



Universidad de Valladolid



ESCUELA DE INGENIERÍAS
INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería de Organización Industrial

Estudio comparativo de metodologías tradicionales y ágiles para proyectos de Desarrollo de Software

Autor
López Gil, Alba

TUTOR

Pajares Gutiérrez, Javier

Departamento de Organización de Empresas y CIM

Valladolid, Septiembre 2018

RESUMEN

En las últimas décadas, el avance de las tecnologías de información y los continuos cambios que se producen, hacen plantearse si los métodos utilizados hasta ahora, denominados tradicionales (PMP, IPMA, PRINCE2, ISO 21500, etc), son los más adecuados para dirigir los proyectos de desarrollo de software. De este planteamiento surgieron las metodologías ágiles, enfocadas a la adaptabilidad a los cambios, con equipos auto-organizados, autonomía a la hora de tomar decisiones e involucramiento del usuario o cliente, el equipo y los interesados. Aunque cada proyecto tenga unas características y unas necesidades propias, se pueden aplicar diferentes métodos o tener en cuenta diferentes factores para tener más probabilidad de que sean finalizados con éxito. De manera concreta, en los proyectos de desarrollo de Software, se han encontrado varios factores que ayudan a aumentar las probabilidades de éxito, denominados FCE. Por ello, en este TFG se realiza un estudio bibliográfico de las metodologías y los FCE de proyectos de desarrollo de software, con el objetivo de realizar una comparativa y obtener qué metodología sería más apropiada dependiendo de las necesidades del proyecto.

PALABRAS CLAVE

Desarrollo de Software, Dirección de proyectos, Metodología, Ágil, Tradicional, Comparativa, factores críticos de éxito, FCE.

ABSTRACT

In the last decades, the advance of information technologies and the continuous changes that occur, raise whether the methods used so far, called traditional (PMP, IPMA, PRINCE2, ISO 21500, etc), are the most appropriate to direct software development projects. Agile methodologies emerged from this approach, focused on adaptability to change, with self-organized teams, autonomy when making decisions and involvement of the user or client, the team and interested parties. Although each project has its own characteristics and needs, different methods can be applied or different factors can be taken into account to be more likely to be completed successfully. Specifically, in software development projects, several factors have been found that help increase the chances of success, called FCE. For this reason, in this TFG a bibliographic study of the methodologies and the FCE of software development projects is carried out, with the aim of making a comparison and obtaining which methodology would be more appropriate depending on the needs of the project.

KEY WORDS

Software Development, Project Management, Methodology, Agile, Traditional, Comparative, critical success factors, FCE.

ÍNDICES

ÍNDICE

1	INTRODUCCIÓN.....	16
1.1	Antecedentes y motivación	16
1.2	Objetivo y alcance.....	16
1.3	Estructura del documento	17
2	HISTORIA DEL DESARROLLO DE SOFTWARE Y PROBLEMÁTICAS	22
2.1	EVOLUCIÓN DEL SOFTWARE.....	22
2.2	EVOLUCIÓN DE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	23
2.3	INGENIERÍA DE SOFTWARE Y PROBLEMÁTICAS	24
3	METODOLOGÍAS TRADICIONALES	28
3.1	PMI – PMBOK.....	28
3.1.1	HISTORIA Y OBJETIVOS DE PMI	28
3.1.2	CARACTERÍSTICAS DEL CICLO DE VIDA DEL PROYECTO SEGÚN LA GUÍA PMBOK	29
3.1.3	PROCESOS Y ÁREAS DE CONOCIMIENTO DE LA GUÍA PMBOK.....	30
3.2	ICB (IPMA COMPETENCIAS BASELINE)	34
3.2.1	HISTORIA Y OBJETIVOS DE IPMA	34
3.2.2	ESTRUCTURA.....	35
3.3	ISO 21500	35
3.3.1	HISTORIA Y OBJETIVO DE LA NORMA ISO 21500	35
3.3.2	ESTRUCTURA.....	36
3.3.3	BENEFICIOS DE UTILIZAR ISO 21500.....	39
3.4	PRINCE2.....	39

3.4.1	HISTORIA Y OBJETIVOS DE PRINCE2	39
3.4.2	ESTRUCTURA DE LA METODOLOGÍA	40
3.4.3	BENEFICIOS DE UTILIZAR PRINCE2	46
4	METODOLOGÍAS ÁGILES.....	50
4.1	FASES DEL ENFOQUE DE GESTIÓN DE PROYECTOS ÁGILES	52
4.2	EL MANIFIESTO AGIL	53
4.3	CARACTERÍSTICAS BÁSICAS DE LOS PROYECTOS CON METODOLOGÍAS ÁGILES 54	
4.4	SCRUM	55
4.4.1	Historia (Origenes de Scrum, Club de Tecnología, 2013)	55
4.4.2	¿Qué es SCRUM?	56
4.4.3	ESTRUCTURA.....	57
4.4.4	Fases de la metodología SCRUM	58
4.5	XP o Extreme Programming.....	60
4.5.1	Historia	60
4.5.2	¿Qué es XP?	62
4.5.3	ROLES EN XP	63
4.5.4	Estructura	65
4.6	DYNAMIC SYSTEMS DEVELOPMENT METHOD	70
4.6.1	HISTORIA.....	70
4.6.2	El marco de proyecto agil de DSDM de 2014 en adelante.....	70
4.6.3	PRINCIPIOS.....	71
4.6.4	Roles y responsabilidades.....	74
4.6.5	ESTRUCTURA.....	78

4.7	CONCLUSIÓN.....	78
5	COMPARACIÓN DE METODOLOGÍAS DE DIRECCIÓN DE PROYECTOS	82
5.1	INTRODUCCIÓN.....	82
5.2	COMPARACION ENTRE LAS METODOLOGIAS TRADICIONALES.....	83
5.2.1	VENTAJAS DE LAS METODOLOGIAS TRADICIONALES.....	90
5.2.2	DESVENTAJAS DE LAS METODOLOGÍAS TRADICIONALES	90
5.3	COMPARATIVA DE LAS METODOLOGÍAS ÁGILES.....	92
5.3.1	VENTAJAS DE LAS METODOLOGIAS AGILES	93
5.3.2	DESVENTAJAS DE LAS METODOLOGIAS AGILES	93
5.4	DIFERENCIAS ENTRE LAS METODOLOGÍAS TRADICIONALES Y ÁGILES.....	94
6	ESTUDIO DE PROYECTOS DE DESARROLLO DE SOFTWARE Y FACTORES DE FRACASO	98
6.1	ESTUDIO DE PROYECTOS DE DESARROLLO DE SOFTWARE	98
6.2	FACTORES DE FRACASO EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE 104	
7	FACTORES DE ÉXITO EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE 110	
7.1	CONCEPTO DE FCE	110
7.2	FCE DE UN PROYECTO DE SOFTWARE	113
7.3	COMPARATIVA DE LAS METODOLOGÍAS RESPECTO A LOS FCE	118

8	CONCLUSIONES	128
8.1	CONCLUSIÓN DE LOS RESULTADOS	128
8.2	FUTURAS EXTENSIONES	130
9	BIBLIOGRAFÍA	134

ÍNDICE DE TABLAS

Tabla 3-1 - Integración de los procesos en las áreas de conocimiento. Extraído de PMBOK6 (2018)	33
Tabla 3-2 - Competencias para la certificación IPMA. IPMA Versión 4.0. (2015)	35
Tabla 3-3 - Procesos ISO 21500. Extraído de HMD Project Manager (2017)	38
Tabla 5-1 - Diferencias entre las características generales entre PMI y PRINCE2. Elaboración propia.....	83
Tabla 5-2 - Diferencias entre los procesos entre PMI y PRINCE2. Extraída de Revista científica. Fernández, K., Garrido, A., Raminéz, Y., y Perdomo, I. (2015).	84
Tabla 5-3 - Diferencias entre las áreas de conocimiento entre PMI y PRINCE2. Extraída de Revista científica. Fernández, K., Garrido, A., Raminéz, Y., y Perdomo, I. (2015).....	85
Tabla 5-4 - Indicadores para la comparación entre los sistemas de certificación. Obtenida de Cardoza, R.A.,Dante, G.C.,De los Ríos,C. I.(2011).	86
Tabla 5-5 - Comparación de sistemas de certificación. Obtenido de Cardoza, R.A.,Dante, G.C.,De los Ríos,C. I.(2011).	89
Tabla 5-6 - Ventajas de las metodologías tradicionales.....	90
Tabla 5-7 - Desventajas de las metodologías tradicionales	91
Tabla 5-8 - Comparación de metodologías ágiles. Traducción extraída de Awad,MA (2005).....	92
Tabla 5-9 - Ventajas de las metodologías ágiles.....	93
Tabla 5-10 - Desventajas de las metodologías ágiles	93
Tabla 5-11 - Diferencia entre metodologías ágiles y tradicionales. Elaboración propia	94
Tabla 7-1 - FCE para un proyecto de software	114
Tabla 7-2 - Comparación de las metodologías respecto a los FCE. Elaboración propia	118

Tabla 7-3 - Comparación entre metodologías ágiles y tradicionales con respecto FCE. Elaboración propia.....	119
Tabla 7-4 - FCE. Área de procesos. Elaboración propia.....	119
Tabla 7-5 - FCE. Recursos Humanos. Elaboración propia.....	121
Tabla 7-6 - FCE. Objetivos y alcance. Elaboración propia.....	122
Tabla 7-7 - FCE. Calidad. Elaboración propia.....	123
Tabla 7-8 - FCE. Tecnología e innovación. Elaboración propia.....	124

ÍNDICE DE FIGURAS

Figura 2.1 - Modelo Waterfall. Extraído de Wikipedia (Agosto, 2018)	23
Figura 3.1 - Ciclo de vida genérico de un proyecto. Extraído de EOI (Abril, 2013)	29
Figura 3.2 - Estructura PRINCE2. Extraída de M. Ferrer (2015).....	40
Figura 3.3 - Temáticas PRINCE2. Extraída de Management Plaza (2017)	42
Figura 4.1 - Proceso Scrum. Extraído de proyectosagiles.org (2008).....	57
Figura 4.2 - Fases en SCRUM. Extraído de OBS Business School (2017)	58
Figura 4.3 - Funcionamiento del método XP. Extraído de Extreme Programming	65
Figura 4.4 - Tarjeta CRC. Extraído de Programación Orientada a Objetos y Talleres (2011).....	67
Figura 4.5 - Esquemas de roles de DSDM. Extraído de Agile Business	75
Figura 5.1 - Diferencias entre metodologías ágiles y tradicionales. Extraída de PMBOK (2017).....	94
Figura 6.1 - Resultado general. Elaboración propia	99
Figura 6.2 - Según tamaño del proyecto. Elaboración propia	100
Figura 6.3 - Comparación ágil vs cascada. Elaboración propia	101
Figura 6.4 - Comparación en proyectos pequeños. Elaboración propia	102
Figura 6.5 - Comparación en proyectos grandes. Elaboración propia	103
Figura 6.6 - Comparación de proyectos grandes y pequeños. Elaboración propia	103

INTRODUCCIÓN

1 INTRODUCCIÓN

1.1 ANTECEDENTES Y MOTIVACIÓN

Las metodologías que se llevan a cabo para gestionar un proyecto, de cualquier sector, son un conjunto de procedimientos que nos ayudan a la planificación, definición de objetivos, actuación y revisión continua de los mismos. Ayudándonos a que llegue a su fin de una manera satisfactoria minimizando los riesgos lo máximo posible.

Durante los últimos años es muy habitual el uso de dichas metodologías desde las más tradicionales a las más novedosas, como son las ágiles. Estas últimas, han nacido por los constantes cambios como consecuencia de la emergencia de las tecnologías de la información y la comunicación (TIC). Van a pasos agigantados y las metodologías que se han utilizado hasta ahora no eran posibles de aplicarse, por la rapidez con la que se producen los cambios.

Para poder obtener una comparativa de las metodologías tradicionales y las ágiles se ha planteado en un tema en concreto. De esta manera se obtendrá cuáles se adaptan mejor a este tipo de proyectos.

Se ha decidido profundizar en el tema de **Desarrollos de Software**. En la actualidad, las empresas demandan y necesitan un software eficiente para mejorar su productividad, calidad, tiempo y costes. Por lo tanto, es una razón más que suficiente para investigar sobre cómo se pueden adaptar las metodologías existentes para crear un software de calidad, de manera que se comporten como una base sólida sobre la que organizar este proyecto, para obtener un resultado satisfactorio.

Para conseguir este objetivo es importante que la metodología de dirección del proyecto sea la más adecuada según las necesidades del mismo. Por ello se ha desarrollado una comparativa de dichas metodologías a través de factores críticos de éxito divididos en diferentes áreas, con lo que se quiere obtener una ayuda para la decisión de qué metodología es la más apropiada.

1.2 OBJETIVO Y ALCANCE

El objetivo principal de este TFG es realizar una comparativa de las metodologías ágiles y tradicionales en los proyectos de desarrollo de software. Para ello se ha hecho un estudio bibliográfico de los factores críticos de éxito de este tipo de proyectos y se ha realizado la comparación a través de los mismos.

A consecuencia de llegar a este objetivo, se han desarrollado otros objetivos adicionales, como son:

- Explicar las metodologías ágiles y tradicionales, principios y características principales, estructuras y roles y responsabilidades.
- Obtener las ventajas y desventajas de las diferentes metodologías de dirección de proyectos.
- Conocer los principales errores y problemáticas de los proyectos de desarrollo de software.
- Obtener los factores de fracaso comunes a los proyectos de desarrollo de software.
- Conocer qué son los factores críticos de éxito y la obtención de los mismos.

1.3 ESTRUCTURA DEL DOCUMENTO

Este proyecto se encuentra dividido en 8 capítulos, en los cuáles se va desarrollando todos los objetivos planteados. Como apoyo a la documentación elaborada a lo largo del trabajo, al final del mismo, se puede consultar la bibliografía utilizada. Los capítulos que componen este Trabajo Fin de Grado son los siguientes:

Capítulo 1 – INTRODUCCIÓN

En este capítulo se presenta el tema a desarrollar, justificando la motivación que dio lugar a la realización del mismo. Definiendo los objetivos que se quieren desarrollar y el alcance del trabajo.

Capítulo 2 – HISTORIA DEL DESARROLLO DE SOFTWARE Y PROBLEMÁTICAS

Por un lado, se trata de manera breve cómo ha evolucionado el software a lo largo de los años, desde su comienzo hasta la actualidad. Y por otro, se habla de la evolución de las metodologías con las que se han desarrollado los proyectos de software.

Además, contiene conceptos referentes al software y su desarrollo y los problemas que se presentan al desarrollar un proyecto de este tipo.

Capítulo 3 – METODOLOGÍAS TRADICIONALES

En este capítulo se hace una descripción completa de las metodologías tradicionales presentándose las características, los procesos y la estructura.

Capítulo 4 – METODOLOGÍAS ÁGILES

En este capítulo, al ser un enfoque más actual, se recopila los principios generales de las metodologías ágiles, características básicas y las fases del enfoque de gestión de estos proyectos. A continuación, de la misma manera que en las tradicionales, se desarrollan algunas de estas metodologías más en profundidad, con sus características, estructuras, procesos a seguir y beneficios de las mismas.

Capítulo 5 – COMPARACIÓN DE METODOLOGÍAS DE DIRECCIÓN DE PROYECTOS

Se realiza una comparación de las metodologías tradicionales entre ellas, obteniendo sus ventajas y desventajas de este uso de metodologías. Después se hace lo mismo con las metodologías ágiles, comparando algunos de los métodos anteriormente expuestos entre ellos. Para más tarde obtener las diferencias entre ambos tipos.

Capítulo 6 – ESTUDIO DE PROYECTOS DE DESARROLLO DE SOFTWARE Y FACTORES DE FRACASO

Se presenta un estudio realizado en 2015 por CHAOS Report de Standish Group, que nos ofrece unos resultados estadísticos reales de los problemas que se presentan en los proyectos de desarrollo de software utilizando los métodos anteriormente desarrollados.

Tras la investigación, se descubren unos factores comunes a los proyectos que no se han completado con éxito. Esto puede ser una ayuda a la hora de saber lo que hay que evitar a la hora de llevar a cabo un proyecto de estas características.

Capítulo 7 - FACTORES DE ÉXITO EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE

En principio se define qué entendemos por Factor Crítico de Éxito (FCE), mostrando diferentes definiciones, encontradas a través de un estudio bibliográfico, planteadas por diferentes autores de los últimos años. Después se ha realizado un estudio bibliográfico de cuáles son esos factores críticos de éxito en el mundo de desarrollo de software. Por último, se ha hecho una selección de los más importantes para llevar a cabo el objetivo principal del proyecto.

En este capítulo se llega al objetivo principal definido en este TFG, se realiza una tabla comparativa de cada una de las metodologías respecto a los FCE, que están divididos en 5 áreas de trabajo, que son: Procesos, Recursos Humanos, Objetivos y Alcance, Calidad y Tecnología e innovación.

Capítulo 8 – CONCLUSIONES

Se da una conclusión general del TFG realizado, planteando de nuevo el objetivo principal del mismo y comprobando si se ha cumplido. Por otra parte, se da una conclusión de los resultados obtenidos de la tabla comparativa. Y para terminar, una evolución en el futuro.

HISTORIA DEL DESARROLLO DE SOFTWARE Y PROBLEMÁTICAS

2 HISTORIA DEL DESARROLLO DE SOFTWARE Y PROBLEMÁTICAS

2.1 EVOLUCIÓN DEL SOFTWARE

Los comienzos del software hasta el día de hoy se podría decir que se divide en cuatro eras:

Primera era: 1950 – 1965

1. Se trabaja con la idea de “Codificar y Corregir”
2. No existe un planteamiento previo
3. No existe documentación de ningún tipo
4. Pocos métodos formales y pocos creyentes en ellos
5. Desarrollo a base de prueba y error

Segunda era: 1965 – 1972

- Se busca simplificar código
- Aparición de Multiprogramación y Sistemas multiusuarios
- Sistemas de tiempo real que apoyan la toma de decisiones
- Aparición de Software como producto
- Inicio de la crisis del software
- Se buscan procedimientos para el desarrollo del software

Tercera era: 1972 – 1985

- Nuevo concepto: sistemas distribuidos
- Complejidad en los sistemas de información
- Aparecen: redes de área local y global y comunicadores digitales
- Amplio uso de microprocesadores

Cuarta era: 1985 – 2000

- Impacto Colectivo de software
- Aparecen: redes de información, tecnologías orientadas a objetos
- Aparecen: redes neuronales sistemas expertos y software de inteligencia artificial
- La información como valor preponderante dentro de las organizaciones.

Se puede hacer esa división respecto a la evolución del software y la importancia que ha ido cobrando a lo largo de los años, evolucionando a pasos agigantados.

2.2 EVOLUCIÓN DE LAS METODOLOGÍAS DE DESARROLLO DE SOFTWARE

Ahora bien, lo que se está tratando en este trabajo, son las metodologías para el desarrollo de estos proyectos. Por tanto, la evolución que ha experimentado es la siguiente (Pardo, 2015):

Una de las primeras metodologías formales de desarrollo de software, es la metodología Waterfall o cascada. Consiste en cumplir etapa por etapa. No se puede avanzar a la siguiente etapa sin haber finalizado la anterior. El cliente propone sus requisitos, se maqueta el diseño para obtener como se va a implementar y por último se verifica y se controla el mantenimiento del producto.

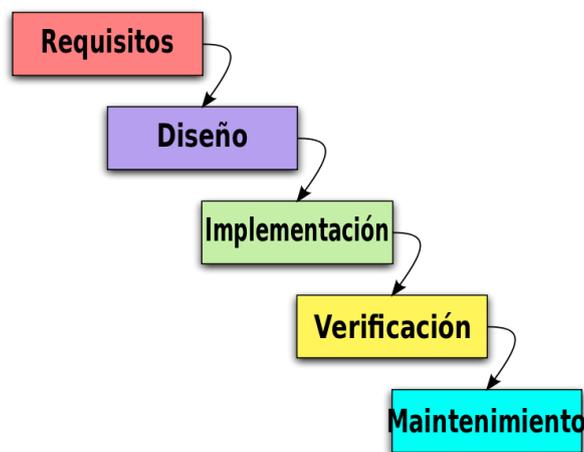


Figura 2.1 - Modelo Waterfall. Extraído de Wikipedia (Agosto, 2018)

Después apareció el Modelo Iterativo o Incremental, se originó a partir de los años 80. Algunos de estos modelos son Espiral, RAD y RUP. Consisten en incrementar sus tareas paso por paso, pero cada tarea tiene un tiempo determinado, por lo tanto hay alguna interactividad entre ellas.

En febrero de 2001, durante una reunión de expertos de la industria de software, nace el término ágil aplicado al desarrollo de software. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales.

A partir de ahí se creó *The Agile Alliance*, una organización sin ánimo de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y promover dichos conceptos por otras organizaciones. El punto de partida fue el **Manifiesto Ágil**, un documento dónde se resume toda esta filosofía.

2.3 INGENIERÍA DE SOFTWARE Y PROBLEMÁTICAS

Se entiende por **Software** al conjunto de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de computación.

La **Ingeniería de Software** es por lo tanto, la aplicación práctica, sistemática, disciplinada y cuantificable del conocimiento científico para realizar el análisis de las necesidades del usuario y obtener el software y la documentación asociada requerida para su desarrollo, operación y mantenimiento de manera rentable, fiable, certificada y que opere en máquinas reales. (Roger S. Pressman, 2010)

El proceso de desarrollo es intensamente intelectual y se ve afectado por la creatividad y juicio de las personas involucradas en el mismo. Aunque un proyecto de desarrollo de software es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de software hay una serie de desafíos adicionales relativos a la naturaleza del producto a obtener, como son la intangibilidad o la fiabilidad del mismo.

Hablamos de un producto **intangible** y normalmente complejo, dado que su cometido es dar respuesta a la abstracción de un problema planteado por personas que normalmente desconocen esta disciplina. Por esta cuestión, definir con exactitud los requisitos a cubrir y consolidarlos desde el inicio se complica con frecuencia, haciendo inevitable el cambio, durante el desarrollo o una vez acabado el mismo.

Un producto software en sí es complejo siendo inviable conseguir el 100% de **fiabilidad** en un programa por muy reducida que sea su funcionalidad. Esto ocurre por la elevada combinatoria asociada a los distintos factores que intervienen en la ejecución del mismo y que impiden una verificación de las todas posibles situaciones que se puedan presentar, son ejemplo de estos factores:

- Datos introducidos por el usuario
- Datos almacenados en el sistema
- Interacción con el software base o sistema operativo
- Interacción o el hardware del sistema sobre el que se ejecuta
- Interacción con otras aplicaciones

METODOLOGÍAS TRADICIONALES

3 METODOLOGÍAS TRADICIONALES

3.1 PMI – PMBOK

3.1.1 HISTORIA Y OBJETIVOS DE PMI

Es una de las organizaciones más grandes y reconocidas a nivel mundial formadas por profesionales de la gestión de proyectos, junto con IPMA, que se explicará más adelante. Con más de 500.000 miembros globales y más de 300 capítulos locales a nivel internacional en esta institución sin ánimo de lucro.

Cuando esta organización se fundó en 1969, lo que se buscaba principalmente era proporcionar un medio para que los gerentes de proyectos pudieran compartir experiencias y debatir sobre problemas comunes.

En la actualidad, lo que persigue esta organización es el progreso de la profesión de dirección de proyectos. Para ello, ha recopilado experiencias previas y ha ido puliendo detalles hasta conseguir unos estándares de trabajo y unas certificaciones mundialmente conocidas, herramientas e investigación académica.

Su actividad se lleva a cabo de manera voluntaria mediante profesionales certificados que están ubicados en más de 80 países, con el fin de crecer en esta profesión, conectar entre profesionales, compartir el conocimiento y la experiencia.

Se basa en la guía PMBOK (Project Management Body Of Knowledge). Es una guía que contiene una serie de acciones a llevar a cabo dentro de un proyecto y buenas prácticas que asesoran sobre la gestión de los proyectos para que éste sea finalizado con éxito. Está en un lenguaje común que es reconocido internacionalmente. Esto no significa que el conocimiento de esta guía sea aplicado siempre igual independientemente del proyecto, sino que el equipo de proyecto debe ser el que decida cómo se aplican estas prácticas adecuándose a su proyecto, basándose en el alcance, industria y cultura del proyecto.

La institución la considera un recurso fundamental para la gestión efectiva de proyectos en cualquier industria. La primera guía PMBOK se publicó en 1987. En este momento, se puede obtener la Sexta Edición, ha sido actualizada recientemente, el 6 de septiembre de 2017, para reflejar los últimos avances sobre prácticas en gestión de proyectos.

