

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías Industriales

Estudio de metodologías ágiles. Aplicación SCRUM

Autor: Manuel Torrado Nevado

Tutor: María Rodríguez Palero

**Dpto. de Organización Industrial y Gestión de
Empresas II**

Escuela Técnica Superior de Ingeniería

Sevilla, 2019



Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Estudio de metodologías ágiles. Aplicación SCRUM

Autor:

Manuel Torrado Nevado

Tutor:

María Rodríguez Palero

Dpto. de Organización Industrial y Gestión de Empresas II

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019

Trabajo Fin de Grado: Estudio de metodologías ágiles. Aplicación SCRUM

Autor: Manuel Torrado Nevado

Tutor: María Rodríguez Palero

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

Agradecimientos

A mis maestros por sus esfuerzos y enseñanzas para que pudiese culminar con éxito esta etapa universitaria. Especial mención a mi tutora María Rodríguez por su dedicación y compromiso en el desarrollo de este trabajo de fin de grado.

A mis compañeros de trabajo por su disposición y ayuda para la realización del documento.

Y sobre todo a mi familia por el apoyo incondicional que me han brindado durante toda esta etapa complicada e importante de mi vida.

GRACIAS.

Resumen

Con la finalidad de progresar y adaptarse a las exigencias de un mercado cada vez más tecnológico y exigente, las industrias y empresas han ido desarrollando a lo largo del tiempo unos métodos y herramientas que permitían mejorar la eficiencia de sus actividades. Surgen de este modo las metodologías de desarrollo. Las más tradicionales, aquellas basadas en un ciclo de vida y procesos secuenciales, supusieron un importante avance en los sistemas de desarrollo de la época, pero actualmente están obsoletas y no pueden hacer frente a las necesidades del mercado contemporáneo. Todo esto desemboca en la búsqueda de nuevas metodologías más competentes y funcionales, las llamadas metodologías ágiles.

Se conoce como metodologías ágiles a aquellos procesos que permiten a una empresa llevar a cabo su trabajo de la manera más eficiente, con el fin de obtener un rendimiento más rápido y flexible de acuerdo a las exigencias del proyecto que se aborde. De esta manera se consigue un aumento de la productividad y una disminución de los costes en el proceso de gestión.

Uno de los métodos más importantes es el conocido como método Scrum. Traducido al español como melé, hace referencia al avance en formación de un equipo de rugby donde todos los componentes avanzan en la misma dirección y ayudándose unos a otros. De este modo, Scrum tiene como objetivo garantizar un método de trabajo en equipo altamente productivo, flexible y con un seguimiento constante que ofrezca a la empresa seguridad y tiempo de reacción ante las dificultades que pueda presentar el proyecto.

En el documento que nos ocupa se hará un estudio de las metodologías de desarrollo tradicionales más influyentes. Posteriormente se analizarán en profundidad algunas de las metodologías ágiles más importantes, con mayor exhaustividad para el método Scrum. Por último, se llevará a cabo la aplicación de este método en un caso real, para la empresa Dable Red S.L., donde se realizará un seguimiento y analizarán los resultados obtenidos, así como su valoración final.

Abstract

In order to progress and adapt to the demands of an increasingly technological and demanding market, industries and companies have developed methods and tools over time to improve the efficiency of their activities. Development methodologies arise in this way. The most traditional ones, those based on a life cycle and sequential processes, represented an important advance in the development systems of the time, but are now obsolete and can not meet the needs of the contemporary market. All this leads to the search for new more competent and functional methodologies, the so-called agile methodologies.

Agile methodologies are those processes that allow a company to carry out its work in the most efficient way, in order to obtain a faster and more flexible performance according to the requirements of the project that is addressed. In this way an increase in productivity and a reduction of costs in the management process is achieved.

One of the most important methods is the one known as the Scrum method. Translated into Spanish as *melee*, it refers to the advance in formation of a rugby team where all the components move in the same direction and helping each other. In this way, Scrum aims to guarantee a method of team work that is highly productive, flexible and constantly monitored, offering the company security and reaction time in the face of difficulties that the project may present.

In the document that occupies us a study of the most influential traditional development methodologies will be made. Subsequently, some of the most important agile methodologies will be analyzed in depth, with greater completeness for the Scrum method. Finally, the application of this method will be carried out in a real case, for the company Dable Red S.L., where a follow-up will be carried out and the results obtained will be analyzed, as well as their final assessment.

Índice

Resumen	ix
Abstract	xi
Índice	xiii
Índice de Tablas	xv
Índice de Figuras	xvii
1 Objetivos	1
2 Introducción	3
3 Historia de las metodologías de desarrollo	5
3.1. <i>Antecedentes históricos</i>	5
3.2. <i>Metodologías de desarrollo</i>	5
3.2.1. Ciclo de vida	5
3.2.2. Tipos de metodologías de desarrollo	7
3.2.2.1 Modelo de cascada	7
3.2.2.2 Modelo de espiral	8
3.2.2.3 Modelo en V	9
3.3. <i>Desventajas de las metodologías de desarrollo</i>	10
4 Metodologías ágiles	11
4.1. <i>Origen</i>	11
4.2. <i>Definición</i>	13
4.3. <i>Ventajas</i>	14
4.3.1. Metodologías ágiles frente a las tradicionales	15
4.4. <i>Métodos ágiles</i>	16
4.4.1. Extreme Programming	16
4.4.2. Kanban	19
4.4.3. Design Sprint	21
4.4.4. Otras metodologías ágiles	23
5 Scrum	25
5.1. <i>Introducción</i>	25
5.2. <i>Historia</i>	26
5.3. <i>Definición</i>	27
5.4. <i>Características</i>	28
5.5. <i>Herramientas</i>	28
5.3.1. Artefactos	28
5.3.2. Roles	31
5.3.3. Eventos	32
5.6. <i>Beneficios de Scrum</i>	35

6. Aplicación de Scrum	37
6.1. <i>Introducción</i>	37
6.2. <i>Entidad objeto de la aplicación. Dable Red S.L.</i>	38
6.2.1. Presentación	38
6.2.2. Actividad a desarrollar	39
6.2.3. Equipo de trabajo e instalaciones	40
6.3. <i>Planificación del proyecto</i>	42
6.3.1. Proyecto	42
6.3.2. Eventos	42
6.3.3. Roles	43
6.3.4. Artefactos	45
6.4. <i>Desarrollo del proyecto</i>	48
6.4.1. Sprint 1	48
6.4.2. Sprint 2	51
6.4.3. Sprint 3	54
6.4.4. Sprint 4	57
7. Conclusiones	61
Referencias	65

Índice de Tablas

Tabla 4-1. Comparación entre metodologías (Fuente: Elaboración propia)	15
Tabla 4-2. Ventajas e inconvenientes del método XP (Fuente: Elaboración propia)	18
Tabla 4-3. Ventajas e inconvenientes del método Kanban (Fuente: Elaboración propia)	21
Tabla 5-1. Empresas que usan Scrum (Fuente: https://proyectosagiles.org/historia-de-scrum/)	27
Tabla 5-2. Ejemplo de Product Backlog (Fuente: Elaboración propia)	29
Tabla 5-3. Ejemplo de Sprint Backlog (Fuente: https://muyagile.com/)	30
Tabla 6-1. Product Backlog del proyecto (Fuente: Elaboración propia)	45
Tabla 6-2. Product Backlog del Sprint 1 (Fuente: Elaboración propia)	48
Tabla 6-3. Product Backlog del Sprint 2 (Fuente: Elaboración propia)	51
Tabla 6-4. Product Backlog del Sprint 3 (Fuente: Elaboración propia)	54
Tabla 6-5. Product Backlog del Sprint 4 (Fuente: Elaboración propia)	57
Tabla 7-1. Comparativa entre los métodos de trabajo (Fuente: Elaboración propia)	62

Índice de Figuras

Figura 3-1. Esquema de ciclo de vida del software (Fuente: http://ciclodevida.net/del-software)	6
Figura 3-2. Esquema del modelo de cascada (Fuente: https://openclassrooms.com)	7
Figura 3-3. Esquema del modelo de espiral (Fuente: https://okhosting.com/)	8
Figura 3-4. Esquema del modelo en V (Fuente: https://www.tutorialspoint.com)	9
Figura 4-1. Manifiesto Ágil, Febrero de 2001 (Fuente: http://www.caminoagil.com)	12
Figura 4-2. Esquema representativo de una metodología ágil (Fuente: https://es.wikipedia.org/)	14
Figura 4-3. Valores de XP (Fuente: oficinaproyectosinformatica.blogspot.com)	17
Figura 4-4. Esquema de Extreme Programming. (Fuente: https://openwebinars.net/)	18
Figura 4-5. Tablero básico Kanban. (Fuente: https://kanbantool.com)	19
Figura 4-6. Tablero Kanban. División de prioridades por filas. (Fuente: https://www.pinterest.es/)	20
Figura 4-7. Tablero Kanban. División de prioridades por colores. (Fuente: https://arrizabalagau...)	20
Figura 4-8. Fases de Google Design Sprint (Fuente: https://designSprintkit.appspot.com/)	22
Figura 5-1. Melé en rugby (Fuente: https://es.wikipedia.org/wiki/Scrum_(rugby))	26
Figura 5-2. Roles de Scrum (Fuente: https://desire.webs.uvigo.es/contenidos/scrum/)	31
Figura 5-3. Diagrama de Sprint Burndown (Fuente: International Scrum Institute)	33
Figura 5-4. Esquema de los eventos Scrum (Fuente: Elaboración propia)	35
Figura 6-1. Logotipo de Dable Red S.L. (Fuente: Dable Red S.L.)	38
Figura 6-2. Instalaciones y equipo de Dable Red S.L. (Fuente: Elaboración propia)	40
Figura 6-3. Nas Server (Fuente: Elaboración propia)	41
Figura 6-4. Scrum board de la oficina (Fuente: Elaboración propia)	41
Figura 6-5. Cronograma del mes de abril (Fuente: Elaboración propia)	43
Figura 6-6. Cronograma del mes de mayo (Fuente: Elaboración propia)	43
Figura 6-7. Esquema de roles del proyecto (Fuente: Elaboración propia)	44
Figura 6-8. Tipos de post-it del proyecto (Fuente: Elaboración propia)	46
Figura 6-9. Sprint Backlog base del proyecto (Fuente: Elaboración propia)	47
Figura 6-10. Sprint Backlog del Sprint 1 (Fuente: Elaboración propia)	48
Figura 6-11. Diagrama Sprint Burndown del Sprint 1 (Fuente: Elaboración propia)	49
Figura 6-12. Clusterización de Campohermoso (Fuente: Elaboración propia)	50
Figura 6-13. Sprint Backlog del Sprint 2 (Fuente: Elaboración propia)	51
Figura 6-14. Diagrama Sprint Burndown del Sprint 2 (Fuente: Elaboración propia)	52
Figura 6-15. Prediseño del cluster 16 (Fuente: Elaboración propia)	53

Figura 6-16. Sprint Backlog del Sprint 3 (Fuente: Elaboración propia)	54
Figura 6-17. Diagrama Sprint Burndown del Sprint 3 (Fuente: Elaboración propia)	55
Figura 6-18. Diseño del cluster 16 (Fuente: Elaboración propia)	56
Figura 6-19. Sprint Backlog (Fuente: Elaboración propia)	57
Figura 6-20. Diagrama Sprint Burndown del Sprint 4 (Fuente: Elaboración propia)	58
Figura 6-21. Carta de empalme y acta de replanteo (Fuente: Elaboración propia)	59

1 OBJETIVOS

“Los objetivos no son un destino, son una dirección. No son órdenes, son compromisos. No determinan el futuro; son medios para movilizar los recursos y la energía capaces de crear el futuro”.

- Peter Druker, -

La innovación y creatividad siempre han sido dos de los factores más importantes e influyentes en el crecimiento económico y tecnológico de cualquier país o entidad. Dentro del mundo empresarial, estos factores son determinantes para el desarrollo de cualquier negocio que busque ser rentable y exitoso. Innovar implica crear nuevas ideas y métodos de trabajos que permitan resolver los problemas de una manera más eficiente a lo que comúnmente se ha hecho hasta el momento. Las metodologías de desarrollo fueron el resultado de esos cambios en los procesos de desarrollo que las empresas introdujeron en sus estrategias de mercado para garantizar el crecimiento y consolidación de sus negocios.

Debido al papel tan importante y determinante que estas metodologías juegan en la actualidad es por lo que este documento que nos ocupa hace un estudio sobre ello. El objetivo que se plantea en él es el de dar a conocer las principales metodologías ágiles que existen, desarrollando sus características, ventajas e inconvenientes y explicando como tratan cada una de ellas de mejorar el rendimiento de las empresas, siempre apoyándose en un pilar fundamental, como es el trabajo en equipo y colaborativo. La implantación de éstas en las empresas y los distintos métodos de aplicación que tienen son puntos que se tienen en cuenta en este estudio.

Entre todas ellas, hay una metodología con mayor importancia y relevancia que el resto, el conocido como método Scrum. Es en este método donde se vuelca principalmente el peso de este trabajo. Es, por tanto, que se lleva a cabo un estudio mucho más amplio y exhaustivo de él, para finalizar con su implementación en un caso real de empresa. Se detallarán las características de este método, su estructura y elementos que lo componen, así como los roles que tomarán los empleados dentro de la empresa y los procesos que se llevarán a cabo a lo largo del ejercicio. Todo esto con el fin de conseguir adaptar el método de trabajo al contexto y necesidades que presenta el proyecto, siempre buscando la flexibilidad de éste y que se obtengan los mayores beneficios posibles.

La última parte de este trabajo se basará en la aplicación del método Scrum en una empresa real (Dable Red SL). Dicha empresa, dedicada al diseño de redes de fibra óptica por todo el territorio nacional y que trabaja con importantes compañías como Orange, Telefonica o MásMóvil, verá adaptada su forma de trabajo de acuerdo con las características que presenta el método Scrum. De modo que su equipo de trabajo, formado por un departamento de ingeniería y otro dedicado a los permisos de instalación, será reestructurado y se asignarán distintos roles y objetivos a cada empleado. Tendrán lugar reuniones periódicas donde se harán balances de las iteraciones que va realizando la empresa, el tiempo de entrega de los productos y de la disponibilidad de recursos con los que cuenta la empresa para poder llevar a cabo su realización. También se llevará a cabo una planificación de cómo será la gestión del proyecto y se realizarán revisiones constantes que aseguren la correcta realización del método y que los objetivos que se van obteniendo son los que se acordaron al principio del ejercicio.

2 INTRODUCCIÓN

“La clave del éxito en los negocios es detectar hacia dónde va el mundo y llegar ahí primero”.

- Bill Gates, -

En un mercado cada vez más complejo y exigente, es de obligada necesidad que las actuales empresas puedan responder a sus clientes con servicios rápidos y de la mayor calidad posible. Las nuevas tecnologías y herramientas que han ido apareciendo en las últimas décadas en el mundo industrial y empresarial han dado lugar a una gestión de proyectos mucho más compleja para las distintas entidades, de modo que la obtención de resultados favorables en la finalización del ejercicio es mucho más difícil y queda fuera del alcance de los métodos más tradicionales.

Se puede definir una metodología de desarrollo como un marco de trabajo que permite planificar, organizar y controlar de la manera más eficiente posible las tareas a realizar durante la ejecución de un ejercicio o proyecto.

Antiguas metodologías de desarrollo, cómo pueden ser el método cascada o el método de solapa, supusieron un antes y un después en el desarrollo a gran escala de los sistemas de negocios y construyeron las bases sobre las que se sustentan las ideas y principios de las actuales metodologías. Con el fin de facilitar y continuar con el desarrollo de los sistemas de una manera más intuitiva y estructurada, estas metodologías fueron implantadas en las organizaciones y entidades más competitivas. La búsqueda de una explotación de negocio óptima era el mayor objetivo en una época de grandes conglomerados empresariales y cualquier herramienta que ayudase a lograr dicha meta era un bien muy cotizado entre las empresas.

Durante décadas, estas metodologías tradicionales fueron las herramientas de desarrollo más factibles, y ayudaron a muchas empresas a prosperar y crecer dentro del mercado. Las tecnologías de desarrollo habían irrumpido con fuerza y las organizaciones especulaban con ellas con el fin de renovarlas y mejorarlas, para así afrontar los nuevos retos que estaban por llegar.

Con la llegada de un mercado caracterizado por clientes más exigente y con requisitos más propensos a cambios y alteraciones, las metodologías tradicionales no consiguieron adaptarse a este nuevo sistema de negocio y se fueron volviendo lentas, ineficientes e inconsistentes. Tanto es así, que actualmente estas metodologías han quedado totalmente obsoletas y desfasadas con respecto a los métodos convencionales, ya que no pueden competir con ellos a la hora de maximizar beneficios y optimizar el funcionamiento de las empresas. Es entonces cuando entran en juego metodologías que facilitan la gestión de los proyectos y consiguen una mejor respuesta a las exigencias que demanda el mercado, las llamadas metodologías ágiles.

3 HISTORIA DE LAS METODOLOGÍAS DE DESARROLLO

“Toda compañía debería trabajar duro para hacer que su propia línea de productos quede obsoleta, antes de que lo haga competencia”.

- Philip Kotler, -

A lo largo de la historia, el hombre ha ido desarrollando diferentes métodos de trabajo con el fin de adaptarse a la evolución continua del sector industrial y a las exigencias de un mercado cada vez más competitivo. Ésto obliga a las empresas a renovarse continuamente para subsistir en cuanto a rentabilidad y desatacar por encima de sus competidores.

3.1. Antecedentes históricos

La década de los 60 supone un antes y un después en las metodologías de desarrollo. Comienzan a parecer los sistemas tradicionales de *ciclo de vida* en los cuales se basan todos los métodos que vendrían a posteriori. En un periodo de tiempo donde el desarrollo industrial seguía al alza y el conglomerado empresarial iba en aumento, se empiezan a desarrollar entornos y marcos de trabajos basados en sistemas de información muy estructurados, metódicos y reiterativos, con el fin de tener siempre identificado el progreso, cuánto se ha avanzado y cuánto queda para la finalización del proyecto.

3.2. Metodologías de desarrollo

3.2.1 Ciclo de vida

Cuando se habla de metodologías de desarrollo es inevitable no asociarlo a su pieza mas trascendente y fundamental, el “ciclo de vida”. El concepto de ciclo de vida ha ocasionado siempre bastante confusión debido a que existen numerosas definiciones e interpretaciones de éste, y son muchos los autores que lo han particularizado y nombrado bajo su criterio y punto de vista.

Para generalizar, se puede definir como ciclo de vida al conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea hasta que se finaliza el software.



