

# Metodologías Ágiles de Desarrollo de Software Aplicadas a la Gestión de Proyectos Empresariales

*Rina Elizabeth López Menéndez de Jiménez*

*Inga. de Sistemas Informáticos, docente Escuela de Ingeniería en Computación, ITCA-FEPADE Sede Central. Email: rina.lopez@itca.edu.sv*

## Resumen

El presente artículo se centra en el conocimiento de las diferentes metodologías ágiles de desarrollo de software utilizadas para llevar a cabo un proyecto empresarial de una forma eficiente, optimizando el uso de los recursos y logrando obtener resultados que cumplan con los requisitos planteados en sus orígenes por los clientes.

## Palabras clave

Metodologías ágiles, programación extrema (XP), sistemas de computación, ingeniería del software, proyectos de desarrollo informático, redes de información, procesos de software.

## Abstract

This article focuses on the knowledge of the various agile software development methodologies used to carry out a business project in an efficient way, optimizing the use of resources and managing to obtain results that meet the requirements originally raised by the clients.

## Keywords

Agile methodologies, extreme programming (XP), computer systems, software engineering, software development Project, information networks, software processes.

## Introducción

Hace varios años las empresas desarrolladoras de software creían que la parte más importante a la hora de construir una solución era contar con un modelado eficiente y las últimas herramientas CASE que existían en el mercado, pero con el paso del tiempo entendieron que esto no era suficiente si no se contaba con un buen desarrollo del proyecto, el cual asegurara un software de calidad y satisfacción del cliente. Debido a esto ahora hoy en día ha nacido un creciente interés por las metodologías de desarrollo de software que agilicen el tiempo de desarrollo y garanticen el uso eficiente de los recursos, aplicadas tanto para empresas grandes con numerosos procesos como a empresas pequeñas que no cuentan con muchas herramientas para llevar a cabo los proyectos.

Ante este escenario, las organizaciones cada vez están optando por la incorporación de metodologías ágiles de desarrollo de software que completen los vacíos metodológicos que las otras formas de desarrollo también llamadas “*metodologías pesadas*” dejaban en la ejecu-

ción de proyectos. Las metodologías ágiles surgen de una iniciativa en el año 2001, tras una reunión celebrada en Utah, EEUU con la participación de un grupo de 17 expertos en el área de desarrollo de software, los cuales manifestaron la importancia que el equipo desarrollador respondiera de forma oportuna a los cambios que puedan surgir a lo largo de la ejecución del proyecto. [5]

## GESTIÓN DE PROYECTOS INFORMÁTICOS EMPRESARIALES

Las organizaciones que quieran generar efectos diferenciadores con respecto a sus competidores deben implementar software en el desarrollo de sus actividades de negocio, todo esto a través de la gestión de proyectos.

*Un Proyecto Informático* es un sistema de cursos de acciones simultáneas y/o secuenciales que incluye per-

sonas, equipamientos de hardware, software y comunicaciones, enfocadas en obtener uno o más resultados deseables sobre un sistema de información. [1]

Existen diferentes tipos de clasificaciones de los proyectos informáticos, entre los cuales se pueden mencionar:

- Software
- Hardware
- Comunicaciones y Redes
- Instalaciones de Hardware
- Auditoria, etc. [1]

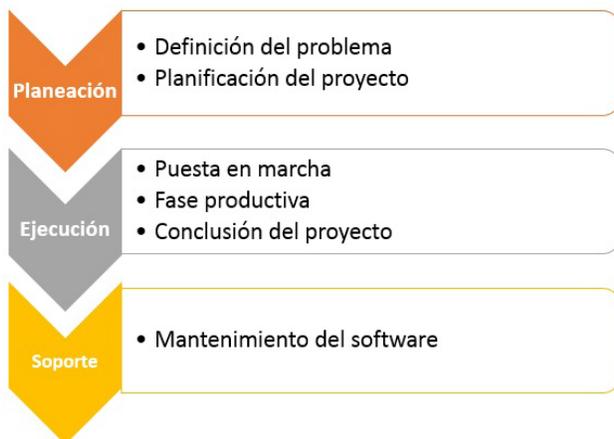


Fig. 1. Fases Principales de un Proyecto de Desarrollo de Software.