3.1.2 CARACTERÍSTICAS DEL CICLO DE VIDA DEL PROYECTO SEGÚN LA GUÍA PMBOK

Todos los proyectos tienen un inicio y un final, pero como se gestione todo lo que se realiza en ese intervalo, varía dependiendo del proyecto. Independientemente del tamaño o la complejidad del proyecto, todos pueden configurarse dentro de este esquema (Figura 3.1).

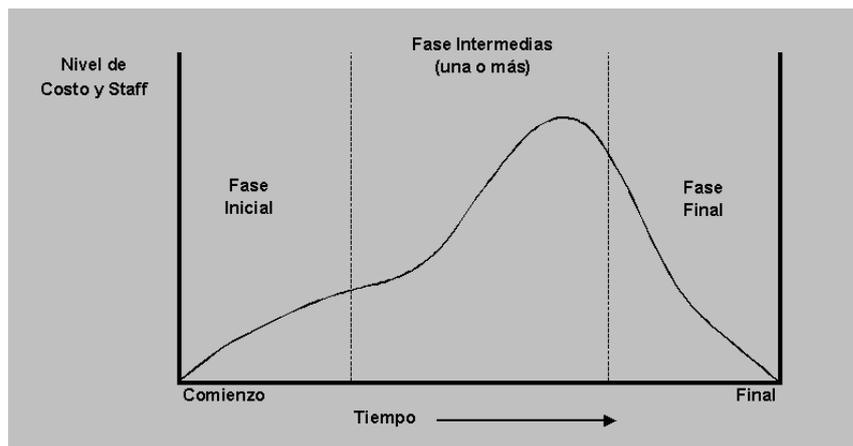


Figura 3.1 - Ciclo de vida genérico de un proyecto. Extraído de EOI (Abril, 2013)

Los ciclos de vida del proyecto generalmente **definen**:

- El comienzo y el final del proyecto
- El trabajo técnico que se debe realizar en cada fase
- Quien está involucrado en cada fase, es decir, organización del equipo y repartición de tareas
- Como se va a controlar el trabajo realizado y cuando se va a dar por aprobado para pasar a la siguiente fase. Este control se hace mediante los entregables descritos anteriormente.
- El riesgo y la incertidumbre al comienzo del proyecto es muy alta, ya que se duda mucho con cumplir los objetivos a tiempo. Una vez que el proyecto está en marcha, la certeza de terminar con éxito aumenta.
- El personal necesario al principio, siempre suele ser bajo y a medida que se va avanzando se necesita más personal para cubrir las necesidades que requiere el proyecto y en su última etapa cae rápidamente.
- El nivel de costos y riesgo de cambios y correcciones. Estos irán aumentando cuando el proyecto se vaya desarrollando.
- La influencia que pueden causar los interesados en el proyecto respecto a las características finales del producto del proyecto y en el coste final del mismo.

- En un comienzo es más alto, y decrece gradualmente a medida que avanza el proyecto.

Muchos equipos de dirección de proyectos, con la experiencia, han ido marcando las etapas en las que se divide un proyecto, y suele ser las que aplican por lo general para todos sus proyectos, es decir, dentro de un mismo sector se tiene tendencia a usar un ciclo de vida que sea usado por los demás dentro de la industria.

Aunque muchos ciclos de vida puedan parecer similares por sus nombre o requisitos del proyecto, incluso requiriendo entregas similares, muy pocos ciclos de vida son iguales. Por ejemplo, el ciclo de vida de desarrollo de software, que lo trataremos más adelante, en una organización puede tener una única fase de diseño, mientras que en otras puede tener distintas fases para el diseño arquitectónico y el detallado.

3.1.3 PROCESOS Y ÁREAS DE CONOCIMIENTO DE LA GUÍA PMBOK

La guía PMBoK se basa en 49 procesos, los cuales no hay que confundir con las etapas o fases en las que se dividirá el proyecto. A continuación, les nombraremos y les dividiremos según el ciclo de vida del proyecto. Estos procesos según la guía PMBOK son:

INICIO

1. Desarrollar el acta de constitución del proyecto.
2. Identificar a los interesados.

PLANIFICACIÓN

3. Desarrollar el plan para la dirección del proyecto.
4. Planificar el involucramiento de los interesados.
5. Planificar la gestión del alcance.
6. Recopilar los requisitos.
7. Definir el alcance.
8. Crear la EDT/WBS.
9. Planificar la gestión del cronograma.
10. Definir las actividades.
11. Secuenciar las actividades.
12. Planificar la gestión de los riesgos.
13. Identificar los riesgos.
14. Realizar el análisis cualitativo de riesgos.
15. Realizar el análisis cuantitativo de riesgos.

16. Planificar la respuesta a los riesgos.
17. Planificar la gestión de recursos.
18. Planificar la gestión de los costos.
19. Estimar los costos.
20. Estimar los recursos de las actividades.
21. Estimar la duración de las actividades.
22. Desarrollar el cronograma.
23. Determinar el presupuesto.
24. Planificar la gestión de la calidad.
25. Planificar la gestión de las comunicaciones.
26. Planificar la gestión de las adquisiciones.

EJECUCIÓN

27. Dirigir y gestionar el trabajo del proyecto.
28. Gestionar el conocimiento del proyecto (Nuevo)
29. Gestionar la participación de los Interesados.
30. Adquirir recursos
31. Desarrollar el equipo.
32. Dirigir al equipo.
33. Gestionar las comunicaciones.
34. Efectuar las adquisiciones.
35. Gestionar la calidad.
36. Implementar la respuesta a los riesgos (Nuevo)

MONITOREO Y CONTROL

37. Monitorear y controlar el trabajo del proyecto.
38. Realizar el control integrado de cambios.
39. Monitorear el involucramiento de los interesados.
40. Controlar el cronograma.
41. Controlar los costos.
42. Monitorear las comunicaciones
43. Monitorear los riesgos.
44. Controlar la calidad.
45. Controlar los recursos (Nuevo)
46. Validar el alcance.
47. Controlar el alcance.
48. Controlar las adquisiciones.

CIERRE

49. Cerrar el proyecto o fase.

Este orden puede cambiar según el proyecto, dependerá de las restricciones y de la naturaleza del mismo.

También se puede dividir en áreas de conocimiento, según PMBOK un área de conocimiento es “un área identificada de la dirección de proyectos definida por sus requisitos de conocimientos y que describe en términos de sus procesos, prácticas, datos iniciales, resultados, herramientas y técnicas que los componen”.

Son 10 áreas de conocimiento, que son:

- 1) Integración del proyecto
- 2) Gestión del alcance del proyecto
- 3) Gestión del tiempo
- 4) Gestión de los costes
- 5) Gestión de la calidad
- 6) Gestión de los Recursos humanos
- 7) Gestión de las comunicaciones
- 8) Gestión de los riesgos
- 9) Gestión de las adquisiciones
- 10) Gestión de los interesados

Cada una de estas diez áreas de conocimiento reúne los procesos y actividades necesarias para que cada una de ellas se lleve a cabo. Es decir, los 49 procesos del PMBOK se dividen en estas áreas formando parte de ellas (véase Tabla 3.1).

Tabla 3-1 - Integración de los procesos en las áreas de conocimiento. Extraído de PMBOK6 (2018)

ÁREAS DE CONOCIMIENTO	GRUPOS DE PROCESOS DE LA DIRECCIÓN DE PROYECTOS				
	INICIO	PLANIFICACIÓN	EJECUCIÓN	MONITOREO Y CONTROL	CIERRE
4. Gestión de la integración	4.1. Desarrollar el Acta de Constitución del Proyecto	4.2. Desarrollar el Plan para la Dirección del Proyecto	4.3. Dirigir y Gestionar el Trabajo del Proyecto 4.4. Gestionar el conocimiento del proyecto	4.5. Monitorear y Controlar el Trabajo 4.6. Realizar el control integrado de cambios	4.7. Cerrar Proyecto o Fase
5. Gestión del Alcance		5.1. Planificar la Gestión del Alcance 5.2. Recopilar los requisitos 5.3. Definir el Alcance 5.4. Crear la EDT/WBS		5.5. Validar el alcance 5.6. Controlar el alcance	
6. Gestión del tiempo		6.1. Planificar la gestión del cronograma 6.2. Definir las actividades 6.3. Secuenciar las actividades 6.4. Estimar la duración de las actividades 6.5. Desarrollar el cronograma		6.6. Controlar el cronograma	
7. Gestión de los costos		7.1. Planificar la gestión de los costos 7.2. Estimar los costos 7.3. Determinar el presupuesto		7.4. Controlar los costos	
8. Gestión de la calidad		8.1. Planificar la gestión de la calidad	8.2. Gestionar la calidad	8.3. Controlar la calidad	
9. Gestión de los RRHH		9.1. Planificar la gestión de recursos 9.2. Estimar los recursos de las actividades	9.3. Adquirir recursos 9.4. Desarrollar el equipo 9.5. Dirigir el equipo	9.6. Controlar los recursos	
10. Gestión de las comunicaciones		10.1. Planificar la gestión de las comunicaciones	10.2. Gestionar las comunicaciones	10.3. Monitorear las comunicaciones	
11. Gestión de los riesgos		11.1. Planificar la gestión de los riesgos 11.2. Identificar los riesgos 11.3. Realizar el análisis cualitativo de los riesgos 11.4. Realizar el análisis cuantitativo de los riesgos 11.5. Planificar la respuesta a los riesgos	11.6. Implementar la respuesta a los riesgos	11.7. Monitorear los riesgos	
12. Gestión de las adquisiciones		12.1. Planificar la gestión de las adquisiciones	12.2. Efectuar las adquisiciones	12.3. Controlar las adquisiciones	
13. Gestión de los interesados	13.1. Identificar a los interesados	13.2. Planificar el involucramiento de los interesados	13.3. Gestionar la participación de los interesados	13.4. Monitorear el involucramiento de los interesados	

A través de esta tabla, se puede ver qué procesos se tienen en cuenta para los proyectos dirigidos a través de la guía PMBOK y cuáles de ellos se usan en cada ciclo de vida del proyecto.

3.2 ICB (IPMA COMPETENCES BASELINE)

3.2.1 HISTORIA Y OBJETIVOS DE IPMA

IPMA es una federación de más de 60 asociaciones miembro, las cuales desarrollan competencias de gestión de proyectos en ubicaciones con gran influencia de esta manera surgen relaciones entre profesionales, con universidades y facultades y compañías de consultoría.

IPMA fue creada en Suiza en 1965, siendo la organización de gestión de proyectos más antigua, por un grupo de profesional en la gestión de proyectos con el objetivo principal de que hubiera comunicación para compartir ideas, tomar contacto y avanzar la carrera entre profesionales de la gestión de proyectos.

Su misión, según IPMA (2014), es *“promover la competencia en toda la sociedad para permitir un mundo en el que todos los proyectos tengan éxito”*. Asume la responsabilidad que tiene de ayudar a organizaciones, sin ánimo de lucro, a aumentar su competencia en dirección de proyectos. Tiene en cuenta, el cumplimiento de un código ético y una conducta profesional para ejercer esta profesión, y también se puede encontrar en sus páginas.

Se ha ido extendiendo desde Europa al resto de continentes. La demanda de los productos y servicios y el número de las Asociaciones Miembro está en constante crecimiento. Cada asociación Miembro es responsable del desarrollo y gestión de su propio programa de competencia y cualificación en la gestión de los proyectos.

La certificación IPMA se basa en el estándar de competencia global, ICB. Este estándar está definido por el IPMA y describe las competencias que deben poseer los gerentes de proyectos. En la actualidad, ICB está por la versión 4, la cual está diseñada para garantizar que cada gerente de proyecto pueda verificar que ha desarrollado competencias en todas las áreas relevantes para asumir este puesto profesional. Este contiene los términos básicos, tareas, habilidades, funciones, procesos, métodos, técnicas y herramientas que se deben usar para llevar a cabo una dirección de proyecto con éxito.

El objetivo de IPMA es estandarizar todos los procesos y actividades básicas que se siguen en la gestión de proyectos, para que este llegue a su fin de la manera más eficaz y eficiente. Además, con los certificados que tiene evalúa las capacidades del equipo y del director de proyecto.

3.2.2 ESTRUCTURA

La versión 4 de ICB consta de 21 elementos de competencia dividida en 3 áreas (véase **Tabla 3.2**):

- *Competencias de personal*
- *Competencias de práctica*
- *Competencias de perspectiva*

Con esta certificación estas asegurando la competencia en conocimientos, experiencia y actitud personal para llevar a cabo la dirección de proyectos. Además, desde su creación está inmersa en un proceso de mejora continua.

Tabla 3-2 - Competencias para la certificación IPMA. IPMA Versión 4.0. (2015)

PERSONAL	PRÁCTICA	PERSPECTIVA
Autorreflexión y autogestión	Diseño	Estrategia
Integridad personal y confianza	Metas, objetivos y beneficios	Gobernanza , estructuras y procesos
Comunicación personal	Alcance	Cumplimiento, estándares y regulaciones
Relaciones y compromiso	Tiempo	Poder e interés
Liderazgo	Información	Cultura y valores
Trabajo en equipo	Calidad	
Conflicto y crisis		
Inventiva		
Negociación		
Orientación hacia resultados		

Por lo tanto, IPMA no da tanto las pautas y actividades a seguir para gestionar un proyecto sino que se basa en el desarrollo de aptitudes, comportamientos y actitudes que debe tener un buen director de proyecto.

3.3 ISO 21500

3.3.1 HISTORIA Y OBJETIVO DE LA NORMA ISO 21500

La ISO 21500 es la norma de “Directrices para la dirección y gestión de proyectos” (AEC, 2015) es una orientación sobre la gestión de proyectos y proporciona una guía que puede ser utilizada por cualquier tipo de organización pública, privada y

organizaciones comunitarias y para cualquier tipo de proyecto independientemente de su tamaño, complejidad o duración. Proporciona una guía que ofrece una descripción detallada de los conceptos y procesos que se consideran necesarios para crear buenas prácticas en la gestión de proyectos. Estas pautas indican la manera en la que una compañía gestiona sus normas en relación a los proyectos, los procesos los tiempos o los riesgos entre otros. (Norma ISO 21500: Guía para la gestión de proyectos; ISOTools 2017)

La norma ISO 21500 se elaboró con el objetivo de orientar a las organizaciones en su gestión. Además, como tiene un alto nivel sobre conceptos y pautas para la gestión de proyectos, los nuevos profesionales y gerentes con experiencia pueden servirse de la norma ISO 21500 para conseguir una mejora en sus proyectos y unos buenos resultados empresariales. En diferencia a otras metodologías, no se crea con el fin de obtener una certificación ni cumplir unos requisitos para este fin, sino solo como una guía para alcanzar el éxito en la gestión de proyectos.

La norma ISO 21500 da soporte a uno de los principales motores económicos mundiales: los proyectos. Esta norma se creó por situaciones que se daban en ese momento dejando en evidencia la necesidad de una norma que ayudara en la gestión de proyectos. Se puso en marcha cuando PMI dejó de desarrollar la gestión de la calidad como norma, dejando este hueco como una oportunidad para crear dicha norma.

Según Mario Coquillat (2014) algunos de los motivos o causas por las que se elaboró esta norma, fueron las siguientes:

- Globalización progresiva de los mercados
- Creciente oferta internacional de proyectos
- Eliminación progresiva de barreras comerciales
- Armonización de estándares existentes
- Establecimiento de principios y procedimientos comunes de gestión de proyectos

El objetivo principal que busca la norma es que el resultado del proyecto sea el mismo independientemente de la persona que lo pone en práctica.

3.3.2 ESTRUCTURA

El índice de esta norma de manera resumida, es el siguiente:

- 1) Objeto y campo de aplicación
- 2) Términos y definiciones

- 3) Conceptos de la dirección y gestión de proyectos
- 4) Procesos de dirección y gestión de proyectos

La estructura de la norma ISO 21500 sigue las pautas del PMBOK. Los tres primeros capítulos de la ISO 21500 tienen el mismo contenido que el PMBOK, el estándar del PMI. Proyecta 39 procesos diferentes a los que aparecen en la guía PMBOK, pero muy similares.

PMI consiguió que su tabla (véase [Tabla 3.3](#)) de procesos estuviera en la norma ISO 21500.

Tabla 3-3 - Procesos ISO 21500. Extraído de HMD Project Manager (2017)

ÁREAS DE CONOCIMIENTO	GRUPOS DE PROCESOS DE LA DIRECCIÓN DE PROYECTOS				
	INICIO	PLANIFICACIÓN	EJECUCIÓN	MONITOREO Y CONTROL	CIERRE
4. Gestión de la integración	4.1. Desarrollar el Acta de Constitución del Proyecto	4.2. Desarrollar el Plan para la Dirección del Proyecto	4.3. Dirigir y Gestionar el Trabajo del Proyecto 4.4. Gestionar el conocimiento del proyecto	4.5. Monitorear y Controlar el Trabajo 4.6. Realizar el control integrado de cambios	4.7. Cerrar Proyecto o Fase 4.8. Recopilar lecciones aprendidas
5. Gestión del Alcance		5.1. Planificar la Gestión del Alcance 5.2. Recopilar los requisitos 5.3. Definir el Alcance 5.4. Crear la EDT/WBS		5.5. Validar el alcance 5.6. Controlar el alcance	
6. Gestión del tiempo		6.1. Planificar la gestión del cronograma 6.2. Definir las actividades 6.3. Secuenciar las actividades 6.4. Estimar la duración de las actividades 6.5. Desarrollar el cronograma		6.6. Controlar el cronograma	
7. Gestión de los costes		7.1. Planificar la gestión de los costos 7.2. Estimar los costos 7.3. Determinar el presupuesto		7.4. Controlar los costos	
8. Gestión de la calidad		8.1. Planificar la gestión de la calidad	8.2. Gestionar la calidad	8.3. Controlar la calidad	
9. Gestión de los recursos	9.0. Establecer el equipo de proyecto	9.1. Planificar la gestión de recursos 9.2. Estimar los recursos de las actividades	9.3. Adquirir recursos 9.4. Desarrollar el equipo 9.5. Dirigir el equipo	9.6. Controlar los recursos	
10. Gestión de las comunicaciones		10.1. Planificar la gestión de las comunicaciones	10.2. Gestionar las comunicaciones	10.3. Monitorear las comunicaciones	
11. Gestión de los riesgos		11.1. Planificar la gestión de los riesgos 11.2. Identificar los riesgos 11.3. Realizar el análisis cualitativo de los riesgos 11.4. Realizar el análisis cuantitativo de los riesgos 11.5. Planificar la respuesta a los riesgos	11.6. Implementar la respuesta a los riesgos	11.7. Monitorear los riesgos	
12. Gestión de las adquisiciones		12.1. Planificar la gestión de las adquisiciones	12.2. Efectuar las adquisiciones	12.3. Controlar las adquisiciones	
13. Gestión de los interesados	13.1. Identificar a los interesados	13.2. Planificar el involucramiento de los interesados	13.3. Gestionar la participación de los interesados	13.4. Monitorear el involucramiento de los interesados	

3.3.3 BENEFICIOS DE UTILIZAR ISO 21500

Los beneficios de la aplicación de esta norma en el desarrollo de un proyecto son: (¿Qué hay que saber de la ISO 21500?,2017)

- Favorecer a la transparencia de conocimientos entre proyectos y organizaciones
- Mejorar las condiciones de ejecución de las distintas etapas de proyectos
- Facilitar los procesos de licitación y eficiencia
- Promover el uso de una terminología de gestión de proyectos coherente y común
- Flexibilidad de los empleados ante la gestión de los proyectos
- Adecuar la capacidad de los equipos de proyecto en entornos internacionales

La norma europea proporciona principios universales de gestión de proyectos que ayudan a lograr objetivos de forma más sencilla. Además, está diseñada para alinearse con las principales normas internacionales de sistemas de gestión de la calidad y de riesgos como la ISO 10006:2003 (gestión de la calidad en los proyectos), ISO 1000:2003 (orienta hacia una gestión de calidad) e ISO 31000:2009 (principios y directrices sobre la gestión de riesgos).

3.4 PRINCE2

3.4.1 HISTORIA Y OBJETIVOS DE PRINCE2

PRINCE2 es un método estructurado de gestión de proyectos. Originalmente desarrollado en 1989 por la Agencia de Computadora Central y de Telecomunicaciones (CCTA) en el Reino Unido, como sistema estándar para la gestión de proyectos de TI. El CCTA fue renombrado como OGC (Oficina de Comercio Gubernamental). El método fue basado en el sistema PROMPT, un método de gestión de proyectos creado por la empresa Simpack Systems Ltd en 1975. El PROMPT fue adoptado por el CCTA en 1979 como el estándar que se utilizaría para todos los proyectos de tecnologías de la información del gobierno. Cuando lanzaron esta metodología, PRINCE2 en 1989, reemplazó con eficacia a sistema PROMPT dentro del ámbito de proyectos del gobierno.

Es un método ampliamente reconocido, que proporciona un lenguaje común a todos los participantes en el proyecto. Incluye descripciones de los roles de gestión y

las responsabilidades asignadas a los participantes en el proyecto esto resulta beneficios a la hora de adaptarlo a un proyecto determinado con un grado de complejidad y necesidad de habilidades de organización conocimientos para llevar a cabo las distintas tareas del proyecto. (E. López 2014).

Esta metodología describe procedimientos para coordinar personas y actividades en un proyecto, cómo diseñar y supervisar el proyecto y los pasos a seguir si ocurre alguna desviación de lo planificado y es necesario realizar ajustes.

3.4.2 ESTRUCTURA DE LA METODOLOGÍA

La estructura (véase Figura 3.2) que sigue esta metodología se compone de **7 principios, 7 procesos y 7 temáticas**, las cuales se explicarán para entender mejor como funciona dicha metodología.

7 PRINCIPIOS

Según E. López (2014) la metodología que PRINCE2 propone se apoya en 7 Principios, enriqueciendo no sólo al proyecto en concreto, sino a toda la organización en la que se desarrolla.

Los principios de la metodología son los siguientes:

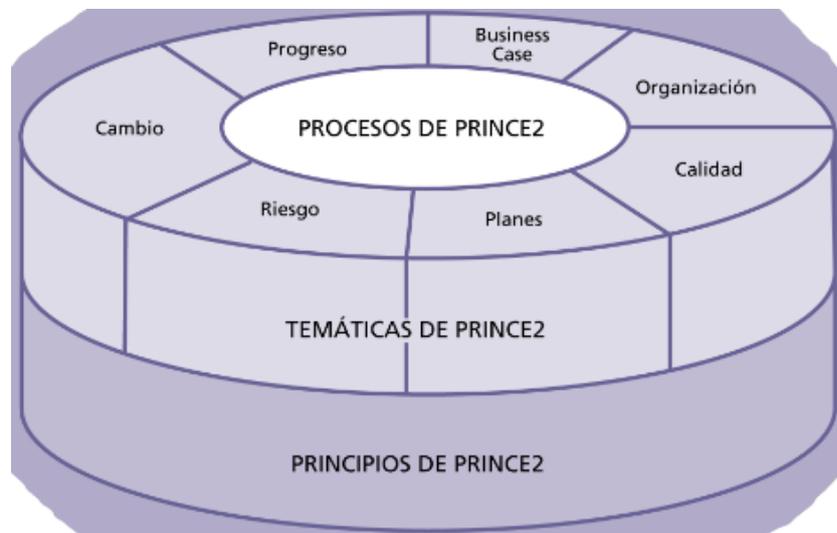


Figura 3.2 - Estructura PRINCE2. Extraída de M. Ferrer (2015)

1. Justificación comercial continua

Se asegura de que hay un motivo justificable para iniciar el proyecto. Esta justificación se identifica y se aprueba.

2. Roles y Responsabilidades definidos

Asegurando que los intereses de los usuarios que van a usar el proyecto, los proveedores y el responsable del área de negocio están representados en la toma de decisiones.

3. Aprender de la experiencia

Se recogen las experiencias anteriores, las que se van obteniendo a lo largo de la ejecución del proyecto, así como las lecciones aprendidas al cierre del mismo.

4. Gestión por fases

Un proyecto que sigue la metodología PRINCE2 se planifica, se supervisa y se controla fase a fase.

5. Gestión por excepción

Es decir, delegar la autoridad suficiente de un nivel de gestión al siguiente, dándole autonomía según unas tolerancias pautadas (de tiempo, coste, calidad, alcance, beneficio y/o riesgo) de manera que, de sobrepasar la tolerancia, se consulte al nivel superior como actuar.

6. Orientación a productos

Centra la atención en la definición y entrega de productos, es decir, un proyecto no son un conjunto de tareas a realizar, si no que entrega productos (que se elaboran tras la ejecución de las tareas que sean necesarias).