Figura 3-1. Esquema de ciclo de vida del software (Fuente: <http://ciclodevida.net/del-software>)

Las distintas etapas por las que pasa una idea durante su ciclo de vida son las siguientes:

- Definición de necesidades o requisitos. Fase en la que se identifican las necesidades de negocio y se definen las características del software a desarrollar. Generalmente el cliente no tiene una idea clara y concisa de lo que realmente necesita, es por ello, que esta fase se encarga de ayudarlo y concretar los requisitos que está buscando.
- Análisis. Durante este paso, y a partir de la información recogida en la etapa anterior, se procede a elaborar el modelo de software más conveniente para llevar a cabo el proyecto. Todo esto incluye saber las limitaciones del producto, aprendizaje para la solventación de los problemas que puedan suceder en el sistema, conocer con antelación los cambios que puede requerir el proyecto durante su desarrollo, etc. Una vez estudiadas las posibilidades de efectuar correctamente el proyecto, se procede a la planificación temporal y los recursos correspondientes del mismo.
- Diseño. En esta etapa se realiza el diseño del producto a partir de la información recogida en las fases precedentes. Se detallan los principios técnicos a partir de los modelos anteriores. Se procede al diseño de los componentes indicando las infraestructuras técnicas, arquitectura de software y estándares de desarrollo. Se implanta y corre el modelo.
- Codificación o Implementación. Una vez finalizado el producto, se presenta al cliente. Se le muestra su funcionamiento y rendimiento y se debate si se ha llegado a conseguir la idea y requisitos que se estaban buscando, consiguiendo de esta forma la satisfacción por ambas partes.
- Pruebas o Verificación. Se ejecutan planes de pruebas definidos con anterioridad y se comprueba que el funcionamiento del producto es adecuado y que cumple con las especificaciones y requerimientos definidos en la etapa de diseño.

- Validación. Cuando las pruebas se cumplen con satisfactoriedad, se da el visto bueno y se valida el funcionamiento del producto.
- Mantenimiento y funcionamiento. Se optimiza el funcionamiento del producto en términos de mayor eficiencia y menor número de errores. Se corrigen problemas y defectos, si procede, y se dota al producto de continuas actualizaciones para garantizar el mejor funcionamiento del mismo.

3.2.2 Tipos de metodologías de desarrollo

Una vez definido el término *ciclo de vida* y las distintas fases que lo componen, es necesario aclarar que existen una gran cantidad de interpretaciones de ellas, así como distintas formas de organizarlas y ejecutarlas. De acuerdo con cómo se lleve a cabo su implementación, existen distintos modelos de desarrollos, los cuales se detallan a continuación.

3.2.2.1 Modelo de cascada

El modelo de cascada es una de las metodologías más antiguas en lo que a ciclo de vida se refiere. Se trata de un proceso que se desarrolla de manera secuencial. Es su principal característica, de la cual viene su nombre, y consiste en un método donde las etapas se van realizando una detrás de otra, es decir, no se puede avanzar de etapa sin antes haber concluido con la anterior.

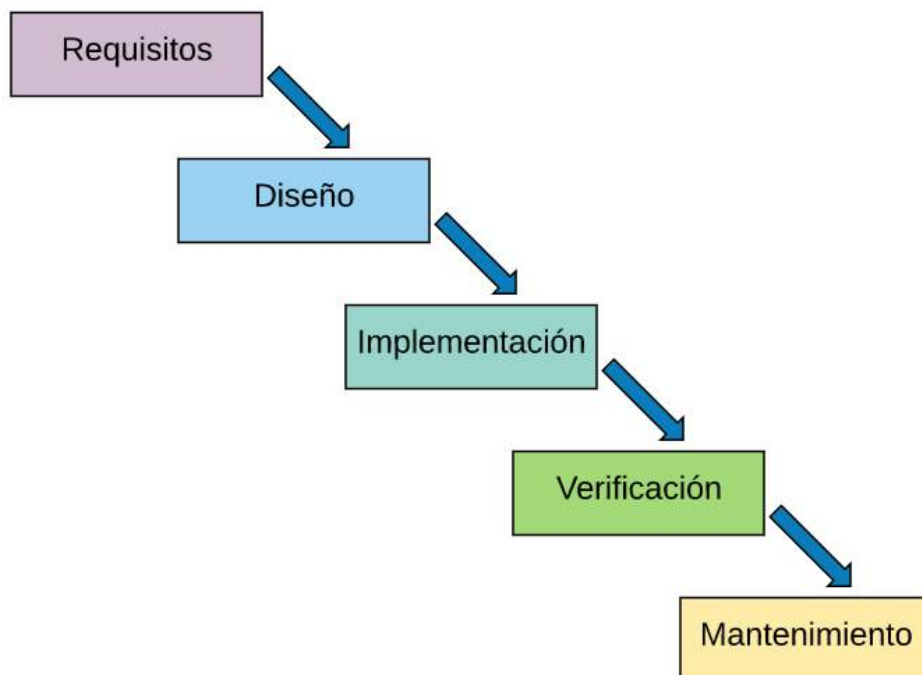


Figura 3-2. Esquema del modelo de cascada (Fuente: <https://openclassrooms.com>)

Este modelo apareció para llevar a cabo proyectos en las industrias y en la construcción, donde modificar o hacer cambios en lo ya construido suponía grandes pérdidas de dinero y tiempo, por lo que se optaba por este modelo secuencial que aseguraba que los procesos que se iban realizando no llevaban consigo errores que habría que solventar a posteriori. Es un modelo ideal para proyectos que están bien definidos y que se sabe a priori que sus requisitos se van a mantener estables a lo largo de su desarrollo.

Como se trata de un modelo fluenciado por el ciclo de vida, basa sus etapas de desarrollo en una recogida de requisitos, un diseño, una implementación, verificación y un mantenimiento, los cuales se realizan siguiendo este orden y sin saltarse ninguna fase.

Sin embargo, esta metodología no es tan eficiente en los tiempos actuales y presenta numerosos inconvenientes. Los clientes normalmente no tienen claro los requisitos iniciales e introducirán modificaciones y añadirán otros nuevos una vez empezado el desarrollo del proyecto, lo que provocará volver a realizar etapas ya superadas y supondría un importante aumento de los costes. También cabe destacar que el producto no se presenta al cliente hasta su total finalización, lo cual crea incertidumbre e inseguridad, puesto que no se tiene un seguimiento que garantice el correcto funcionamiento durante su desarrollo.

3.2.2.2 Modelo de espiral

Para hacer frente a los inconvenientes que presentaba el modelo de cascada, surge como alternativa el modelo de espiral. Este modelo, de desarrollo incremental, hace uso de un sistema iterativo basado en espirales, de modo que el producto se está trabajando y mejorando continuamente hasta que el resultado sea el deseado.

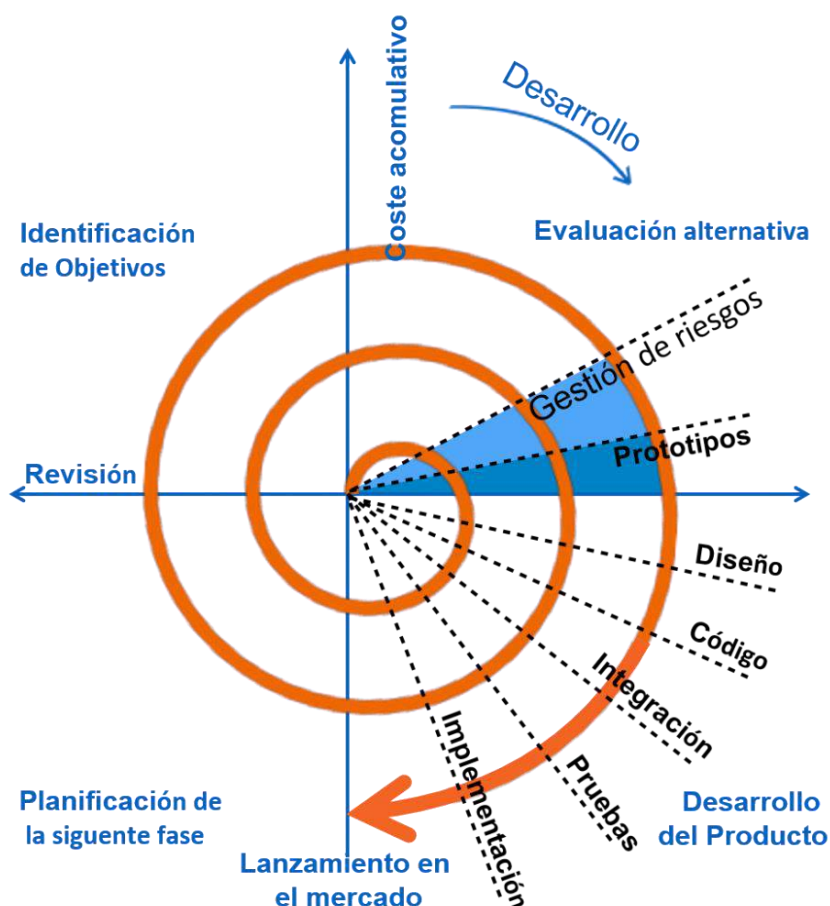


Figura 3-3. Esquema del modelo de espiral (Fuente: <https://okhosting.com>)

Al contrario que el modelo de cascada, el modelo de espiral permite la vuelta atrás en sus fases y se pueden hacer el número de iteraciones o ciclos que se consideren oportunas, de este modo se consigue disminuir los riesgos a la vez que avanzan el desarrollo del proyecto. Debido a esto, el enfoque de este proceso es muy distinto al del método anterior, teniendo éste una dirección orientada al análisis de riesgos.

Pero todo esto tiene un precio y es que el método se vuelve muy costoso y exige un equipo bastante competente y con un alto nivel de formación para poder hacer frente a cualquier problema o contratiempo que se puedan encontrar. También existe el riesgo de que se produzcan bucles en el desarrollo y el proyecto se alargue más de lo previsto si se toman decisiones erróneas.

3.2.2.3 Modelo en V

Modelo también conocido como modelo de validación y verificación. Al igual que el modelo en cascada, no permite el retroceso entre sus fases, pero este modelo propone un seguimiento y evaluación que se realiza de forma paralela al desarrollo normal de cada etapa. De este modo, se asegura no pasar a la etapa siguiente hasta que no se haya evaluado y validado la etapa actual, evitando así futuros problemas y errores que serían más difícil de solventar en un período más avanzado del desarrollo del proceso.

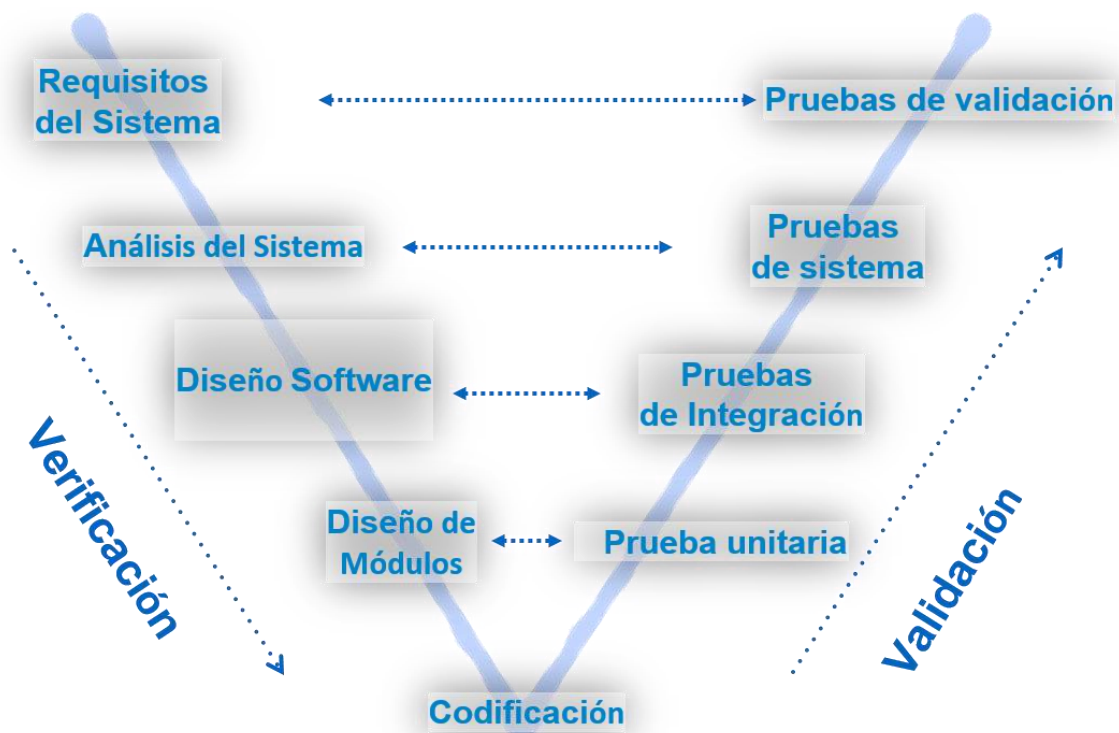


Figura 3-4. Esquema del modelo en V. (Fuente: <https://www.tutorialspoint.com>)

No obstante, el modelo sigue presentando la rigidez y poca flexibilidad que caracteriza al modelo en cascada, por lo que siguen existiendo importantes riesgos ante cambios fortuitos e imprecidibles del cliente.

3.3. Desventajas de las metodologías de desarrollo

Cómo se ha podido ver en el apartado anterior, las metodologías tradicionales presentan numerosos inconvenientes que limitan su funcionamiento en el actual marco laboral.

El modelo de cascada se caracteriza por ser un modelo muy rígido, que no permite retroceder ni avanzar etapas a conveniencias del desarrollador. Al ser un método secuencial, obliga a finalizar una fase para poder pasar a la siguiente. Además, el producto no es presentado al cliente hasta su completo desarrollo, sin dar posibilidad de testeo ni evaluación durante su desarrollo. Todo esto se traduce en un continuo estado de incertidumbre e inseguridad, que en un mercado como el actual, donde el cliente se muestra indeciso y exigente, supone a las empresas correr un riesgo innecesario y con un alto precio a pagar.

El modelo en V tiene como objetivo solventar ese estado de inseguridad a través de continuas evaluaciones y pruebas que se realizan a la par que el desarrollo del producto, pero sigue siendo un método que no puede hacer frente a decisiones impredecibles y fortuitas del cliente, puesto que sigue tratándose de un proceso secuencial sin posibilidad de cambio hasta su finalización.

Por su parte, el modelo de espiral mejora las desventajas anteriores al tratarse de un proceso iterativo y que permite la repetición de las etapas hasta que ambas partes, clientes y desarrolladores, estén conformes con el producto final. No obstante, se corre el riesgo de entrar en bucles demasiado largos y costosos por no llegar a un acuerdo que contente a las dos partes. Además, este método reclama un equipo de trabajo muy competente y cualificado que sea capaz de responder a las necesidades y cambios que exijan los clientes. Todo esto conlleva unos procesos de formación que añaden más gastos aún a la gestión del desarrollo.

Debido a la poca competitividad que estos métodos ofrecen a las actuales empresas, éstas se han visto obligadas a reinvertirse y buscar otras alternativas que les ofrezcan más oportunidades de subsistir y progresar en su desarrollo industrial. De ésta forma, se da lugar a la aparición de nuevos métodos más competentes en los años posteriores.

4 METODOLOGÍAS ÁGILES

“La agilidad puede tomar diferentes formas, pero básicamente consiste en la habilidad de adaptarse rápidamente o incluso anticiparse al contexto y liderar un cambio. La agilidad, en el sentido más amplio de su definición, afecta al diseño estratégico, las operaciones, la tecnología e innovación, y la habilidad que tiene una organización de innovar en la incorporación de productos, procesos y modelos de negocios”.

- Michael Cusmano, profesor de la MIT Sloan School of Management, -

Con la búsqueda de un nuevo entorno y marco de trabajo que se ajuste mejor a las necesidades y exigencias del mercado actual, ven la luz nuevos métodos de desarrollos mucho más avanzados y competentes que los tradicionales. El movimiento ágil irrumpe para quedarse.

4.1. Origen. Manifiesto Ágil

El termino “Metodologías ágiles” como tal, tiene su origen un 17 de Febrero de 2001, donde diecisiete expertos en modelos de desarrollo de software basados en procesos se reúnen en Snowbird (Utah) para debatir sobre las nuevas técnicas de desarrollo de software que estaban apareciendo años anteriores. Convocados por Kent Beck, el cual había publicado años antes un libro sobre una nueva metodología (Extreme Programing), deciden introducir una nueva terminología que definiese a las nuevas metodologías que estaban surgiendo a lo largo de la década de los 90. Metodologías que sustituirían a las tradicionales y que supondrían una alternativa al carácter rígido y pesado que definía a los modelos anteriores. De este modo, se apostó por el término “ágil” para su referencia.

En dicha reunión, quedan determinados y definidos los principios en los que se basan las metodologías ágiles, todos ellos resumidos en cuatro postulados que se recogen en el “Manifiesto Ágil”. No obstante, muchos métodos basados en la metodología ágil, como pueden ser Scrum (1986), Crystal Clear o el ya nombrado Extreme Programing (1999), fueron creados con anterioridad a este manifiesto.



