

Resumen del texto Metodologías Ágiles de Desarrollo Software

Erick de Jesus Hernández Cerecedo

21 de septiembre de 2019

1. Introducción

El siguiente documento contiene un resumen del texto “Metodologías Ágiles de Desarrollo de Software Aplicadas a la Gestión de Proyectos Empresariales” (López G, 2015) de donde se rescatan los temas más destacados.

2. Gestión de Proyectos Informáticos Empresariales

Un Proyecto informatico es un sistema de cursos de acciones simultaneas y/o secuenciales que involucran personas, equipamientos de hardware, software y comunicaciones, enfocadas en obtener uno o más resultados deseables sobre un sistema de informacion. Las organizaciones o empresas emplean este tipo de proyectos con el fin de generar diferenciadores respecto a sus competidores y la forma más eficiente es implementar un sistema de gestión de proyectos.

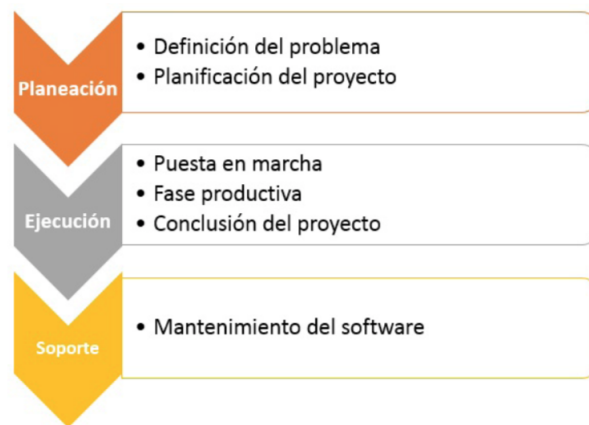


Figura 1: Fases Principales de un Proyecto de Desarrollo de Software

Clasificación de los proyectos informáticos:

- Software

- Hardware
- Comunicaciones y Redes
- Instalaciones de Hardware
- Auditoria, etc.

3. Fases Principales de un Proyecto de Desarrollo de Software

A. Planeación

En esta fase se plantean los objetivos del proyecto, considerando las tres dimensiones sobre las que se apoya todo proyecto.

- Calidad
- Costo
- Tiempo de curación

B. Ejecución

En esta fase se pone en práctica lo planeado en la fase anterior, es decir una mala planeación traerá resultados negativos a la ejecución de un proyecto.

C. Soporte

También llamada fase mantenimiento viene después de la implementación, consiste en mantener al software operando en óptimas condiciones y verificando que no existan fallas.

4. Metodologías Ágiles de Desarrollo

Son una iniciativa de un conjunto de expertos en el área de desarrollo de software con el fin de optimizar el proceso de creación del mismo, el punto de partida fue el manifiesto “Ágil”.

Manifiesto Ágil

Este documento engloba principios y valores que hacen diferente un proyecto de software ágil de uno tradicional, está regida por doce principios que ayudan a que el proceso se vuelva menos complejo y responda de manera oportuna a los cambios que surjan, siempre contando con el punto de vista del cliente.

Según el manifiesto se valora:

- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- Desarrollar software que funcione más rápido que la documentación del mismo.



Figura 2: Proceso Ágil de Desarrollo de Software

- La colaboración con el cliente más que la negociación de su contrato.
- Responde a los cambios más que seguir con el plan establecido.

Principales Metodologías Ágiles

A. Scrum

Scrum se basa en la teoría de control de procesos empírica, que asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Existen tres pilares fundamentales que soportan el control del proceso empírico los cuales son:

- Transparencia
- Inspección
- Adaptación

La metodología *Scrum* describe cuatro eventos importantes que componen cada una de las entregas:

- *Reunion de planificación del sprint (Sprint Planing Meeting)*
- *Scrum Diario (Daily Scrum)*
- *Revision del Sprint (Sprint Review)*
- *Retrospectiva del Sprint (Sprint Retrospective)*

Scrum se centra en la división del trabajo complete (Product Backlog) en distintos apartados o bloques que pueden ser abordados en periodos cortos de tiempo (1 -4 semanas), los cuales son denominados Sprint.

