya no exclusivamente – al sector de TI: el 62% de los proyectos de desarrollo de software en Alemania trabajan en parte de forma ágil, el 16% lo hace incluso de forma continua, mientras que solo el 22% de los proyectos trabajan exclusivamente con métodos convencionales.

La filosofía ágil subraya la capacidad de cada persona y la autoorganización de los equipos de desarrollo. Mediante «métodos ágiles» se intentan eliminar las deficiencias de los modelos de desarrollo de software anteriores, que se consideran ya demasiados rígidos. Dado que una planificación rígida y continua ya no es practicable, los requisitos para el software no se fijan desde el principio, sino que se desarrollan a medida que se va entendiendo el sistema de software. Esto debe lograrse sobre todo a través de una intensa comunicación con los clientes. Para facilitarla, el Extreme Programming utiliza, entre otros, los llamados «User Stories», relatos de usuarios para describir las características del sistema o las funciones principales. La planificación ha de realizarse en cooperación y sin decisiones separadas del gerente del proyecto. La cooperación de un grupo para planificar el volumen del próximo lanzamiento o distribución (de software) se denomina «Planning Game». Los *Planning Games* y la estimación de los gastos de proyecto se repiten cíclicamente (proceso iterativo).

En una reunión en Utah en 2001 se eligió el término *ágil* para este tipo de desarrollo de software y se redactó el «manifiesto ágil» (www.agilemanifesto.org/iso/de/manifesto.html): **«Exploramos caminos mejores para desarrollar el software haciéndolo nosotros mismos y ayudando a otros a hacerlo. Con estas actividades hemos aprendido a apreciar estos valores:**

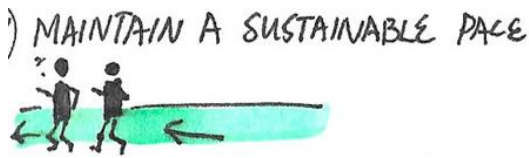
- **Los individuos y las interacciones** son más importantes que los procesos y herramientas
- **Un software operativo** es más importante que una documentación exhaustiva
- **La colaboración con el cliente** es más importante que las negociaciones contractuales
- **Reaccionar a los cambios** es más importante que seguir un plan.»

Los *Principios Ágiles* son un hilo conductor para el trabajo ágil. A veces los principios ágiles también se denominan como método. **En el *Manifiesto Ágil* se enumeran doce principios.**¹

- Satisfacción del cliente mediante entregas tempranas y continuas de software útil
- Los procesos ágiles aprovechan los cambios (incluso en una fase avanzada de desarrollo) para dar una ventaja competitiva al cliente.
- Entrega de software operativo en periodos regulares, preferentemente en periodos de tiempo cortos (pocas semanas o meses)
- Colaboración casi diaria de expertos y diseñadores durante el proyecto (p.ej.: posesión de un código común o Collective Code Ownership)
- Ofrecer el entorno y el apoyo que requieren los individuos motivados para cumplir sus funciones
- Transmisión de información, a ser posible, en una conversación cara a cara
- Funcionamiento del software como medida de progreso más importante
- Respetar un tiempo de trabajo constante de clientes, diseñadores y usuarios para lograr un desarrollo sostenible

- Atención constante a la excelencia técnica y al buen diseño
- La simplicidad es esencial
- Autoorganización de los equipos en la planificación y realización
- Autorreflexión de los equipos sobre la propia conducta adaptativa respecto a la mejora de la eficiencia

El principio del ritmo sostenible: sin horas extra



Uno de los doce principios ágiles es el del ritmo sostenible (sustainable pace). [Kent] Beck conoce la „40-hour week“ (semana de 40 horas) como una de las doce prácticas del Extreme Programming. Las horas extra son un indicio de problemas serios en el proyecto que han de ser

localizados y solucionados. Y debido a esas horas extra constantes, los diseñadores apenas pueden codificar sencilla y claramente. Se vuelven improductivos y programan más errores. Roman Pichler, como personalidad destacada, describe algo similar: En *Scrum* – el método ágil más difundido – no se aceptan las horas extra periódicas.

El »desarrollo de software ágil es un término genérico para una serie de métodos y practicas, que se basan en los valores y principios del Manifiesto de Desarrollo de Software Ágil.« (Agile Alliance, 2018)

En **Scrum**, el método ágil más difundido, existen las funciones *Product Owner*, *Team* y *Scrum Master*. El Product Owner toma la perspectiva del cliente y describe en el llamado *Product Backlog* los requisitos que han de cumplirse para conseguir el objetivo del proyecto. Este documento se elaborará de manera continuada y en él se priorizarán los requisitos según el progreso total.

Los requisitos se establecerán en el llamado sprint (2-4 semanas), que siempre tiene la misma duración. En el *Sprint Planning*, el equipo de diseñadores se pone de acuerdo con el Product Owner sobre los requisitos. El Team elegirá las tareas que tratará en el próximo sprint. Estimaré el trabajo que requieren, para garantizar que sean factibles en el periodo elegido del Sprint.

Los resultados del sprint se presentarán al Product Owner (cuando proceda también a clientes/usuarios) en el *Sprint Review*. Solo funcionalidades probadas y “operativas” se considerarán como realizadas.

El equipo se autoorganiza completamente para todo el proceso y ha de disponer de los recursos y las capacidades correspondientes. En retrospectiva, después del sprint, el Team reflexionará sobre el proceso laboral y la cooperación, para determinar medidas de mejora. El Scrum Master, como experto del método, apoyará al Team para que pueda cumplir con las reglas fijadas por él mismo y eliminar los obstáculos.

Dr. Nadine Müller, ver.di (DE)