Figura 4-1. Manifiesto Ágil, Febrero de 2001 (Fuente: <http://www.caminoagil.com>)

En el Manifiesto Ágil, firmado por los críticos: Kent Beck, James Grenning, Jim Highsmith, Andrew Hunt, Mike Beedle, Steve Mellor, Arie van Bennekum, War Cunningham, Robert C. Martin, Brian Marick, Alistair Cockburn, Jeff Sutherland, Ken Schwaber, Dave Thomas, Ron Jeffries, Jon Kern y Martin Fowler, queda escrito lo siguiente:

“Estamos poniendo al descubierto mejores métodos para desarrollar software, haciéndolo y ayudando a otros a que lo hagan. Con este trabajo hemos llegado a valorar:

- *A los **individuos y su interacción**, por encima de los procesos y las herramientas.*
- *El **software que funciona**, por encima de la documentación exhaustiva.*
- *La **colaboración con el cliente**, por encima de la negociación contractual.*
- *La **respuesta al cambio**, por encima del seguimiento de un plan.*

Aunque hay valor en los elementos de la derecha, valoramos más los de la izquierda.”¹

¹ Manifiesto Ágil, los cuatro postulados de los métodos ágiles. Escrito por Kent Beck y resto de críticos, 17 de Febrero de 2001

A partir de estos 4 valores fundamentales se establecen un conjunto de 12 principios en los que se recogen las ideas más importantes de esta metodología. Los principios son los siguientes:

- “1.- **Satisfacer al cliente** a través de la entrega temprana y continua de software de valor. Es la mayor prioridad.
- 2.-**Son bienvenidos los requisitos cambiantes**, incluso en etapas tardías del desarrollo. Los procesos ágiles se doblan al cambio como ventaja competitiva para el cliente.
- 3.-**Iteraciones constantes**. Entregar con frecuencia software que funcione, en periodos de un par de semanas hasta un par de meses.
- 4.-**Trabajo colaborativo**. Los responsables del negocio y los desarrolladores deben trabajar juntos de forma cotidiana durante el proyecto.
- 5.-**Motivación del equipo**. Construcción de proyectos en torno a individuos motivados, dándoles la oportunidad, el entorno y el respaldo que necesitan y procurándoles confianza para que realicen la tarea.
- 6.- **Contacto directo con los clientes**. La forma más eficiente y efectiva de comunicar información de ida y vuelta dentro de un equipo de desarrollo es mediante la conversación cara a cara.
- 7.-**Medida de progreso**. El software que funciona es la principal medida del progreso.
- 8.-**Desarrollo sostenido**. Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben mantener un ritmo constante de forma indefinida.
- 9.- **Búsqueda de la excelencia**. La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- 10.-**La simplicidad**, como arte de maximizar la cantidad de trabajo que se hace, es esencial.
- 11.- **Autorregulación**. Las mejores arquitecturas, requisitos y diseños emergen de equipos que se autoorganizan.
- 12.- **Revisión permanente**. En intervalos regulares, el equipo reflexiona sobre la forma de ser más efectivo y perfecciona su comportamiento en consecuencia.”²

Años después, y con la intención de promover el “movimiento ágil” en proyectos y otras aplicaciones, varios de estos críticos se unen para fundar *La Alianza Ágil*.

4.2. Definición

Se puede definir el desarrollo ágil como aquellas metodologías de desarrollo de software que se basan en un proceso iterativo e incremental, y que permiten adaptar el entorno y marco laboral a las condiciones del proyecto, con el fin de conseguir flexibilidad y una respuesta rápida para que su desarrollo sea lo más efectivo posible. Tanto las necesidades como las soluciones van evolucionando a lo largo del desarrollo.

Los equipos de trabajos que emplean esta metodología se caracterizan por ser multidisciplinarios y enfatizar la comunicación cara a cara frente a la documentación.

Al igual que las metodologías de desarrollo tradicionales, el desarrollo ágil se compone de un ciclo de vida iterativo que está compuesto por las etapas: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación.

² Los 12 principios recogidos en el Manifiesto Ágil. Escrito por Kent Beck y resto de críticos, 17 de Febrero de 2001

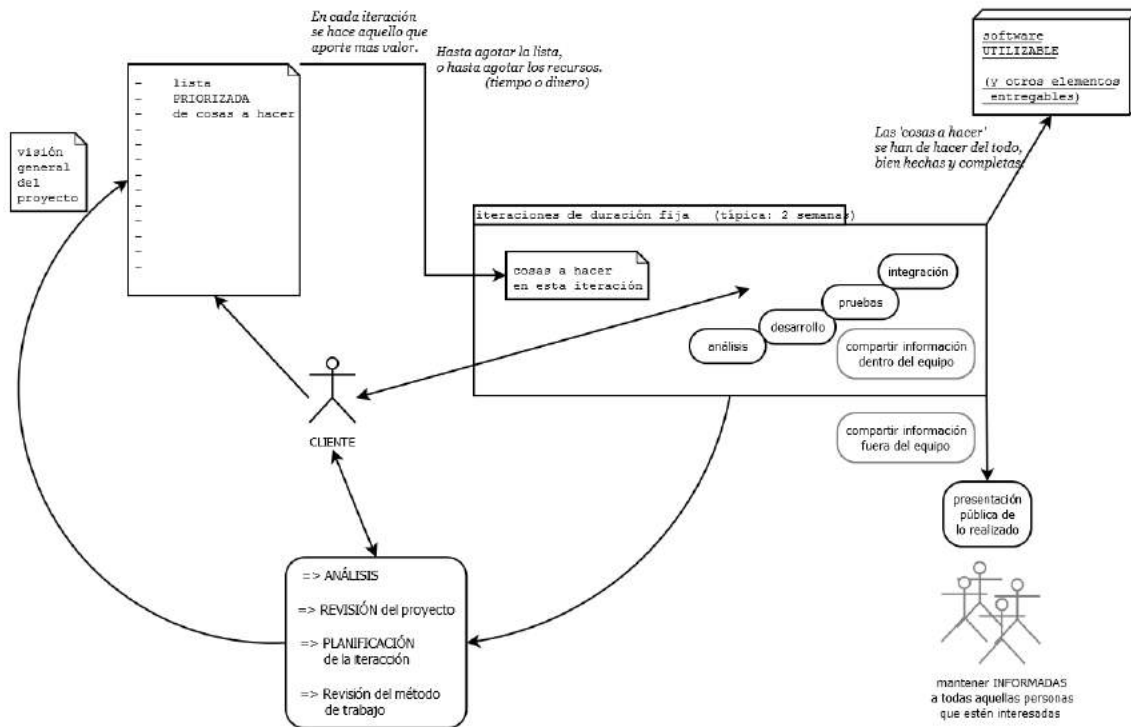


Figura 4-2. Esquema representativo de una metodología ágil (Fuente: https://es.wikipedia.org/wiki/Desarrollo_ágil_de_software)

De entre todas las etapas que componen el desarrollo del proyecto, la fase “finalizado” (done) es la que toma mayor importancia debido a que con cada iteración se busca aumentar el número de software (producto) que funciona y que está exento de errores.

Con esta forma de trabajo se pretende la división del proyecto en pequeñas partes para agilizar su completación y entrega, con el fin de desarrollar servicios y productos de calidad que satisfagan las necesidades de clientes cuyos requisitos varían a una velocidad cada vez mayor.

4.3. Ventajas

Son numerosas las ventajas que aporta la metodología de desarrollo ágil, entre ellas se pueden destacar las siguientes:

- Un trabajo de mayor eficiencia y velocidad. El hecho de trabajar con un proyecto dividido en particiones más pequeñas agiliza la producción y ofrece entregas de calidad en intervalos de tiempos más cortos.
- Ahorro de tiempo y costes. Al tratarse de un método de trabajo que se basa en la rapidez y eficiencia, permite el cumplimiento de los plazos de entregas acordados, así como llevar a cabo un adecuado presupuesto durante el desarrollo del proyecto.
- Mejora de la calidad del producto. Debido a la continua comunicación que existe entre los desarrolladores y el cliente a lo largo del desarrollo, se asegura que el producto final cumpla las expectativas que el cliente buscaba. De este modo se intenta que el producto sea lo más perfecto en cuanto a calidad se refiere.

- Posibilidad de reacción rápida frente a problemas y errores. Gracias a la continua evaluación que se realiza a lo largo del proyecto es posible detectar fallos y darles una solución de manera rápida y concisa.
- Clientes más satisfechos. Las ventajas anteriores se traducen en una mejora significativa de la satisfacción del cliente, el cual se sentirá importante en el desarrollo del proyecto debido a su compromiso e implicación.
- Equipo de desarrollo más implicado y motivado. Conocer el estado del proyecto en cualquier instante permite al equipo debatir y aceptar los compromisos y requisitos del mismo.
- Inversiones más rentables. Cabe destacar que las tempranas entregas al cliente le permitirá explotar antes sus productos o servicios, con lo cual, el retorno de la inversión será más rápido.

4.3.1 Metodologías ágiles frente a las tradicionales.

A continuación se muestra una comparativa entre ambas metodologías:

Tabla 4-1. Comparación entre metodologías (Fuente: Elaboración propia)

Metodologías ágiles	Metodologías tradicionales
Flexibilidad y capacidad de respuesta rápida ante modificaciones en el proyecto.	Rigidez y capacidad de respuesta lenta ante modificaciones en el proyecto.
Clientes más comprometidos e implicados en el desarrollo del proyecto.	Mínima interacción entre clientes y desarrolladores.
Pequeños equipos de trabajo (un promedio de 10 integrantes).	Grandes grupos de trabajo y con distintas distribuciones.
Tiempos de entregas más cortos.	Tiempos de entregas más extendidos.
Gran diversidad de roles en el equipo de desarrollo.	Mínima diversidad de roles en el equipo de desarrollo.
Pocas normas y políticas, procesos menos controlados.	Gran cantidad de normas y políticas, procesos muy controlados.
Garantía de que el producto cumpla las expectativas del cliente (evaluación continua).	No hay seguridad de que el producto cumpla todos los requisitos del cliente.
Posibilidad de respuesta ante cambios.	Obligado seguimiento estricto de lo planeado al comienzo del desarrollo.
Poca dependencia de la arquitectura de software.	Dependencia mediante modelos de la arquitectura software.

4.4. Métodos ágiles

En la actualidad existen diversos métodos que enfocan su gestión de desarrollo en las metodologías de desarrollo ágil. Entre las opciones más usadas e importantes se encuentran: programación extrema (XP), Kanban o SCRUM, de la cual se hará un exhaustivo estudio en este trabajo. No obstante, existen otros métodos menos conocidos pero no por ello menos importantes. Un claro ejemplo es Design Sprint, que es una metodología de diseño que actualmente utiliza una de las empresas más importante del mundo, Google. A continuación se detallan algunos de estos métodos.

4.4.1 Extreme Programming (XP)

Extreme Programming, también conocido por sus siglas XP, es una metodología formulada en 1999 por Kent Beck y que se recoge en su libro *Extreme Programming Explained: Embrace Change*. Esta metodología está enfocada en potenciar las relaciones personales entre los componentes del equipo de trabajo y entre éstos y los clientes, de este modo se fomenta un clima de colaboración, trabajo en equipo y comunicación. Con esto se pretende eliminar los tiempos muertos o que éstos sean los mínimos posibles.

Se trata de una herramienta perfecta para equipos de trabajos reducidos debido a su enfoque en las relaciones entre los miembros, por lo que es muy oportuna para aquellas empresas que están empezando y todavía no están consolidadas.

Extreme Programming se caracteriza por ser un método basado en la simplicidad y satisfacción al cliente. Sus 5 valores más representativos son:

- **Comunicación.** Toda la comunicación entre desarrolladores y clientes se efectúa cara a cara, intercambiando frecuentemente opiniones y puntos de vistas en cada una de las reuniones que se concertan.
- **Respeto.** Los desarrolladores muestran un gran respeto a su trabajo, lo que se traduce en un compromiso con el nivel de calidad del producto final, buscando de este modo el mejor diseño posible del mismo. En cuanto al entorno de trabajo, se pretende un clima de colaboración y apoyo, respetando en cada momento a los demás compañeros y sus opiniones. Esto mejora los niveles de motivación y compromiso hacia el proyecto.
- **Retroalimentación.** El feedback continuo que se lleva a cabo entre desarrolladores y clientes permite detectar, localizar y solucionar posibles fallos del sistema. De igual modo, el cliente tiene la posibilidad de testear periódicamente el funcionamiento del producto y contrastar el cumplimiento de sus requisitos.
- **Coraje.** Un equipo exigente consigo mismo y persistente en el buen funcionamiento de su trabajo es fundamental para el desarrollo óptimo del proyecto. No dejar para mañana el trabajo que se pueda hacer hoy, o no tener miedo a la hora de refactorizar, son algunas de las ideas que representa este valor.
- **Simplicidad.** Un diseño simple facilita las comunicaciones entre el equipo, ya que es más fácil de comprender e implementar en el proyecto. Además, XP se preocupa por atender y solucionar los problemas más recientes que se vayan produciendo sin tener en cuenta los que se puedan producir en un futuro. De este modo, se ahorra tiempo y esfuerzo en futuros requerimientos que podrían variar o no llegarán a sucederse.



Figura 4-3. Valores de XP (Fuente: oficinaproyectosinformatica.blogspot.com)

Con estos valores, la programación extrema busca adaptar sus proyectos a sistemas simples, con códigos claros y sin documentación. De este modo se consigue un proceso más eficiente y una optimización de los tiempos de desarrollo.

Las etapas que componen el método XP son las siguientes:

- Planificación del proyecto. Junto al cliente se plantean los requisitos y se planifica el desarrollo del proyecto.
- Diseño del proyecto. Se elabora un diseño simple del proyecto ayudándose de prototipos.
- Codificación. Se lleva a cabo una programación en parejas para mejorar los resultados en cuanto a eficiencia y calidad.
- Pruebas. Se realizan pruebas unitarias de manera periódica para una integración continua del producto. Antes del lanzamiento se tienen que superar las pruebas de aceptación.



Figura 4-4. Esquema de Extreme Programming. (Fuente: <https://openwebinars.net/>)

Entre las ventajas e inconvenientes del método XP se pueden destacar:

Tabla 4-2. Ventajas e inconvenientes del método XP (Fuente: Elaboración propia)

VENTAJAS	INCONVENIENTES
Excelente comunicación entre miembros del equipo y cliente.	Sólo recomendable para proyectos de corto plazo.
Programación organizada.	Elevadas comisiones en caso de fallo.
Posibilidad de adaptar a cualquier lenguaje de programación.	Posibilidad de que el proyecto crezca conforme a las entregas.
Total control del cliente en sus prioridades.	No se tiene información del coste y tiempo total.
Pruebas y evaluación continua durante el proceso.	Buscar disponibilidad del cliente.
Tasa de fallos muy pequeña.	
Ahorro de tiempo y dinero.	
Buena implementación en las nuevas tecnologías.	

4.4.2 Kanban

Kanban es una metodología de desarrollo que tiene sus inicios en la industria manufactura. Taiichi Ohno (1912-1990) fue un ingeniero industrial japonés conocido por implementar por primera vez este sistema en las fábricas automovilísticas de Toyota, cuyas ideas de producción industrial se recogen en su libro: *Toyota Production System: Beyond Large-Scale Production*. Toyota llegó a convertirse en un referente de la industria automovilística gracias a este sistema, el cual se sigue usando hoy en día por muchas empresas actuales.

No obstante, fue David J Anderson, el que en su obra: *Kanban: Cambio evolutivo exitoso para su negocio de tecnología*, formula definitivamente este sistema dentro de las metodologías de desarrollo de software. En él se define el método Kanban como “*un enfoque incremental y evolutivo de cambios en sistemas y procesos para organizaciones*”³.

David J. Anderson expone tres principios fundamentales que se deben cumplir en Kanban:

1. *Empieza con lo que estás haciendo*
2. *Decide que vas a seguir cambios incrementales y evolucionarios*
3. *Respetar el proceso en curso, sus roles, responsabilidades y títulos*⁴

Con este sistema se pretende la división del proceso de desarrollo del producto en tareas bien definidas y delimitadas, que se deben realizar con la mayor calidad posible y en el orden prescrito para poder avanzar a la siguiente y de este modo garantizar un proceso de fabricación de calidad. El objetivo es obtener un proceso lo más eficiente posible a través de la organización y el trabajo equipo. El hecho de trabajar en tareas más concretas y sencillas permite obtener un producto más fiable y de calidad, y unos tiempos de desarrollos más cortos.

Como bien indica su nombre en japonés (“Kan” significa visual y “ban” tarjeta), Kanban es una metodología caracterizada por lo visual y que usa un tablero o pizarra donde se exponen las tareas definidas en tarjetas, las cuales se irán organizando según la fase de desarrollo en la que se encuentre en ese momento.



Figura 4-5. Tablero básico Kanban. (Fuente: <https://kanbantool.com>)

Partiendo de las tareas básicas: “Para hacer”, “En curso” y “Hecho”, el tablero se puede complementar con más estados que se consideren oportunos para el proyecto en cuestión. Algunos de los más usados son “Backlog”, “Pendiente de validación” o “En producción” entre otros.

³ Definición de Kanban según David J Anderson en *Kanban: Cambio evolutivo exitoso para su negocio de tecnología*, Nov 2016.

⁴ Principios fundamentales definidos por David J Anderson.

El número de tareas en cada fase del proceso (Work In Process, WIP) es limitado, de modo que en una fase no puede haber más tarjetas de las permitidas. En caso de que el WIP sea máximo en una fase, se tendrá que finalizar una de las tareas para poder introducir otra. Esta restricción sirve al equipo como método de autocontrol e indica si el desarrollo del producto lleva un flujo adecuado y sin atascos debido a una mala organización.

También es usual organizar las tareas en prioridades u orden de desarrollo colocándolas en diferentes filas o utilizando tarjetas de distintos colores, de este modo, no se podrá continuar con el resto de tareas hasta que las más prioritarias no estén realizadas.

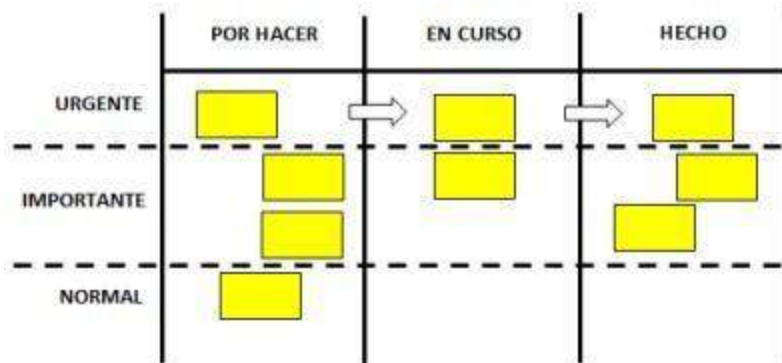


Figura 4-6. Tablero Kanban. División de prioridades por filas. (Fuente: <https://www.pinterest.es/>)

Las tareas que se encuentren en la fila “Urgente” tendrán más prioridad y deberán realizarse antes que el resto de tareas. De igual modo ocurre con la fila “Importante” respecto a la fila “Normal”.

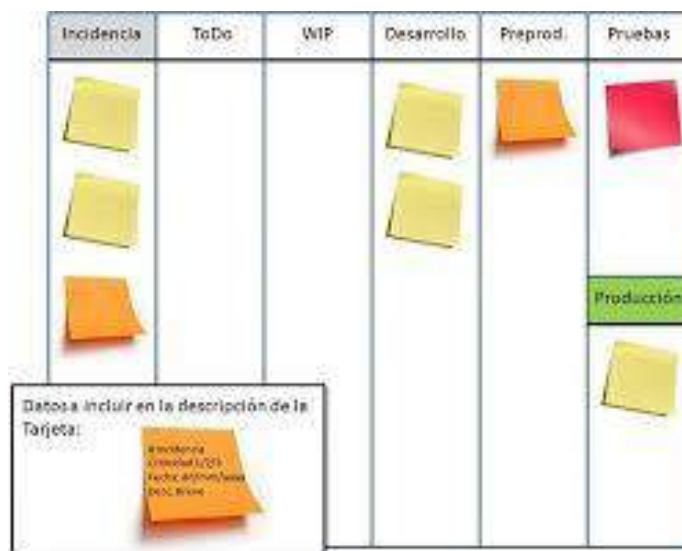


Figura 4-7. Tablero Kanban. División de prioridades por colores. (Fuente: <https://arrizabalagauriarte.com>)

En este caso, las tareas irán escritas en tarjetas de diferentes colores: rojo, naranja y amarillo, siendo las rojas las más prioritarias y las amarillas las menos importantes.