7. Adaptación

Asegurando que la metodología PRINCE2 y los controles a aplicar se basen en el tamaño, complejidad, importancia, capacidad y nivel de riesgo del proyecto.

7 TEMÁTICAS

PRINCE2 se refiere a las áreas de gestión como temáticas (Figura 4.3). Como Project Manager se debe supervisar estas áreas en todo momento (Management Plaza, 2015):



Figura 3.3 - Temáticas PRINCE2. Extraída de Management Plaza (2017)

1. **Business Case** → el PORQUÉ del proyecto

Los proyectos comienzan con una idea que se considera que aporta valor a la organización. En esta temática, se establece como convertir esa idea inicial en una propuesta y como mantener el foco del proyecto a lo largo de toda su vida para comprobar que se mantenga alineada con los intereses de la organización. Esta temática sustenta el principio de *Justificación comercial continua*.

2. **Organización** → el QUIÉN del proyecto

Esta temática describe las funciones y responsabilidades de un equipo de gestión de proyectos que utiliza la metodología PRINCE2. Se tiene en cuenta la comunicación entre el equipo del proyecto, así como la comunicación con el resto de partes interesadas en el proyecto (stakeholders). Esta temática sustenta el principio *Roles y responsabilidades definidos*.

3. Calidad → el QUÉ del proyecto

Esta temática se encarga de dar las pautas de cómo debe desarrollarse la idea inicial para asegurarse:

- Que todos los participantes comprenden la calidad del producto que debe ser entregado
- Como se gestionara para asegurarse de que dichos requisitos de calidad se realicen

Da importancia a reconocer y actuar sobre las lecciones aprendidas durante todo el proyecto. Esta temática sustenta el principio *Aprender de la experiencia*.

4. Planes → el CÓMO, CUÁNTO Y CUÁNDO del proyecto

Esta temática complementa a la de Calidad; incluye la descripción de los pasos necesarios para desarrollar los planes y las técnicas propuestas por PRINCE2. Esta temática proporciona un procedimiento de planificación, y establece los distintos niveles de planificación necesarios de acuerdo a los tres niveles de gestión del equipo del proyecto. Esta temática sustenta el principio *Gestión por excepción*.

5. Riesgos → ¿Y SI...?

Esta temática explica como PRINCE2 gestiona la incertidumbre en sus planes y en el entorno del proyecto. La identificación de riesgos te permitirá tomar decisiones informadas sobre qué proyectos empezar, cuales continuar y cuales no se pueden llevar a cabo y hay que detenerlos. Esta temática sustenta el principio *Justificación comercial continua*.

6. Cambio → CUÁL ES EL IMPACTO en el proyecto

Los proyectos en su mayoría, a medida que se van ejecutando van surgiendo problemas que hay que afrontar para minimizar el impacto. Esta temática establece como analizar los problemas y cómo controlar eficazmente el cambio. Describe como evaluar y actuar sobre cuestiones que puedan tener un impacto potencial sobre cualquiera de los aspectos iniciales del proyecto. Esta temática sustenta el principio *Orientación a los productos*.

7. Progreso → el DÓNDE estamos, a DÓNDE vamos, y a DÓNDE nos dirigimos

Esta temática describe el proceso de toma de decisiones para aprobar los planes, supervisarlos y el procedimiento “escalado” que hay que aplicar si el proyecto no avanza según lo esperado. Por último, determina si debe continuar y cómo hacerlo. Esta temática sustenta a los principios *Gestión por excepción*, *Gestión por fases* y *Justificación comercial continua*.

Para una buena aplicación de la metodología PRINCE2 todas estas temáticas deben ser aplicadas a la gestión del proyecto, pero sin olvidar que deben adaptarse al proyecto y al entorno.

7 PROCESOS

Se define proceso como el conjunto estructurado de actividades diseñadas para lograr un objetivo específico. (M. Ferrer, 2015). PRINCE2 tiene un enfoque basado en procesos para la gestión de proyectos. Hay 7 procesos que guían al equipo a través del proyecto, y cada uno de estos procesos está formado por un conjunto de actividades que ayudan a dirigir, gestionar y entregar el proyecto.

Hay que tener en cuenta que, para llevar a cabo estos procesos, se tienen que tener los conocimientos necesarios obtenidos a través de las temáticas anteriormente explicadas.

Estos procesos se ejecutan cronológicamente en el orden que se van a exponer. Como cualquier proceso, toma una o varias entradas y proporciona unas salidas definidas.

Los 7 procesos son:

1. **Puesta en marcha del proyecto** → antes de poner el proyecto en marcha, el director del proyecto y el ejecutivo deben realizar una serie de actividades para asegurar la viabilidad del mismo. Se conoce como el proceso de pre-proyecto, ya que se realiza antes de que comience el proyecto. Consiste en establecer los objetivos del proyecto, asignar un equipo para la gestión del mismo y se crea un plan de escenario para ejecutar la etapa de inicio.
2. **Inicio del proyecto** → define lo que se va a entregar, la calidad, el tiempo y los costes del proyecto. También se hace en esta etapa el análisis de riesgos y la necesidad de recursos para llevar a cabo el proyecto. Esta etapa proporciona la Documentación de Inicio del Proyecto o PID, el Plan de Proyecto y el Business Case.
3. **Dirección del proyecto** → de este proceso se encarga la Junta del Proyecto, su función es autorizar etapas del proyecto y gestionar el proyecto en general mediante el uso del estilo de gestión por excepción. Se extiende desde el inicio del proyecto hasta su fin. Se encargan de controlar que se estén cumpliendo los objetivos acordados y todos los planes descritos en la fase de Inicio del Proyecto.

4. **Control de fases** → un proyecto se divide en fases para la gestión y el control del mismo con mayor eficiencia. El Project Manager se encarga del control de las fases, que consiste en supervisar el trabajo, tomar acciones correctivas, observar los cambios, comunicarse con las partes interesadas y presentación de informes. Según el manual de PRINCE2 su propósito es: “Asignar el trabajo que se debe realizar, hacer un seguimiento de dicho trabajo hacer frente a las cuestiones, informar a la junta de proyectos sobre el progreso y llevar a cabo rectificaciones para garantizar que la fase se mantenga dentro de las tolerancias establecidas”.
5. **Gestión de entrega de productos** → es donde se ejecutan los paquetes de trabajo y se lleva a cabo el proyecto. De esta tarea se encarga el director del equipo, entrega al Team Manager los paquetes de trabajo y cuando se terminan se vuelven a entregar al Director del Equipo.
6. **Gestión de límites de fase** → los límites de fase tienen dos funciones principales: la presentación de informes sobre el rendimiento de la etapa existente y la planificación de la siguiente fase. Por lo tanto la Junta de Proyecto puede comprobar si la fase se ha comportado según lo previsto en el Plan de la Fase.
7. **Cierre del proyecto** → es la última fase de un proyecto. PRINCE2 sugiere una serie de actividades a realizar para preparar el cierre del proyecto, como son; el Informe del Proyecto Final, Lecciones Aprendidas, Informa y registro de aceptación. El proyecto es cerrado por el Ejecutivo de la Junta del Proyecto y no por el Director del Proyecto. Mediante este proceso se verifica la aceptación por el usuario de los productos del proyecto. Se asegura que se pueda prestar asistencia respecto de los productos cuando se disuelva el proyecto. Se evalúan todos los beneficios ya materializados y se garantiza que se ha hecho lo necesario para dar respuesta a todas las cuestiones y riesgos pendientes mediante acciones a realizar recomendadas.

3.4.3 BENEFICIOS DE UTILIZAR PRINCE2

Según Alfred Maeso (2014) como Project Manager, afirma que es uno de los métodos con los que se pueden resolver la mayor parte de problemas a los que se tiene que enfrentar en la mayoría de sus proyectos.

1. **Buena definición de roles y responsabilidades** → PRINCE2 ofrece una descripción completa de todos los roles que deben participar en la dirección de proyectos, identificando sus funciones y responsabilidades. La gestión de la participación de los principales interesados del proyecto ya está incluida dentro del método. Por eso todos los participantes tienen claro lo que se espera de ellos.
2. **Niveles de gestión y control en los proyectos** → PRINCE2 define los niveles de dirección y toma de decisiones necesarios para un proyecto: Gestión corporativa o de programa, (definen objetivos estratégicos y la necesidad de realizar proyectos para conseguirlos), nivel de dirección (toma las decisiones de mayor nivel del proyecto: go/no go), nivel de gestión (realiza la planificación y toma las decisiones operativas para cumplir con el plan) y nivel de entrega (realiza el trabajo técnico, y garantiza la calidad e integridad de los resultados). Además, PRINCE2 utiliza varios sistemas de control para garantizar que el proyecto cumple con los objetivos previstos. Entre ellos, la división del proyecto en fases de gestión, el reporting regular y periódico sobre el estado de proyecto, las evaluaciones a final de fase, el cierre estructurado. Pero una de las cosas más características es el sistema de delegación y escalado. Se trata de que cada nivel de gestión delega a su nivel inferior la responsabilidad de llevar a cabo las funciones que le corresponden dentro de unos límites. Si se pasan estos límites tiene que volver al nivel superior para tomar la mejor decisión posible. Por lo tanto, con este método, el Project manager tiene el soporte de todo el equipo y toma las decisiones que le corresponden.
3. **Foco en el negocio y en el encaje organizacional:** un proyecto es el medio mediante el cual las organizaciones interesadas consiguen realizar los cambios necesarios para conseguir sus objetivos estratégicos. Esa visión debe mantenerse presente durante toda la duración del proyecto. Entonces, el Project Manager conoce lo que la organización espera del proyecto y puede actuar en consecuencia.
4. **Simplicidad y capacidad de adaptación:** es un método simple: gira en torno a 7 principios, 7 temáticas y 7 procesos, además define los principales productos de

gestión que se utilizan en los proyectos y también su contenido recomendado. Esta estructura está pensada para ser lo suficientemente simple y adaptable para poderse utilizar en todo tipo de proyectos. Es decir, PRINCE2 debe aportar sencillez y beneficios al proyecto, adaptándose a las características de cada proyecto.

5. **Mejora continua:** PRINCE2 incluye en el método todo lo necesario para iniciar en la organización un proceso de mejora continua en la dirección de proyectos. por lo tanto, el Project Manager podrá dirigir cada vez mejor sus proyectos.

Finalmente, uno de los mejores beneficios que tiene este método es que es compatible y complementario con las buenas prácticas de dirección de proyectos universalmente conocidas y documentadas en la Guía PMBO y la ISO 21500.

Hay que tener en cuenta a la hora de utilizar PRINCE2 que es tan “solo” un método de gestión de proyectos basado en unos principios, procesos y temáticas que te ayudan a que el proyecto sea finalizado con éxito, pero esto no es suficiente. Además es necesario poner en práctica las habilidades de liderazgo comunicación, técnicas y herramientas especializadas de dirección de proyectos para garantizar el éxito del proyecto.

METODOLOGÍAS ÁGILES

4 METODOLOGÍAS ÁGILES

Esta metodología originalmente era para gestionar la creación y desarrollo de softwares. Surge este enfoque como respuesta a las necesidades que no podían cubrir las metodologías tradicionales. Los problemas que se pretende solucionar con las metodologías ágiles son el de retrasar decisiones y la planificación adaptativa; permitiendo avanzar con el desarrollo de software a gran escala.

Aunque estas metodologías empezaron por las necesidades que surgían a la hora de gestionar proyectos de sistemas de información, las empresas han empezado a **usar los principios ágiles de distintas formas** y han llegado a aplicar la metodología en sus procesos de desarrollo, creación, logística, etc. para convertirse en empresas más eficaces y funcionales. De esta manera son capaces de adaptarse a las nuevas expectativas de los usuarios y a las exigencias del mercado.

Por definición, las metodologías ágiles son aquellas que nos aportan gran flexibilidad a los cambios manteniendo las condiciones del proyecto, consiguiendo de esta manera una respuesta más rápida y consiguiendo afrontar los nuevos retos de forma más eficaz, reduciendo costes e incrementando la productividad.

La metodología ágil se caracteriza porque los equipos de proyecto son multidisciplinares y se organizan según las necesidades de cada proyecto. Cada etapa del proyecto es llevada por un equipo diferente, e irá evolucionando según las necesidades del propio proyecto. Para que esta manera de trabajar funcione, la comunicación entre los miembros que forman el proyecto es fundamental.

Los principios básicos de todas las metodologías ágiles les podríamos resumir en los siguientes.

En primer lugar, con estas metodologías **mejoramos la satisfacción del cliente**, ya que en esta manera de gestionar los proyectos juega un papel muy importante e influyente en todas las etapas del proyecto. En cada etapa, se informa al cliente de los avances realizados como de los problemas que se han tenido, así el cliente, puede sumar su experiencia para optimizar los procesos seguidos y solucionar los problemas que han surgido. Como hay una constante comunicación con el cliente, se pueden evitar malentendidos en la creación del producto pedido, y de esta manera se evitarán fallos y retrasos.

Con esta constante colaboración se consigue **la motivación e implicación del equipo** de desarrollo del proyecto. Esta mejora viene dada porque todos los miembros del equipo pueden conocer el estado del proyecto en todo momento, de modo que

todas las ideas de sus integrantes son tenidas en cuenta y basadas en la realidad. **Aceptando que los requisitos cambien** incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar venta competitiva al cliente.

Además, a intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo para la ajustar y perfeccionar su comportamiento en consecuencia.

Además otra de las mejoras fundamentales que nos da este tipo de metodologías es el **ahorro de tiempo y de costes**. Se trabaja de una manera mucho más eficaz y sin olvidar el presupuesto acordado y sus tiempos de entrega que se han definido al principio del proyecto.

Esta mejora viene del resultado de que se **trabaja con mayor velocidad y eficiencia**. Se realizan entregas parciales, de esta manera podemos **detectar errores lo antes posible** y **eliminar características innecesarias** del producto. Así se va viendo una evolución funcional del producto.

Finalmente, las metodologías ágiles permiten que el retorno de la inversión se produzca más rápido, ya que con las entregas parciales se va mostrando las funcionalidades principales del producto, lo que **rentabiliza la inversión de manera más acelerada**.

Por otro lado, uno de los grandes mitos que existe alrededor de las metodologías ágiles es, que al utilizarlas se debe renunciar a la planificación y a una previsión de resultados. Esta falsa creencia suele venir dada por una mala interpretación del manifiesto ágil.

Es muy importante obviamente contar con un plan de proyecto y una previsión, pero en lo que hace hincapié este manifiesto, es en que esa planificación debe ser revisada constantemente para poder mejorarlo y adaptarlo a los posibles cambios.

Los defensores de este tipo de metodologías consideran que los cambios en los requisitos y contratiempos son un aspecto natural e inevitable. Y creen que es más importante la capacidad de adaptación a estos cambios en cualquier punto de vida del proyecto que, intentar definirlos desde el inicio e invertir tiempo en controlarlos y requisitos.

Además, cabe destacar que en la actualidad, para muchos equipos de desarrollo el uso de metodologías tradicionales les resulta muy lejano a su forma de trabajo considerando difícil de asumir la introducción e inversión asociada en formación y herramientas para manejar este tipo de metodologías.

A continuación, se va a detallar el manifiesto ágil, para profundizar más en qué consiste este tipo de metodologías y que principios siguen.

4.1 FASES DEL ENFOQUE DE GESTIÓN DE PROYECTOS ÁGILES

La metodología ágil se basa en dividir el proyecto en fases (sprints), cada una de estas fases dan un producto terminado, el cual puede ser usado. Estas fases se suceden hasta haber conseguido el total de las funcionalidades definidas para el producto. De forma esquemática estas fases se componen de lo siguiente (Chandana, 2015):

- 1. Fase de Visualizar** → es la fase inicial de la gestión del proyecto en las metodologías ágiles. Cuando se aprueba un caso de negocios y el proyecto se pone en marcha, los miembros del equipo ágil colabora para crear una visión real y convincente del proyecto a desarrollar. En esta fase se identifica la visión del proyecto, los objetivos a conseguir y se identifica a los interesados. El equipo se organiza para conseguir todo lo planteado.
- 2. Fase de Especular** → en esta fase se formaliza la visión del proyecto convirtiéndose en requisitos, se realiza el enfoque general para cumplir estos requisitos y se presenta un plan de lanzamiento de alto nivel para el producto. Hay dos actividades clave en esta fase:
 - El equipo de proyecto debe tener claro los requisitos del proyecto y la prioridad de los mismos para empezar a trabajar
 - Hay que crear un plan con las limitaciones de tiempo y el plan de iteración para la implementación de la visión.
- 3. Fase de Explorar** → los miembros del equipo estudian varias alternativas para implementar cumplir con los requisitos del proyecto. Se realizan entregas y pruebas de trabajo. El equipo trabaja de manera iterativa en la fase de exploración, toman un conjunto de características y lo aceptan en un plan para una iteración. Esta fase va de la mano con la de adaptación, en a que el equipo aprende de las experiencias de desarrollo.
- 4. Adaptar** → lo principal de esta fase es recibir los comentarios y adaptarse a la situación en función de los requisitos del cliente. Por lo tanto, el equipo de

proyecto revisa los resultados de la ejecución, la situación actual y el rendimiento. De esta manera se adapta a las nuevas situaciones y revisa el plan para adaptarse. En esta fase, se pueden dar cambios en el enfoque del proyecto, procesos, entorno, los objetivos, etc., de acuerdo con los requisitos de los clientes.

5. **Cierre** → esta es la última fase del marco ágil. Se concluye el proyecto captando las lecciones clave del proyecto.

Esta forma de llevar a cabo un proyecto, da la posibilidad de ajustar los objetivos y los requisitos con cada sprint, asegurándose que el final del producto cumplirá con las expectativas del cliente. Además, como cada sprint es un producto funcional, se puede comercializar y avanzar ingresos generados por el proyecto.

4.2 EL MANIFIESTO AGIL

(Beck et al, 2001)

Los 12 principios de este manifiesto son:

- I. La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.
- II. Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- III. Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- IV. La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- V. Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.
- VI. El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.
- VII. El software que funciona es la medida principal de progreso.
- VIII. Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.
- IX. La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- X. La simplicidad es esencial.
- XI. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.
- XII. En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.

4.3 CARACTERÍSTICAS BÁSICAS DE LOS PROYECTOS CON METODOLOGÍAS ÁGILES

Las características básicas de los proyectos gestionados con metodologías ágiles son las siguientes (Sanz, MDAP Executive master in Project Management, 2014) (Figueroa, 2015):

- Incertidumbre: la dirección indica la necesidad estratégica que se desea cubrir (sin entrar en detalles), ofreciendo máxima libertad al equipo de trabajo.
- Equipos auto-organizados: no existen roles especializados
- Autonomía: libertad para la toma de decisiones.
- Auto-superación: de forma periódica se evalúa el producto que se está desarrollando.
- Auto-enriquecimiento: transferencia del conocimiento.
- Fases de desarrollo solapadas: Las fases no existen como tal sino que se desarrollan tareas/actividades en función de las necesidades cambiantes durante todo el proyecto. De hecho, en muchas ocasiones no es posible realizar un diseño técnico detallado antes de empezar a desarrollar y ver algunos resultados. Por otra parte, las fases tradicionales efectuadas por personas diferentes no favorece el trabajo en equipo y pueden llegar a generar más inconvenientes que ventajas (p.ej. un retraso en una fase, afecta a todo el proyecto).
- Control sutil: establecimientos de puntos de control para realizar un seguimiento adecuado sin limitar la libertad y creatividad del equipo. Así mismo, se recomienda:
 - Evaluar el ambiente laboral, siendo fundamental la elección de personas que no generen conflictos.
 - Reconocer los méritos mediante un sistema de evaluación justo y entender los errores como puntos de mejora y aprendizaje.
 - Potenciar la interacción entre el equipo y el negocio, para que puedan conocer las necesidades de primera mano.
 - Difusión y transferencia del conocimiento: alta rotación de los miembros de los equipos entre diferentes proyectos. Por otra parte, potenciar el acceso libre a la información y documentación.

4.4 SCRUM

4.4.1 HISTORIA (ORIGENES DE SCRUM, CLUB DE TECNOLOGIA, 2013)

El término Scrum (traducido del inglés como melé) fue acuñado y definido por Ikujiro Nonaka e Hirotaka Takeuchi en los años 80, cuando las principales empresa de desarrollo tecnológica empezaban a dominar el mercado y a definir conductas de trabajo. Ambos publicaron en 1986 en la Harvard Business Review este artículo “El nuevo juego para el desarrollo de productos”. Este artículo de 1986 fue una influencia para desarrollar muchos de los conceptos que dieron nacimiento a lo que hoy llamamos Scrum.

El avance de las formaciones de las melés en partidos de rugby, inspiró a Nonaka y Takeuchi para bautizar una nueva forma de trabajar que ya venía dándose en muchas empresas tecnológicas como Honda, Canon y Fuji-Xerox.

Nonaka y Takeuchi utilizan esta metáfora del Rugby de Scrum para describir el desarrollo del producto: «El enfoque de las ‘carrera de relevos’ para el desarrollo de productos entra en conflicto con el objetivo de obtener la máxima velocidad y flexibilidad. En su lugar un enfoque como el rugby – donde el equipo intenta avanzar como equipo, enviando el balón hacia atrás y luego avanzar – sirve mejor a los desarrollos competitivos que se ven hoy en día».

El desarrollo del software fue el primero en aplicar la metodología Scrum. Anteriormente, este sector utilizaba para planificar y gestionar sus proyectos con métodos en cascada o secuencial. En ellas se planifican varias fases con unos plazos establecidos y se van ejecutando.

En 1993, Jeff Sutherland y su equipo en Easel Corporation crearon el proceso de Scrum para ser utilizado en los procesos de desarrollo de software combinando los conceptos del artículo de 1986 con los conceptos del desarrollo orientado a objetos, un control de procesos empírico, desarrollo iterativo e incremental, procesos de software y mejora de la productividad, así como el desarrollo de sistemas complejos y dinámicos. En 1995, Ken Schwaber publica el primer informe sobre Scrum en OOPSLA. Desde esa fecha, Schwaber and Sutherland, juntos y separados, han producido y publicado varias especificaciones para Scrum, incluyendo Desarrollo de Software Ágil con Scrum, Gestión de Proyectos Ágiles con Scrum y la Guía de Scrum.

En conclusión, Scrum propone una metodología donde el equipo debe avanzar en conjunto, de nada sirve tener partes del producto terminado, si el producto en sí no está terminado. De esta manera la velocidad para el desarrollo de productos lo

marcará el que tiene la menor velocidad en el equipo. La filosofía de **Scrum** impulsa y resalta el trabajo en equipo, el aprendizaje constante y una estructura de desarrollo dinámico, que es flexible a los cambios durante el proceso de desarrollo.

4.4.2 ¿QUÉ ES SCRUM?

Scrum es un método para trabajar en equipo a partir de iteraciones o Sprints. El objetivo de esta metodología es planificar y controlar proyectos donde hay gran incertidumbre por los cambios que suceden a última hora.

La planificación suele hacerse por semanas. Al final de cada sprint o iteración, se va revisando el trabajo validado de la anterior semana. En función de los resultados obtenidos, se priorizan y planifican las actividades en las que invertiremos nuestros recursos en el siguiente sprint.

La metodología Scrum se centra en ajustar sus resultados y responder a las exigencias reales y exactas de cliente. Se va revisando cada entregable porque los requisitos van cambiando a corto plazo. Por eso, los sprints duran un mínimo de una semana y un máximo de cuatro semanas.

Entre las principales características de la metodología Scrum destaca que es un desarrollo incremental en lugar de la clásica planificación del desarrollo completo de un producto o servicio. Sus equipos de trabajo se caracterizan por ser **auto-organizados**. Y se centra en el producto final, en la calidad del mismo.

Además, en la metodología Scrum se solapan diferentes fases de desarrollo, en lugar de llevar a cabo una planificación secuencial o de cascada.

Roles de Scrum

La metodología Scrum tiene unos roles y responsabilidades principales, asignados a sus procesos de desarrollo. Estos son:

- **Product Owner.** Se asegura de que el proyecto se esté desarrollando acorde con la estrategia del negocio. Escribe historias de usuario, las prioriza, y las coloca en el Product Backlog. Debe conocer la velocidad del equipo, para realizar estimaciones de cuando estarán implementadas las necesidades en el producto. Es el responsable de cancelar el sprint si ocurre un imprevisto extremo.

- **Scrum Master o Facilitador.** Debe participar en las reuniones y asegurarse de que cumplan el tiempo y el objetivo establecido. Elimina los obstáculos que impiden que el equipo cumpla con su objetivo. Se encarga también de que todo el equipo siga la metodología Scrum y que sea entendido por todos.
- **Development team Member.** Los encargados de crear el producto para que pueda estar listo con los requerimientos necesarios. Se recomienda que sea un equipo multidisciplinar, de no más de 10 personas. Sin embargo, empresas como Google disponen de unos 15.000 desarrolladores trabajando en una rama del código.