### A. Planeación

En esta fase se tienen que plantear los objetivos del proyecto identificando todos los recursos necesarios para poderlos alcanzar. Considerando las 3 dimensiones sobre las que se apoya todo proyecto.

- Calidad
- Costo
- Tiempo de duración

### B. Ejecución

Durante esta fase se trata de poner en práctica lo planeado en la fase anterior, la ejecución se verá fuertemente influida por la planeación, es decir una mala planeación traerá resultados negativos a la fase de ejecución.

### C. Soporte

La fase de soporte o mantenimiento es la que viene después de la implantación y consiste en mantener funcional el sistema informático, operando en óptimas

condiciones y siempre verificando que no existan posibles fallas.

## METODOLOGÍAS ÁGILES DE DESARROLLO

El término ágil surge como 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 cual era caracterizado por ser rígido y con mucha documentación. [5]. El punto de partida fue el manifiesto ágil, el cual es un documento donde se detalla todo lo que involucra la filosofía “ágil”.

### Manifiesto Ágil

Este es un documento que engloba principios y valores que hacen diferente un proyecto de desarrollo de software ágil de uno en su forma tradicional.



Fig. 2. Proceso Ágil de Desarrollo de Software. [6]

Según el manifiesto ágil se valora a:

- ✓ Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.
- ✓ Desarrollar software que funcione más que la documentación del mismo.
- ✓ La colaboración con el cliente más que la negociación de su contrato.
- ✓ Responde a los cambios más que seguir con el plan establecido.[5]

Esta metodología ágil está regida además por doce principios que ayudan a que el proceso de desarrollo se vuelva menos complejo y responda de manera oportuna a los cambios que surgen a lo largo del mismo, siempre contando con el punto de vista del cliente.

## PRINCIPALES METODOLOGÍAS ÁGILES

### A. Scrum

Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Esta metodología emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo, realiza entregas del proyecto en sí.

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 Planning Meeting)*
- *Scrum Diario (Daily Scrum)*
- *Revision del Sprint (Sprint Review)*
- *Retrospectiva del Sprint (Spring Retrospective)[7]*

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

### Equipo de desarrollo de Scrum

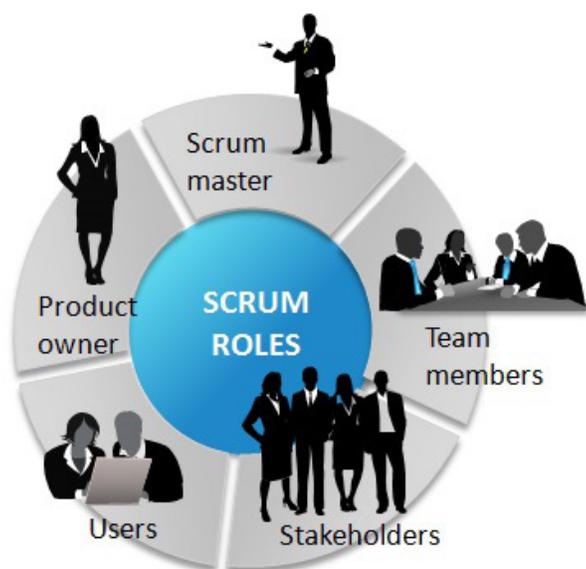


Fig. 3. Equipo de Trabajo de la Metodología Scrum. [11]

El equipo de desarrollo consiste en un conjunto de profesionales en el área que desempeñan su trabajo con el fin de proporcionar un producto terminado (Sprint), este equipo es formado de manera integral, contando con diversidad de competencias y cumple la característica de ser auto-dirigidos, sin reconocimiento de títulos, con libertad de decisión sobre las entregas.