La visualización en tiempo real del flujo de trabajo y su progreso, junto con un límite de WIP establecido, permite la supervisión en todo momento del proyecto, lo cuál permite su análisis y la posibilidad de introducir cambios en el sistema de producción que tengan efectos positivos en el resultado final.

Es importante aclarar que la pizarra o tablero Kanban se puede implementar en otras metodologías de desarrollo de software sin la necesidad de que las ideas y características de este método se lleguen a aplicar. Scrumban (Kanban + Scrum) es un ejemplo de integración de la pizarra Kanban.

Las ventajas e inconvenientes que presenta el método Kanban:

Tabla 4-3. Ventajas e inconvenientes del método Kanban (Fuente: Elaboración propia)

VENTAJAS	INCONVENIENTES
Sistema de producción flexible frente a la demanda.	Sólo recomendable para procesos de producción repetitivos.
Optimización de los tiempos de producción y de entrega.	No permite plazos de abastecimientos grandes.
Se evitan procesos innecesarios que aumenten el tiempo del desarrollo.	El almacenamiento y manejo de stock puede resultar muy costoso.
Control y supervisión del proyecto en todo momento.	
Productos fiables y de calidad.	
Garantía de un flujo de trabajo adecuado y sin atascos.	
Fácil integración con otras metodologías de desarrollo.	

4.4.3 Design Sprint

Design Sprint es una metodología ágil de diseño desarrollada por Google en 2010 y que hoy en día es utilizada por muchas empresas emergentes como pueden ser Uber, Slack, Pocket o Nest, las cuales buscan la mayor rentabilidad para sus negocios. Design Sprint puede adaptarse a empresas de todos los tamaños y a un gran número de equipos multidisciplinares.

Este método se caracteriza por acotar el ciclo de diseño de un producto a una semana, con el fin de obtener una rápida visión del producto final y que se acorde con los requisitos demandados por el cliente. Para ello, se hace uso de procesos denominados *Sprints*, los cuales Google definen de la siguiente manera:

*“Un Sprint de diseño es un proceso de cinco fases que ayuda a responder preguntas críticas de negocio a través de prototipos rápidos y pruebas de usuario. Los Sprints permiten a los equipos alcanzar metas y entregas claramente definidas y obtener aprendizajes clave rápidamente. El proceso ayuda a generar innovación, fomenta el pensamiento centrado en el usuario, alinea a tu equipo bajo una visión compartida y te permite lanzar el producto más rápido.”*⁵

⁵ Definición de Sprint recogida en DesignSprintKit, plataforma de aprendizaje de Google

Los Sprints se dividen en 5 fases, las cuales se pretenden abordar en 5 días consecutivos (una fase cada día). En estas sesiones se busca la participación del mayor número de miembros del equipo posible (ingenieros, responsables de marketing, diseñadores, etc.) para así tener una gran variedad de puntos de vista diferentes y lograr más rápido una solución efectiva. Se recomienda la presencia de un Sprint Master, encargado de dirigir y gestionar el proceso para que se garantice una correcta ejecución del mismo.

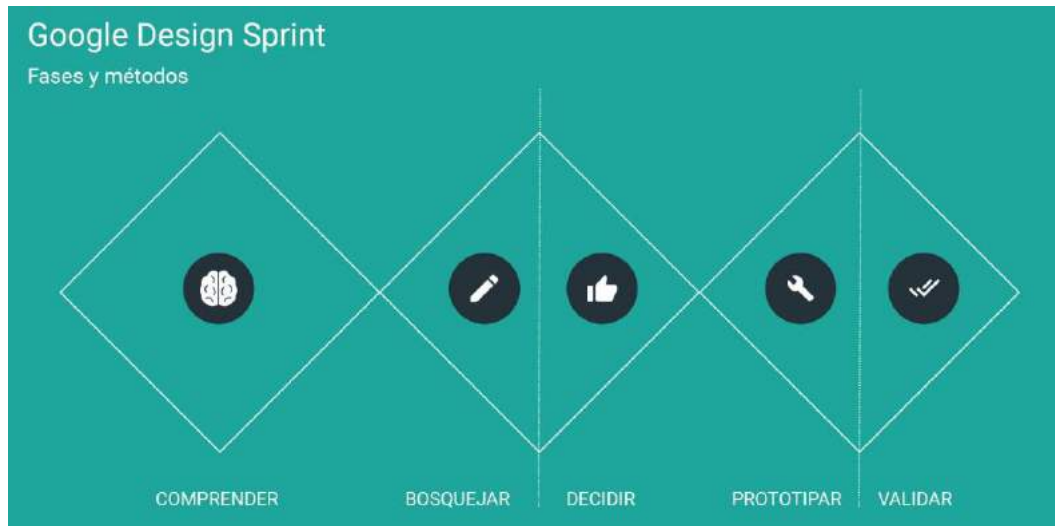


Figura 4-8. Fases de Google Design Sprint (Fuente: <https://designSprintkit.appspot.com/>)

Las 5 fases que componen el proceso son:

- **Día 1 – Entendimiento.** En el primer día se plantea el objetivo y sus problemas desde todos los puntos de vista posibles con el fin de concretar una base de conocimiento común y comprendida por todos. Se debaten varios puntos como: oportunidades de negocios, público al que va destinado, competencias, etc.
- **Día 2 – Propuestas.** Se exponen y se desarrollan distintas soluciones trabajadas de forma individual o en pequeños equipos de trabajo. Éstas deben de estar bien detalladas y entendibles para el resto de componentes, sin tener aún en cuenta su viabilidad.
- **Día 3 – Decisión.** Durante este día se estudian las propuestas del día anterior y se escogen aquellas que presenten mayor probabilidad de cumplir los objetivos.
- **Día 4 – Prototipado.** Se lleva a cabo el diseño y construcción de prototipos que reflejen de la manera más fiel posible la propuesta elegida en la fase anterior. Al final de este día debe estar todo listo para la fase final del día siguiente, para ello se prepara un guión, una presentación y se revisa que el prototipo funcione correctamente.
- **Día 5 – Validación.** En el último día se realizan pruebas y se expone lo trabajado hasta ahora delante de un público seleccionado, del cual se busca reacciones y sensaciones. Los resultados obtenidos son de gran importancia ya que marcarán el camino a seguir del proyecto.

Al finalizar un Design Sprint se obtienen tangibles como pueden ser prototipos, notas, diagramas, gráficos, informes de pruebas, etc. los cuales permiten afrontar con más decisión y claridad el desarrollo del proyecto, ya que nos da una visión más definida y compartida del producto que se va a desarrollar.

4.4.4 Otras metodologías ágiles

Dentro del amplio catálogo de metodologías ágiles se pueden encontrar otras de menos relevancia pero que también merecen ser nombradas. A continuación se muestran algunas de ellas, así como sus principales características.

El método Scrum no será incluido en la lista ya que se trata de una de las metodologías más importantes y será estudiada en profundidad en el siguiente apartado.

- **Agile Inception**

Técnica orientada en lograr un consenso entre todos sus trabajadores con el fin de reducir las incertidumbres y los riesgos del proyecto. Para ello se llevan a cabo numerosas reuniones durante largos períodos de tiempo y en ellas se debaten la viabilidad y el nivel de riesgo del proyecto que se pretende abordar.

- **Lean Software Development (LSD)**

Esta técnica japonesa tiene su origen en la empresa Toyota y fue una estrategia de fabricación muy exitosa en las industrias durante el período de recuperación de la segunda guerra mundial, tanto es así que a día de hoy se sigue haciendo uso de ella para el desarrollo de software.

Taiichi Ohno, ingeniero precursor del sistema Kanban, fue también el artífice de esta metodología. Su técnica se fundamentó en siete principios:

1. *Eliminar desperdicios.*
2. *Amplificar el aprendizaje.*
3. *Decidir lo más tarde posible.*
4. *Entrega lo más rápido posible.*
5. *Capacitar, potenciar, al equipo.*
6. *Construir con integridad.*
7. *Ver el todo.*⁶

- **Crystal Clear**

Alistair Cockburn crearía esta metodología que a su vez se subdivide en varios tipos de métodos en función del número de participantes del proyecto.

- Clear (de 0 a 6 personas).
- Amarillo (de 7 a 20 personas).
- Naranja (de 21 a 40 personas).
- Rojo (de 41 a 80 personas).
- Marron (de 81 a 200 personas).

⁶ Los 7 principios de LSD establecidos por Taiichi Ohno

Cada método posee unas técnicas determinadas según el tipo de proyecto en cuestión pero todos comparten unos principios y valores característicos de la metodología.

Esta metodología hace hincapié en el factor humano y, por tanto, sus principios están enfocados al equipo del proyecto y la relación entre sus componentes. Las entregas frecuentes, la comunicación entre los trabajadores y la organización en roles son algunas de las propiedades de este método ágil basado en ciclos iterativos de desarrollo incremental.

- **Open Unified Process (OpenUP)**

Se trata de un método desarrollado por varias empresas tecnológicas y que se caracteriza por ser un proceso mínimo y suficiente, lo cual quiere decir que sólo se llevan a cabo tareas y procesos del proyecto que tengan contenidos estrictamente necesarios y fundamentales.

Open UP permite asignar distintos roles a su equipo de desarrollo, como pueden ser: analista, arquitecto, desarrollador, stakeholders, tester, etc. Éstos trabajan conjuntamente el proyecto, el cual se divide en iteraciones planificadas en períodos de tiempo de una semana. Se basa en un desarrollo incremental y centrado en la arquitectura para minimizar los riesgos y mejorar la organización del proceso.

5 SCRUM

“Scrum es como tu suegra, resalta todos tus fallos”.

-Ken Schwaber -

Scrum es actualmente la metodología ágil más influyente e importante dentro del panorama empresarial. Son muchas las empresas que optan por su implantación en el interior de sus departamentos para optimizar la calidad de sus proyectos y minimizar los costes y tiempos invertidos en ellos. Es por esto por lo que se dedica un capítulo integro al análisis en profundidad de este método.

5.1. Introducción

Son muchas las compañías que hacen uso del método Scrum para la gestión de procesos en sus departamentos, tanto es así que hoy en día lo usan más del 70% de los equipos de desarrollo. A pesar de ser un método que vió la luz por la década de los noventas, Scrum ha sabido evolucionar y adaptarse a los retos y exigencias del mundo actual. No obstante, Scrum no se caracteriza por ser un método simple y sencillo de aplicar, muy pocas empresas consiguen sacarle el máximo partido a Scrum y muchos de los beneficios que aporta el método se pierden por el camino. Aquellas empresas que consiguen su implementación pueden ver como su productividad y eficiencia mejoran de 4 a 10 veces más, utopía que está al alcance de muy pocas.

5.2. Historia

El origen de Scrum puede situarse en 1986, Ikujiro Nonaka y Takeuchi plasmaron en el artículo “El nuevo juego para el desarrollo de productos” la manera de producir de importantes empresas manufactureras como eran Canon, Honda, HP y Fuji-Xerox entre otras. Estas empresas producían a nivel mundial con un enfoque progresivo y ayudándose de equipos de trabajos auto-organizados. Tanta era la importancia que le daban al trabajo colaborativo que optaron por denominar al método Scrum (melé en español) ya que les recordaba a la posición que tomaban los equipos de rugby al empezar una jugada, una formación en masa donde todos los jugadores se ayudan entre sí para conseguir la posesión del balón. Nonaka y Takeuchi expresaron la analogía y semejanza del rugby y Scrum con la siguiente metáfora:

“El enfoque de “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 avanza, sirve mejor a los desarrollos competitivos que se ven hoy en día.”⁷



Figura 5-1. Melé en rugby (Fuente: [https://es.wikipedia.org/wiki/Scrum_\(rugby\)\)](https://es.wikipedia.org/wiki/Scrum_(rugby)))

Gracias a este artículo se consigue establecer las ideas y principios bases del método Scrum y serviría de apoyo e influencia para que posteriores autores definiesen el método tal y como lo conocemos hoy en día.

Años más tardes, en 1993 se utilizó el método Scrum por primera vez y se hizo de la mano de Jeff Sutherland, el cual lo aplicó en el desarrollo de software para su empresa Easel Corporation.

Sin embargo, sería en 1995 cuando se publicaría el primer informe del método Scrum como tal. Ken Schwaber expuso en OOPSLA 95 su libro “Scrum Development Process”, en él se describía detalladamente los principios e ideas de Scrum y cómo habían sido aplicados por Jeff Sutherland en su empresa Easel Corporation, y por él mismo en su compañía Delphi.

Allá por los años noventas, la situación del mercado dejaba mucho que desear. La planificación de proyectos suponía años de desarrollos y elevados costes que no garantizaban los resultados deseados. Un claro ejemplo fue el desastroso software para la gestión del sistema sanitario Diraya que la Junta de Andalucía intentó desarrollar, en el cual se invirtieron unos 45 millones de euros y que años más tarde se tuvo que volver a diseñar ya que el funcionamiento que ofrecía no era el idóneo.

⁷ Analogía del rugby y Scrum. Ikujiro Nonaka y Takeuchi (1986)

Para intentar revertir esta situación era necesario que métodos ágiles como Scrum se dieran a conocer y se propagaran sus ideas y principios por todo el mundo. Ésto llegaría con el manifiesto ágil que Kent Beck y diecisiete críticos más firmaron en 2001. Algunos de estos participantes formaron la *Alianza ágil*, una organización sin fines de lucro que se encargaría de difundir y promover por todo el mundo la metodología ágil para el desarrollo de aplicaciones.

Desde entonces, miles de empresas por todo el mundo han optado por aplicar el método Scrum en sus proyectos para el desarrollo de productos, englobando tanto a pequeñas empresas como a multinacionales.

Tabla 5-1. Empresas que usan Scrum (Fuente: <https://proyectosagiles.org/historia-de-scrum/>)

Sectores	Ejemplos de empresas que utilizan Scrum
Media y Telcos	BBC, BellSouth, British Telecom, DoubleYou, Motorola, Nokia, Palm, Qualcomm, Schibsted, Sony/Ericsson, Telefonica I+D, TeleAtlas, Verizon
Software, Hardware	Adobe, Autentia, Biko2, Central Desktop, Citrix, Gailén, IBM, Intel, Microfocus, Microsoft, Novell, OpenView Labs, Plain Concepts, Primavera, Proyectalis, Softhouse, Valtech, VersionOne
Internet	Amazon, Google, mySpace, Yahoo
ERP	SAP
Banca e inversión	Bank of America, Barclays Global Investors, Key Bank, Merrill Lynch
Sanidad y Salud	Patientkeeper, Philips Medical
Defensa y Aeroespacial	Boeing, General Dynamics, Lockheed Martin
Juegos	Blizzard, High Moon Studios, Crytek, Ubisoft, Electronic Arts
Otros	3M, Bose, GE, UOC, Ferrari

5.3. Definición

Se identifica Scrum como un proceso de gestión de proyectos que simplifica y facilita el desarrollo de los productos y garantiza la satisfacción y cumplimiento de todos los requisitos y necesidades que exigen los clientes. El método Scrum se basa en la experiencia y en la observación de los hechos, es decir, se trata de un proceso empírico que continuamente tiene que hacer frente a lo imprevisible y dar solución a los problemas que se van sucediendo.

Se trata de un marco de trabajo que, como todas las metodologías ágiles, promueve el trabajo en equipo y donde la comunicación entre los miembros que componen la plantilla es fundamental para obtener resultados más eficientes.

Durante el proceso de desarrollo se llevan a cabo de forma regular un conjunto de prácticas, las cuales se apoyan unas sobre otras y que son definidas al comienzo de la planificación, buscando siempre la máxima capacidad de producción. Se realizan, de manera regular, entregas parciales del producto final al cliente para que éste pueda dar su valoración, comprobar que todo va según lo previsto o realizar modificaciones, lo cual aporta una gran satisfacción y beneficio al cliente. Por ello, Scrum está especialmente orientado a proyectos complejos con necesidades variables y pocas definidas, o para resolver situaciones donde el cliente no está recibiendo lo que necesita, los plazos de entregas son excesivamente largos, los costes son demasiados elevados o incluso exista la necesidad de reaccionar ante la competencia.

5.4. Características

A continuación se exponen un conjunto de características que definen el enfoque y visión de esta metodología:

1. Una de las principales características es que se trata de un método basado en el desarrollo incremental e iterativo, a diferencia de otros procesos donde la planificación y producción se realizan de forma continua y completa. Esto quiere decir que el desarrollo del producto se realiza por partes, entregas que se realizan periódicamente y que van incrementando su funcionalidad con respecto a la anterior. Cada una de estas entregas o iteraciones se denominan *Sprint*, y se realizan de manera continua hasta que el cliente de por terminada la ejecución del proyecto.
2. A diferencia de los procesos secuenciales o basados en el método cascada, las tareas o fases de desarrollo en Scrum se llevan a cabo de manera simultánea, solapándose unas con otras, para de este modo conseguir agilizar el proceso y dotarlo de rapidez.
3. El equipo ágil trabaja dando prioridad a aquellos procesos o requisitos más importantes para el cliente.
4. El continuo seguimiento y revisiones durante el desarrollo aportan al proceso transparencia e información constante, características muy significativas de un equipo ágil.
5. Con este método se busca que el producto resultante sea de la mayor calidad posible y para ello pone absoluta confianza en las capacidades y conocimientos de equipos de trabajo auto-organizados, así como la colaboración entre los miembros, quedando la calidad de los procesos empleados en segundo plano.

5.5. Herramientas

Para que Scrum se aplique y ejecute correctamente se hace uso de un conjunto de herramientas y funciones que diferencian a esta metodología de las demás. Scrum se apoya en un conjunto de eventos y artefactos que se llevan a cabo a lo largo del proceso de desarrollo por un equipo de trabajo en los que se imparten distintos roles.

5.5.1 Artefactos

Existen 3 tipos de elementos o artefactos dentro del marco de trabajo de Scrum, los cuales están destinados a dotar al proyecto de transparencia de información necesaria para la toma de decisiones. Los artefactos son los que siguen:

1. **Product Backlog.** Es el artefacto más importante y sobre el que recae toda la información relacionada con el producto. Se trata de un inventario de tareas, características y requisitos necesarios para la realización del producto. Se crea a partir de una visión inicial del proyecyo y está en continuo

crecimiento y evolución a lo largo del desarrollo. Por tanto, es conveniente comenzar con los requerimientos más prioritarios al comenzar con Scrum e ir añadiendo elementos conforme se vayan estableciendo más necesidades del producto.