4.4.3 ESTRUCTURA

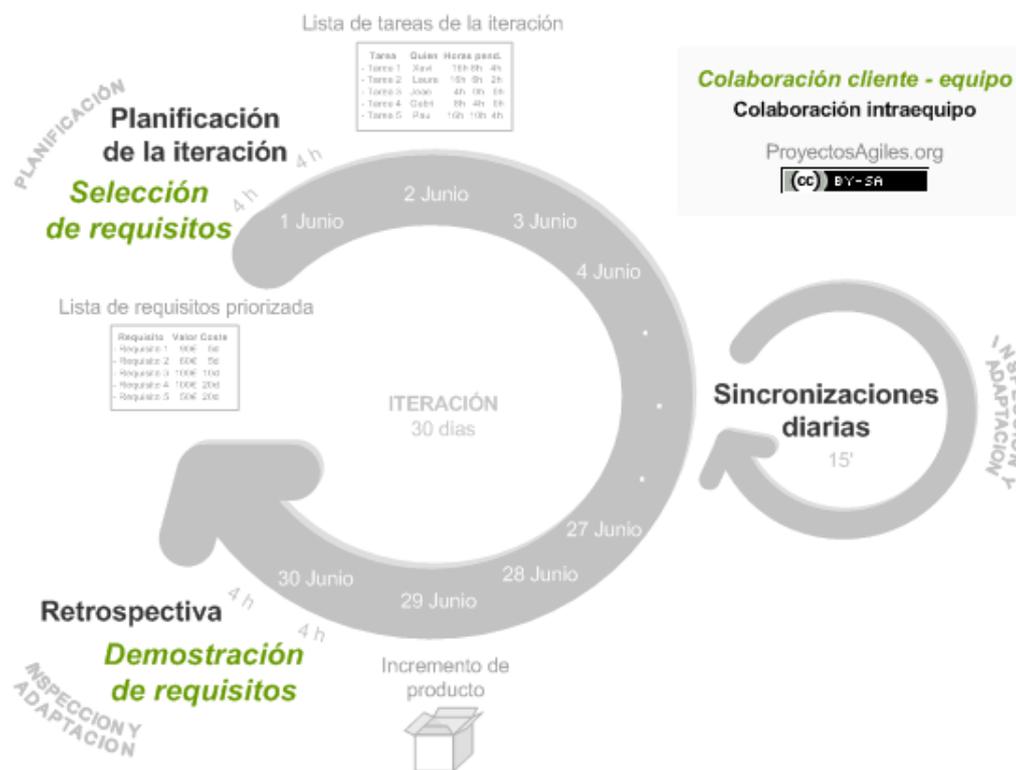


Figura 4.1 - Proceso Scrum. Extraído de proyectosagiles.org (2008)

En scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

4.4.4 FASES DE LA METODOLOGIA SCRUM

Lo principal de la metodología Scrum son los Sprints (**Figura 4.2**), el cual es un intervalo prefijado durante el cual se crea un producto funcional que formara parte del producto final. A lo largo del desarrollo hay sprints consecutivos de duracion constante. Se podria decir que cada sprint es un mini proyecto de no mas de un mes de duracion. Cada sprint cuenta con un diseño, un plan flexible de desarrollo, y unos requisitos que daran lugar al producto deseado.

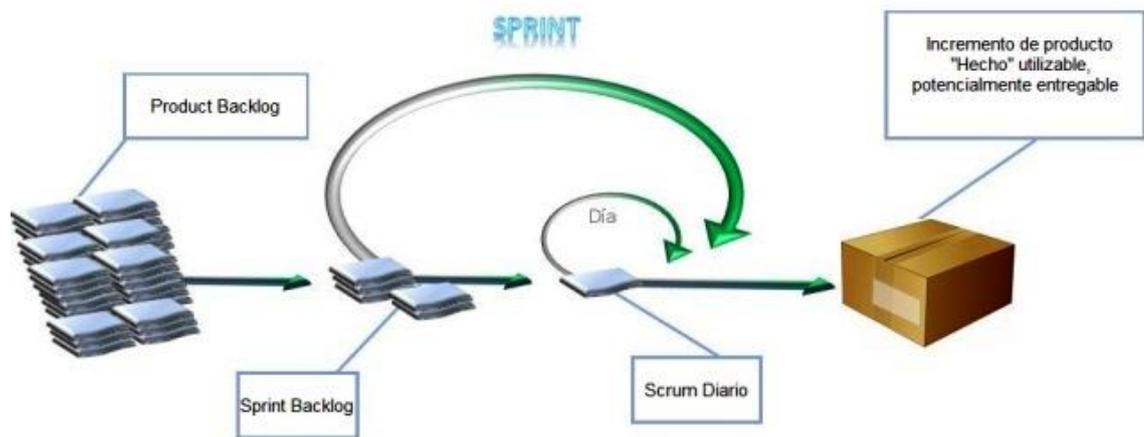


Figura 4.2 - Fases en SCRUM. Extraído de OBS Business School (2017)

Cada Sprint puede tener una serie de **eventos o etapas** (OBS Business School, 2017). Los más comunes son:

1. Reunión para la planificación del Sprint

El trabajo a realizar para cada sprint se prevé en esta reunión. Es un evento o etapa en el que colabora todo el equipo Scrum y es de duración variable, por ejemplo para un sprint de un mes tiene unas ocho horas de duración. La reunión en general, consta de dos partes, donde queda definido:

- Qué se va a entregar en el próximo sprint
- Cómo se va a realizar el trabajo propuesto

En conclusión, queda definido el Objetivo del Sprint.

2. Scrum diario

Es un evento de 15 minutos cuyo fin es que el equipo de Desarrollo sincronice actividades, y establezca un plan para las próximas 24 horas. Se basa en poner en común y sincronizar actividades para elaborar el plan del día. Esto se realiza mediante la inspección del trabajo desde el último Scrum Diario, lo que aporta datos de la progresión del sprint y de la tendencia del progreso. Este evento aporta comunicación diaria entre todos los interesados, y elimina la necesidad de realizar otras reuniones. Es una reunión clave de inspección y adaptación, dónde se identifican y eliminan obstáculos y se promueve la rápida toma de decisiones.

3. Trabajo de desarrollo durante el Sprint

Cuando el sprint esta en curso se debe asegurar que: no se realizan cambios y por lo tanto, los objetivos se están cumpliendo. No se disminuyen los objetivos de calidad. Se mantiene un feedback constante con el cliente o dueño del proyecto, para poder renegociar el alcance del proyecto y que se mantenga claro en todo momento.

4. Revisión del Sprint

Se lleva a cabo al final del Sprint, consiste en una reunión con el cliente o dueño del proyecto y todo el equipo Scrum, en la que se estudia y revisa el Product Backlog del Sprint. Se definen los aspectos a cambiar, en caso necesario, de mayor valor o probables para planificarlo en el siguiente Sprint. Es una reunión informal, y la presentación del incremento esta destinada a obtener retroalimentación fomentar la colaboración. Esta etapa incluye los siguientes elementos:

- Los asistentes son el Equipo Scrum y los interesados clave invitados por el Dueño de Producto;
- El propietario del producto identifica lo que se ha “hecho” y lo que no se ha “hecho”
- El equipo de desarrollo discute lo que funcionó bien durante el Sprint, qué problemas hubo y cómo se resolvieron;
- El equipo de desarrollo demuestra el trabajo que se ha “hecho” y responde preguntas sobre el Incremento;
- El propietario del producto analiza el estado actual del Product Backlog, y estima fechas de finalización basado en el progreso hasta la fecha

- Todo el grupo colabora en qué hacer a continuación, de modo que la revisión del Sprint ofrece valiosos aportes a las subsiguientes reuniones de planificación de Sprint.
- Se hace una revisión de cómo el mercado o el uso potencial del producto podría haber cambiado lo que es de más valor para hacer a continuación
- Se hace una revisión de la línea de tiempo, presupuesto, capacidades potenciales y mercado para la próxima entrega prevista del producto

El resultado de la revisión del Sprint es un Product Backlog revisado que define los ítems del Product Backlog de mayor valor o probables para el siguiente Sprint. El Product Backlog también se puede ajustar en general para satisfacer las nuevas oportunidades.

5. Retrospectiva del proyecto

Oportunidad del equipo de desarrollo para mejorar su proceso de trabajo y aplicar los cambios en los siguientes Sprints. El propósito de la retrospectiva de Sprint es:

- Revisar cómo fue el último Sprint en lo que respecta a las personas, relaciones, procesos y herramientas
- Identificar y ordenar los temas principales que salieron bien y las potenciales mejoras
- Crear un plan para la implementación de mejoras con respecto a cómo el Equipo Scrum hace su trabajo.

4.5 XP O EXTREME PROGRAMMING

4.5.1 HISTORIA

(Ingeniería de Software, septiembre 2008)

La Programación Extrema, como proceso de creación de software diferente al convencional, nace de la mano de Kent Beck (gurú de la XP y autor de los libros más influyentes sobre el tema).

Chrysler Corporation hacía tiempo que estaba desarrollando una aplicación de nóminas, pero sin demasiado éxito por parte de la gente que tenía en el proyecto. El verano de 1996, Beck entró en nómina en la compañía y se le pidió hacer esta

aplicación como trabajo. Es en esta aplicación cuando nace la Programación Extrema como tal.

Beck reconoció que el proceso (o metodología) de creación de software o la carencia de este era la causa de todos los problemas y llegó a la conclusión que para proporcionar un proceso que fuera flexible era necesario realizar ciertos cambios en la estructura o manera de hacer de los programadores, los cuales se tenían que acomodar al cambio a realizar.

Él tenía varias ideas de metodologías para la realización de programas que eran cruciales para el buen desarrollo de cualquier sistema.

Las ideas primordiales de su sistema las comunicó en la revista C++ Magazine en una entrevista que ésta le hizo el año 1999. En ésta decía que él estaba convencido que la mejor metodología era un proceso que enfatizase la comunicación dentro del equipo, que la implementación fuera sencilla, que el usuario tenía que estar muy informado e implicado y que la toma de decisiones tenía que ser muy rápida y efectiva.

Los autores (o mejor dicho, los propulsores como el propio Kent Beck, Ward Cunningham o Ron Jeffries entre otros) de la Programación Extrema, fueron a la web Portland Pattern Repository y empezaron a hablar de ella y promocionarla, de lo que era y cómo realizarla. Estos propulsores de la XP hablaban de ella en cada ocasión que tenían y en cada página que, poco o mucho hablara de temas de programación.

Este hecho, llegó a molestar a buena parte de la comunidad que intentaba discutir sobre temas de programación. Fue tanta esta molestia que nació el fenómeno XP Free Zone (zona libre de XP) en determinadas webs como petición de no hablar de Programación Extrema en ella.

La discusión sobre temas de diseño de modelos de programación sobre los cambios recientes se hizo tema difícil porque la mayoría de la actividad fue relacionada con la Programación Extrema. Eventualmente, y debido a esto, la mayoría de la gente que solía discutir sobre los temas de diseño de modelos de programación fue apartándose de este ambiente para discutir sus ideas en otros ambientes más “reservados”. La comunidad empezó a referirse a estos sitios como a Salas Wiki (Wards Wiki).

4.5.2 ¿QUÉ ES XP?

La Programación Extrema es exitosa porque enfatiza la satisfacción del cliente. En lugar de ofrecerle todo lo que pueda desear en una fecha muy lejana en el futuro, este proceso ofrece el software que necesita a medida que lo necesita. La Programación Extrema les permite a sus desarrolladores responder con confianza a los requisitos cambiantes de los clientes, incluso a fines del ciclo de vida.

Enfatiza en la importancia del trabajo en equipo, tanto los gerentes como clientes y desarrolladores deben colaborar para obtener los resultados esperados del proyecto. XP implementa en un entorno simple pero eficaz que permite a los equipos ser altamente productivos.

Extreme Programming mejora un proyecto de software de cinco maneras esenciales:

- Comunicación
- Simplicidad
- Retroalimentación
- Respeto
- Valentía

Para que esta metodología se lleve de manera óptima, el equipo se comunica constantemente con sus clientes y entre ellos.

El equipo se auto-organiza para llevar a cabo el trabajo. Es el responsable de mantener el diseño simple y limpio. Se hacen entregas cada cierto tiempo. De esta manera, obtienen retroalimentación al probar su software desde el primer día e implementan los cambios que se sugieren. Con esta base, los programadores extremos pueden responder valientemente a los requisitos y la tecnología cambiantes.

El aspecto más sorprendente de la programación extrema son sus reglas simples. Las reglas pueden parecer incómodas e incluso ingenuas al principio del proyecto, pero se basan en unos principios sólidos.

Estas reglas establecen expectativas entre los miembros del equipo, pero no son el objetivo final. Definen un entorno que promueve la colaboración y el empoderamiento del equipo, ese es su objetivo. Una vez logrado, el trabajo en equipo productivo continuará incluso a medida que las reglas cambien para ajustarse a las necesidades específicas de su empresa.

4.5.3 ROLES EN XP

Tracker → Es el encargado de seguimiento. Proporciona retroalimentación al equipo. Debe verificar el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado comunicando los resultados para mejorar futuras estimaciones.

Customer → Es el cliente o usuario que escribe las historias de usuario y le asigna la prioridad a cada una centrándose en qué aporta mayor valor al negocio. También es el encargado de realizar las pruebas funcionales para validar la implementación presentada.

Programmer → Es el programador más importante del equipo XP. Este es el que se encarga de escribir las pruebas unitarias y el código del sistema.

Coach → Es responsable del proceso global y se encarga de guiar a los miembros del equipo para seguir el proceso correctamente.

Tester → Ayuda al cliente a escribir las pruebas funcionales. Ejecuta pruebas regularmente difunde los resultados en el equipo y es responsable de las herramientas de soporte para pruebas.

Big Boss → Es el vínculo entre clientes y programadores. Su labor esencial es la coordinación.

Consultor → Es un miembro externo del equipo con un conocimiento específico en algún tema necesario para el proyecto ayuda al equipo a resolver un problema específico y puede que no siempre haya un consultor como parte del equipo.

Manager → Se encarga de agendar las reuniones, se asegura de que el proceso de juntas sea seguido, registra los resultados de las reuniones para futuros reportes para el Tracker. Asiste a las reuniones y trae información importante, mantiene al equipo feliz y productivo.

Doomsayer → Se asegura de que todos los miembros conozcan los riesgos del proyecto.

Asegurándose:

- De que todo el mundo sabe los riesgos que algo implica.
- Que las malas noticias no se oculten ni sean pasadas por alto.
- Que las malas noticias no sean exageradas.

Este rol debe tener cuidado porque puede bajar la productividad del equipo pero será responsable de los cambios al proyecto gracias a los riesgos. Se debe ser constructivo al respecto.

Algunos roles pueden ser combinados en la misma persona; por ejemplo la misma persona puede ser el Manager y el Tracker y otros es recomendable que no sean combinados: Programmer-Tracker, Programmer-Tester, Customer-Programmer, Coach-Tracker, por mencionar algunos. El Manager no debería combinarse con algún otro rol además de Tracker. Ron Jeffries opina que el Coach no debería combinarse con el Programador, pero que podría terminar de una manera satisfactoria. Solo que no lo suficientemente buena.

También existen anti roles de Extreme Programming tienden a meterle ruido a la realización del proyecto.

Standards and Methodology Guy → Normalmente un ex-académico la clase de persona que nunca ha hecho que un día valga la pena, código difícil en su vida, pero tiene el poder de decirte que estás haciendo mal y por qué el emplear Extreme Programming puede no funcionar y no funcionará. Puede haber unas discrepancias con el Manager.

GodHead → El tipo de persona que quiere ser el Chief Architect (rol que no existe en esta metodología) o el Manager pero no lo es. Gusta de estar en el centro de todo. Memoriza suficientes detalles del sistema para abalanzarse dentro de cada discusión acerca del diseño. Toma responsabilidad en todo lo que ve. Y el resto del equipo teme tocar su código o el diseño sin él.

4.5.4 ESTRUCTURA



Figura 4.3 - Funcionamiento del método XP. Extraído de Extreme Programming

PLANEACIÓN

Esta actividad consiste en definir los requisitos que permiten que los miembros técnicos del equipo XP entiendan el contexto del negocio para el software y recogen las características principales y la funcionalidad que se requiere. Con todas estas condiciones, se crean las historias de usuario que describen el software que se va a elaborar.

Cada una de estas historias es escrita por el cliente, poniendo todos los requisitos necesarios y dándoles la prioridad que tienen para el negocio. Después, los miembros del equipo XP evalúan cada historia y le asignan un coste, medido en semanas de desarrollo. Las historias no pueden durar más de 3 semanas, si el equipo lo cree conveniente puede mandar al cliente descomponerla en historias más pequeñas. Es destacable que en cualquier momento se pueden crear historias nuevas.

Los clientes y los desarrolladores trabajan juntos para decidir cómo agrupar las historias en la siguiente entrega. Una vez que se llega a un compromiso por todas las

partes interesadas sobre la entrega, el equipo XP se organiza de alguna de las siguientes maneras:

- Todas las historias se implementarán de inmediato
- Las historias con más valor entraran a la programación de actividades y se implementaran en primer lugar
- Las historias con más riesgo se implementaran primero

Después de la primera entrega del proyecto (conocida también como implemento de software) el equipo XP calcula su velocidad de trabajo, y miran cuantas historias de usuario han sido capaces de implementar. Estos datos los usan para:

- Ayudar a estimar las fechas de entrega y programar las actividades para las entregas posteriores
- Determinar si se ha hecho un gran compromiso para todas las historias durante todo el desarrollo del proyecto

Con ello se modifican, si es necesario, el contenido de las entregas o las fechas de la entrega final. A medida que avanza el proyecto, el usuario puede introducir nuevas historias, modificar las ya realizadas, descomponerlas o eliminarlas. Por ello el equipo XP tiene que estar preparado para cualquier cambio.

DISEÑO

El diseño de XP es siempre sencillo. Este diseño guía la implementación de una historia tal y como se describe. XP estimula el uso de tarjetas CRC (clase-responsabilidad-colaborador) (Figura 4.8) como un mecanismo eficaz para pensar en el software en un contexto orientado a objetos. Estas tarjetas identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software. Son el único producto de trabajo de diseño que se generan como parte del proceso XP.

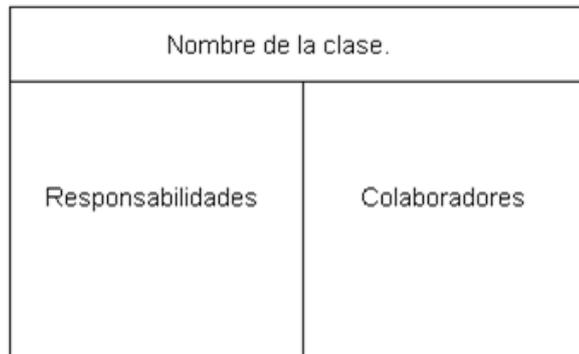


Figura 4.4 - Tarjeta CRC. Extraído de Programación Orientada a Objetos y Talleres (2011)

Si el diseño de una historia es muy complejo, XP recomienda la creación de un prototipo operativo. Entonces, se implementa ya evalúa el prototipo del diseño, llamado solución en punta. El objetivo es disminuir el riesgo cuando comience la implementación verdadera y validar las estimaciones originales para la historia que contiene el problema de diseño.

XP aboga por el rediseño, técnica de construcción que también es un método para la optimización del diseño. Según Fowler (1963) el rediseño es:

“Rediseño es el proceso mediante el cual se cambia un sistema de software en forma tal que no altere el comportamiento externo del código, pero si mejore la estructura interna. Es una manera disciplinada de limpiar el código que minimiza la probabilidad de introducir errores. En esencia, cuando se rediseña, se mejora el diseño del código después de haber sido escrito.”

El objetivo del rediseño es controlar dichas modificaciones, sugiriendo pequeños cambios en el diseño que “son capaces de mejorarlo en forma radical”. Sin embargo, se debe tener en cuenta que el esfuerzo que requiere el rediseño aumenta en forma notable con el tamaño de la aplicación.

Un concepto central en XP es que el diseño ocurre tanto antes como después de que comienza la codificación. Rediseñar significa que el diseño se hace de manera continua conforme se construye el sistema. En realidad, la actividad de construcción en sí misma dará al equipo XP una guía para mejorar el diseño.

CODIFICACIÓN

Después de que las historias han sido desarrolladas y de que se ha hecho el trabajo de diseño preliminar, el equipo no inicia la codificación, sino que desarrolla una serie de pruebas unitarias a cada una de las historias que se van a incluir en la entrega en curso (incremento de software).

Una vez creada la prueba unitaria, el desarrollador está capacitado para centrarse en lo que debe implementarse para pasar la prueba. Una vez que el código está terminado, se le aplica de inmediato una prueba unitaria, con lo que se obtiene retroalimentación instantánea para los desarrolladores.

Uno de los aspectos de los que más se habla en la XP es la programación por parejas. XP recomienda que dos personas trabajen juntas en una estación de trabajo con el objeto de crear código para una historia. Esto da un mecanismo para la solución de problemas en tiempo real y para el aseguramiento de la calidad también en tiempo real. También mantiene a los desarrolladores centrados en el problema que se trate. En la práctica, cada persona adopta un papel un poco diferente.

A medida que las parejas de programadores terminan su trabajo, el código que desarrollan se integra con el trabajo de los demás. En ciertos casos, esto lo lleva a cabo a diario un equipo de integración. En otros, las parejas de programadores tienen la responsabilidad de la integración. Esta estrategia de “integración continua” ayuda a evitar los problemas de compatibilidad e interfaces y brinda un ambiente que ayuda a descubrir a tiempo los errores.

PRUEBAS

La creación de pruebas unitarias antes de que comience la codificación es un elemento clave del enfoque de XP. Las pruebas unitarias que se crean deben implementarse con el uso de una estructura que permita automatizarlas, de manera que puedan ejecutarse en repetidas veces y con facilidad. Esto estimula una estrategia de pruebas de regresión.

A medida que se organizan las pruebas unitarias individuales en un “grupo de prueba universal” [Wel99], las pruebas de la integración y validación del sistema puede efectuarse a diario.

Esto da al equipo XP una indicación continua del envase y también lanza señales de alerta si las cosas marchan mal. Wells [Wel99] dice: “corregir pequeños problemas cada cierto número de horas toma menos tiempo que resolver problemas enormes justo antes del plazo final.”

Las pruebas de aceptación XP, también llamadas pruebas del cliente son especificadas por el cliente y se centran en las características y funcionalidad generales del sistema que son visibles y revisables por parte del cliente. Las pruebas de aceptaciones derivan de las historias de los usuarios que se han implementado como parte de la liberación del software.

VENTAJAS

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.
- Solución de errores de programas
- Versiones nuevas
- Implementa una forma de trabajo donde se adapte fácilmente a las circunstancias

DESVENTAJAS

- Es recomendable emplearlo solo en proyectos a corto plazo
- Altas comisiones en caso de fallar
- Imposible prever todo antes de programar
- Demasiado costoso e innecesario

4.6 DYNAMIC SYSTEMS DEVELOPMENT METHOD

4.6.1 HISTORIA

DSDM fue desarrollado en el Reino Unido en los años 90 por un consorcio de proveedores y de expertos en la materia del desarrollo de sistemas de información (IS), el consorcio de DSDM, combinando sus experiencias de mejores prácticas. El consorcio de DSDM es una organización no lucrativa y proveedor independiente, que posee y administra el *framework*. La primera versión fue terminada en enero de 1995 y publicada en febrero de 1995. El framework para el Negocio Centralizado Desarrollado lanzado en mayo de 2003.

En julio de 2006, el framework para el Negocio Centralizado Desarrollado lanzado en mayo de 2003 se puso a disposición de los usuarios para ver y usar; sin embargo, cualquier persona que revele DSDM debe seguir siendo miembro del consorcio sin fines de lucro.

En 2014, el manual de DSDM se puso a disposición en línea y público. Además, se pueden descargar plantillas para DSDM.

En octubre de 2016, el consorcio DSDM cambió su nombre por Agile Business Consortium. Una organización sin ánimo de lucro, fue pionera en Agile y continúa inspirando nuevos desarrollos y pensando, como el papel de la innovación en el corazón del Framework for Business Agility.

4.6.2 EL MARCO DE PROYECTO AGIL DE DSDM DE 2014 EN ADELANTE

El DSDM Agile Project Framework, según el Manual DSDM (2014) es el enfoque líder, probado y ágil que brinda la gobernanza y el rigor junto con la agilidad y flexibilidad que las organizaciones exigen hoy en día.