B. Extreme Programming (XP)

La programación extrema es una metodología que se basa en una serie de reglas y principios que se han utilizado a lo largo de toda

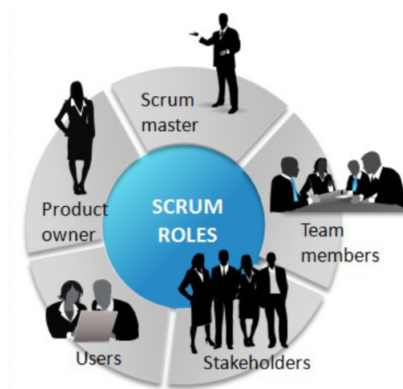


Figura 3: Equipo de Trabajo de la Metodología Scrum

la historia del desarrollo del software, en el que cada una de ellas se aplican conjuntamente, se le énfasis a las tareas que agreguen valor de manera que creen un proceso ágil.

Se engloba en 12 principios básicos los cuales se agrupan en cuatro categorías grandes:

- *Retroalimentación a Escala Fina*, en esta fase se encuentran diversos principios, realización de pruebas, proceso de planificación en parejas.
- *Proceso Continuo en lugar de por lotes*, permite la integración continua, refactorización y entregas pequeñas.
- *Entendimiento compartido*, en esta categoría se definen criterios como el de crear un diseño fácil, las tarjetas CRC y la creación de la metáfora del sistema o historia completa.
- *Bienestar del programador*, se rige por la filosofía que un programador cansado, exhausto crea código de mala calidad, por eso se recomienda que los desarrolladores tengan 40 horas de trabajo a la semana y muy pocas horas de trabajo extra.

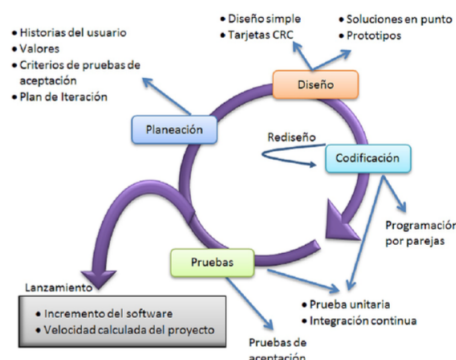


Figura 4: Macro de trabajo de la metodología XP.

C. Crystal Clear

Crystal es una metodología en la cual se establecen códigos de color como parte de la definición de la complejidad de la misma, si es más oscuro entonces el método es más pesado, todo esto en función de su criticidad y tamaño.

Lista de riesgos potenciales:

- **C:** Pérdida de confort debido a un fallo del sistema.
- **D:** Pérdida de dinero discrecional (nuestro dinero)
- **E:** Pérdida de dinero esencial (Este es el dinero del cual no se puede disponer).
- **C:** Pérdida de vidas por el fallo del sistema.

Los número indican la cantidad de personas que son coordinadas en el proyecto, de acuerdo a lo siguiente:

- Clear es para equipos de 8 personas o menos.
- Amarillo para equipos de 10 - 20 personas.
- Naranja para equipos de 20 - 50 personas.
- Rojo es para equipos de 50 - 100 y así sucesivamente pasando por el marrón y violeta.

L6	L20	L40	L80
E6	E20	E40	E80
D6	D20	D40	D80
C6	C20	C40	C80
Clear	Yellow	Orange	Red

Figura 5: Criticidad de la metodología Crystal.

5. Plataformas y Arquitecturas

De la misma forma en que se cuenta con diversas metodologías de ágiles de desarrollo de software, se cuenta con plataformas en las que se pueden ejecutar, algunas de estas herramientas traen una versión de prueba y otra de paga además de herramientas en la nube.

OpenProject

IceScrum

TeamWork

X Planner

Agile Mantis

6. Metodologías Tradicionales VRS Ágiles

La incorporación de nuevas tecnologías y formas de llevar a cabo el proceso de desarrollo de software ha venido revolucionando de tal manera que se han dejado los procesos largos y documentación exhaustiva, por métodos más enfocados en el cliente y en el equipo de desarrollo.

Aspectos	Ágil	Dirigido por Modelos
Personas	Alta prioridad; se facilita relación cliente desarrollador	No prioritario: El modelo del espacio del problema es la base de la discusión entre cliente-desarrollador
Proceso	Prioridad media; Incremental y evolutivo	Tiende al proceso en cascada; poco incremental
Tecnología	Baja prioridad: Solo cobra importancia al final	Es relevante; se usa para la generación del software (usando un PSM)
Modelos	Artefacto secundario; Se producen cuando es absolutamente necesario	Artefacto prioritario; Fuente de la implementación
Software	Artefacto prioritario; Es la única medida de progreso	Artefacto secundario; depende del espacio de la solución

Cuadro 1: Tabla comparativa Método Ágil vs Tradicional