La manipulación y gestión de este documento queda en manos del Product Owner, el único responsable de su contenido.

El Product Backlog es un artefacto vivo, esto quiere decir que el inventario podrá sufrir modificaciones mientras el proyecto esté en desarrollo. Estos cambios están sujetos a feedbacks, intereses de negocios, respuestas a la competencia, etc.

Los ítems recogidos en el Product Backlog poseen los siguientes atributos:

- **Descripción** del ítem. Da a conocer las tareas a realizar mediante una pequeña descripción de las mismas.
- **ID**. Permite registrar de manera abreviada los distintos ítems.
- Orden de **prioridad** del desarrollo del ítem. Se basa en los beneficios que aporte, los riesgos, costes, etc. Se puede expresar de forma numérica o mediante una clasificación escrita, como ocurre en este caso.
- **Estimación** de esfuerzo y complejidad de desarrollo. La estimación es un proceso bastante complejo y hay que tener en cuenta que nunca será del todo precisa debido a la gran cantidad de variables a evaluar. No obstante, añade información importante sobre las tareas y ayuda a planificar los Sprints de una manera más equitativa en cuanto a esfuerzo. La técnica de estimación más utilizada es el Planning Poker, la cual consiste en reuniones donde todos los componentes del equipo Scrum valoran el esfuerzo y complejidad de cada ítem de manera individual. Posteriormente, cada miembro da a conocer su veredicto y se hace consenso para acordar la estimación más representativa para todos. Hay muchas formas de valorar la estimación (porcentajes, series de números, etc.), en este caso se hará uso de una escala numérica, siendo 1 el mínimo esfuerzo y 10 el máximo.
- **Valor** o peso del ítem. Relacionado con la importancia que tiene la tarea en el Sprint. Es un parámetro que ayuda a identificar aquellas tareas con las que se debe de ser más cauteloso, aquellas que son más importantes. Al igual que la estimación, se valora mediante una escala numérica del 1 al 10.

Tabla 5-2. Ejemplo de Product Backlog (Fuente: Elaboración propia)

Prioridad	ID	Descripción	Estimación (1-10)	Valor (1-10)
Muy alta	001	Prediseño	7	8
Media	002	Análisis y previsión de daños	3	5
Baja	003	Interfaz de usuario	6	6
Alta	004	Gestión de direcciones	2	3

2. **Sprint Backlog**. Tabla que recoge los elementos o ítems del Product Backlog a desarrollar durante un Sprint (iteración de desarrollo) e indica el estado en el que se encuentran durante el proceso. Permite dar una visión general del trabajo a realizar durante la iteración y es el equipo de desarrollo el encargado de su gestión.

El Sprint Backlog aporta información y seguimiento de los distintos ítems, de este modo se puede saber en cualquier momento el estado de cada uno de ellos, si se encuentra en espera, en desarrollo o si se ha finalizado.

Al igual que el Product Backlog, se trata de un artefacto vivo que permite modificaciones y alteraciones mientras se trabaja sobre él. El equipo de desarrollo irá actualizando las estimaciones y objetivos del Sprint a medida que se avanza en la iteración, para así obtener resultados más exactos y puntuales.

La representación del Sprint Backlog puede ser llevada a cabo mediante tableros o pizarras Kanban, de este modo se consigue integrar ambos métodos y facilitar el desarrollo del proceso. La combinación de ambas metodologías puede denominarse como *Scrumban*.

Tabla 5-3. Ejemplo de Sprint Backlog (Fuente: <https://muyagile.com/product-Backlog-y-Sprint-Backlog/>)

Sprint Backlog			
Producto Web de Compra de Libros	Pendiente	En Progreso	Finalizado
Nº Sprint: 04	Historia #1	Tarea 1.2 Tarea 1.3	Tarea 1.1 Tarea 1.4
Objetivo del Sprint <i>El objetivo de este Sprint es que el usuario pueda completar una búsqueda de libros por autor y ordenarla por precio de compra ascendente y descendente, así como por año de publicación.</i>			Historia #2 Tarea 2.1 Tarea 2.2 Tarea 2.3
	Historia #3 Tarea 3.5	Tarea 3.3 Tarea 3.4	Tarea 3.1 Tarea 3.2
	Tarea Técnica #04 Subtarea 4.1	Subtarea 4.2	
	Spike #1 Bug 001 Bug 012 Bug 005		

3. **Incremento.** Entrega parcial del producto que se obtiene como resultado al finalizar el Sprint Backlog. Se trata de un entregable de una nueva versión del producto, el cual debe de estar terminado completamente y listo para su despliegue. Como su nombre indica, todo producto terminado al final de un Sprint debe ser potencialmente más funcional que el anterior, ya que se trata de un método ágil incremental.

Antes de la entrega del producto al cliente, el Product Owner tiene la última palabra y es el encargado de decidir si se lanza o no la versión, todo bajo su criterio.

5.5.2 Roles

Cada uno de los miembros que componen un equipo Scrum lleva asignado un papel y unas responsabilidades que debe impartir durante el proyecto. Estos roles son imprescindibles para la realización exitosa de cada una de las fases del proyecto y son los siguientes:

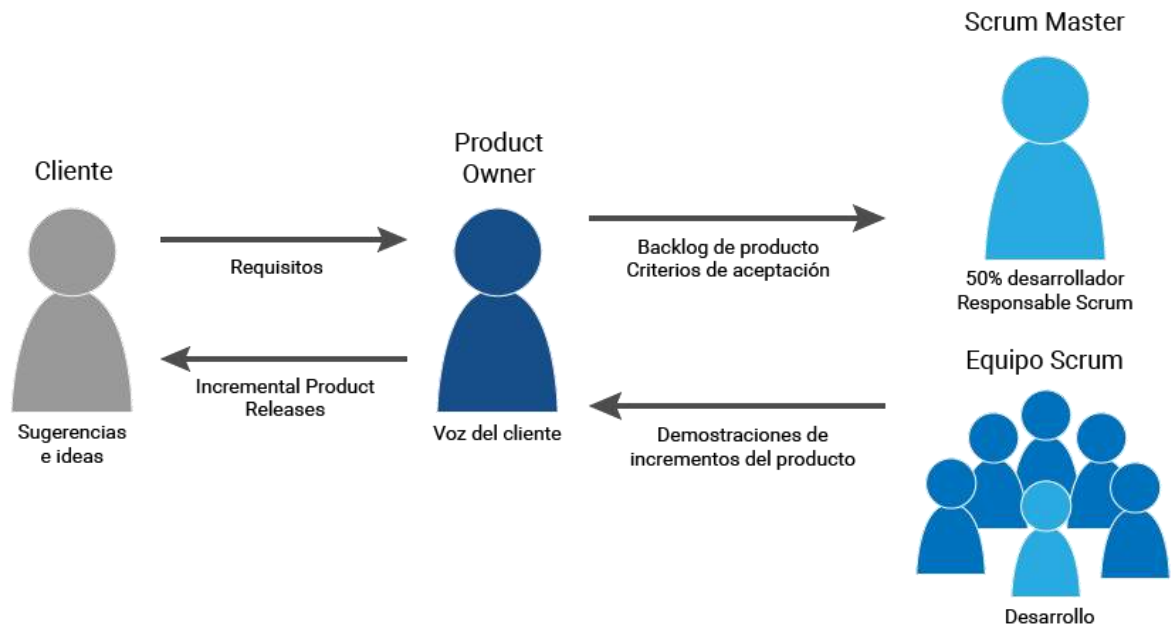


Figura 5-2. Roles de Scrum (Fuente: <https://desire.webs.uvigo.es/contenidos/scrum/>)

1. **Product Owner (Dueño del producto).** Se trata del rol central del proyecto, sobre el que recae el peso del negocio. Es la figura que representa a los accionistas y clientes, incluso en algunas ocasiones puede ser el propio cliente quien tome esta responsabilidad. Entre las funciones y responsabilidades del Product Owner se encuentran:
 - Asegura la viabilidad del negocio optimizando el proyecto y maximizando el valor del producto y la inversión.
 - Actúa de interlocutor transmitiendo las necesidades, peticiones y requerimientos de los clientes.
 - Gestión del Product Backlog. Garantizar que esté estructurado, detallado y priorizado.
 - Supervisa cada entrega del producto y sugiere modificaciones y adaptaciones para las siguientes iteraciones.
 - Tiene el poder de tomar cualquier decisión referente al producto y al desarrollo del proyecto del mismo.
2. **Scrum Master.** Es la persona sobre la que recae el liderazgo del equipo Scrum, encargada de impulsar el método y asegurar que se cumplen sus bases y principios. Otras de sus responsabilidades son:
 - Velar por el cumplimiento de normas y fases de la metodología.
 - Gestionar el proceso y sus iteraciones, garantizando su correcta ejecución y ayudando a facilitar las mecánicas de trabajo.

- Eliminar cualquier obstáculo o contratiempo que pueda perjudicar las entregas del producto.
 - Se encarga de las tareas de formación y aprendizaje.
 - Motivar e incentivar a los componentes del equipo de trabajo.
 - Concertar y planificar las reuniones y eventos.
 - Colaborar con el Product Owner para maximizar el retorno de la inversión (ROI)
3. **Scrum Team (Equipo de desarrollo).** Habitualmente suele estar formado por un equipo de entre 3 a 9 trabajadores. Son los responsables del desarrollo del producto y de que se lleven a cabo las diferentes entregas incrementales al final de cada iteración. Entre los profesionales que componen el equipo podemos encontrar: ingenieros, diseñadores, programadores, arquitectos, agentes de marketing, etc. Se caracterizan por ser equipos auto-organizados y auto-gestionados. Sus funciones son:
- Cumplir con la planificación del proyecto y desarrollar las diferentes tareas que se les ha sido asignadas.
 - Cumplir con los plazos de entrega y requisitos (Product Backlog) de cada iteración, aportando siempre un incremento de funcionalidad a cada producto finalizado.
 - Poner a disposición de la empresa sus conocimientos y capacidades para cualquier necesidad.

5.5.3 Eventos

Con el fin de reducir el número de reuniones no planificadas y fomentar un clima de colaboración y comunicación, existen 5 eventos imprescindibles dentro del marco de trabajo de Scrum, todos ellos con una duración programada y fija denominada *Timebox*. Los eventos son los siguientes:

1. **Sprint.** Evento más importante y sobre el que se apoya el método Scrum. Se puede definir Sprint como cada una de las iteraciones, de tiempo definido (normalmente entre 15 y 30 días), que se llevan a cabo a lo largo del proceso y hasta que se consigue la finalización completa del producto. Cada Sprints solo puede iniciarse cuando el anterior haya concluido, no se pueden cometer dos Sprints de forma simultánea. Este evento puede considerarse como un metaevento ya que contiene a los demás eventos que se llevan a cabo en Scrum.

Una vez que el Sprint está en proceso no es aconsejable su alteración ya que podría perjudicar en gran medida al equipo de desarrollo y pondría en peligro la consumación de los objetivos. De igual modo, una prolongación de tiempo de más de un mes para la finalización del Sprint provocaría un aumento de riesgos y complejidad. Por ello, es importante que a la hora de planificar un Sprint queden bien definidos y aclarados todos los objetivos y procedimientos del mismo.

En caso de que sea necesaria una cancelación de un Sprint antes de su finalización, es el Product Owner el encargado de tomar la decisión. No obstante, puede ser aconsejado e influenciado por otras partes del equipo, ya sea el Scrum Master, el equipo de desarrollo o incluso el cliente. La cancelación de un Sprint debe considerarse como última medida de actuación, ya sea por obsolescencia del producto o por falta de viabilidad del proyecto, puesto que provoca malestar, desanimo y falta de motivación al equipo de trabajo, así como gastos y consumo de recursos innecesarios.

Es frecuente que muchas empresas hagan seguimientos de los Sprints para dotar al proceso de más transparencia y margen de maniobra. La monitorización es una de las técnicas más usadas por las empresas y consiste en representar el progreso del Sprint en una gráfica. *Sprint Burndown* es el nombre que recibe el diagrama que más se plantea en esta técnica. En él, los días en los que transcurre el Sprint se muestran en el eje de abscisas, mientras que la cantidad de trabajo pendiente de realizarse se refleja en el eje de ordenadas. De este modo, el equipo de desarrollo tiene una visión general de la masa de trabajo por hacer y el tiempo restante para realizarlo, con la posibilidad de comunicar al Product Owner posibles adaptaciones que fuesen necesarias. Es recomendable que también se refleje una estimación del progreso ideal que sirva como referencia a la hora de comparar con el progreso real.

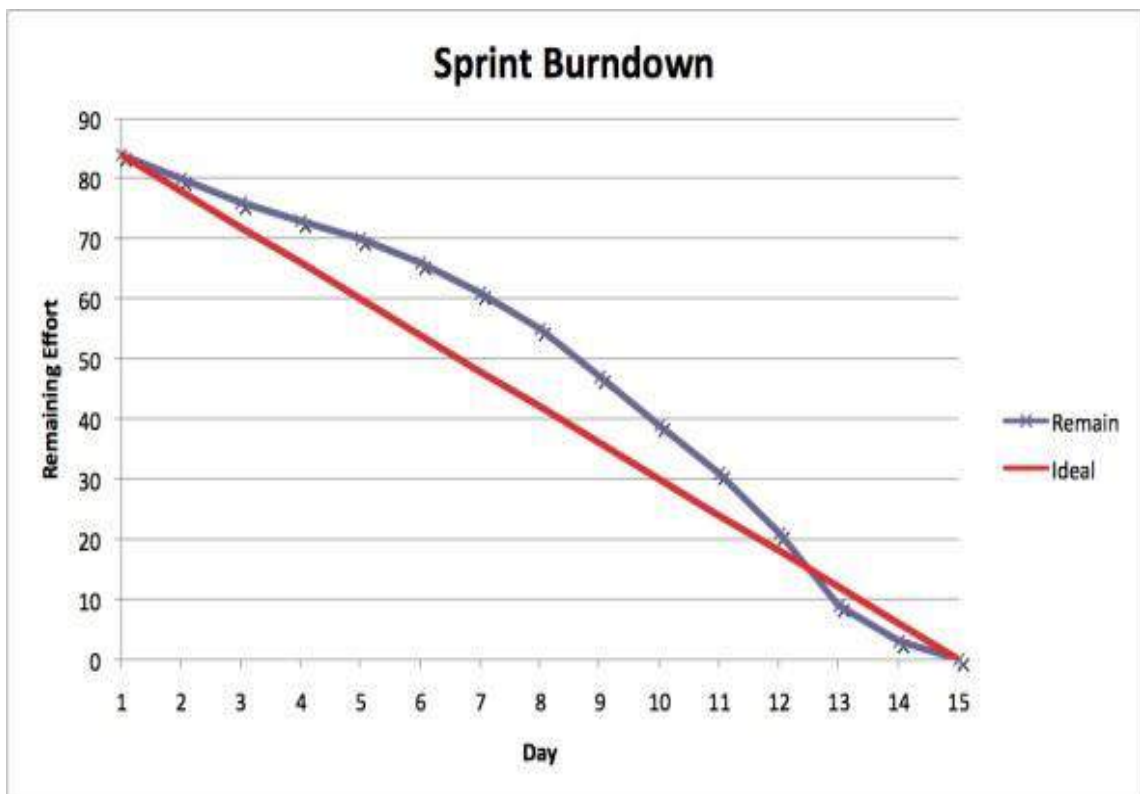


Figura 5-3. Diagrama Sprint Burndown (Fuente: [International Scrum Institute](#))

2. **Planificación de Sprint.** La primera parte de un Sprint es una reunión en la que se planifica el Sprint. Suele tener una duración de no más de 8 horas. En ella, el equipo Scrum establece los siguientes parámetros:
 - Se escogen los ítems del Product Backlog que se realizarán en el Sprint y se establecen las tareas que se llevarán a cabo para completar cada uno de ellos. Se obtiene como resultado el *Sprint Backlog*.
 - Se define el objetivo que se debe cumplir al finalizar el Sprint, obteniéndose el *Sprint Goal*. Este objetivo servirá de guía y otorgará al equipo de desarrollo un motivo o razón para la creación del incremento.

Antes de dar por concluida la reunión, el Scrum Master debe asegurarse de que el equipo de desarrollo haya comprendido el planning y tenga claro el propósito del mismo para que no haya confusiones durante el proceso y se produzcan contratiempos.

3. **Scrum diario.** Se trata de una breve reunión (no más de 15 minutos) que tiene lugar al comienzo de cada jornada de trabajo y en la que solo participa el equipo de desarrollo. No obstante, el Scrum Master se encarga de la supervisión de las reuniones, garantizando que se realice a diario y que se cumplan los criterios de duración y participación de todos los integrantes.

Durante la reunión se realiza una inspección del trabajo realizado, se sincronizan las tareas y se actualiza el Sprint Backlog. El equipo debate sobre lo trabajado el día anterior y lo que se trabajará en ese día. También se informa de los errores y contratiempos que se han ido produciendo y se intenta dar una solución.

Es importante que las reuniones diarias sean estandarizadas y se adecuen a un mismo modelo de hora y lugar para así evitar confusiones y retrasos.

4. **Revisión de Sprint.** Esta fase, con una duración aproximada de cuatro horas, se realiza al finalizar el Sprint. El Product Owner, con ayuda del equipo Scrum, muestra al cliente el incremento que se ha obtenido como resultado del Sprint y se valora el resultado final del mismo, considerando si se han cumplido o no los objetivos. Además, el cliente puede agregar nuevas tareas al Product Backlog, efectuando así un feedback del producto.

Entre las tareas que se llevan a cabo durante la revisión se pueden destacar:

- El Product Owner se encarga de reunir al equipo Scrum y a las partes interesadas.
- El equipo de desarrollo detalla como se ha realizado el Sprint, comentando lo que funcionó correctamente y explicando como se solventaron los problemas que ocurrieron.
- El Product Owner hace balance del estado del Product Backlog, comunicando que ítems se han finalizado y cuáles no. También establece las fechas de los siguientes incrementos que se realizarán posteriormente.
- Se abre debate entre el equipo Scrum y clientes para que estos últimos puedan hacer preguntas sobre el incremento y propongan adaptaciones si fuese necesario (feedback).
- Se realiza una inspección del estado del mercado, cronogramas y presupuestos para mejorar el lanzamiento del producto o tomar medidas para futuros incrementos.

5. **Retrospectiva del Sprint.** Tras la revisión de cada Sprint se procede a hacer un balance general (aproximadamente de 3 horas de duración) de la iteración. El equipo de desarrollo junto al Scrum Máster sacan conclusiones de los resultados y proponen cambios y mejoras para los próximos Sprint, procurando que la implementación de Scrum sea lo más exitosa posible.