El DSDM Agile Project Framework se puede usar de manera independiente o combinada con otros métodos reconocidos como PRINCE2 y PMI. También es ideal como un envoltorio para enfoques Agile más limitados para garantizar que se aborde todo el ciclo de vida del proyecto.

El manual que apoya esta metodología, está destinado a apoyar el estudio y el desarrollo profesional continuo en el campo de la gestión de proyectos ágiles. Muestra cómo se puede aplicar el ciclo de vida para abordar las necesidades del negocio, mitigar el riesgo y maximizar el retorno de la inversión.

4.6.3 PRINCIPIOS

La metodología ágil DSDM sigue ocho principios (Agile Business, 2014). Comprometer cualquiera de los siguientes principios socava la filosofía de DSDM e introduce un riesgo para el resultado exitoso del proyecto. Si un equipo no sigue todos estos principios, no obtendrá todos los beneficios del enfoque. El valor colectivo de los principios de DSDM permite a las organizaciones entregar soluciones empresariales de mejor valor en colaboración.

Los ocho principios son:

1. **Enfoque en la necesidad del negocio** → no hay que olvidar a la hora de tomar decisiones del fin primordial del proyecto. No se puede perder la visión del negocio. Es decir, un proyecto es un medio para un fin, no un fin en sí mismo. Para cumplir con este objetivo los equipos de DSDM:

- Comprenden las verdaderas prioridades comerciales
- Establecen un caso comercial válido
- Garantizan el patrocinio y el compromiso empresarial continuo
- Garantían la entrega de la subsección mínima utilizable

2. **Entrega a tiempo** → dar una solución dentro del tiempo acordado es muy deseable para el proyecto, y en muchos casos el factor de éxito más importante. Ya que, la tardía entrega de un proyecto puede ser un inconveniente si tiene que cumplir plazos legales o es una oportunidad de mercado. Para cumplir con este principio, los equipos de DSDM necesitan:

- Timebox (es el tiempo máximo para conseguir unos objetivos) el trabajo
- Enfoque en las prioridades comerciales
- Siempre llegue a los plazos
- Construya confianza a través de una entrega predecible

La combinación de timeboxing y la priorización de MoSCoW permiten a los equipos de DSDM proteger los plazos mientras se definen las características y crear una reputación para una entrega oportuna y predecible.

3. **Colaboración** → para llevar a cabo el proyecto a su fin con éxito es muy importante que el equipo trabaje con espíritu de cooperación activa y compromiso. Esta actitud de equipo da como resultado mayor comprensión,

velocidad y propiedad compartida, lo que permite a los equipos funcionar a un nivel que supera la suma de sus partes. Para cumplir este principio, los equipos de DSDM necesitan:

- Compromiso de las partes interesadas durante todo el proyecto
- Participación proactiva de los representantes de las empresas
- Asegurarse de que todos los miembros están capacitados para tomar decisiones
- Construir una cultura de equipo único

4. Nunca comprometa la calidad → el nivel de calidad debe ser acordado al inicio del proyecto. Todo el trabajo realizado tiene que cumplir con este mínimo de calidad en todo momento. La solución tiene que ser “lo suficientemente buena” para ser utilizada de manera efectiva. Para cumplir este principio los equipos DSDM necesitan:

- Acordar el nivel de calidad antes de que comience el desarrollo
- Asegurarse de que la calidad no se convierte en una variable
- Hacer pruebas desde el inicio y durante todo el desarrollo
- Diseñar y documentar apropiadamente

Asegurar que las pruebas se integren correctamente en el proceso de Desarrollo iterativo, con revisiones periódicas durante todo el ciclo de vida del proyecto, ayuda al equipo de DSDM a crear una solución de calidad. Los productos de revisión y control de calidad creados a medida que avanza el proyecto ayudan a demostrar que la calidad de la solución cumple con el estándar esperado.

En un proyecto de TI, el uso de técnicas de desarrollo basadas en pruebas también puede mejorar significativamente la calidad de la solución, asegurando que se comprenda la aceptabilidad de la solución antes de que comience el desarrollo.

5. Crecer progresivamente desde bases firmes → DSDM aboga primero por comprender el alcance del problema comercial que se va a resolver y la solución propuesta. Una vez que se han establecido las bases del desarrollo, se entrega la solución final incrementalmente a fin de ofrecer un beneficio empresarial real tan pronto como sea posible. Para cumplir este principio, los equipos DSDM necesitan:

- Analizar apropiadamente y con suficiente detalle el diseño para crear bases sólidas
- Evaluar las prioridades y analizar con cada incremento, de manera rápida, la viabilidad del proyecto

Los equipos de DSDM implementan este principio a través de la aplicación apropiada de un ciclo de vida del proyecto, que proporciona una base sólida de conocimiento durante las fases de Factibilidad y Fundamentos antes de construir la solución de forma incremental durante la fase de Desarrollo Evolutivo.

6. **Desarrollar iterativamente** → DSDM usa una combinación de desarrollo iterativo, demostraciones frecuentes y revisión exhaustiva para alentar la retroalimentación oportuna. Adoptar cambios como parte de este proceso evolutivo permite al equipo converger en una solución comercial precisa. Para cumplir este principio, los equipos de DSDM necesitan:

- Recopilar comentarios de negocios en cada iteración
- Reconocer que la mayoría de los detalles deberían surgir durante el desarrollo
- Aceptar el cambio
- Usar el desarrollo iterativo para fomentar la creatividad, la experimentación y el aprendizaje

Dentro de las limitaciones de tiempo y costo, el cambio se fomenta activamente a fin de desarrollar la solución más adecuada. DSDM usa la iteración y la revisión constante para asegurarse de que lo que se está desarrollando es lo que la empresa realmente necesita. Los ciclos de retroalimentación deben formar parte del proceso para la evolución de todos los productos del proyecto.

7. **Comunicar claro y continuado** → una mala comunicación entre las partes interesadas del proyecto puede llevar al fracaso del mismo. Las prácticas de DSDM están diseñadas para mejorar este principio. Para cumplir este principio, los equipos de DSDM necesitan:

- Comunicación cara a cara a todos los niveles
- Reuniones diarias del equipo

- Usar prácticas de comunicación visual como prototipos. Estas prácticas son mucho más efectivas que el uso de documentos textuales de gran tamaño, que a veces se escriben por razones distintas a la consecución de los objetivos comerciales del proyecto.
- Crear y mantener la documentación necesaria
- Conocer las expectativas de los interesados durante todo el proyecto
- Transparencia en todas las intervenciones

8. Demostrar control → es primordial tener control de todo el proyecto durante todo el ciclo de vida. Esto se suele lograr haciendo referencia a un plan para el trabajo que se realiza, alineado con los objetivos comerciales acordados. Para cumplir con este principio, los equipos de DSDM, especialmente el Gerente de proyecto y el Líder de equipo, necesitan:

- Visibilidad de los progresos y de lo planificado
- Medir el progreso de las entregas
- Gestionar proactivamente
- Utilizar un nivel apropiado de formalidad y dar la información pertinente
- Evaluar la viabilidad del proyecto en función de los objetivos comerciales

El uso de Timeboxes bien definidos, con puntos de revisión constantes, y la preparación de los Fundamentos de Gestión y los Planes de Timebox, están diseñados para ayudar al Gerente del Proyecto y al resto del equipo del proyecto a seguir este principio.

4.6.4 ROLES Y RESPONSABILIDADES

Las personas que trabajan juntas de manera efectiva son la base de cualquier proyecto exitoso. DSDM reconoce esto y asigna funciones y responsabilidades claras a cada persona en un proyecto, representando los intereses comerciales, la solución / intereses técnicos, los intereses de la administración y los intereses del proceso. Todos los que participan en un proyecto de DSDM trabajan muy juntos para romper las posibles barreras de comunicación.

Las mejores soluciones surgen de equipos empoderados y auto-organizados. Sin embargo, estos equipos, y las personas dentro de ellos, deben asumir activamente la responsabilidad de su empoderamiento dentro de los límites que se han acordado. Al mismo tiempo, es importante que ellos:

- Respeten el conocimiento, las habilidades, la experiencia y las aportaciones del resto del equipo y todos los interesados.
- Asuman la responsabilidad de su trabajo y la dependencia de otros miembros
- Mejoren la colaboración del grupo y los procesos de trabajo

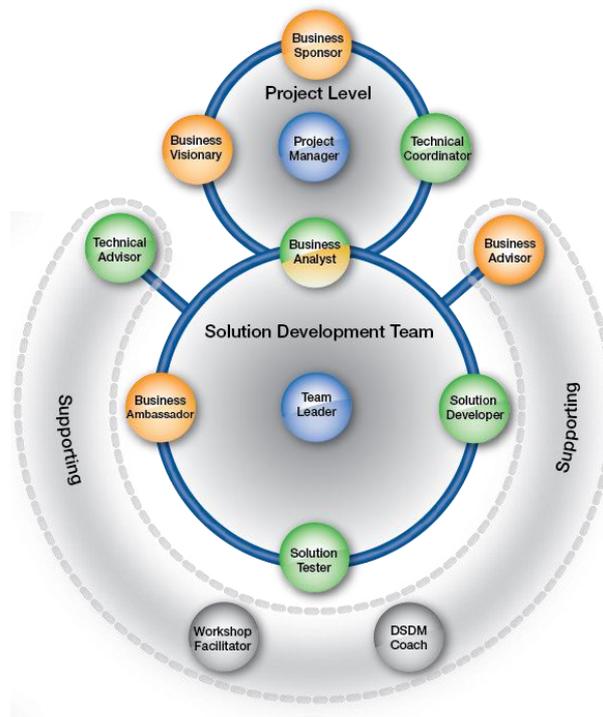


Figura 4.5 - Esquemas de roles de DSDM. Extraído de Agile Business

Según Rodríguez, M (2017) los roles están agrupados por intereses:

Naranja → Business-Oriented Roles → Perspectiva de negocio

- **Business Ambassador:** Persona clave para representar a negocio en el equipo de desarrollo. Participa de manera muy activa en la creación y priorización de requisitos durante *Feasibility* y *Foundations*, es el responsable del detalle y la priorización durante el desarrollo. Rol muy parecido al Product Owner, toma las decisiones de negocio respecto al equipo.

- **Business Visionary:** nos proporciona la visión del proyecto, interpretando las necesidades del Sponsor y de los futuros usuarios de la solución, es el responsable de comunicarlo y marcar la dirección a seguir a todos los integrantes del equipo. Para evitar posibles duplicidades o inconsistencias en la visión, se recomienda que este rol lo lleva a cabo una única persona.
- **Business Sponsor:** nivel más senior a nivel de proyecto. Es el responsable del Business Case y del presupuesto del proyecto.
- **Business Advisor:** Rol de soporte que nos puede proporcionar conocimiento de negocio específico, que no encontramos en otros miembros del equipo de proyecto. Considerado un *Subject Matter Expert* de negocio, puede representar a usuarios, focus groups, o aspectos legales / compliance a tener en cuenta.

Verde → Solution/technical oriented Roles → Perspectiva técnica

- **Technical Advisor:** Rol de soporte parecido al Business Advisor, pero más orientado a aspectos técnicos a tener en cuenta en la solución. Por ejemplo, requisitos a cumplir una vez esté puesta en producción, conocimiento profundo de la tecnología utilizada, o soporte técnico.
- **Solution Tester:** DSDM hace énfasis en definir y asegurar un nivel de calidad adecuado, y el Solution Tester es la persona responsable de definir y ejecutar los test según la estrategia acordada. Muy parecido al de Team Member en Scrum, pero especificando skills más orientados a testing.
- **Solution Developer:** Es capaz, junto con sus compañeros del equipo de desarrollo, de coger los requisitos y transformarlos en un incremento de la solución que cumpla las necesidades técnicas y de negocio. Rol muy parecido al de Team Member en Scrum, pero especificando skills más orientados a desarrollo.
- **Technical Coordinator:** Principal autoridad técnica del proyecto, nos asegura que estamos diseñando una solución alineada con las necesidades técnicas de la organización, que disponemos de los skills adecuados en los roles técnicos, y que éstos trabajan de manera adecuada. Es el equivalente al *Business Visionary* pero desde una perspectiva técnica.

Azul → Management/leadership orientes Roles → Perspectiva de gestión y liderazgo

- **Project Manager:** Además de proporcionar un liderazgo siguiendo principios ágiles al Solution Development Team, su papel se centra en la gestión del entorno de trabajo en el que evoluciona la solución. Facilitador y coordinador de la gestión del proyecto, delega en el equipo los detalles y escala los posibles problemas que puedan surgir que vayan más allá de la capacidad de decisión del equipo.
- **Team Leader:** Ayuda a que el equipo sea lo más productivo posible, actuando como líder a su servicio. Rol parecido al Scrum Master, idealmente es elegido por sus compañeros del Solution Development Team, y facilita las diferentes reuniones del equipo.

Gris → Process-oriented Roles → Perspectiva de procesos

- **Workshop Facilitator:** Los workshops son una de las prácticas recomendadas en DSDM, de modo que disponer de una persona neutral, con los skills necesarios para facilitar este tipo de talleres, es esencial para su éxito.
- **DSDM Coach:** En una transición ágil, la adopción de ciertas técnicas y procesos puede ser difícil, y más aún lo es el cambio de *mindset*. La aportación de un *coach* con experiencia práctica en agile, y con conocimiento del framework en el que nos vamos a apoyar, es un factor clave de éxito de esta transición.

Mezcla de colores → son los roles que abarca dos áreas de interés separadas.

- **Business Analyst:** Único rol con intereses duales, tiene suficiente conocimiento del negocio como para entender las necesidades, y suficiente conocimiento técnico como para asegurar que la solución cumple tanto los requisitos funcionales como los no funcionales. Es un miembro del Solution Development Team, y a la vez a nivel de proyecto.

En total, DSDM tiene definido 13 roles, muchos más que los que tienen otras metodologías. También es verdad que DSDM está cubriendo explícitamente un alcance mayor que otras metodologías, incluyendo aspectos que hay que tratar como los de antes de comenzar el proyecto.

Por otra parte, una persona que forma parte del equipo DSDM puede tener más de un rol, cubriendo siempre todas las responsabilidades.

4.6.5 ESTRUCTURA

DSDM consiste en 3 fases: fase del pre-proyecto, fase del ciclo de vida del proyecto, y fase del post-proyecto.

La Fase I: PRE-PROYECTO

En esta fase se decide lo que se pretende conseguir y quienes van a participar para conseguirlo.

La Fase II: CICLO DE VIDA DEL PROYECTO

Esta fase se divide en 5 etapas:

1. Estudio de viabilidad: Se evalúa si la aplicación es viable, para el proceso teniendo en cuenta los requisitos básicos del negocio y sus restricciones asociadas.
2. Estudio de la empresa: Se establecen los requisitos funcionales y de la información con el fin de permitir un valor al negocio; Además, se define la arquitectura básica de la aplicación.
3. Iteración del modelo funcional: Todos los prototipos del MDSO están diseñados para evolucionar hacia la aplicación entregable.
4. Diseño e iteración de la estructura: Se revisa la construcción de prototipos durante la iteración del modelo funcional. En algunas ocasiones la iteración de construcción de diseño suceden de forma concurrente.
5. Implementación: Se entrega una versión del sistema con el fin de capacitar al usuario y evaluar detalladamente los documentos de sistema.

La Fase III: POST-PROYECTO

Esta fase se trata de asegurar la validez del producto que se ha creado, realizando pruebas y viendo si es lo esperado.

4.7 CONCLUSIÓN

Según se ha visto anteriormente, las metodologías se dividen en dos grupos: **las tradicionales y las ágiles**. Las metodologías tradicionales se centran en el cumplimiento de los procesos y su control, siguiendo unas etapas y procesos perfectamente definidos. Estos métodos en muchos casos han resultado ser efectivos y exitosos, sobre todo cuando se trata de un proyecto de grandes dimensiones respecto al tiempo y a los recursos necesarios.

Por otro lado, las metodologías ágiles, dan mucha más importancia a las personas que van a participar en el proyecto, siendo fundamental la comunicación entre todos sus miembros: cliente, director de proyecto, equipo del proyecto, interesados. No se centran en una entrega final con el producto terminado, sino que busca un desarrollo iterativo e incremental. Estas nuevas metodologías están dando muy buenos resultados por la adaptabilidad y flexibilidad que tiene hacia los cambios, ya sean de características del producto requisitos o tiempo de desarrollo.

El objetivo de este trabajo es, comprobar que factores son influyentes a la hora de que un proyecto de desarrollo de software termine de manera exitosa o no.

COMPARACIÓN DE METODOLOGÍAS DE DIRECCIÓN DE PROYECTOS

5 COMPARACIÓN DE METODOLOGÍAS DE DIRECCIÓN DE PROYECTOS

5.1 INTRODUCCIÓN

Como hemos visto en los capítulos anteriores, se hace una clasificación de las metodologías desde dos enfoques diferentes de gestión de proyectos: tradicionales y ágiles.

Una buena metodología de gestión de proyectos guiará al director del proyecto a través de un conjunto de actividades controladas, gestionadas y visibles para lograr los resultados del proyecto (Oficina de Comercio Gubernamental, 2009). Pero, la metodología por sí misma no es una condición suficiente o incluso necesaria para el éxito del proyecto. La metodología de gestión de proyectos debe estar alineada con la organización propia y de los clientes. Es importante notar que es la metodología la que debe ser adaptada al proyecto y no viceversa.

Se podría pensar que lo importante para que un proyecto tenga éxito, no es suficiente con el uso de una buena metodología. Sino que es importante conocer varias de estas, y de diferentes enfoques para que la organización pueda adaptar la mejor al proyecto, según convenga.

Las metodologías tradicionales se centran en el cumplimiento de los procesos y su control, siguiendo unas etapas y procesos perfectamente definidos. Estos métodos en muchos casos han resultado ser efectivos y exitosos, sobre todo cuando se trata de un proyecto de grandes dimensiones respecto al tiempo y a los recursos necesarios.

Tradicionalmente, las metodologías de gestión de proyectos como PMBOK y PRINCE2 han tenido una fuerte orientación predictiva. Es decir, a partir del detalle del producto que se quiere elaborar (análisis funcional/técnico, requerimientos funcionales/técnicos, etc,) se definen fases/actividades perfectamente planificadas en el tiempo en base a los recursos disponibles. A partir de esta proyección inicial, el objetivo durante el transcurso del proyecto es conseguir que se cumpla aquello que se había previsto, calendarios, costes, calidad y tiempo (Fabiola, 2015).

Por otro lado, las metodologías ágiles, dan mucha más importancia a las personas que van a participar en el proyecto, siendo fundamental la comunicación entre todos sus miembros: cliente, director de proyecto, equipo del proyecto, interesados. No se centran en una entrega final con el producto terminado, sino que busca un desarrollo iterativo e incremental. Estas nuevas metodologías están dando muy buenos resultados por la adaptabilidad y flexibilidad que tiene hacia los cambios, ya sean de características del producto requisitos o tiempo de desarrollo. Según Pérez

(2014) los proyectos gestionados con metodologías ágiles se inician sin un detalle cerrado de lo que va a ser construido. A nivel comercial, los proyectos pueden ser vendidos como servicios y no como productos.

5.2 COMPARACION ENTRE LAS METODOLOGIAS TRADICIONALES

Entonces, si se dividen en dos grandes grupos: tradicionales y ágiles. ¿Qué diferencia hay entre metodologías del mismo enfoque? El objetivo de este capítulo es aclarar las diferencias principales que hay entre estos métodos y el enfoque que le da cada uno de ellos a los proyectos, cómo se aplica, y a qué.

Vamos a analizar más en profundidad dos de las metodologías antes explicadas: PMP y PRINCE2. (Véase [Tabla 5.1](#))

Según el análisis de características generales:

Tabla 5-1 - Diferencias entre las características generales entre PMI y PRINCE2. Elaboración propia

PMI	PRINCE2
Estados Unidos	Reino Unido
PMI	AXELOS
Importa “la forma”	Importa “el resultado”
Colección de buenas prácticas para la gestión de proyectos	Método de gestión de proyectos
Descriptivo	Prescriptivo
Impulsado por los requisitos del cliente	Impulsado por un caso de negocio
Cada tema se puede consultar aisladamente del resto	Un conjunto integrado de procesos y componentes (no son elementos aislados que se puedan aplicar de manera independiente)
Orientado a los gerentes de proyectos	Cubre los roles de la gestión de proyectos definiendo los roles y sus responsabilidades
Cubre las competencias interpersonales	No cubre las competencias interpersonales
Incluye la gestión por fases	Se divide el proyecto en una serie de fases que facilita su planificación asignación de tareas, supervisión y control.

Según los procesos:

Tabla 5-2 - Diferencias entre los procesos entre PMI y PRINCE2. Extraída de Revista científica. Fernández, K., Garrido, A., Ramirez, Y., y Perdomo, I. (2015).

PMI	PRINCE2
Iniciación	Puesta en marcha de un proyecto Dirección del proyecto
Planificación	Inicio del proyecto Gestión de los límites de la fase Gestión de la entrega de productos
Ejecución	Control de la fase Gestión de la entrega de productos
Seguimiento y control	Dirección del proyecto Control de la fase Gestión de los límites de la fase
Cierre	Gestión de los límites de la fase Cierre del proyecto

Según las áreas de conocimiento definidas por el PMBOK y los procesos y componentes de PRINCE2:

Tabla 5-3 - Diferencias entre las áreas de conocimiento entre PMI y PRINCE2. Extraída de Revista científica. Fernández, K., Garrido, A., Ramez, Y., y Perdomo, I. (2015).

PMI	PRINCE2
Integración	Justificación continua del negocio Gestión por excepción
Tiempo	Plan de proyecto plan de fase, plan de equipo, plan de revisión de beneficios
Alcance	Planes, caso de negocio, progreso Técnica de planificación basada en el producto donde se realiza la estructura de descomposición del producto (EDT), se escriben las descripciones detalladas y se realiza el diagrama de flujos vs WBS (es el proceso de subdividir los entregables del proyecto y el trabajo del proyecto en componentes más pequeños y más fáciles de manejar)
Costos	Aborda de manera general algunos aspectos sobre la gestión del valor ganado
Calidad	Orientación al producto Lecciones aprendidas y mejora continua Gestión de la configuración
Riesgos	Similitud en cuanto a la gestión de riesgos
Comunicaciones	Progreso Estrategia de gestión de las comunicaciones
Recursos involucrados	humanos Se definen los roles y responsabilidades del equipo de gestión del proyecto y se hace mención sobre la gestión de recursos materiales
Adquisición	No cubierto
Interesados	Estrategia de gestión de las comunicaciones

Una de las diferenciaciones que tienen las metodologías tradicionales es respecto a las condiciones para conseguir su certificación.

Para comparar estas metodologías se han utilizado unos indicadores para que nos proporcione las diferencias principales entre estas metodologías.

Tabla 5-4 - Indicadores para la comparación entre los sistemas de certificación. Obtenida de Cardoza, R.A., Dante, G.C., De los Ríos, C. I. (2011).

INDICADORES	LEYENDA
1. Acreditación del modelo	Permite diferenciar los sistemas de acuerdo a procesos de validación que hayan superado: 1: No certificado por ninguna norma 2: Acreditado por la Norma ISO 17024 3: Certificado en la Norma ISO 9001 y en la ISO 17024
2. Niveles de certificación: cantidad	Diferencia los sistemas según los niveles de certificación: 1: Establece 2 niveles de certificación 2: Establece 3 de certificación 3: Establece 4 niveles de certificación
3. Niveles de certificación: accesibilidad	Diferencia los sistemas según permiten acceder de un nivel de certificación a otro: 1: No es necesario poseer un nivel inferior para acceder a uno superior 2: Es necesario poseer un determinado nivel inferior para acceder a los niveles superiores
4. Alcance de la certificación	Diferencia los sistemas según si los aspirantes pueden acceder a certificarse en: 1: Dirección de proyectos y programas 2: Dirección de proyectos, programas y carteras
5. Orientación de la dirección de proyectos	Permite diferenciar los sistemas según cómo dividen la dirección de proyectos en sus guías y estándares: 1: Por competencias 2: Por procesos 3: Por segmentos de proyecto
6. Forma de la organización	Diferencia los sistemas según el tipo de organización que forman: 1: Confederación de asociaciones 2: Asociación de membrecías 3: Organismo con organizaciones de entrenamiento acreditadas

Tabla 5-4 – Continuación. Cardoza, R.A., Dante, G.C., De los Ríos, C. I. (2011).