*Scrum Master*, es un líder que está bajo el servicio del equipo scrum, este miembro ayuda al equipo y a los clientes externos a comprender las interacciones que pueden ser de ayuda y cuáles no lo son, además él es el encargado de asegurar que el equipo adopte las teorías, prácticas y reglas de la metodología scrum.

*Product Owner*, es la persona responsable de transmitir al equipo de desarrollo la visión del producto que se desea crear, aportando la perspectiva de negocio.

*Stakeholders*, conjunto de personas que no forman parte directa del proceso de desarrollo pero que si deben ser tenidos en cuenta, por ser personas interesadas en el mismo, tales como directores, gerentes, comerciales etc.

*Usuarios*, al igual que los Stakeholders no forman parte del proceso de creación directamente (podrían estar en la fase de revisión de entregables si se considera necesario). Son los destinatarios finales de la aplicación a desarrollar, el público objetivo del mismo. [12]

*Equipo de desarrollo*, equipo responsable de desarrollar y entregar el producto. Mantiene una organización horizontal en la que cada miembro del equipo se auto-gestiona y organiza libremente en la definición y ejecución de los distintos sprints.

### 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 la historia del desarrollo de software, aplicando conjuntamente cada una de ellas de manera que creen un proceso ágil, en el que se le dé énfasis a las tareas que agreguen valor y quiten procedimientos que generan burocracia en el mismo.

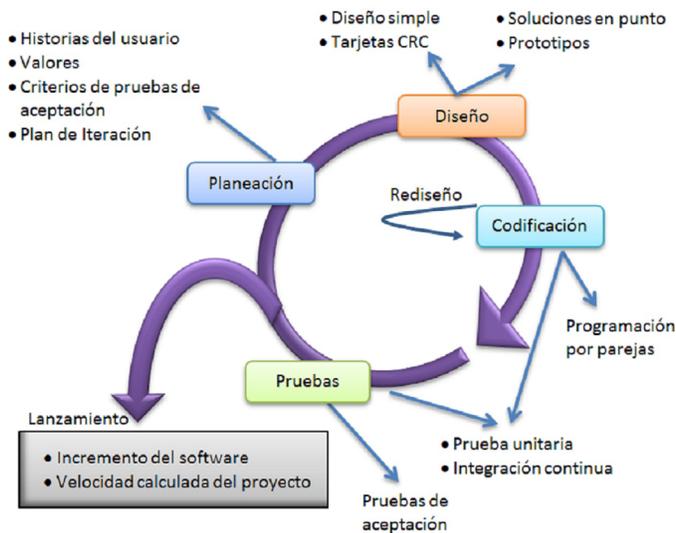


Fig. 4. Marco de trabajo de la metodología XP. [13]

La programación extrema se engloba en doce principios básicos, los cuales a su vez se agrupan en cuatro categorías grandes, entre ellas se pueden mencionar:

- ✓ *Retroalimentación a Escala Fina*, en esta fase se encuentran diversos principios como los de realización de pruebas, proceso de planificación, el cliente en el sitio y programación en parejas.
- ✓ *Proceso Continuo en lugar de por lotes*, permite la integración continua, refactorización (Evaluar el diseño del sistema a los largo de todo el proyecto y codificar si es necesario) y entregas pequeñas.
- ✓ *Entendimiento compartido*, en esta categoría se definen criterios como el de crear un diseño fácil, las tarjetas CRC (Clase, Responsabilidad y Colaboración) 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 extras de trabajo.

### 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; cuánto más crítico es el sistema más rigor se necesita. Además cristal sugiere que se defina un color para cada proyecto en función de su criticidad y tamaño. No existe una metodología cristal en general, sino existe una metodología cristal para cada tipo de proyecto. Tal como se muestra en la siguiente figura.