El Scrum Máster también alentará y motivará al equipo de desarrollo para que se logre una mejora en las tareas y procesos de desarrollo, de modo que se garantice una mayor eficiencia y calidad de producción en futuros Sprints.

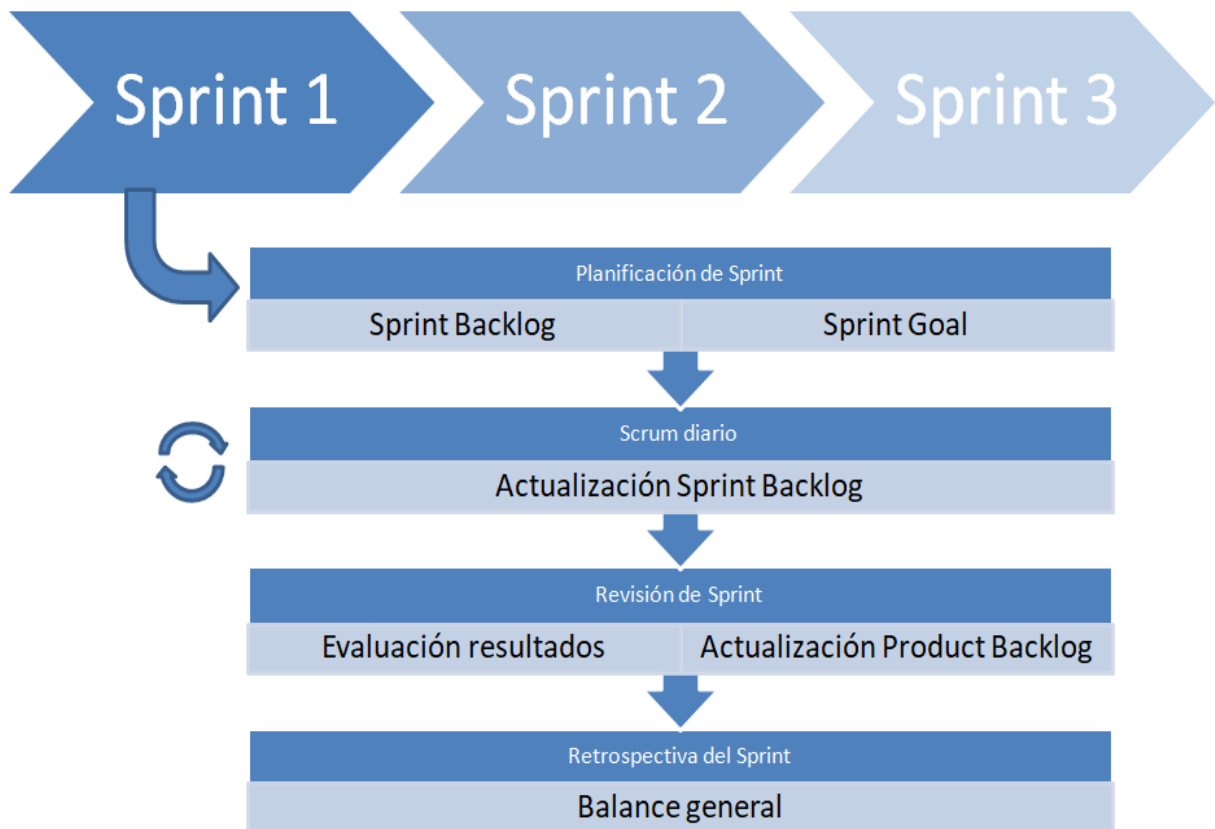


Figura 5-4. Esquema de los eventos Scrum (Fuente: Elaboración propia)

5.6. Beneficios del método Scrum

Como ya se ha podido ver, Scrum es la metodología ágil más influyente e importante del mercado actual y esto es gracias a los numerosos beneficios que este método aporta a todo aquel que logre su correcta implementación. Algunos de ellos son:

- Cortos períodos de entregas. Propicia ventajas como:
 - Resultados anticipados.
 - Flexibilidad y tiempo de respuesta ante alteraciones del cliente o mercado.
 - Gestión de las expectativas de los clientes. Participación directa del cliente en cada iteración.
 - Gestión sistemática de riesgos y retorno de inversión (ROI).
- Se fomenta un clima de colaboración y compromiso que se refleja en un aumento de productividad.
- El continuo seguimiento y revisiones garantizan una mayor calidad del producto elaborado.
- Alta transparencia y control de las etapas de desarrollo.
- El marco de trabajo promueve la auto-organización y autonomía.
- El conocimiento de la velocidad de producción y las funcionalidades de cada rol reducen los riesgos en cuanto al desarrollo del producto.
- Alineamiento entre el equipo y el cliente. Trabajo cooperativo.
- Alto índice de cumplimiento de expectativas del cliente.

- Soporte a la innovación.
- Equipo motivado y creativo.
- Maximización del retorno de inversión.

6 APLICACIÓN DE SCRUM

“El mundo cambia rápidamente. Los grandes ya no serán los que vencen a los pequeños; serán los ágiles quienes superen a los lentos.”.

- Rupert Murdoch -

En el siguiente capítulo se llevará a cabo la aplicación del método Scrum a un caso práctico particular. Con la ayuda de todos los conocimientos de la metodología que se han expuesto en el capítulo anterior, se pretende implantar un modelo de trabajo acorde con los principios y valores de Scrum, hacer un seguimiento a lo largo de su desarrollo y evaluar los resultados finales del proyecto. Con ello se pretende buscar una demostración de la influencia que puede llegar a alcanzar una correcta ejecución del método en cualquier marco de trabajo.

6.1. Introducción

Como ya anunció el Scrum Máster David Matthew: *“Scrum se parece al póquer; puedes aprender las reglas en 10 minutos, pero toma mucho más tiempo ser bueno en él”*, y es que se trata de un método simple pero no fácil de aplicar.

A continuación se pondrán en práctica todos los conceptos, herramientas y valores que se han visto en el anterior estudio realizado. Se aplicará el método a una empresa real, Dablered S.L., la cual se encarga de diseño de redes de telecomunicación (FTTH) de ámbito nacional. Se establecerán roles dentro del equipo de ingenieros y se hará una planificación del proyecto, definiendo el número de Sprints que se llevarán a cabo, así como su duración. Se hará uso de herramientas y artefactos característicos del método y se adaptarán al entorno de la empresa y a sus instalaciones. Con la finalización del periodo de aplicación del método, se realizará un balance general y se evaluarán los resultados obtenidos.

Aunque Scrum fue desarrollada como una metodología enfocada al desarrollo de software, en la actualidad, este método se ha expandido a otros sectores de la industria. Los proyectos que se ejecutan a día de hoy presentan un nivel de exigencia y complejidad mucho mayor que los realizados en décadas anteriores. Se desarrollan en contextos mucho más versátiles, y por ello, Scrum se ha ido ajustando según las necesidades empresariales con el fin de poder ser aplicado en cualquier ámbito industrial.

Con la aplicación llevada a cabo en este apartado se pretende demostrar la versatilidad que ofrece esta metodología, la cual permite ser adaptada a proyectos no necesariamente relacionados con el desarrollo de software, como puede ser el sector de la telecomunicación en este caso.

6.2. Entidad objeto de la aplicación. Dable Red S.L.

En este apartado se presenta la empresa que llevará a cabo la aplicación del método para uno de sus proyectos. Se detallarán sus funciones y tareas, así como sus instalaciones y plantilla de trabajadores. Todo de ello será adaptado para un uso de la metodología lo más óptimo posible.

6.2.1 Presentación



Figura 6-1. Logotipo de Dable Red S.L. (Fuente: Dable Red S.L.)

Dable Red S.L. es una empresa sevillana fundada en 2017 que trabaja dentro del sector tecnológico y de las telecomunicaciones. Su principal actividad es el diseño de redes de telecomunicaciones (FTTH), aunque pretende ampliar su campo laboral a otros sectores tecnológicos, como las energías renovables, más concretamente en instalaciones de plantas y cubiertas fotovoltaicas.

La empresa consta de un departamento de ingeniería compuesto de 8 profesionales, más otro departamento encargado de la obtención de permisos para la instalación de equipos, compuesto por 3 integrantes. La gerencia y dirección de la empresa corre a cargo de un director ejecutivo.

Actualmente, la empresa opera junto a importantes compañías de telefonía: Orange, Movistar, MásMovil y Jazztel, y ofrece servicio a municipios, localidades y ciudades de la comunidad de Andalucía. Para ello cuenta con un equipo de replanteo que se encarga de desplazarse a la población y recoger información de sus calles, viviendas y comercios, así como de conseguir los permisos de los habitantes, necesarios para instalar en sus propiedades.

6.2.2 Actividad a desarrollar

El método Scrum será aplicado al departamento de ingeniería y a sus funciones, obviando de este modo las tareas que llevan a cabo el resto de departamentos.

Las tareas y trabajos desarrollados por el equipo de ingeniería son los siguientes:

- Clusterización de la población. En primer lugar se procede a la división de la población en zonas (clusters), las cuales deben cumplir unos requisitos necesarios para poder llevar a cabo su diseño. Algunos de esos requisitos son:
 - Cantidad de UUIIS⁸ dentro del rango permitido (entre 107 y 213).
 - Disponibilidad de una CEA⁹ por cluster, a partir de la cual ofrecer servicio.
 - Porcentaje de permisos conseguidos mayor del 60% del total de propietarios.
 - Posibilidad de ofrecer servicio a toda propiedad incluida en el cluster, evitando obstáculos y impedimentos.
- Realización de prediseños. Consiste en la realización de un croquis representando en él la supuesta distribución de cables, conductos y CTOs¹⁰ que se proponen instalar para dar servicio al cluster. Se obvian detalles e información excesiva.
- Realización de diseños. Tras la aprobación del prediseño se procede a realizar el diseño definitivo. Se plasma el diseño sobre la herramienta software en la que se trabajará. Se representa la distribución de cables y conductos por las calles de la población y se colocan las CTOs en las propiedades escogidas para dar servicio a las UUIIs. Posteriormente se procede a la fusión de cables y a la asignación de direcciones de cada equipo.
- Documentación reglamentaria. En último lugar se realiza la documentación reglamentaria que se entregará al cliente junto al diseño final de cada cluster. En dicha documentación se recoge lo siguiente:
 - Plano LLD: Plano general del cluster generado por el software de diseño. En él se detalla toda la red de fibra óptica diseñada. Mediante etiquetas se indican los nombres, tipos y características de los distintos elementos de la red.
 - Carta de empalme: Libro Excel donde se recogen y detallan todas las fusiones realizadas en las CTOs, CEA y cables. Todo queda explicado al más mínimo detalle para que el instalador no tenga problemas a la hora de fusionar.
 - Lista de materiales. En esta lista se muestra todo el material y equipo que será necesario para la instalación. Esto incluye: CTOs, cables, conductos, cajas de empalmes, arquetas, splites, etc. De cada uno se detalla la identificación, cantidad, longitud, dimensión, dirección y estado.
 - Actas de replanteo. Fichas en las que recogen toda la información de la propiedad en la que se va instalar una CTO. Se detalla la dirección, el número de plantas, el tipo de propiedad (local o vivienda), tipo de instalación (fachada, interior, pedestal o arqueta) y se adjuntan fotos de la ubicación.
 - Área de influencia. En este entregable queda reflejado el número total de UUIIs a las que se le dan servicio y el área de influencia de cada CTO (direcciones de todas las propiedades afectadas).

El ejercicio se dará por finalizado cuando el diseño y la documentación hayan sido entregados a la empresa instaladora y den su aprobación para llevar a cabo la instalación.

En este caso, se hará un estudio de la aplicación del método durante el desarrollo de diseño de una población completa.

⁸ Se refiere a UUII como a la unidad que representa a un posible cliente al que ofrecer servicio, ya sea una vivienda o local.

⁹ Caja de telecomunicación desde la que se realiza el despliegue de cables, tanto de alimentación como de distribución.

¹⁰ Caja terminal óptica a la que se conecta las fibras ópticas y a partir de la cual se distribuyen los cables de altas que darán servicio.

6.2.3 Equipo de trabajo e instalaciones

Dable Red S.L. cuenta con un equipo de profesionales altamente cualificados del sector de las telecomunicaciones (FTTH), entre los que se encuentran ingenieros de telecomunicaciones, ingenieros industriales e ingenieros de automatización. El departamento de ingeniería cuenta con un total de 8 componentes, uno de ellos ejerce de jefe de ingeniería. Bajo la supervisión general se encuentra un director ejecutivo responsable de la gerencia y dirección de la empresa, así como de garantizar la viabilidad del proyecto y descubrir futuras oportunidades de negocio.

El equipo de trabajo opera en una oficina dotada de ordenadores personales para cada uno de los profesionales, además se cuenta con todos los softwares y programas operativos que cada compañía telefónica exige para el diseño de sus redes.



Figura 6-2. Instalaciones y equipo de Dable Red S.L. (Fuente: Elaboración propia)

También se dispone de un Nas Server (servidor) que permite la compartición de archivos en tiempo real, sin necesidad de transpasarlos de un equipo a otro. Esto es de gran utilidad ya que el hecho de que los trabajadores tengan siempre a mano todos los archivos realizados (por ellos mismos o por otro compañero) facilita las tareas y permite un trabajo cooperativo más eficiente, uno de los puntos fuertes que ofrece Scrum.

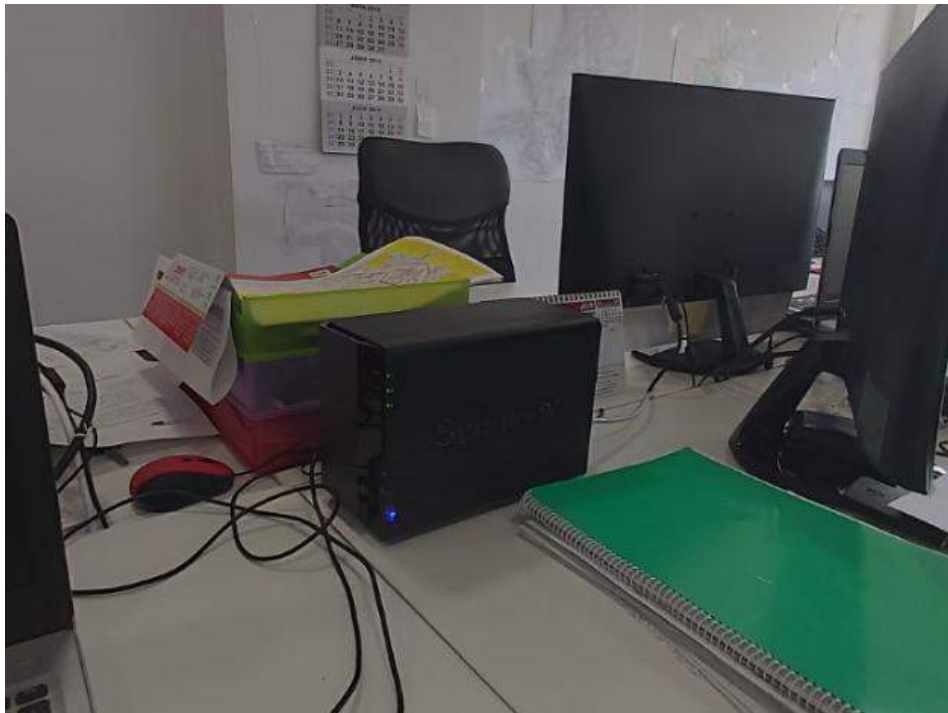


Figura 6-3. Nas Server (Fuente: Elaboración propia)

Cabe destacar la presencia de una pizarra o tablero que será usado como Scrum board y que permitirá representar el Sprint Backlog correspondiente de cada iteración. Para el seguimiento de las tareas e ítems del Product Backlog se usarán post-it de diferentes colores para poder diferenciar los tres niveles de prioridad, los cuales se irán colocando en la pizarra para tener una mejor visualización del estado de las tareas del Sprint.



Figura 6-4. Scrum board de la oficina (Fuente: Elaboración propia)

6.3. Planificación del proyecto

Una vez descrita la entidad objeto de la aplicación, se procede a la definición y planificación del proyecto que será desarrollado, el cual será materia de estudio y se evaluarán sus resultados finales con el fin de ver la influencia e impacto del método Scrum como herramienta de trabajo.

6.3.1 Proyecto

El proyecto que se llevará a cabo es el diseño de una red de fibra óptica que de servicio en su totalidad a la localidad de Campohermoso, perteneciente al municipio de Níjar, Almería. Las tareas a realizar por el equipo de ingeniería son las expuestas en el apartado anterior las cuales se refieren a la clusterización de la localidad y realización de diseño y documentación pertinente. El ejercicio concluye con la entrega y validación del diseño, planos y documentación.

Para el proyecto que nos ocupa, MásMovil será la compañía cliente que contrate los servicios de Dable Red S.L. y pone total confianza en su equipo de trabajo para la primera fase de despliegue y servicio de fibra óptica en la provincia de Almería.

La duración del proyecto impuesta por el cliente es de 2 meses, con fecha de inicio el día 1 de abril de 2019 y fecha de finalización el día 31 de mayo de 2019.

6.3.2 Eventos

Las entregas parciales en cortos períodos de tiempo es una de las ventajas que ofrecía Scrum y que otorgaba al proyecto flexibilidad y margen de reacción. El departamento de ingeniería y el director ejecutivo acuerdan con el cliente un total de 4 entregas parciales de incrementos a lo largo de los 2 meses de desarrollo, con el fin de dotar al cliente de una mayor información del estado del producto y le permita llevar a cabo las adaptaciones que considerase oportunas.

Para una correcta planificación que simplifique y agilice las entregas parciales, el equipo de ingeniería decide realizar 4 Sprints, cada uno de ellos con un *Timebox*, o duración programada, de 15 días. Dentro de cada Sprint se realizarán cada uno de los eventos característicos de Scrum: planificación de Sprint, Scrum diario, revisión de Sprint y retrospectiva del Sprint. Los Sprints planificados son los siguientes:

- Sprint 1 (del 1 al 15 de abril). Clusterización
- Sprint 2 (del 16 al 30 de abril). Presideño
- Sprint 3 (del 1 al 15 de mayo). Diseño
- Sprint 4 (del 16 al 31 de mayo). Documentación

A continuación se muestran los cronogramas correspondientes a cada mes de desarrollo. En ellos se representan todos los eventos en una línea temporal, donde se indican sus fechas de inicio y fin.