INDICADORES	LEYENDA
<p>7. Requisitos del modelo: experiencia en dirección de proyectos</p>	<p>Establece los niveles de experiencia en dirección de proyectos que se solicitan para acceder al proceso de certificación:</p> <ul style="list-style-type: none"> 1: No exige experiencia para ningún nivel de certificación 2: Se exige un mínimo de años de experiencia para algunos niveles y para los niveles inferiores no es obligatoria
<p>8. Requisitos el modelo: formación o educación exigida en dirección de proyectos</p>	<p>Establece la exigencia de formación o educación inicial en dirección de proyectos para acceder al proceso de certificación:</p> <ul style="list-style-type: none"> 1: No se exige formación o educación inicial 2: Se exige formación o educación
<p>9. Proceso de certificación: sistema de evaluación</p>	<p>Diferencia los sistemas de evaluación que se realizan:</p> <ul style="list-style-type: none"> 1: Sólo evaluación escrita (examen de conocimientos) 2: Evaluación escrita y entrevista presencial con los evaluadores
<p>10. Medida de la experiencia en dirección de proyectos</p>	<p>Establece la unidad de medida de la experiencia en dirección de proyectos que se requiere para acceder a ciertos niveles de certificación:</p> <ul style="list-style-type: none"> 1: No se pide experiencia 2: Se pide experiencia en años 3: Se pide experiencia en número de proyectos
<p>11. Validez</p>	<p>Diferencia los sistemas de acuerdo al tiempo de validez de los certificados expendidos:</p> <ul style="list-style-type: none"> 1: Certificados de 3 y 5 años 2: Certificados sólo de 5 años
<p>12. Propósito de las guías y estándares: proyectos</p>	<p>Establece si el propósito por el que han sido desarrollados los estándares y guías es brindar conocimiento y prácticas para la dirección de proyectos individuales:</p> <ul style="list-style-type: none"> 1: No tiene guías y estándares con este propósito 2: Si tiene guías y estándares con este propósito

Tabla 5-4 - Continuación. Obtenida de Cardoza, R.A., Dante, G.C., De los Ríos, C. I. (2011).

INDICADORES	LEYENDA
<p>13. Propósito de las guías y estándares: organizaciones</p>	<p>Establece si el propósito por el que han sido desarrollados los estándares y guías es brindar conocimiento y prácticas para la dirección de proyectos empresariales:</p> <p>1: No tiene guías y estándares con este propósito</p> <p>2: Si tiene guías y estándares con este propósito</p>
<p>14. Propósito de las guías y estándares: personas</p>	<p>Establece si el propósito por el que han sido desarrollados los estándares y guías es el desarrollo, evaluación y certificación de personas:</p> <p>1: No tiene guías y estándares con este propósito</p> <p>2: Si tiene guías y estándares con este propósito</p>
<p>15. Modificación de guías y estándares</p>	<p>Diferencia los sistemas que permiten que sus asociaciones nacionales modifiquen un porcentaje del contenido de los estándares y guías:</p> <p>1: No permite modificaciones en sus estándares y guías</p> <p>2: Si permite modificaciones en sus estándares y guías</p>
<p>Tabla 5.4 → Indicadores para la comparación entre los sistemas de certificación. Extraída de “Comparación de cuatro sistemas de certificación del ámbito de la dirección de proyectos. Cardoza</p>	

Tabla 5-5 - Comparación de sistemas de certificación. Obtenido de Cardoza, R.A.,Dante, G.C.,De los Ríos,C. I.(2011).

	PMI	IPMA	PRINCE2
1. Acreditación del modelo	3	3	2
2. Niveles de certificación: cantidad	2	3	1
3. Niveles de certificación: accesibilidad	1	1	2
4. Alcance de la certificación	1	2	1
5. Orientación de la dirección de proyectos	2	1	2
6. Forma de la organización	2	1	3
7. Requisitos del modelo: experiencia en dirección de proyectos	2	2	1
8. Requisitos el modelo: formación o educación exigida en dirección de proyectos	2	1	1
9. Proceso de certificación: sistema de evaluación	1	2	1
10. Medida de la experiencia en dirección de proyectos	2	2	1
11. Validez	1	2	2
12. Propósito de las guías y estándares: proyectos	2	2	1
13. Propósito de las guías y estándares: organizaciones	2	1	2
14. Propósito de las guías y estándares: personas	2	2	1
15. Modificación de guías y estándares	1	2	1

La ISO 21500 no se ha incluido en la comparación de certificaciones, porque no es una metodología como las demás. Es una norma que se elaboró con el objetivo de orientar a las organizaciones en su gestión. No se crea con el fin de obtener una certificación ni cumplir unos requisitos para este fin sino solo como una guía para alcanzar el éxito en la gestión de proyectos.

5.2.1 VENTAJAS DE LAS METODOLOGÍAS TRADICIONALES

Se han recogido una serie de ventajas respecto al enfoque tradicional, que a partir de bibliografía y las referencias indicadas, he elaborado una tabla recopilación con todas ellas.

Tabla 5-6 - Ventajas de las metodologías tradicionales

El inicio del proyecto arranca con todo aprobado adecuadamente	MDAP (2017)	Executive	Master
Ciclo en cascada	MDAP (2017)	Executive	Master
Se planifica todo el proyecto	MDAP (2017)	Executive	Master
Se definen parámetros de control de la calidad	MDAP (2017)	Executive	Master
Persiguen llevar una documentación exhaustiva de todo el proyecto	Conectatics (2017)		
Se definen las etapas claramente y los roles de cada uno	Conectatics (2017)		
Basadas en normas provenientes de estándares	Roger Humberto (2012)		
Proceso más controlado	Roger Humberto (2012)		
Nivel de incertidumbre bajo	Figueroa (2015)		
Adaptación a proyectos grandes (por tamaño, nº de miembros, complejidad...)	Figueroa (2015)		

5.2.2 DESVENTAJAS DE LAS METODOLOGÍAS TRADICIONALES

Se han recogido una serie de desventajas respecto al enfoque tradicional, que a partir de bibliografía y las referencias indicadas, he elaborado una tabla recopilación con todas ellas.

Tabla 5-7 - Desventajas de las metodologías tradicionales

Cierta resistencia a los cambios, lo que provoca altos costes al implementar algún cambio	Roger Humberto (2012) Conectatics (2017)
Numerosas políticas y normas	Roger Humberto (2012)
Ejecuta las etapas una sola vez lo que se define en cada etapa es inamovible y hasta que no finaliza con éxito una etapa no se pasa a la siguiente	Conectatics (2017)
El usuario no ve el producto hasta el final	Conectatics (2017)
Poco feedback, ya que solo hay una entrega final	Elaboración propia
Establecer objetivos desde el inicio supone en muchos casos limitaciones porque el proyecto es volátil y obliga tomar decisiones desde el inicio con mucha incertidumbre	Elaboración propia

5.3 COMPARATIVA DE LAS METODOLOGÍAS ÁGILES

Tabla 5-8 - Comparación de metodologías ágiles. Traducción extraída de Awad,,MA (2005)

	PUNTOS CLAVE	CARACTERÍSTICAS ESPECIALES	DEBILIDAD IDENTIFICADA
SCRUM	Equipos de desarrollo independientes, pequeños, auto-organizados, ciclos de lanzamiento de 30 días	Imponer un cambio de paradigma de “definido y repetible” a “vista de desarrollo de nuevos productos de Scrum”	Si bien Scrum detalla específicamente cómo administrar el ciclo de lanzamiento de 30 días, la prueba de integración y aceptación no se detalla
XP	Desarrollo dirigido por el cliente, pequeños equipos, construcciones diarias	Refactorización: el rediseño continuo del sistema para mejorar su rendimiento y capacidad de respuesta también cambia	Si bien las prácticas individuales son adecuadas para muchas situaciones, la visión general y la gestión
DSDM	Aplicación de controles RAD, uso de timeboxing y equipos DSDM habilitados	Primer método de desarrollo de software realmente ágil, uso de creación de prototipos, varias funciones de usuario: “embajador”, “visionario” y “asesor”	Si bien los métodos están disponibles, solo los miembros del consorcio tienen acceso a documentos técnicos que tratan sobre el uso real del método

5.3.1 VENTAJAS DE LAS METODOLOGIAS AGILES

Tabla 5-9 - Ventajas de las metodologías ágiles

Rápida respuesta ante los cambios	Wordpress
Modelo flexible	
Gestión de historias que constituyen la visión	MDAP Executive Master (2017)
Parte de exploración y testeo del mercado	MDAP Executive Master (2017)
Adaptación en base a los resultados de la exploración de mercado	MDAP Executive Master (2017)
Entregas parciales del producto	Conectatics (2017)
Intervención del cliente en el proceso	

5.3.2 DESVENTAJAS DE LAS METODOLOGIAS AGILES

Tabla 5-10 - Desventajas de las metodologías ágiles

Estructura muy débil como consecuencia de su flexibilidad	Pastrana (2014)
La necesidad de equipo puede suponer una desventaja si trabajamos con equipos como colaboradores	Pastrana (2014)
Dependientes de la presencia del mismo equipo de principio a fin	Burke ,R (2013)
Falta de documentación del diseño	Wordpress
Problemas derivados de la comunicación oral	
Restricciones en cuanto a tamaño de los proyectos	
Si el proyecto fracasa, el aprendizaje de errores se queda en los propios desarrolladores, no hay suficiente documentación	
Necesidad de trabajar en el mismo sitio, ya que las metodologías ágiles se basan en la comunicación diaria	Roger Humberto (2012)

5.4 DIFERENCIAS ENTRE LAS METODOLOGÍAS TRADICIONALES Y ÁGILES



Figura 5.1 - Diferencias entre metodologías ágiles y tradicionales. Extraída de PMBOK (2017)

Con todos estos datos explicados en los puntos anteriores, de otras fuentes y ejemplos se desarrolla una tabla resumen de las diferencias principales y generales que hay entre las metodologías tradicionales y las ágiles.

Tabla 5-11 - Diferencia entre metodologías ágiles y tradicionales. Elaboración propia

	Ágiles	Tradicionales
Enfoque	Adaptación	Predictivo
Éxito de medición	Valor del negocio	Conformación de planificar
Tamaño del proyecto	Pequeño	Grande
Estilo de gestión	Descentralizada	Autocrático
Perspectiva para el cambio	Cambio y adaptabilidad	Cambio y sostenibilidad
Cultura	Liderazgo - Colaboración	Comandos de control
Documentación	Bajo	Pesado
Cliente	Parte del equipo	Interactúa mediante reuniones
Énfasis	Orientada a las personas	Orientado a los procesos
Ciclos	Muchos	Limitado
Planificación por adelantado	Minimo	Exhaustivo
Retorno de la inversión	A principios de proyecto	Fin de proyecto
Tamaño del equipo	Pequeños	Grandes

Hablando a nivel global, podríamos decir que las metodologías tradicionales se centran en la planificación, mientras que las ágiles ponen su foco en la ejecución. En lo que se refiere al Project Manager, tradicionalmente es una figura que controla el proyecto y dirige a los miembros del equipo, mientras que en el mundo ágil los

equipos están auto-organizados y auto-dirigidos. Una posee líneas base de tiempo, coste y alcance, la otra es adaptable al cambio, y su restricción principal es la calidad del producto y la rapidez... No podemos concluir que haya una mejor que otra, simplemente distintas, nacidas para diferentes tipos de proyecto.

Las metodologías ágiles son una alternativa interesante para superar las debilidades de las metodologías convencionales, pero, al igual que los computadores no son en sí mismos la solución a los problemas de procesamiento de información, éstas no son la solución a todos los problemas que enfrenta el desarrollo de software. Con el surgimiento de las metodologías ágiles, el concepto de etapa se desvanece dando paso a la idea de actividades, las cuales pueden ser organizadas a comodidad del equipo de trabajo, en paquetes pequeños conservando las mismas labores e identidad de las etapas concebidas en las primeras metodologías. Es claro que la informática y la computación son ciencias nuevas, sin embargo ya es tiempo de que se les concedan espacios diversos y espontáneos en los que las personas que participan en los proyectos del área puedan sentirse más productivos con base en las circunstancias particulares de cada proyecto. (Canós, J. H., & Letelier, M. C. P. P. (2012). Metodologías ágiles en el desarrollo de software.)

ESTUDIO DE PROYECTOS DE DESARROLLO DE SOFTWARE Y FACTORES DE FRACASO

6 ESTUDIO DE PROYECTOS DE DESARROLLO DE SOFTWARE Y FACTORES DE FRACASO

6.1 ESTUDIO DE PROYECTOS DE DESARROLLO DE SOFTWARE

Para extraer unos factores que se puedan utilizar para hacer una comparativa de por qué hay proyectos que fracasan y otros que se resuelven con éxito. Se desea comprobar en qué medida existen unos proyectos y otros y cuáles son las variables que hacen que finalicen de esta manera.

Se ha accedido a un estudio de 2015, las últimas actualizaciones, del CHAOS Report de Standish Group, que ofrece unos resultados estadísticos sobre la base de 50.000 proyectos encuestados y analizados, proyectos repartidos por todo el mundo, con diferentes tamaños de proyectos, complejidades diferentes y con equipos grandes y pequeños. Estas estadísticas están basadas en metodologías ágiles y tradicionales de desarrollo de software.

Se trata de un estudio realizado desde 2011 hasta 2015.

En primer lugar vamos a ver la progresión de los resultados de los proyectos en estos años de manera general. (Figura 6.1)

Para ello vamos a definir tres resultados:

- **Exitoso:** clasificamos aquí a los proyectos que han resultado satisfactorios para el cliente, consiguiendo el alcance definido y cubriendo todas las necesidades. Además en un tiempo estimado con el cliente y dentro del presupuesto previsto.
- **Complicado:** son aquellos que no han cumplido algunas de estas premisas, pero tampoco se pueden considerar como proyectos fallidos.
- **Fallido:** son aquellos que no cumplieron ninguna de las tres restricciones.

RESULTADO GENERAL

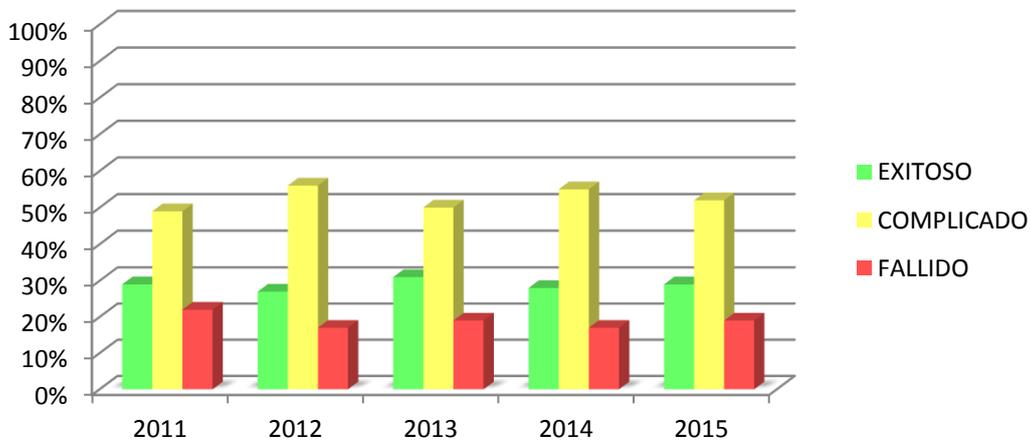


Figura 6.1 - Resultado general. Elaboración propia

Como podemos observar, no hay gran diferencia entre los proyectos realizados en 2011 y 2015, por lo que se puede deducir que no se han encontrado puntos clave de éxito o de fracaso. Ya que no se ha conseguido una mejora en los cinco años. Además se observa que la mayoría de proyectos se quedan en proyectos complicados, esto significa que no se consigue satisfacer todas las restricciones propuestas al inicio del mismo.

Este dato puede resultar interesante, ya que parece que en principio la metodología usada no tiene mucha influencia, ni los ciclos de vida que se plantean en cada proyecto.

Pero se sigue con el estudio, y se intenta dividir según el tamaño del proyecto (Figura 6.2), con el objetivo de encontrar algún patrón que determine como maximizar la posibilidad de éxito y minimizar el fracaso.

Según tamaño del proyecto

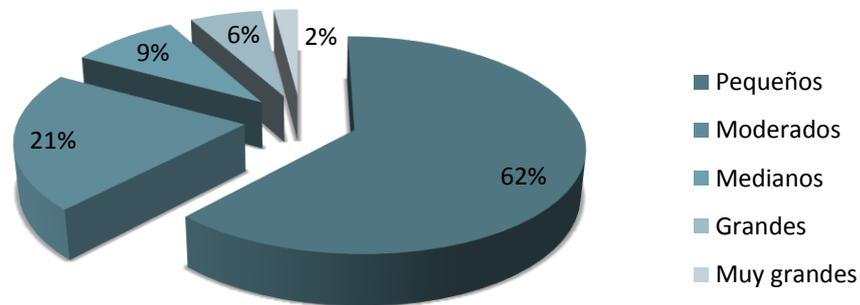


Figura 6.2 - Según tamaño del proyecto. Elaboración propia

Se puede apreciar que cuanto más grande sea el proyecto más posibilidades de fracaso tiene. Esto es obvio, ya que cuanto más grande es más complejo es de organizar y de seguir unas pautas, ya que se maneja mucha información y mucha carga de trabajo. Es más difícil cumplir con los objetivos marcados desde el inicio de tiempo y coste.

Por lo tanto, se puede aprender que para tener más posibilidades de éxito en proyectos muy grandes, se puede dividir y planificar como proyectos más pequeños.

Otra comparativa que ofrece este estudio es en función de la metodología escogida, ágil contra tradicional (Figura 6.3). En este caso lo que nos muestra es la comparación entre **ágil y en cascada**, es una metodología tradicional usada para el desarrollo de software.

Hay que tener en cuenta que como el desarrollo de software generalmente se realiza con modelo en cascada y en los últimos años se ha ido introduciendo la metodología ágil, hay muchos más que siguen el primer modelo. Por tanto, la estadística puede ser un poco engañosa.

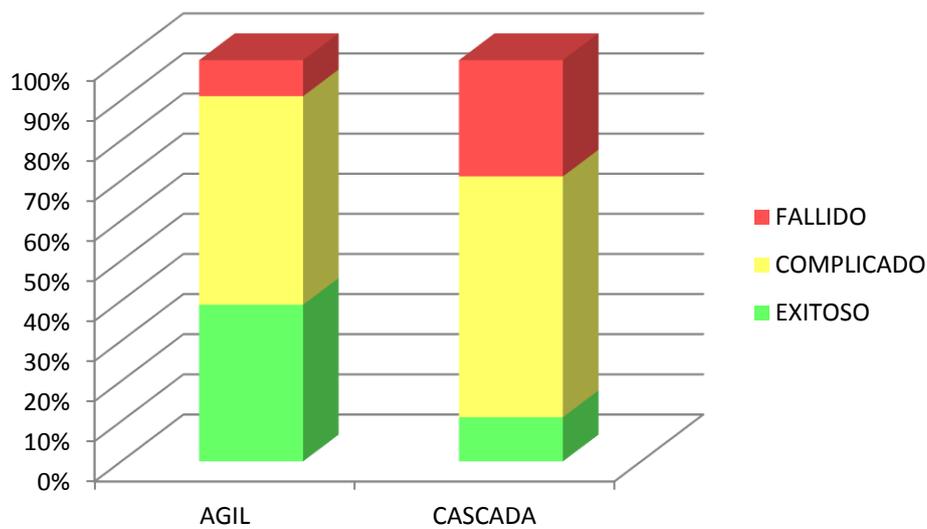


Figura 6.3 - Comparación ágil vs cascada. Elaboración propia

Las conclusiones que se obtienen es que con la metodología ágil hay más probabilidades de terminar el proyecto con éxito y menos de fracaso. Con el método de cascada, hay muy pocos éxitos y muchos más complicados.

No se puede decir que la metodología agil es mejor, porque no especifica el tamaño del proyecto.

En el siguiente gráfico, sí que se va a desglosar por tamaño y se obtendrá más información fiable de qué tipo de método es mejor dependiendo del tamaño del proyecto. (Figura 6.4)

PROYECTOS PEQUEÑOS

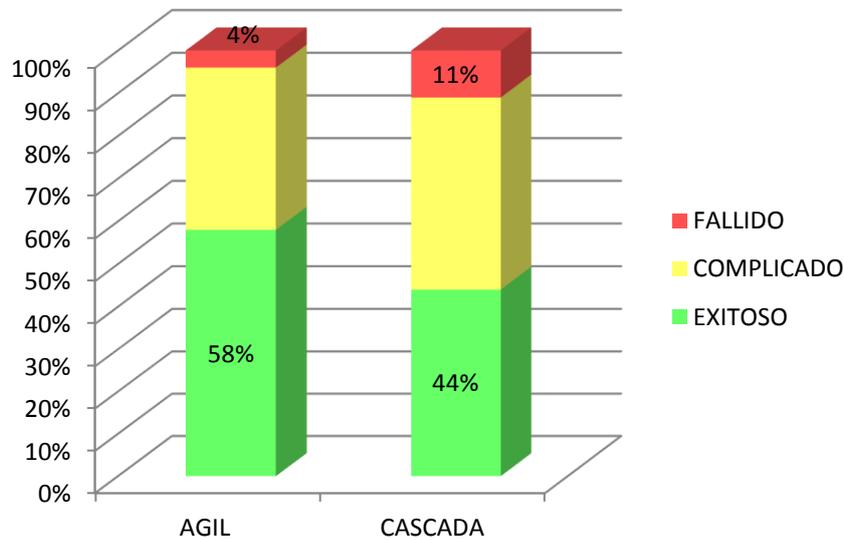


Figura 6.4 - Comparación en proyectos pequeños. Elaboración propia

En primer lugar, se analizan los proyectos pequeños, se observa un porcentaje bastante elevado en las dos metodologías, siendo un poco mayor en las ágiles. Y los fracasos bastante escasos en los dos casos. Por lo tanto, podríamos decir que mientras sean proyectos de tamaño pequeño, la metodología no influye en exceso.

Por otro lado, en los proyectos grandes (figura 6.5) hay muy poco porcentaje de éxito si el método que elegimos es el modelo en cascada. Y un alto porcentaje de fracaso. Mientras que en las ágiles, en comparación, hay más porcentaje de éxito y menos de fracaso.

En conclusión, si el proyecto es grande sí que es mejor utilizar una metodología ágil, frente a las tradicionales. Mientras que el proyecto sea pequeño, es independiente la metodología a elegir.

PROYECTOS GRANDES

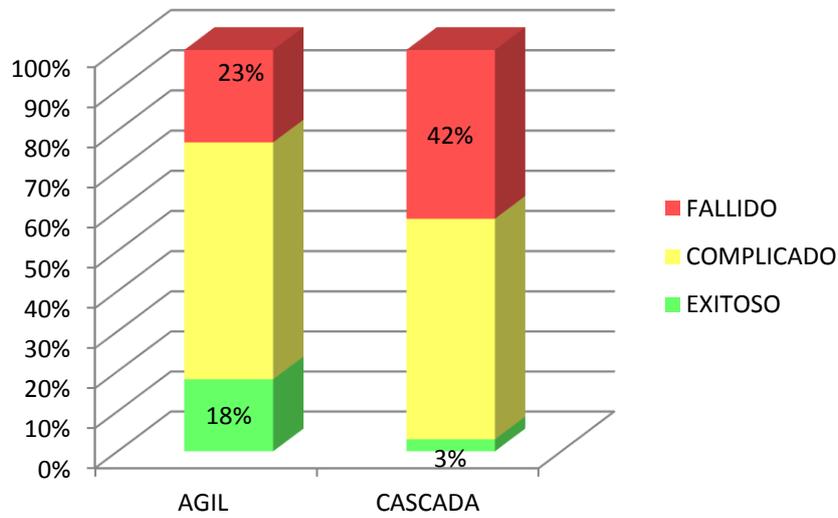


Figura 6.5 - Comparación en proyectos grandes. Elaboración propia

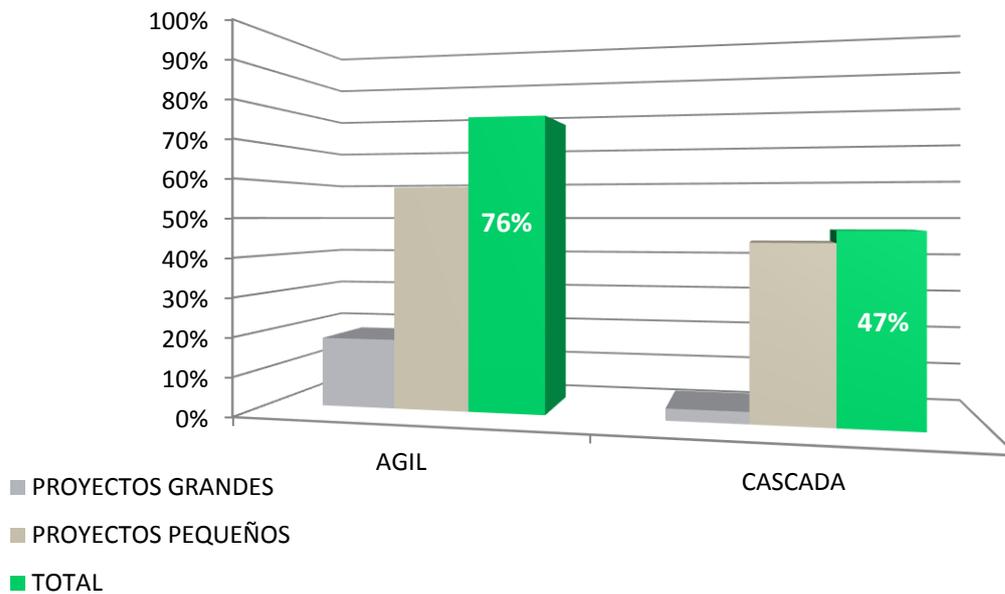


Figura 6.6 - Comparación de proyectos grandes y pequeños. Elaboración propia

Gracias a este estudio, se ve que la mejor manera de llevar un proyecto es dividirlo en proyectos más pequeños, siguiendo las prácticas ágiles y que se adapten al proyecto de la manera más adecuada.