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

Fig. 5. Criticidad de la metodología Crystal [8]

Cada una de las letras que se muestran en la figura anterior hace referencia a los 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)
- L:** pérdida de vidas por el fallo del sistema [8]

A su vez los números 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* para equipos de 50-100 y así sucesivamente pasando por el marrón y violeta.[8]

Como las metodologías descritas anteriormente Crystal es regida por principios que hacen su utilización eficiente, entre los cuales se pueden mencionar:

- ✓ Cada proyecto necesita un grado diferente de compensación (Trade Off).
- ✓ Entre más pequeño sea el proyecto es mejor la forma de coordinación.

- ✓ Cada uno de los proyectos necesita diferentes medios de comunicación.
- ✓ Debe existir retroalimentación y comunicación efectiva, reduciendo así los problemas en entregas fallidas.
- ✓ Los puntos dulces aceleran el desarrollo, se debe de contar con personas capaces y dedicadas a su trabajo, que se preocupen por hacer las entregas a tiempo y conforme a lo que el cliente solicita.

## Roles y Productos en Crystal Clear

En Crystal Clear existen ocho diferentes roles asignados a los miembros del equipo (*sponsor ejecutivo, usuario experto, diseñador líder, diseñador-programador, experto del negocio, coordinador, tester, writer*), los primeros cuatro roles deberían ser asignados a personas diferentes, mientras que los últimos se podrían asignar a personas dentro del mismo proyecto.

**Tabla 1. Roles y Productos en Crystal. [9]**

Roll- Último Responsable	Productos
<b>Sponsor (patrocinador, quien financia)</b>	La declaración de la Misión con el Trade-off de Prioridades.
<b>Equipo</b>	La estructura y las convenciones del equipo. Los resultados del trabajo de reflexión.
<b>Coordinador, con ayuda del equipo</b>	El Mapa del Proyecto, El Plan de Entrega, El Estado del Proyecto, La Lista de Riesgo, El Plan y Estado de la Iteración, La visualización del Calendario-Cronograma.
<b>Experto del negocio y usuario experto juntos</b>	La lista de los objetivos por actor: Los Casos de Uso. El archivo de Requerimientos: El modelo del rol del usuario.
<b>Líder de diseño (diseñador líder)</b>	La descripción de la Arquitectura
<b>Diseñadores-programadores (incluyendo al líder de diseño)</b>	Borradores de pantalla, Modelo de Dominio Común, Esquemas y notas de diseño, Código fuente, Código de Migración, Las Pruebas, El sistema empaquetado.
<b>Tester</b>	Reporte de errores en ese momento

## PLATAFORMAS Y ARQUITECTURAS

Así como contamos con diferentes metodologías ágiles de desarrollo de software, se cuentan también con diferentes plataformas en las cuales se puedan ejecutar, siendo las principales:

*OpenProject*: incluye gestión de timeline, workpackacges, gestión documental, Wikis, News y Foros para colaboración y existen plugins para gestión de costos o metodologías ágiles como SCRUM.

*IceScrum*: como su nombre sugiere está enfocada a soportar la metodología ágil Scrum. Sprints, Product Management, Release Plan, Sandbox, Timeline o Cuadros de Mandos, esta herramienta cubre con cierta holgura la mayor parte de las prácticas ágiles. Hay una versión gratuita y una versión extendida de pago.

*TeamWork Project*: herramienta online para gestionar distintos proyectos y equipos de trabajo. Accesible desde móviles. Ofrece un sistema de ticketing (issue tracking), scheduling, gestión y planificación de recursos, gestión documental, y facilidades para metodologías ágiles como por ejemplo: Scrum y Kanban. [10]

*X Planner*: X-Man (Extreme Manager) es una herramienta fácil para gestión y seguimiento de proyectos ágiles. Si trabajas con un formato ágil tipo XP o Scrum, merece la pena echarle un vistazo, porque además es un programa “limpio”.

*Agile Mantis*: es un plugin utilizado en el programa MantisBT y este trae todos los componentes necesarios para desarrollar la metodología Scrum como pedidos pendientes de productos, historias de usuarios, sprint, etc.