- **Product Owner.** El rol de jefe de proyecto recae sobre el *director ejecutivo*. Es el dueño de la empresa y la figura más importante en cuanto a razones de negocio y viabilidad del proyecto. Es el encargado de mantener contacto directo con los clientes y de negociar posibles nuevos proyectos y oportunidades de negocio.
- **Scrum Máster.** El papel de líder del equipo Scrum es asignado al *jefe del departamento de ingeniería*. Sus años de experiencia en la empresa le convierten en el mejor candidato ya que ha adquirido mayores conocimientos del sector que el resto de sus compañeros. También es el que mejor conoce a la plantilla de trabajo y el que más trato ha tenido con cada uno de los componentes. Es el encargado de la supervisión y la gestión de todo lo relacionado con el método Scrum, así como garantizar un buen clima de trabajo y motivar al resto del equipo.
- **Scrum team.** El equipo Scrum está formado por todos los *ingenieros* que trabajan en el departamento, incluido el jefe de ingeniería, el cual también formará parte de este equipo de desarrollo. Es una plantilla formada por un total de 8 profesionales muy cualificados y con amplios conocimientos del sector. Es fundamental el buen ambiente de compañerismo que existe entre ellos, donde la ayuda y cooperación siempre está a la orden del día.
- **Cliente.** Por último, *Másmovil* es el cliente al que Dable Red ofrece su servicio en este proyecto. Ya se había trabajado con anterioridad con la compañía telefónica y los buenos resultados obtenidos hacen que se siga depositando la confianza necesaria en Dable Red para acometer sus futuros proyectos.

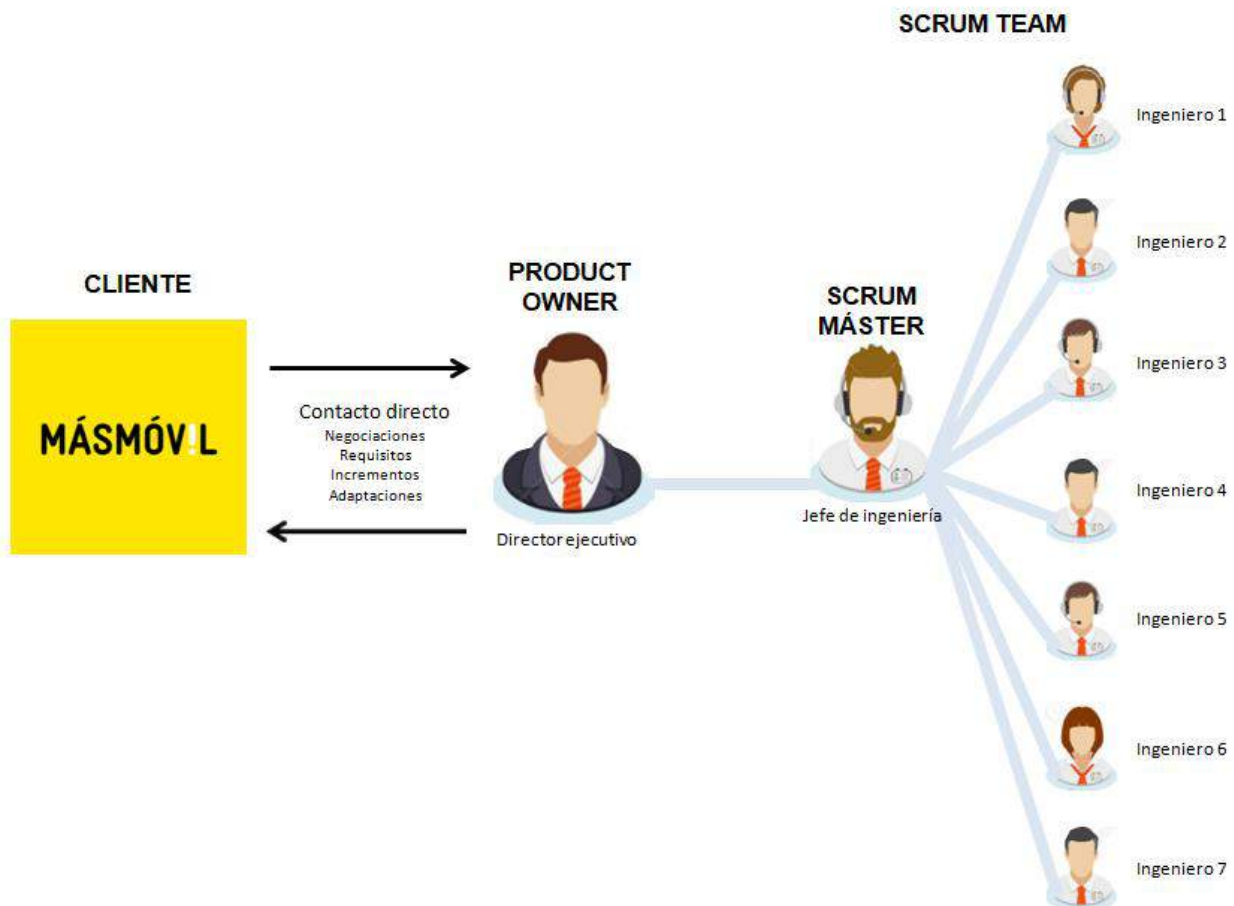


Figura 6-7. Esquema de roles del proyecto (Fuente: Elaboración propia)

6.3.4 Artefactos

En cuanto a los artefactos que se usan durante el proyecto, el equipo cuenta con lo siguiente:

- **Product Backlog.** Desarrollado por el Product Owner y el Scrum Máster. En él se recogen todas las tareas que se realizarán a lo largo del proyecto, así como su descripción y algunos atributos. A continuación se muestra el Product Backlog utilizado:

Tabla 6-1. Product Backlog del proyecto (Fuente: Elaboración propia)

Prioridad	ID	Descripción	Estimación (1-10)	Valor (1-10)
Alta	1A	Direcciones y continuidad de cables en planos.	4	5
Media	1B	Representación y recuento de UUIIs en planos.	5	5
Baja	1C	Clusterización.	7	8
Alta	2A	Agrupación de UUIIs y asignación de CTOs en planos.	9	9
Media	2B	Despliegue de cables y conductos en planos.	8	9
Baja	2C	Plasmación del prediseño en software.	6	10
Alta	3A	Representación de elementos secundarios (arquetas, pasos aéreos, conductos, etc.) en software.	4	6
Media	3B	Delimitación de áreas de influencias.	3	7
Media	3C	Fusiones de cables.	8	8
Media	3D	Etiquetado de elementos.	2	5
Baja	3E	Generación de plano de diseño completo en software.	5	10
Media	4A	Cartas de empalmes.	8	9
Media	4B	Lista de materiales.	4	7
Media	4C	Actas de replanteo.	7	8
Media	4D	Áreas de influencias.	6	8

- Sprint Backlog.** Se desarrolla un Sprint Backlog con los ítems correspondientes para cada iteración, los cuales se representan en la pizarra ubicada en la oficina. Esta técnica es característica de la metodología Kanban, la cual hace uso de tableros y pizarras para la supervisión y seguimiento del proceso de desarrollo. Por ello, es posible considerar el método como *Scrumban*. Los estados por los que deben pasar cada ítems son: “pendiente”, “en proceso” y “realizado”, a excepción del Sprint 2 que debido a su complejidad posee tres estados más: “en revisión”, “rechazado” y “aprobado”. Las tareas se recogen en post-it de diferentes colores para señalar su prioridad y en ellos se indican los IDs y las descripciones, como a continuación se muestra:



Figura 6-8. Tipos de post-it del proyecto (Fuente: Elaboración propia)

Los diferentes clusters que se deben diseñar se organizan por filas y en cada una de ellas se disponen de las mismas tareas a realizar (ítems en post-it). Los ítems prioritarios deben estar en el estado “realizado” para que los ítems con un nivel inferior de prioridad puedan empezar a procesarse. No existe limitación de ítems (WIP) en un mismo estado, el número de tareas está restringido por las prioridades. Se considera que el Sprint está finalizado cuando todos los ítems de todos los clusters se encuentren en el estado “realizado”.

De este modo, cada Sprint dispone de un Sprint Backlog “base” que sigue el siguiente esquema¹¹:

¹¹ Este esquema no procede en el Sprint 1, el cual tiene como objetivo la determinación de los clusters (clusterización)










<i>SPRINT X</i>	<i>Pendiente</i>	<i>En proceso</i>	<i>Realizado</i>
<i>Cluster 01</i>			
<i>Cluster 02</i>			
<i>Cluster 03</i>			
<i>Cluster 04</i>			
<i>Cluster 05</i>			

Figura 6-9. Sprint Backlog base del proyecto (Fuente: Elaboración propia)

- **Incrementos.** Hay un total de 4 incrementos a entregar a lo largo del proyecto, uno a la finalización de cada Sprint. Cada incremento es funcionalmente superior al anterior y necesitan ser evaluados y validados por el cliente para que se puede empezar a producir el siguiente. Los incrementos y sus fechas de entrega son los siguientes:
 1. Clusterización (15 de abril). Se debe entregar al cliente un plano de la localidad dividida en zonas que cumplen con los requisitos necesarios para ser diseñadas. También se da a conocer el número de UIIs total y el correspondiente a cada cluster.
 2. Prediseño (30 de abril). Debido a la complejidad e incertidumbre que existe en esta iteración, se entregan propuestas de prediseños de cada cluster justo después de haberse realizado, con el fin de tener tiempo de reacción y poder realizar correcciones o rediseños en caso de que el cliente no de su aprobación. El incremento consiste en la aprobación de todos los prediseños.
 3. Diseño (15 de mayo). En base a los prediseños entregados en el anterior Sprint, se realizan los diseños completos y se entregan al cliente al final de la iteración
 4. Documentación (31 de mayo). Por último se hace entrega de toda la documentación complementaria al diseño.

6.4. Desarrollo del proyecto

Una vez presentado el proyecto y descrito su planificación se procede a la aplicación del método Scrum para la gestión de su desarrollo. A continuación se realiza el seguimiento y evaluación de cada uno de los Sprints que se llevan a cabo.

6.4.1 Sprint 1

Fecha: 01/04/2019 – 15/04/2019

Objetivo: Clusterización

Product Backlog:

Tabla 6-2. Product Backlog del Sprint 1 (Fuente: Elaboración propia)

Prioridad	ID	Descripción	Estimación (1-10)	Valor (1-10)
Alta	1A	Direcciones y continuidad de cables en planos	4	5
Media	1B	Representación y recuento de UIIs en planos	5	5
Baja	1C	Clusterización	7	8

Sprint Backlog:

A diferencia del resto de Sprint Backlogs, en este caso no se trabaja con clusters puesto que es el fin de esta iteración. En su lugar, se divide la población en 5 zonas distintas (norte, este, oeste, sur y central) para así poder repartir mejor la carga de trabajo. En cuanto a los estados de desarrollo, este Sprint Backlog trabaja con los 3 principales que se definieron en la planificación del proyecto: “pendiente”, “en proceso” y “realizado”. Los ítems utilizados son 3 para cada zona, y cada uno de ellos con un nivel de prioridad diferente.



Figura 6-10. Sprint Backlog del Sprint 1 (Fuente: Elaboración propia)

Monitorización (Sprint Burndown):

El primer Sprint cuenta con el volumen de trabajo más bajo de todos, un total de 15 tareas, 3 tareas a realizar por cada una de las 5 zonas. Debido al bajo número de tareas a realizar y que éstas no suponían mucha complejidad ni un elevado tiempo de realización, el equipo de trabajo consiguió un ritmo de trabajo mejor al ideal, consiguiendo que el volumen de trabajo real se mantuviese siempre por debajo del volumen ideal a lo largo del Sprint.

De este modo, el equipo de desarrollo finalizó todo el trabajo en el undécimo día. Los cuatro días restantes supusieron un amplio período de tiempo ocioso, el cual podría haber sido aprovechado para otras tareas de futuros Sprints. Este será un punto a debatir en la retrospectiva de esta iteración para evitar que este suceso se vuelva a repetir en futuros ejercicios.

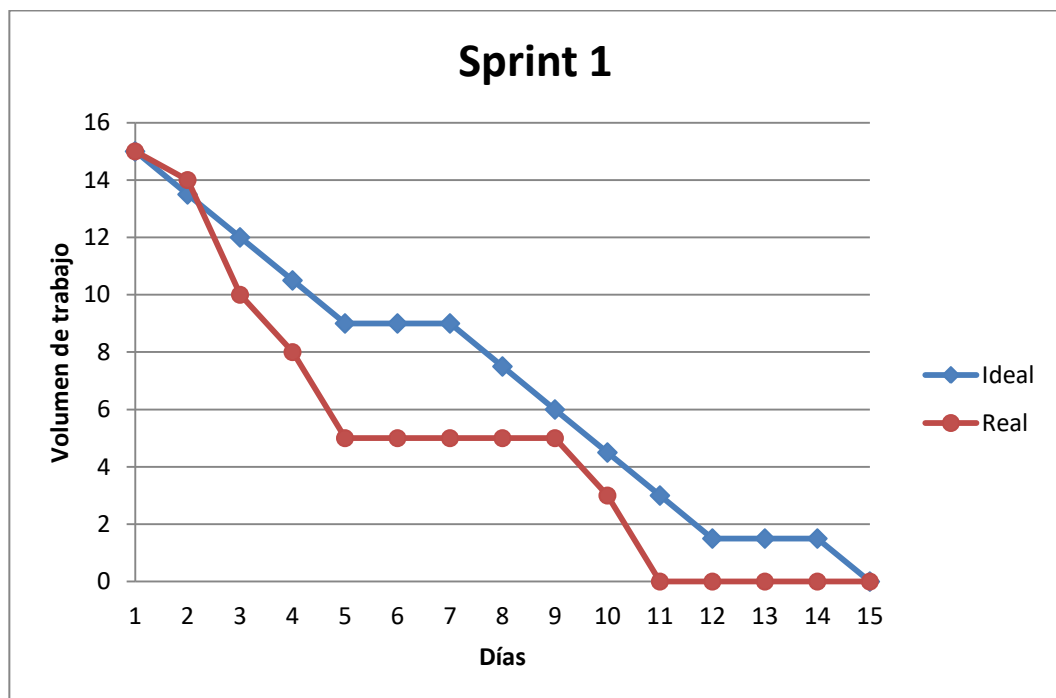


Figura 6-11. Diagrama Sprint Burndown del Sprint 1 (Fuente: Elaboración propia)

Entregable:

Se entrega al cliente el plano de la población dividida en los distintos clusters con los que se van a trabajar. Como se puede ver en la imagen, se definieron un total de 19 clusters en toda la población.

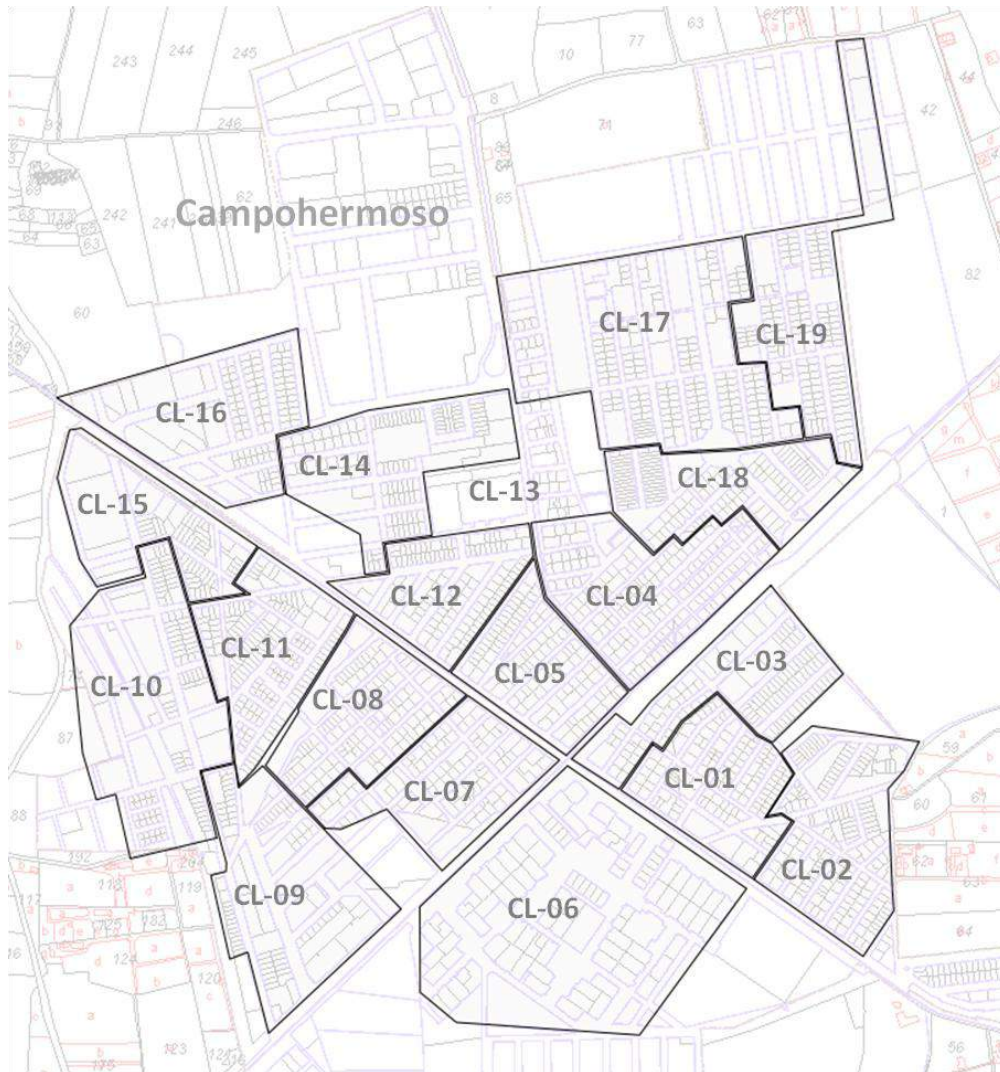


Figura 6-12. Clusterización de Campohermoso (Fuente: Elaboración propia)

Retrospectiva del Sprint:

Durante el último día, el Scrum Máster y su equipo hacen retrospectiva del Sprint 1 y llegan a la conclusión de que ha sido una iteración con un balance favorable, con buena eficiencia en resultados y tiempo. No obstante, el excesivo tiempo ocioso de los últimos días debido a la pronta finalización de las tareas promueve a disminuir el número de días en futuros Sprints de clusterización, para así invertir ese tiempo en otras iteraciones que lo necesiten más.