Para poder sacar unas conclusiones más fiables, se estudiará de una manera más profunda cuáles son los factores que influyen en el resultado de los proyectos.

6.2 FACTORES DE FRACASO EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE

Según la investigación se han encontrado unos puntos clave comunes que suelen estar entre todos los proyectos con resultado fallido.

En primer lugar se encuentran factores respecto a la organización del proyecto:

a. Falta de patrocinio ejecutivo

Para que los proyectos puedan llevarse a cabo, no solo hace falta una organización, también los recursos necesarios para que el proyecto se pueda desarrollar.

Esta falta de compromiso se puede deber a que los patrocinadores tengan una sobrecarga laboral, por la cultura organizacional de la empresa. Cuando esto sucede en el 33% de los proyectos fallidos es porque las capacidades para eliminar obstáculos se ven afectadas, un 45% de los proyectos se ve afectado por el retraso del cronograma y un 28% porque el equipo se ve retrasado por la manera de tomar decisiones. También por falta de comunicación o por falta de desarrollo profesional de los mismos.

Según un estudio de PMI y la consultora BCG de octubre de 2014, el patrocinio ejecutivo ha adquirido más importancia en los últimos años. Sin embargo, conseguir un patrocinio de alta calidad es muy difícil.

La falta de patrocinio es un motivo de fallo pero tan importante como esto es el exceso del mismo.

b. Cultura de organización demasiado tradicional

Se entiende por cultura organizacional el conjunto de normas, valores y creencias que guían el comportamiento del personal dentro de una empresa. Aunque existen tantos modelos organizacionales como empresas, pero según Roger Harrison ((Harvard Business Review) define cuatro tipos de orientaciones, que son:

- Organizaciones orientadas **al poder**: se caracterizan por el dominio (interno/externo) y el control directivo. Se aplica el principio de autoridad y la organización se basa en una estructura piramidal. Se da mucha importancia a la planificación.

- Organizaciones orientadas **a la norma**: se basa en procedimientos estructurados, aplicando siempre las normas y métodos adecuados y la racionalidad y planificación. En estas organizaciones hay exceso de burocratización.
- Organizaciones orientadas **a los resultados**: en este tipo de organización se busca conseguir el objetivo, y no se centra tanto en seguir los métodos. Da mucha importancia al personal y al trabajo participativo, de esta manera se consigue un trabajo más eficaz.
- Organizaciones orientadas **a las personas**: lo más valioso son las personas que forman la organización, es el mayor recurso para que los proyectos salgan delante de una manera satisfactoria.

Una vez vistas este tipo de organizaciones que pueden existir, se puede decir que la estructura orientada al poder que era la más general hace unos años, se tiene que ir perdiendo para convertirse en organizaciones donde el objetivo principal sea la satisfacción del cliente y todos los miembros tengan claro hacia dónde dirigirse cooperando y sintiéndose participes del éxito.

c. Tamaño de la organización demasiado grande

Esto puede suponer una dificultad a la hora de la organización, ya que para coordinar a muchas personas se tiene que tener una visión global del proyecto y transmitírselo al equipo. El director de proyecto debe ser:

- Organizado y eficiente en la planificación
- Líder
- Jefe de equipo
- Motivador
- Capacidad de influencia
- Dotes de comunicación
- Obtener del equipo el mayor rendimiento posible
- Capaz de resolver problemas

Conseguir un buen director de proyecto es complicado, ya que aparte de todas estas capacidades debe ser un perfil que cuente con un perfil técnico que le permita ostentar un mejor entendimiento de los procesos.

Respecto a las personas también vemos algunos factores que hacen que el proyecto no sea completado con éxito.

- d. Falta de habilidades necesarias
- e. Falta de conocimientos en la gestión de proyectos

- f. Falta de trabajo en equipo
- g. Resistencia al trabajo en grupo o de manera individual
- h. Mal relación con el cliente

Respecto a los procesos los fallos que nos podemos encontrar son:

- i. Alcance del proyecto mal definido
- j. Requisitos del proyecto mal definidos
- k. Planificación del proyecto mal definida
- l. Falta de progreso ágil, retrasando así el proyecto
- m. Falta de la presencia del cliente
- n. Rol del cliente mal definido

FACTORES DE ÉXITO EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE

7 FACTORES DE ÉXITO EN LOS PROYECTOS DE DESARROLLO DE SOFTWARE

7.1 CONCEPTO DE FCE

El concepto de FCE nace y se desarrolla desde el ámbito de la disciplina de la dirección estratégica y el management empresarial. El concepto FCE o factores críticos de éxito, fue mencionado por primera vez por Daniel (1961) quien sostenía la necesidad de eliminar temas que no estuvieran directamente relacionados con el éxito de una organización y así conformar sistemas de información eficientes que ayudasen a los directivos de la planificación y gestión de las organizaciones.

El primero que utilizó este término como sistema fue John F. Rockart en un artículo publicado en la revista *Harvard Business Review* (publicado en la edición española en 1981) donde se presenta un nuevo método desarrollado por un equipo de investigación aplicada de la *Sloan School of Management* del Instituto Tecnológico de Massachussets (MIT) que llamaron *Método de los Factores Críticos de Éxito*.

Rockart (1979) desarrolló la perspectiva de Daniel proponiendo el método de los FCE para ayudar a los directivos a especificar sus necesidades de información acerca de los aspectos que son críticos para sus organizaciones de forma que los sistemas de información pudiesen ser desarrollados para satisfacer esas necesidades. Es decir, desarrolla la idea de *cubrir las necesidades de información para los directivos* basándose en FCE, sosteniendo que este concepto es de gran utilidad para la conformación de sistemas de información eficientes. Rockart (1979:85) define FCE de la siguiente forma:

“Los FCE son, para cualquier negocio, un número limitado de áreas en las que los resultados, si éstos son satisfactorios, garantizarán un rendimiento competitivo exitoso para la organización. Estas son áreas principales donde las cosas deben ir bien para que el negocio florezca: si los resultados en dichas áreas no son los adecuados, los esfuerzos de la organización, para ese período, no estarán definidos y los objetivos de gestión no podrán ser alcanzados.”

Destaca también la diferencia entre Factor de Éxito (FE) y FCE. Un FE es algo que puede ocurrir (o no) para conseguir un objetivo. Un FE se considera que es crítico (FCE) cuando su cumplimiento es necesario para alcanzar los objetivos.

Rockart plantea que lo que llama la atención en los sistemas de información es que suelen ofrecer demasiada información que se asimila sólo parcialmente y que, gran parte de ella, no tiene importancia o es poco relevante, concluyendo que el

problema suele estar en la definición exacta de qué datos son los que un gestor necesita. Es decir, introduce la idea de la necesidad de incorporar el criterio de *discriminación y selectividad* de la información, que debe orientarse hacia FCE, para que los sistemas de información sean operativos, eficaces y eficientes.

En definitiva, tanto Daniel como Rockart se centraron en la importancia que supone el suministro de la mejor información para una gestión más efectiva de la planificación y el control. Por tanto, su principal contribución fue haberse centrado en áreas críticas. Siguiendo un orden cronológico, se exponen a continuación diversas definiciones encontradas en la literatura en torno al concepto de FCE:

- “Los FCE son aquéllas variables en las que la dirección puede influir a través de sus decisiones y que pueden afectar, significativamente, las posiciones competitivas generales de las distintas compañías de una industria” (Hofer y Schendel, 1978:77).
- FCE son “características, condiciones o variables que cuando son mantenidas, sostenidas y administradas correctamente pueden tener un impacto significativo en el éxito de una compañía en una industria en particular” (Leidecker y Bruno, 1984:24).
- FCE son “condiciones, circunstancias, actividades o eventos que, específicamente, están limitados en número a las áreas en las que los resultados, si éstos son satisfactorios, podrán asegurar el rendimiento competitivo y con éxito de una organización” (Jenster, 1987: 102).
- FCE son “objetivos, declaraciones finales, características, condiciones o variables que son críticas para alcanzar la misión de la organización y el éxito de la organización” (Hardaker y Ward 1987:114).
- FCE son “el número limitado de áreas en las que los resultados, si éstos son satisfactorios, podrán asegurar el rendimiento competitivo de la organización” (Daft, 1988:618).
- “Los factores más importantes relacionados con el éxito son aquéllos que están relacionados con los objetivos y metas de la compañía” (Pollalis y Grant, 1994:12).

A pesar de que las definiciones y conceptos obtenidos de diferentes autores parecen discrepar en algunos aspectos, se observa un amplio consenso en una serie de características comunes que ayudan a explicar la naturaleza y alcance de los FCE pudiéndose afirmar que:

- Son áreas, aspectos o sucesos limitados de éxito que aseguran en gran medida un rendimiento competitivo.

- Están directamente relacionados con el logro de la visión, misión y objetivos a largo plazo
- Pueden ser factores internos o externos
- Son áreas de resultado en las que el rendimiento se puede medir, controlar y evaluar

Para que los FCE sean estratégicamente manejables es imprescindible identificar los indicadores que puedan usarse para su medición y evaluación (Leidecker y Bruno, 1984; Goodstein *et al*, 1992; Thompson y Strickland, 2002). Así, cada FCE debe contar con uno o más *indicadores clave* que provean la medida y un estándar para medir y evaluar su rendimiento. Los indicadores clave de rendimiento más efectivos son aquellos diseñados de acuerdo con el proceso donde son usados de forma que faciliten una lectura comprensible y continua del rendimiento.

Los indicadores, objetivos o subjetivos, son, generalmente, una mezcla de índices de éxito que pueden ir desde ratios de liquidez o rentabilidad hasta resultados de opiniones de los clientes, la moral de los trabajadores o las actitudes de los agentes implicados tanto internos como externos.

Los FCE varían de una industria a otra, e incluso de época a época dentro de la misma industria a medida que cambian sus fuerzas impulsoras y las condiciones competitivas. Muy rara vez una industria posee más de tres o cuatro factores clave para el éxito en un momento determinado, incluso entre esos tres o cuatro, uno o dos por lo común exceden en importancia a los otros (THOMPSON, 2004:109).

La pregunta que se deriva en este punto es: **¿Cuáles son los FCE competitivos de una industria?** Los FCE de una industria son aquellos aspectos que afectan más a la capacidad de los miembros del sector para prosperar en el mercado, es decir, los elementos particulares de la estrategia, los atributos del producto, los recursos, las competencias, las habilidades competitivas y los resultados de negocios que marcan la diferencia entre utilidades y pérdidas, lo que finalmente significa el éxito o el fracaso competitivo. Por su propia naturaleza, los FCE son tan importantes que todas las empresas de la industria les deben prestar una gran atención, debido a que son los requisitos previos para el éxito, para decirlo de otra forma, los FCE son las reglas que determinan que una empresa sea financiera y competitivamente exitosa (THOMPSON, 2004:109).

Si el diagnóstico de los FCE de la industria es erróneo, a largo plazo se puede incrementar el riesgo de que una estrategia sea mal dirigida, por el contrario si una empresa tiene una adecuada percepción de los FCE de la industria puede lograr una ventaja competitiva sostenible, concentrando su estrategia en los FCE y dedicando sus

energías a ser mucho mejor que los rivales en uno o más de estos factores. De hecho, las compañías que sobresalen en un FCE en particular disfrutan de una posición de mercado más poderosa como recompensa a sus esfuerzos. Ser significativamente superior que los rivales en uno o más de los FCE, significa una oportunidad excepcional para lograr una ventaja competitiva. De manera que, utilizar uno o más de los FCE de la industria como piedra angular de la estrategia de la compañía y tratar de lograr una ventaja competitiva sostenible, al tratar de sobresalir en algún FCE particular, constituye un enfoque fructífero (THOMPSON, 2004:109).

Complementando lo anterior, en una consultora se ha adoptado la sigla MECORA (medible, específico, consensuado, oportuno, realista y alineado) que describe los requisitos a reunir de un FCE como son:

- La medición debe ser distinta a los métodos tradicionales para evaluar la gestión. Esto significa que debe ser crucial para el cumplimiento de la estrategia planteada y su alineación con la misión de la empresa.
- Debe haber una relación causa-efecto identificable entre acciones y resultados
- Debe proveer un marco realista para que los empleados alcancen los objetivos buscados
- Debe promover y ayuda a alcanzar el cambio cultural en el comportamiento del personal
- Debe poder distinguirse entre desempeños eficaces y desempeños ineficaces
- Debe ser cuantificable en un alto alcance reflejado observaciones y documentaciones en los comportamientos relacionados al trabajo
- Debe retroalimentar la información hacia un estándar deseado
- Debe ser percibido de manera clara y sin ambigüedad por los empleados
- Debe tener un seguimiento de sus resultados
- Debe poseer un marco temporal apropiado
- Debe tener en cuenta las capacidades internas las influencias del entorno
- Debe ser un proceso continuo, de permanente aprendizaje y retroalimentación
- Todos los aspectos (áreas) del negocio deben poder ser cuantificables y registrables

7.2 FCE DE UN PROYECTO DE SOFTWARE

No hay ningún método universal en la investigación de los FCE. Consultores e investigadores han utilizado numerosos métodos para su identificación como entrevistas, análisis de las actividades relacionadas con la organización, listas de

fuentes de los factores basadas en la literatura, cuestionarios enviados por correo y entrevistas en combinación con cuestionarios.

Uno de los medios que se puede utilizar para identificar FCE relacionados con el sector TIC es la revisión de documentación. Por lo tanto, se ha elaborado la siguiente tabla con los FCE encontrados a través de la consulta bibliográfica:

Leyva Vázquez, M., & Rosado Rosello, R., & Febles Estrada, A. (2012).

Bermúdez (2003)

Bañales y Adam (2007)

Stankovic, D., Nikolic, V., Djordjevic, M., y Cao, D. (2008)

Tabla 7-1 - FCE para un proyecto de software

Autor	FCE para un proyecto de software
Radosevich (1999)	El líder
Caralli (2004)	Apoyo del gobierno Calidad Recursos humanos Marketing Innovación Liderazgo Apoyo de la dirección
Achanga (2006)	Liderazgo Apoyo
Kassicieh (1998)	Educación Entrenamiento
Sohal (2000)	Entrenamiento de liderazgo
Newton (1995)	Entrenamiento Desarrollo de los gerentes
Bañuelas (2002)	Habilidades básicas
Antony (2002)	Desarrollo de sus empleados Trabajo en equipo

Tabla 7-1 - FCE para un proyecto de software

Autor	FCE para un proyecto de software
Standish Group (1995)	Involucramiento del usuario Apoyo por parte de la alta gerencia Definición clara de requerimientos Planeación apropiada Expectativas realistas Metas intermedias alcanzables Equipo de trabajo competente Sentido de pertenencia al proyecto Visión y objetivos claros Equipo comprometido y disciplinado
Nicholas (2001)	Compromiso Involucramiento de los participantes Buena comunicación Calidad en el intercambio de información
Nasr, Diekmann y Kuprenas (2000)	Adecuada planeación y especificaciones Alcance de trabajo bien definido Solicitud de información Tasas de baja productividad Estimaciones precisas Cambio de ordenes Calendarización realista Retrabajo
Guido y Clements (1999)	Objetivos y misión del proyecto claramente definidos Apoyo de la alta gerencia Un administrador del proyecto competente Un equipo de proyecto competente Recursos suficientes Involucrar y consultar al cliente o usuario final Buena comunicación Responsabilidad del cliente Monitoreo apropiado y retroalimentación Tecnología apropiada

Tabla 7-1 - FCE para un proyecto de software

Autor	FCE para un proyecto de software
<p>Leyva, Rosado y Febles (2012)</p>	<p>Apoyo de la alta gerencia Participación de los usuarios Conocimientos técnicos de los usuarios Definición del alcance y prioridades Calidad de las fuentes de datos Entrenamiento de los usuarios Definición y requisitos Tiempo Compromiso del cliente con la solución Disponibilidad tecnológica</p>
<p>Stankovic, D., Nikolic, V., Djordjevic, M., y Cao, D. (2008)</p>	<p>Fuerte apoyo ejecutivo Compromiso del patrocinador y director Buena comunicación Equipos auto-organizados Equipos con experiencia y conocimiento Reuniones diarias Búsqueda de un diseño simple Entregas parciales Tamaño del proyecto</p>
<p>William LA (2010)</p>	<p>Involucramiento del usuario Soporte de administración ejecutiva Clara definición de requerimientos Planeación adecuada Expectativas realistas Pequeñas entregas del proyecto Persona competente Apropiación del proyecto Visión y objetivos claros Trabajo duro</p>

Como se puede observar, en muchos casos coinciden los FCE de unos autores y otros. Otros hacen pequeñas aportaciones, pero no por ello menos importantes.

Para hacer una comparación clara se elaborará una nueva tabla con los más importantes y relevantes, y serán los que se sometan al estudio de comparación entre metodologías. He decidido clasificar los factores críticos de éxito en grupos según donde afecten.

PROCESOS

1. Calendarización realista
2. Adecuada planeación y especificaciones
3. Recursos suficientes
4. Buena comunicación
5. Tiempo
6. Reuniones diarias

RECURSOS HUMANOS

7. Apoyo de la dirección
8. Liderazgo
9. Desarrollo de los gerentes
10. Habilidades básicas
11. Desarrollo de sus empleados
12. Involucramiento del usuario
13. Involucramiento de los participantes
14. Responsabilidad y compromiso del cliente
15. Equipos auto-organizados
16. Equipos con experiencia y conocimiento

OBJETIVOS Y ALCANCE

17. definición clara de requerimientos
18. Metas intermedias alcanzables
19. visión y objetivos claros
20. Alcance del trabajo bien definido
21. Tamaño del proyecto

CALIDAD

22. Monitoreo apropiado y retroalimentación
23. Calidad de las fuentes de datos
24. Entregas parciales

TECNOLOGÍA E INNOVACIÓN

25. Tecnología apropiada
26. Conocimientos técnicos de los usuarios
27. Disponibilidad tecnológica

7.3 COMPARATIVA DE LAS METODOLOGÍAS RESPECTO A LOS FCE

En este capítulo se va a realizar una comparación de las metodologías respecto a los factores críticos de éxito (FCE) anteriormente seleccionados. Para ello, se ha escogido un sistema de puntuación con un rango del 1 al 3. Asignado el 1 cuando la metodología no tiene en cuenta ese factor, un 2 cuando lo tiene en cuenta pero no directamente, y un 3 cuando es uno de los principios o requisitos que tiene el método para aplicarlo correctamente.

Se ha creado una tabla comparativa (tabla 7.2), que es la siguiente:

Tabla 7-2 - Comparación de las metodologías respecto a los FCE. Elaboración propia

FACTORES CRÍTICOS DE ÉXITO	METODOLOGÍAS TRADICIONALES				METODOLOGÍAS ÁGILES		
	PMP	IPMA	ISO 21500	PRINCE2	SCRUM	XP	DSDM
PROCESOS	13	12	13	12	17	17	17
1. Calendarización realista	2	2	2	2	3	3	3
2. Adecuada planeación y especificaciones	3	2	3	3	3	3	3
3. Recursos suficientes	2	2	2	2	2	2	2
4. Buena comunicación	2	2	2	1	3	3	3
5. Tiempo	3	3	3	3	3	3	3
6. Reuniones diarias	1	1	1	1	3	3	3
RECURSOS HUMANOS	17	18	16	14	23	26	24
7. Apoyo de la dirección	2	2	2	2	2	2	2
8. Liderazgo	3	3	2	3	3	3	2
9. Desarrollo de los gerentes	2	3	2	1	2	2	2
10. Habilidades básicas	3	1	3	1	2	3	2
11. Desarrollo de sus empleados	1	3	1	1	3	3	2
12. Involucramiento del usuario	1	1	1	1	2	3	3
13. Involucramiento de los participantes	1	1	1	1	2	2	3
14. Responsabilidad y compromiso del cliente	1	1	1	1	2	3	3
15. Equipos auto-organizados	1	1	1	1	3	3	2
16. Equipos con experiencia y conocimiento	2	2	2	2	2	2	3
OBJETIVOS Y ALCANCE	13	10	10	11	12	12	12
17. Definición clara de requerimientos	3	3	2	2	3	3	3
18. Metas intermedias alcanzables	1	2	1	2	3	3	3
19. visión y objetivos claros	3	2	2	3	2	2	2
20. Alcance del trabajo bien definido	3	1	3	2	2	2	2
21. Tamaño del proyecto	3	2	2	2	2	2	2
CALIDAD	5	5	4	6	8	8	9
22. Monitoreo apropiado y retroalimentación	3	3	2	3	3	3	3
23. Calidad de las fuentes de datos	1	1	1	1	2	2	3
24. Entregas parciales	1	1	1	2	3	3	3
TECNOLOGÍA E INNOVACIÓN	4	4	4	5	8	8	6
25. Tecnología apropiada	1	1	1	2	3	3	2
26. Conocimientos técnicos de los usuarios	2	2	2	2	3	3	2
27. Disponibilidad tecnológica	1	1	1	1	2	2	2
TOTAL	104	98	94	96	136	142	136

Como se observa hay un resultado general, el cual proporciona la comparación total entre las diferentes metodologías.

En este caso, la metodología que aparece con mayor puntuación total es la metodología ágil XP. Pero esta tabla da información más detallada.

En cuanto a cuál es la mejor metodología en el grupo de las tradicionales y de las ágiles por parte de las tradicionales, la que obtiene mayor puntuación general es PMP y de las ágiles como hemos podido observar, XP.

Tabla 7-3 - Comparación entre metodologías ágiles y tradicionales con respecto FCE. Elaboración propia

FACTORES CRÍTICOS DE ÉXITO	METODOLOGÍAS TRADICIONALES				METODOLOGÍAS ÁGILES		
	PMP	IPMA	ISO 21500	PRINCE2	SCRUM	XP	DSDM
TOTAL	102	98	94	92	132	142	136

Esto no significa que solo se puedan usar esas metodologías en concreto, o que te asegures el éxito del proyecto. La decisión de la utilización de una metodología u otra, la tomará el responsable del proyecto.

Además, puede que una metodología se adapte mejor según las necesidades requeridas por el proyecto.

Se va a analizar en mayor profundidad cada uno de los grupos en los que se ha dividido los FCE. Con esta división de FCE en diferentes áreas se pretende saber cuál de las metodologías cumple mejor con estos factores de éxito, por tanto, cual de ella sabe tratar mejor cada una de estas áreas y según las necesidades del proyecto ayude a tomar la decisión de una metodología u otra.

Se va a comenzar por el grupo de procesos (véase tabla 7.4)

Tabla 7-4 - FCE. Área de procesos. Elaboración propia

FACTORES CRÍTICOS DE ÉXITO	METODOLOGÍAS TRADICIONALES				METODOLOGÍAS ÁGILES		
	PMP	IPMA	ISO 21500	PRINCE2	SCRUM	XP	DSDM
PROCESOS	14	13	15	13	15	15	16
1. Calendarización realista	3	3	3	3	2	2	2
2. Adecuada planeación y especificaciones	3	2	3	3	3	3	3
3. Recursos suficientes	2	2	3	2	2	2	2
4. Buena comunicación	2	2	2	1	3	3	3
5. Tiempo	3	3	3	3	2	2	3
6. Reuniones diarias	1	1	1	1	3	3	3

Esta área de procesos está formada por los siguientes FCE:

1. Calendarización realista
2. Adecuada planeación y especificaciones
3. Recursos suficientes
4. Buena comunicación
5. Tiempo
6. Reuniones diarias

Todos ellos se refieren a los procesos llevados a cabo en un proyecto. La calendarización y planeación de los procesos y las tareas a llevar a cabo tienen que ser realistas y cumplirse en el tiempo establecido. Esto se consigue a través de una buena comunicación del equipo, director del proyecto, cliente, participantes e interesados. También es importante asegurarse de poseer los recursos necesarios para completar ese proyecto con éxito.