Algunas de las herramientas descritas anteriormente traen una versión de prueba y otra de paga, además se cuentan con herramientas en la nube.

## 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 manera tal que se han dejado de lado los largos procesos y documentación exhaustiva, por métodos más enfocados en el cliente y en el equipo de desarrollo. Basándose en este enfoque

se realiza una comparación de la forma antigua de realizar el proceso versus el actual. A continuación se muestra una tabla resumen con aspectos importantes de ambas metodologías.

**Tabla 2. Aspectos diferenciadores entre procesos ágiles y dirigidos por modelos. [5]**

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

Según un análisis realizado en los buscadores de mayor presencia en Internet, se encontraron las preferencias de metodologías de desarrollo:

**Tabla 3. Metodologías con mayor presencia en la red. [6]**

	1	2	3	4	5
Google	SCRUM (3420000)	XP (1190000)	TDD(492000)	CM(244000)	APM(170000)
Yahoo	SCRUM (5120000)	XP(4470000)	CM(2930000)	TDD(2800000)	APM(766000)
Live	SCRUM(1970000)	XP (1470000)	TDD(1040000)	CM(724000)	AM(538000)

## Conclusiones

Se realizó un estudio y análisis de las diferentes metodologías de desarrollo de software utilizadas en la actualidad y que forman parte esencial en los proyectos empresariales; los cuales se han ido innovando para cubrir de manera eficiente los requerimientos de los clientes; a la vez se hizo en análisis de las metodologías antiguas orientadas a la documentación versus las metodologías ágiles más enfocadas en las personas y en actividades que agregan valor.

De igual forma, se llevó a cabo un estudio de las diferentes herramientas de software libre y de paga que se tienen para la implementación de metodologías ágiles.

## Referencias

### LIBROS

[1] G. Maigua, E. López, Buenas prácticas en la dirección y gestión de proyectos informáticos, Argentina: Editorial de la Universidad Tecnológica Nacional – edUTecNe, 2012, p.13.  
 [2] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley, Pearson Education, 2000.  
 [3] R. Pressman, Ingeniería de software: Un enfoque práctico, (3rd. Ed.) México: Mc Graw-Hill, 1993.  
 [4] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile software development methods. Review and analysis, Finlandia: VTT, 2002.

### REPORTES TÉCNICOS

[5] P. Torres, E. López, “Metodologías ágiles en el desarrollo de software,” Alicante, Nov. 2003.  
 [6] A. Palacios, V. Merchán, “Guía de fundamentos para la dirección de proyectos de desarrollo de software con enfoque pmi y los métodos ágiles,” Ecuador, 2014.  
 [7] K. Schwaber, J. Sutherland, “La guía de Scrum,” Nov.2013.

[8] S. G. Rivadeneira, “Metodologías ágiles enfocadas al modelado de requerimientos,” Unidad Académica Rio Turbio - Universidad Nacional de la Patagonia Austral, Argentina, May. 2012.  
 [9] L. Gimson, “Metodologías ágiles y desarrollo basado en el conocimiento,” Universidad Nacional de La Plata, Argentina, Jun. 2012.  
 [10] M. Villanueva. (2014, Abr.). Quality and Technology. Disponible: <http://www.calidadytecnologia.com/2014/04/herramientas-Open-Source-Gestionar-Proyectos.html>  
 [11] J. J. Grajeda. (2012, Feb.). ¿Qué es Scrum?, Disponible: <http://aprendiendosoftware.blogspot.com.es/2012/02/que-es-scrum.html>  
 [12] V. Digital (2012, Sept.). Scrum la metodología de desarrollo ágil por excelencia, Disponible: <http://vassdigital.com/blog/scrum-la-metodologia-de-desarrollo-agil-por-excelencia/>  
 [13] Rojas (2013, Jun.). Programación Extrema, CodeJobs, Disponible: <http://www.codejobs.biz/es/blog/2013/06/05/programacion-extrema-xp#sthash.7z7S2a2S.dpbs>