

Tarea Ayudantía ISW

Carla Arteaga
Paula Lineros
Juan Carlos Pérez

0.1. Metodología Ágil: Scrum

El desarrollo Ágil es una respuesta metodológica a las dificultades del actual desarrollo de Software, con el fin de organizar de manera efectiva y lograr un mejor resultado del software proporcionado.

Las Metodologías Ágiles se basan en un desarrollo iterativo e incremental, teniendo en cuenta un orden, por el cual los requerimientos y las soluciones progresan, por lo que disminuye el tiempo entre iteraciones.

En cada iteración debe existir: Una planificación, Análisis de Requerimientos, Diseño, Codificación, Revisión y Documentación. Al término de cada iteración se debe reevaluar las precedencias del proyecto.

Scrum es una metodología Ágil. En un tipo de referencia, que tiene una serie de prácticas y roles, del cual se puede partir, para definir el proceso que se llevará a cabo durante el proyecto. Los roles principales en Scrum son el Scrum-Master, que trabaja de forma similar al Jefe de proyecto, el ProductOwner, que representa a los stakeholders (interesados externos o internos), y el Team que incluye a los desarrolladores.

Existen varias implementaciones de sistemas para gestionar el proceso de Scrum, que van desde notas amarillas post-it y pizarras hasta paquetes de software. Una de las mayores ventajas de Scrum es que es muy fácil de aprender, y requiere muy poco esfuerzo para comenzarse a utilizar.

En pocas palabras, Scrum:

1. Divide su organización en pequeños equipos interfuncionales y autoorganizados.
2. Divide su trabajo en una lista de productos pequeños y concretos. Ordenar la lista por orden de prioridad y estimar el esfuerzo relativo de cada elemento.
3. Divide el tiempo en corto de longitud fija iteraciones (normalmente 1-4 semanas), con código potencialmente entregable demostrado después de cada iteración.
4. Sobre la base de los conocimientos adquiridos mediante la inspección de la liberación después de cada iteración, optimiza el plan de lanzamiento y actualización de las prioridades de la colaboración con el cliente.
5. Optimiza el proceso por el que tiene una retrospectiva después de cada iteración.

0.2. Metodología Tradicional

CMMI es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software. En la actualidad hay dos áreas de interés cubiertas por los modelos de CMMI: Desarrollo y Adquisición.

La versión actual de CMMI es la versión 1.2. Existen, a su vez, tres sabores dentro de la versión disponible:

CMMI para Desarrollo (CMMI-DEV o CMMI for Development), que trata sobre los procesos de desarrollo de productos y servicios de software.

CMMI para la adquisición (CMMI-ACQ o CMMI for Acquisition), que trata sobre la gestión de la cadena de suministro, adquisición y contratación externa en los procesos del gobierno y la industria.

CMMI para servicios CMMISVC o CMMI for Services actualmente en borrador está diseñado para cubrir todas las actividades que requieren gestionar, establecer y entregar servicios. Dentro del sabor CMMI-DEV, existen dos modelos:

CMMI-DEV CMMIDEV IPPD Integrated Product and Process Development. Las prácticas de CMMI deben adaptarse a cada organización en función de sus objetivos de negocio.

Las organizaciones no pueden ser certificadas CMMI. Por el contrario, una organización es evaluada usando un método de evaluación conocido como SCAMPI y recibe una calificación (o acreditación) de nivel 2 a 5 si sigue los niveles de madurez; se comienza ya estando en el nivel 1.

También es posible que la organización seleccione áreas de proceso y en vez de por niveles de madurez puede obtener los niveles de capacidad en cada una de las áreas de Proceso obteniendo el "Perfil de Capacidad" de la Organización.

El modelo CMMI v1.2 CMMIDEV contiene las siguientes 22 áreas de proceso:

1. Análisis de Causas y Resolución (CAR)
2. Gestión de la configuración (CM)
3. Análisis de Decisiones y Resolución (DAR)
4. Gestión Integrada de Proyectos (IPM)
5. Medición y Análisis (MA)
6. Innovación y Despliegue Organizacional (OID)
7. Definición de procesos organizacionales (OPD)
8. Enfoque Organizacional en Procesos (OPF)
9. Rendimiento de Procesos Organizacionales (OPP)
10. Formación Organizacional (OT)
11. Monitorización y Control de Proyecto (PMC)
12. Planificación de proyecto (PP)
13. Aseguramiento de calidad de Procesos y Productos (PPQA)
14. Integración de Producto (PI)

- 15. Gestión Cuantitativa de Proyectos (QPM)
- 16. Gestión de Requerimientos (REQM)
- 17. Desarrollo de Requerimientos (RD)
- 18. Gestión de Riesgos (RSKM)
- 19. Gestión de Acuerdos con Proveedores (SAM)
- 20. Solución Técnica (TS)
- 21. Validación (VAL)
- 22. Verificación (VER)

0.3. Tabla Comparativa

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo
Grupos pequeños (menor a 10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos mediante reuniones
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Cuadro 1: Tabla comparativa metodologías ágiles y tradicionales