En principio, el pensamiento general nos lleva a que las metodologías tradicionales cumplen mejor con la planificación y tienen más en cuenta la calendarización con cada una de las tareas desde el inicio hasta el final del proyecto. Pero en la siguiente comparación y con los FCE seleccionados vemos que no es así. Para el caso de proyectos de desarrollo de Software no es tan importante la calendarización de principio a fin de proyecto, porque en este tipo de proyectos, según hemos visto, una de las tareas más complejas es fijar, desde el inicio, los requisitos finales del producto.

Por lo tanto, vemos que las puntuaciones más altas son de las metodologías ágiles. En todas por igual ya que la manera de planificar y llevar a cabo los requisitos y las necesidades del software son similares.

En cuanto a las tradicionales, la que más destaca es la PMP y la ISO 21500, ya que tienen las mismas pautas que el PMBOK. IPMA y PRINCE2 con puntuaciones más bajas. IPMA no describe como se adecua la planificación de los requisitos. PRINCE2 no tiene en cuenta la comunicación como factor clave. Esta última puede que esté de manera indirecta, ya que al tener unos roles y responsabilidades bien definidas, hace más fácil la comunicación en el equipo.

No obstante se puede observar que no hay mucha diferencia a la hora de planificación de los procesos en las metodologías tradicionales. Pasa lo mismo que entre las metodologías ágiles. Pero si hay una diferencia más significativa entre los dos grupos de metodologías, siendo más adecuadas las metodologías ágiles, por su comunicación constante y sus reuniones diarias.

El siguiente área a comparar es Recursos Humanos (tabla 7.5)

Tabla 7-5 - FCE. Recursos Humanos. Elaboración propia

FACTORES CRÍTICOS DE ÉXITO	METODOLOGÍAS TRADICIONALES				METODOLOGÍAS ÁGILES		
	PMP	IPMA	ISO 21500	PRINCE2	SCRUM	XP	DSDM
RECURSOS HUMANOS	17	18	16	14	23	26	24
7. Apoyo de la dirección	2	2	2	2	2	2	2
8. Liderazgo	3	3	2	3	3	3	2
9. Desarrollo de los gerentes	2	3	2	1	2	2	2
10. Habilidades básicas	3	1	3	1	2	3	2
11. Desarrollo de sus empleados	1	3	1	1	3	3	2
12. Involucramiento del usuario	1	1	1	1	2	3	3
13. Involucramiento de los participantes	1	1	1	1	2	2	3
14. Responsabilidad y compromiso del cliente	1	1	1	1	2	3	3
15. Equipos auto-organizados	1	1	1	1	3	3	2
16. Equipos con experiencia y conocimiento	2	2	2	2	2	2	3

En este apartado se encuentran los siguientes FCE:

7. Apoyo de la dirección
8. Liderazgo
9. Desarrollo de los gerentes
10. Habilidades básicas
11. Desarrollo de sus empleados
12. Involucramiento del usuario
13. Involucramiento de los participantes
14. Responsabilidad y compromiso del cliente
15. Equipos auto-organizados
16. Equipos con experiencia y conocimiento

Estos FCE hacen referencia a las personas involucradas en el proyecto tanto externas como internas. Se tienen que tener en cuenta tanto las habilidades básicas del equipo como las técnicas. El desarrollo de software requiere muchos conocimientos técnicos pero sin olvidar las habilidades individuales y el desarrollo del equipo.

En este área se puede observar en primer lugar que la más destacada es XP respecto al resto de metodologías, muy seguidas de las otras dos metodologías ágiles. No hay una diferencia muy significativa.

Sin embargo, si las comparamos con las metodologías tradicionales, sí que hay muchos puntos de diferencia. Esto se debe principalmente a la organización de los equipos, el involucramiento del usuario y los participantes y la responsabilidad y compromiso del cliente. Estos son factores propios de las metodologías ágiles y no de

las tradicionales, y en este tipo de proyectos de desarrollo de software son clave. Por lo tanto, en este caso nos quedamos con las metodologías ágiles, priorizando el XP.

Si miramos las metodologías tradicionales de forma aislada, se observa que la mejor en esta área sería IPMA. La principal diferencia con el resto de metodologías de este tipo es el desarrollo de sus empleados y los gerentes. Esto es gracias a que en IPMA hay competencias de conducta y dirigen comportamientos y actitudes del personal para que esta metodología se lleve a cabo de manera óptima. Además, gracias a su sistema de certificaciones, plantea un claro plan de carrera profesional. Estas certificaciones no solo se alcanzan por tener conocimientos, sino también por el desempeño continuado en el tiempo de las funciones y competencias de un profesional.

La siguiente área que vamos a analizar es Objetivos y Alcance, que está compuesto por los siguientes FCE:

- 17. Definición clara de requerimientos
- 18. Metas intermedias alcanzables
- 19. Visión y objetivos claros
- 20. Alcance del trabajo bien definido
- 21. Tamaño del proyecto

Estos factores corresponden a qué se necesita en el proyecto, cuáles son los requisitos, qué mueve a realizar este proyecto. En la mayoría de los proyectos de desarrollo de software el cliente no es experto en este tema, por lo tanto, no tiene muy claro cómo llegar a su objetivo, pero si lo que quiere. Por lo tanto, al inicio del proyecto se plantean los requisitos del proyecto, y se van revisando constantemente para que sea lo pedido. Aun así, estos objetivos y requisitos pueden cambiar a medida que se va desarrollando el proyecto. En este caso, como ya se sabe, las metodologías ágiles son las que mejor se adaptan a los cambios. Veamos qué obtenemos de la comparación.

Tabla 7-6 - FCE. Objetivos y alcance. Elaboración propia

FACTORES CRÍTICOS DE ÉXITO	METODOLOGÍAS TRADICIONALES				METODOLOGÍAS ÁGILES		
	PMP	IPMA	ISO 21500	PRINCE2	SCRUM	XP	DSDM
OBJETIVOS Y ALCANCE	13	10	10	11	12	12	12
17. Definición clara de requerimientos	3	3	2	2	3	3	3
18. Metas intermedias alcanzables	1	2	1	2	3	3	3
19. visión y objetivos claros	3	2	2	3	2	2	2
20. Alcance del trabajo bien definido	3	1	3	2	2	2	2
21. Tamaño del proyecto	3	2	2	2	2	2	2

Como se puede observar en este caso, la mayor puntuación corresponde a la metodología tradicional PMP. Todos los FCE que se han considerado, los tiene en cuenta esta metodología, excepto las metas intermedias. Las metodologías tradicionales no hacen entregas intermedias, y en el desarrollo de software es muy importante ir haciendo pruebas constantes de lo que compondrá el producto final.

Realmente, las metodologías ágiles no se encuentran muy lejos de la puntuación más alta, tan solo están un punto por debajo. Esto se debe a que en este caso sí que hay una definición clara de las metas intermedias que se quieren conseguir, pero se tienen menos claros la visión del proyecto, su alcance y el tamaño. Según se ha comprobado con numerosos proyectos una de las causas de fracaso se da cuando el proyecto es grande.

La conclusión obtenida en esta área es un poco incierta, ya que aunque PMP sea la de mayor puntuación, las metas intermedias son bastante importantes y útiles en este tipo de proyectos para ir visualizando el producto final en cada entrega y poder hacer todas las mejoras posibles. Por lo tanto, dependería del proyecto y del equipo del mismo, la elección de la metodología.

La siguiente área a comparar es la calidad y los FCE seleccionados para este grupo son:

- 22. Monitoreo apropiado y retroalimentación
- 23. Calidad de las fuentes de datos
- 24. Entregas parciales

Son solo tres factores pero no por ello menos importantes. La calidad es muy importante en la actualidad y tan importante es tener la calidad deseada como no pasarse y añadir un coste extra al producto que no es el principal.

Tabla 7-7 - FCE. Calidad. Elaboración propia

FACTORES CRÍTICOS DE ÉXITO	METODOLOGÍAS TRADICIONALES				METODOLOGÍAS ÁGILES		
	PMP	IPMA	ISO 21500	PRINCE2	SCRUM	XP	DSDM
CALIDAD	5	5	4	6	8	8	9
22. Monitoreo apropiado y retroalimentación	3	3	2	3	3	3	3
23. Calidad de las fuentes de datos	1	1	1	1	2	2	3
24. Entregas parciales	1	1	1	2	3	3	3

En el caso de la calidad, el que obtiene mayor puntuación total es la metodología DSDM, muy seguida de SCRUM y XP. Se diferencian de las metodologías tradicionales en que no existen entregas parciales en estas, por lo tanto no se puede medir una calidad de estas entregas.

Otro de los factores que vemos que menos aparece o menos tienen en cuenta las metodologías, en ambos casos es la calidad de las fuentes de datos. No se tiene en cuenta, de donde proviene la información o al menos las metodologías no lo especifican. Dejando este factor en manos del equipo de proyecto y el director.

Y por último, pero no menos importante, el área de Tecnología e innovación, contiene los siguientes FCE:

- 25. Tecnología apropiada
- 26. Conocimientos técnicos de los usuarios
- 27. Disponibilidad tecnológica

Esta área es muy importante en el desarrollo de software aportando herramientas y competencias fundamentales para llevar a cabo este tipo de proyectos. A través de la tecnología disponible se puede aumentar las posibilidades de desarrollo y competitividad en un mundo globalizado por las tecnologías de información.

Tabla 7-8 - FCE. Tecnología e innovación. Elaboración propia

FACTORES CRÍTICOS DE ÉXITO	METODOLOGÍAS TRADICIONALES				METODOLOGÍAS ÁGILES		
	PMP	IPMA	ISO 21500	PRINCE2	SCRUM	XP	DSDM
TECNOLOGÍA E INNOVACIÓN	4	4	4	5	8	8	6
25. Tecnología apropiada	1	1	1	2	3	3	2
26. Conocimientos técnicos de los usuarios	2	2	2	2	3	3	2
27. Disponibilidad tecnológica	1	1	1	1	2	2	2

En este caso, podemos observar que las metodologías ágiles tienen más en cuenta el uso de la tecnología apropiada. Esto se debe a que este tipo de metodologías se han desarrollado para llevar a cabo tipos de proyectos más innovadores y nuevos como son las nuevas tecnologías que ahora mismo existen en la actualidad.

Además, los equipos que se forman en SCRUM y XP tienen más en cuenta los conocimientos y habilidades de los usuarios, ya que tienen que tener conocimientos de programación muy avanzados, para trabajar en el tiempo previsto y ser competitivos respecto al resto. En las metodologías tradicionales, esto es debilidad para dirigir proyectos de desarrollo de software.

CONCLUSIONES

8 CONCLUSIONES

El objetivo principal fue desde un inicio ver qué metodologías son más apropiadas para dirigir un proyecto de desarrollo de software, desde el punto de vista de los factores críticos de éxito de este tipo de proyectos. No se trata de encontrar la mejor metodología, sino tener una ayuda a la hora de decidir qué metodología utilizar, sabiendo cuáles son sus fortalezas y debilidades, y elegir la que mejor se adapte a las necesidades del proyecto.

Para poder llegar a este punto primeramente se ha contextualizado e introducido la problemática en sí. Se ha realizado una búsqueda de las metodologías utilizadas para la dirección de proyectos, documentadas con su historia, características principales, estructura y procesos, roles y responsabilidades, ventajas y desventajas.

En el siguiente capítulo se ha llevado a cabo la comparativa de estas metodologías, con el objetivo de marcar las diferencias y similitudes, al igual que sus ventajas y desventajas generales.

Se ha mostrado un estudio realizado en 2015 por CHAOS Report de Standish Group, que ofrece unos resultados estadísticos basadas en metodologías ágiles y tradicionales de desarrollo de software. Este estudio nos ha mostrado datos reales de la problemática de este tipo de proyectos y la poca mejoría que han sufrido en los últimos años.

Posteriormente, se han obtenido factores de fracaso comunes en los proyectos de desarrollo de software, dejando en evidencia los fallos que se cometen a la hora de dirigir y desarrollar este tipo de proyectos.

Finalmente, se ha realizado una búsqueda bibliográfica de los factores críticos de éxito, que serán los utilizados para hacer la comparación frente a las diferentes metodologías, y obtener unos resultados de esta investigación.

8.1 CONCLUSIÓN DE LOS RESULTADOS

Con los resultados anteriores se puede afirmar que las metodologías ágiles son mejores para dirigir proyectos de desarrollo de software. Ya que han obtenido la mayor puntuación en cuatro de las cinco áreas propuestas en esta investigación.

Las áreas son:

PROCESOS

RECURSOS HUMANOS

CALIDAD

TECNOLOGÍA E INNOVACIÓN

Cierto es, que los factores críticos de éxito que contienen estas áreas, no siempre han salido favorecidos por el lado de las ágiles. Algunos de estos puntos son mejores en algunas metodologías tradicionales como hemos podido observar en la tabla comparativa.

El área en el que no han sido mejores las metodologías ágiles ha sido “Objetivos y Alcance” del proyecto. La manera de abordar este tema de las metodologías tradicionales es más efectiva, ya que estas metodologías están preparadas para proyectos de tamaño mayor, que es uno de los factores clave de éxito que forman esta área, y una de las causas por la que fracasan este tipo de proyectos. Además en estas metodologías se da más importancia a la definición del alcance, objetivos y visión del proyecto.

A priori parece que en cualquier proyecto la definición de estas tareas desde un principio, facilitará el desarrollo del mismo. En el caso de los proyectos de desarrollo de software, es necesario tener claros los requisitos del proyecto y que estos estén claros en el equipo de trabajo, pero es muy importante la adaptabilidad a los cambios. Esto no nos lo proporcionan las metodologías tradicionales.

Se podría decir que no están completos ni un tipo ni otro, por lo que sería interesante que las metodologías ágiles adoptaran estos puntos en los que se desenvuelven mejor las tradicionales, y de esta manera mejorar en el punto de “Objetivos y Alcance”.

Esta conclusión no significa que siempre se tengan que usar las ágiles y las tradicionales queden obsoletas, sino que para esto tipo de proyectos, las metodologías ágiles tienen en cuenta más factores críticos de éxito. No nos están asegurando el éxito, ni mucho menos en todos los proyectos, pero sí que hay más posibilidades de que el proyecto llegue a su fin de manera satisfactoria.

Esta decisión siempre quedará en manos de los responsables del proyecto. Viendo que es lo que mejor se adapta a las necesidades y requisitos de ese proyecto en concreto.

8.2 FUTURAS EXTENSIONES

Hasta este punto del trabajo fin de grado, se ha llegado al objetivo principal expuesto al inicio del mismo. Pero como posibles ampliaciones o evoluciones que se pueden desarrollar en el futuro, se propone:

- Ampliar la tabla de FCE y estudiarlos con más profundidad, añadiendo más áreas importantes a la hora de dirigir un proyecto de software.
- Realizar un proyecto de este tipo, aplicando las metodologías según estos resultados.
- Investigar sobre nuevas metodologías que puedan mejorar las debilidades que presentan las aquí mostradas o incluso plantear la opción de que la mezcla o combinación de los dos tipos de metodologías podría funcionar.

BIBLIOGRAFÍA

9 BIBLIOGRAFÍA

Agile Business (2014). The DSDM Agile Project Framework. Obtenido de <https://www.agilebusiness.org/content/roles-and-responsibilities>

Agile Lucero. My stakeholders are missing. <http://agilelucero.com/scrum/my-stakeholders-are-missing/>

Alfred Maeso (2014). “Los 5 principales beneficios de utilizar PRINCE2”. Obtenido de <https://www.netmind.es/knowledge-center/principales-beneficios-de-utilizar-prince2/>

Alonso Ferreras, V.H (2009). Factores críticos de éxito y evaluación de la competitividad de destinos turísticos.

Asociación Española para la Calidad. “Norma ISO 21500”. Obtenido de <https://www.aec.es/web/guest/centro-conocimiento/norma-iso-21500>

Awad, MA (2005). A Comparison between Agile and Traditional Software Development Methodologies. Obtenido de <https://es.scribd.com/document/353767437/A-Comparision-Between-Agile-and-Traditional-SW-Development-Methodologies>

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M.,... & Kern, J. (2001). Manifesto for agile software development.

Canós, J. H., Letelier, P., & Penadés, M. C. (2003). Metodologías ágiles en el desarrollo de software. *Metodologías Ágiles en el Desarrollo de Software*, 1(10), 1-8.

Cardoza, R.A., Dante, G.C., De los Ríos, C. I. (2011). Comparación de cuatro sistemas de certificación del ámbito de la dirección de proyectos. Obtenido de http://oa.upm.es/12809/1/INVE_MEM_2011_107522.pdf

Chandana (2017). Entrenamiento de certificación ágil: Marco de gestión de proyectos ágiles.. Obtenido de <https://www.simplilearn.com/apm-framework-article>

Coquillat. M (2014). “Origen de La Norma ISO 21500”. Obtenido de <https://es.scribd.com/document/364709313/Origen-de-La-Norma-ISO-21500>

Daft, D.R. (1988) “Management”, Dryden Press, Chicago

Daniel, D.R. (1961): “Management information crisis”, Harvard Business Review, 39, Septiembre-Octubre, pp. 110-121

Desarrollo en cascada. (2018, 24 de agosto). Wikipedia, La enciclopedia libre. Desde https://es.wikipedia.org/w/index.php?title=Desarrollo_en_cascada&oldid=110170741.

EOI (Abril, 2013). Ciclo de vida de un proyecto. Obtenido de <http://www.eoi.es/blogs/madeon/2013/04/15/ciclo-de-vida-de-un-proyecto/>

Extreme Programming. Obtenido de <https://iswugxp.wordpress.com/historia/>

Fernández, P.K. (2014). Guía comparativa de Metodologías ágiles. Obtenido de <https://uvadoc.uva.es/bitstream/10324/1495/1/TFGB.117.pdf>

Fernández, K., Garrido, A., Ramínez, Y., y Perdomo, I. (2015). PMBOK y PRINCE 2, similitudes y diferencias. *Revista Científica*, 23, 111-123.

Figuroa, R. (2015). Metodologías tradicionales vs Metodologías ágiles. Universidad Técnica particular de Loja.

Fowler, M (1963). Obtenido de https://es.wikipedia.org/wiki/Martin_Fowler

González Bañales, D., & Rodenes Adam, M. (2007). Factores críticos de éxito de la industria del software y su relación con la orientación estratégica de negocio: un estudio empírico-exploratorio. *JISTEM: Journal of Information Systems and Technology Management*, 4 (1), 47-70.

Hardaker, M. y Ward, B. J. (1987) "Getting things done", *Harvard Business Review*, 65: 112-120

HDM Project Managers (2017). Comparativa Tablas de Procesos PMI e ISO 21500. Obtenido de <https://uv-mdap.com/blog/aplicacion-practica-de-los-estandares-del-pmi/>

HDM Project Managers (2015). Características diferenciadoras entre metodologías tradicionales y ágiles. Obtenido de <http://www.uv-mdap.com/blog/caracteristicas-diferenciadoras-entre-metodologias-tradicionales-y-agiles/>

HDM Project Managers (2015). Metodologías tradicionales o ágiles. Que metodología usar en mi proyecto. Obtenido de <http://www.uv-mdap.com/blog/metodologias-tradicionales-o-agiles-que-metodologia-usar-en-un-proyecto/>

Hofer, C.W. y Schendel, D. (1978) "Strategy formulation: analytical concepts", *St. Paul, West Information Strategy: the executive's Journal*, 11 (1): 12-17

Ingeniería de software (Septiembre, 2008). Historia de la programación Extrema. Obtenido de <https://ingenieriadesoftware.blogia.com/2008/090201-historia-de-la-programacion-extrema.php>

IPMA Versión 4.0 (2015). Individual Competence Baseline for Project, Programme and Portfolio Management. Obtenido de <http://products.ipma.world/ipma-product/icb/read-icb/>

IsoTools (Julio 2017). "Norma ISO 21500: Guía para la gestión de proyectos" en <https://www.isotools.org/2017/07/11/norma-iso-21500-guia-gestion-proyectos/>

Jenster, P. (1987) "Using critical success factors in planning", Long Range Planning, 20(4): 102-109

Kenneth S. Rubin. Essential Scrum: A Practical Guide to the Most Popular Agile Process. Obtenido de <http://www.clubdetecnologia.net/blog/2013/origenes-de-scrum/>

Leidecker, J.K. y Bruno, A.V. (1984) "Identifying and using critical success factors", Long Range Planning 17 (1): 15-21

Leyva Vázquez, M., & Rosado Rosello, R., & Febles Estrada, A. (2012). Modelado y análisis de los Factores Críticos de Éxito de los proyectos de software mediante Mapas Cognitivos Difusos. Ciencias de la Información, 43 (2), 41-46.

López. E. EOI (2014). PRINCE2 - Otra metodología para la gestión de proyectos. Obtenido de <http://www.eoi.es/blogs/mcalidadon/2014/12/20/prince2-otra-metodologia-para-la-gestion-de-proyectos/>

M. Ferrer (2015). Introducción a los 7 procesos de un proyecto en <http://blirte.es/prince2-introduccion-a-los-7-procesos-de-un-proyecto/>

Management Plaza (2017). "Temáticas PRINCE2" en <http://managementplaza.es/blog/tematicas-prince2/>

Método de desarrollo de sistemas dinámicos. (2017, 17 de noviembre). Wikipedia, La enciclopedia. Obtenido de https://es.wikipedia.org/w/index.php?title=M%C3%A9todo_de_desarrollo_de_sistemas_din%C3%A1micos&oldid=103503033.

Moya. J. (2017). Metodología Ágil vs Metodología Tradicional. Obtenido de <https://pmi-mad.org/index.php/socios/articulos-direccion-proyectos/1288-metodologia-agil-vs-metodologia-tradicional>

Pardo, Rodolfo (2015). Obtenido de <http://slides.com/rodipardo/desarrollo-software-metodologias>

OBS Business School (2017). Las 5 etapas en los sprints de un desarrollo Scrum. Obtenido de <https://www.obs-edu.com/es/blog-investigacion/project-management/las-5-etapas-en-los-sprints-de-un-desarrollo-scrum>

OBS Business School (2017). ¿Qué hay que saber de la ISO 21500? en <https://www.obs-edu.com/es/blog-project-management/temas-actuales-de-project-management/que-hay-que-saber-de-la-iso-21500>

PMBOK (2018). Mapa de procesos PMBOK6. Obtenido de http://pmbok.certificacionpm.com/pmbok?certification_id=pmbok6

PMBOK (2017). La Guía del PMBOK 6ta edición y las metodologías ágiles. Obtenido de <http://www.pmoinformatica.com/2017/12/pmbok-6-metodologias-agiles.html>

PMI. La participación de los patrocinadores ejecutivos. (2014). Obtenido de https://www.pmi.org/-/media/pmi/documents/public/pdf/learning/thought-leadership/pulse/executive-sponsor-engagement.pdf?sc_lang_temp=es-ES

Pollalis, Y. y Grant, J.H. (1994) "Information resource and corporate strategy development",

Programación Orientada a Objetos y Taller (2011). Tarjetas CRC. Obtenido de <http://jviezca.blogspot.com/2011/09/tarjetas-crc.html>

ProyectosAgiles.Org (2008). Qué es Scrum. Obtenido de <https://proyectosagiles.org/que-es-scrum/>

Rockart, J.F. (1979) "Chief executives define their own data needs", Harvard Business Review 57: 81-92

Rodríguez. M (2017). Roles y responsabilidades DSDM Agile Project Framework. Obtenido de <https://www.netmind.es/knowledge-center/roles-y-responsabilidades-en-el-dsdm-agile-project-framework/>

Roger Humberto (Abril, 2012). Metodologías de desarrollo de software tradicionales vs ágiles. Obtenido de <http://masteringenieriasoft.blogspot.com/2012/04/metodologias-de-desarrollo-de-software.html>

Roger S. Pressman, 2010. Ingeniería del software (7ªED). 1-11

Stankovic, D., Nikolic, V., Djordjevic, M., y Cao, D. (2008). Un estudio de encuesta de factores críticos de éxito en proyectos de software ágil en empresas de TI de la ex Yugoslavia. Revista de Sistemas y Software, 81

THOMPSON, Arthur & Strickland A.J. (2004). Administración Estratégica: Texto y Casos. (13ª Ed.) McGrawHill Interamericana, México. ISBN: 007-249395-X

Viezca. J (2011). Programación orientada a objetos y taller. Obtenido de <https://jviezca.blogspot.com/2011/09/tarjetas-crc.html>

Williams, LA (2010). Metodologías y prácticas ágiles de desarrollo de software. Avances en computadoras, 80

12 Manage The Executive Fast Track. “La metodología PRINCE2” en https://www.12manage.com/methods_ccta_prince2_es.html