6.4.2 Sprint 2

Fecha: 16/04/2019 – 30/04/2019

Objetivo: Prediseño

Product Backlog:

Tabla 6-3. Product Backlog del Sprint 2 (Fuente: Elaboración propia)

Prioridad	ID	Descripción	Estimación (1-10)	Valor (1-10)
Alta	2A	Agrupación de UUOOs y asignación de CTOs en planos	9	9
Media	2B	Despliegue de cables y conductos en planos	8	9
Baja	2C	Plasmación del prediseño en software	6	10

Sprint Backlog:

Este Sprint es el más complejo de todos y el que requiere más cuidado y atención. La aprobación de los prediseños es fundamental para poder realizar las siguientes iteraciones ya que son la base del diseño y la documentación. Por ello, este Sprint Backlog cuenta con tres estados más: “en revisión”, “rechazado” y “aprobado”. Con la entrega parcial de cada prediseño tras su realización y la posterior evaluación por parte del cliente se consigue tener margen de corrección y con ello garantizar que todos los prediseños queden aprobados al finalizar el Sprint.

Cada fila del Sprint Backlog representa cada uno de los 19 clusters a prediseñar, con sus respectivas 3 tareas a realizar.

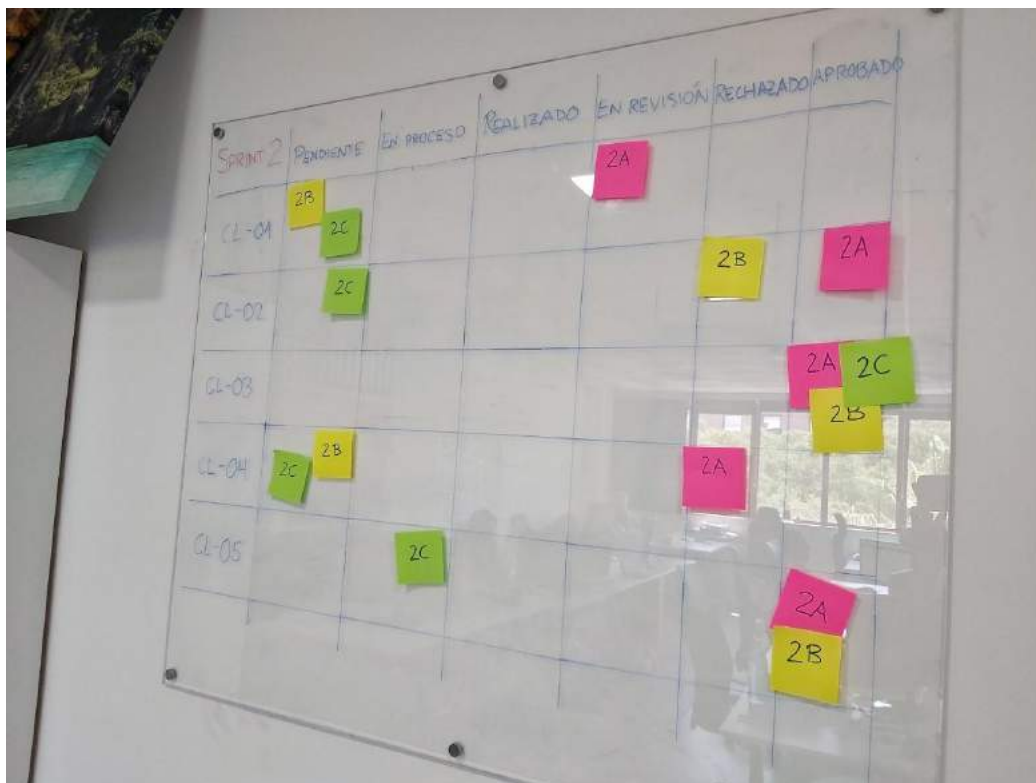


Figura 6-13. Sprint Backlog del Sprint 2 (Fuente: Elaboración propia)

Monitorización (Sprint Burndown):

La iteración cuenta con un volumen de trabajo total de 57 tareas, 3 tareas a realizar en cada uno de los 19 clusters. Durante los primeros días de desarrollo, el equipo consiguió un ritmo de trabajo bastante bueno, manteniendo el volumen de trabajo por debajo del ideal, pero el día 19 comenzaon a llegar rechazos de algunos entregables que provocaron contratiempos en el curso del desarrollo y alteró de manera negativa el ritmo de trabajo. El rechazo de las tareas más prioritarias no permitía al equipo de desarrollo avanzar con el resto de tareas, las cuales se iban acumulando. Fue en el día 22 cuando más rechazos se produjeron debido a las dificultades que presentaban los clusters que se diseñaban en ese momento, tanto es así, que se aumentó el volumen de trabajo con respecto al día anterior debido a que se tuvo que rediseñar. Para intentar revertir la situación, algunos de los clusters que presentaban más problemas se trabajaron de manera colectiva (de 2 a 3 trabajadores) y los contactos directos con el cliente aumentaron para conseguir una supervisión más exhaustiva y con más frecuencia. Gracias al seguimiento y trabajo colaborativo que caracteriza a Scrum, se consiguió aportar al Sprint un aumento de productividad que redirigió el curso del desarrollo y disminuyó el volumen de trabajo, acercándolo al nivel que presentaba el ideal. De este modo, con alguna que otra dificultad, se consigue acabar el Sprint con todas las tareas realizadas y aprobadas por el cliente.

Este Sprint presentó una carga de trabajo demasiado alta para el tiempo de realización que se estimó en la planificación del mismo. Ésto dió lugar a situaciones de agobio y fatiga. Es necesario acordar una mejora en la retrospectiva del Sprint para revertir esta situación en futuros proyectos.

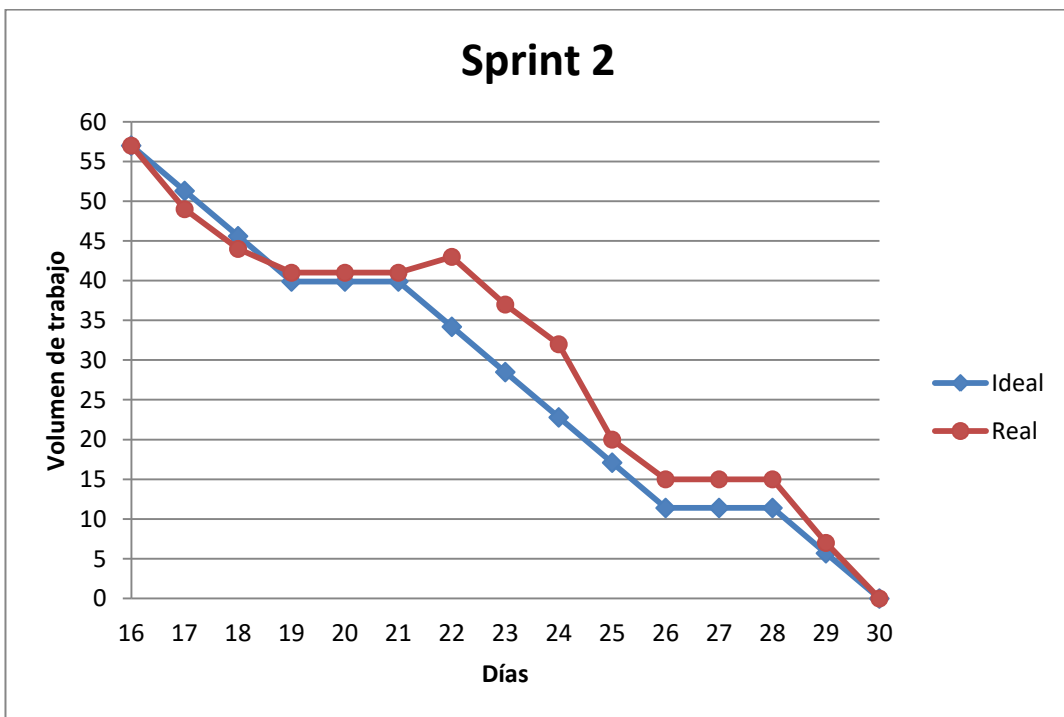


Figura 6-14. Diagrama Sprint Burndown del Sprint 2 (Fuente: Elaboración propia)

6.4.3 Sprint 3

Fecha: 01/05/2019 – 15/05/2019

Objetivo: Diseño

Product Backlog:

Tabla 6-4. Product Backlog del Sprint 3 (Fuente: Elaboración propia)

Prioridad	ID	Descripción	Estimación (1-10)	Valor (1-10)
Alta	3A	Representación de elementos secundarios (arquetas, pasos aéreos, conductos, etc) en software	4	6
Media	3B	Delimitación de áreas de influencias	3	7
Media	3C	Fusiones de cables	8	8
Media	3D	Etiquetado de elementos	2	5
Baja	3E	Generación de plano de diseño completo en software	5	10

Sprint Backlog:

Este Sprint Backlog cuenta con una gran número de ítems a realizar por cluster, 5 en total, pero 3 de ellos poseen el mismo nivel de prioridad (media), lo cual concede al equipo un margen de libertad a la hora de decidir el orden de realización de dichas tareas. Los estados de desarrollo se rigen al esquema estándar: “pendiente”, “en proceso” y “realizado”.



Figura 6-16. Sprint Backlog del Sprint 3 (Fuente: Elaboración propia)

Monitorización (Sprint Burndown):

El tercer Sprint posee la carga de trabajo más elevada de todas, con un total de 95 tareas, aunque la mayoría de ellas presentan un bajo nivel de estimación, es decir, una baja implicación de esfuerzo y complejidad. Como se puede observar en el Product Backlog del Sprint, a excepción de la tarea 3C (fusiones de cables), ninguna de ellas supera el valor 5 de estimación.

El volumen de trabajo real se mantuvo muy similar al ideal durante toda la iteración, no obstante, la carga de trabajo fue más elevada de lo prevista durante la realización de las fusiones de cables, tarea que requería más tiempo y dedicación. Con la finalización de éstas, el ritmo de trabajo aumentó y se logró acabar el ejercicio con soltura y desahogo, manteniendo una carga de trabajo menor a la ideal.

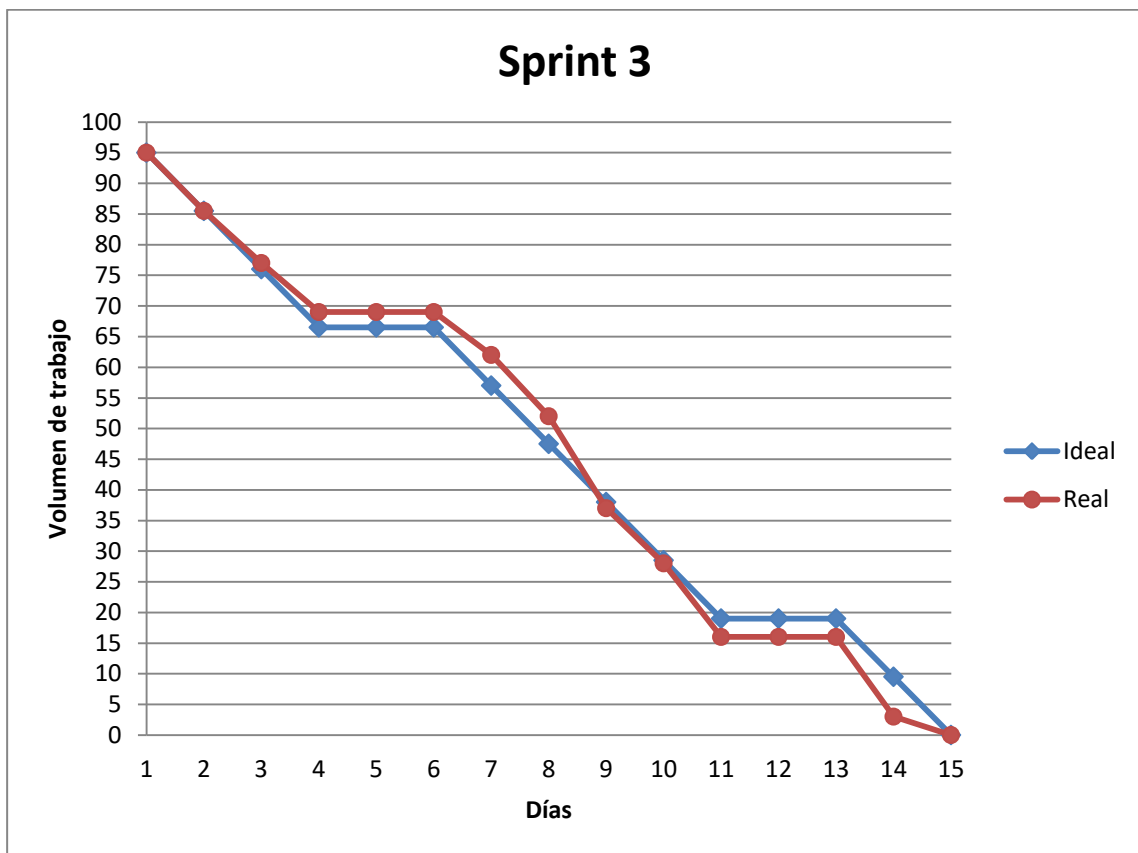


Figura 6-17. Diagrama Sprint Burndown del Sprint 3 (Fuente: Elaboración propia)

Entregable:

Se entrega al cliente el plano en software del diseño completo de Campohermoso. Como se muestra en la imagen del diseño del cluster 16, se complementa el prediseño validado en el Sprint anterior con la representación del resto de elementos de instalación (arquetas, conductos, pasos aéreos) y la colocación de etiquetas identificatorias.

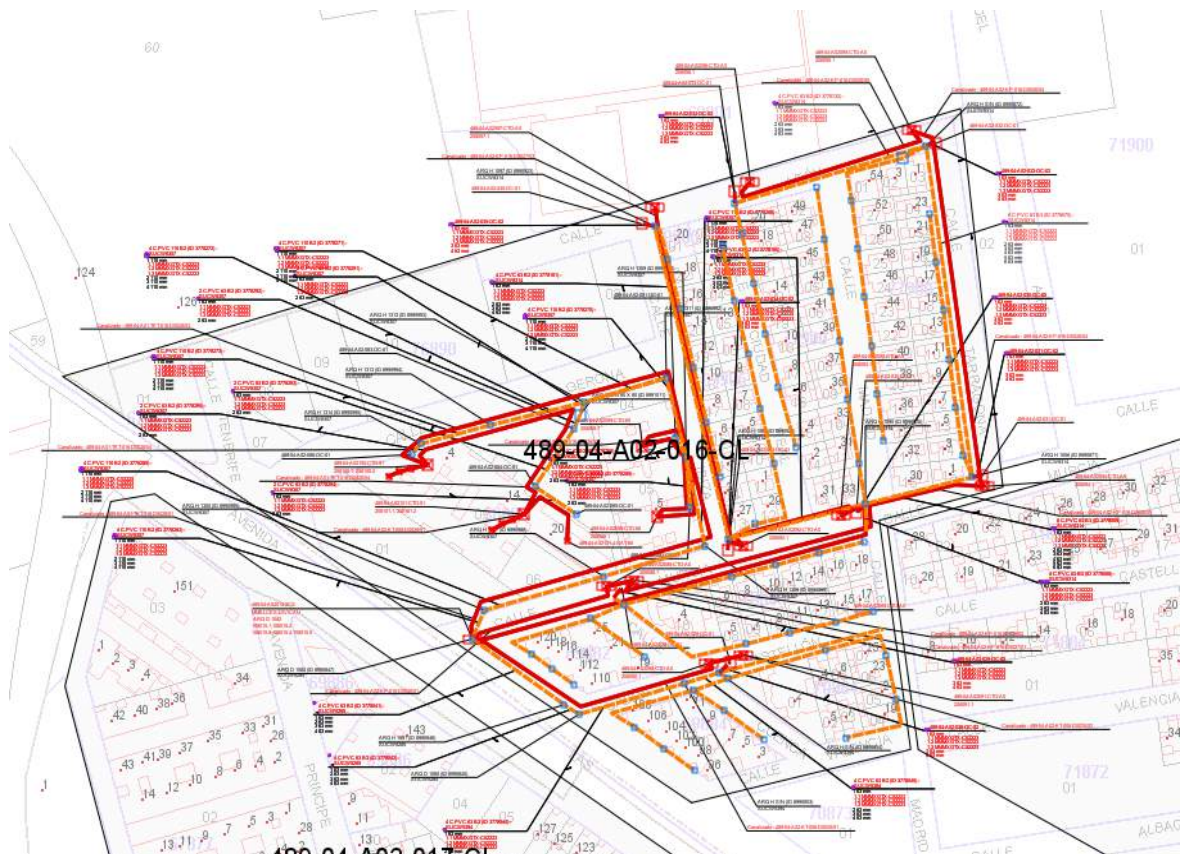


Figura 6-18. Diseño del cluster 16 (Fuente: Elaboración propia)

Retrospectiva del Sprint:

Una eficiente ejecución del Sprint 2 es de suma importancia para el correcto desarrollo de esta iteración. Garantizar unos prediseños aptos y funcionales evita contratiempos y adversidades durante la realización de los diseños finales, como ocurre en este caso. De este modo, el equipo Scrum evalúa positivamente el ejercicio realizado, con un balance favorable y cumpliendo con las expectativas esperadas, unos resultados óptimos y un ritmo de trabajo satisfactorio. No se pretenden cambios para futuros Sprints de diseños.

6.4.4 Sprint 4

Fecha: 16/05/2019 – 31/05/2019

Objetivo: Documentación

Product Backlog:

Tabla 6-5. Product Backlog del Sprint 4 (Fuente: Elaboración propia)

Prioridad	ID	Descripción	Estimación (1-10)	Valor (1-10)
Media	4A	Cartas de empalmes	8	9
Media	4B	Lista de materiales	4	7
Media	4C	Actas de replanteo	7	8
Media	4D	Áreas de influencias	6	8

Sprint Backlog:

La principal característica de este Sprint Backlog radica en la falta de prioridades entre sus tareas. Al no haber distinción en el orden de realización, el equipo de desarrollo tiene total libertad para su auto-organización y gestión, quedando exentos de cualquier dependencia. En cuanto a los estados de desarrollo, se respeta el esquema base definido en la planificación.



Figura 6-19. Sprint Backlog (Fuente: Elaboración propia)

Monitorización (Sprint Burndown):

El último Sprint del proyecto cuenta con una carga de trabajo total de 75 tareas. Debido a que todas las tareas tienen la misma prioridad en esta iteración, el desarrollo de la misma se efectuó de manera muy liviana, aportando total libertad a los trabajadores para elegir el orden de realización de cada uno de los ítems.

El nivel de volumen de trabajo se mantuvo por debajo del ideal en todo momento, aumentando la diferencia entre ambas curvas a lo largo del periodo de tiempo.

Al igual que ocurrió en el Sprint 1, se consiguió acabar con toda la carga de trabajo antes de lo previsto, volviendo a aparecer excesivo tiempo ocioso en los últimos días de la iteración. Punto a mejorar para los próximos ejercicios..

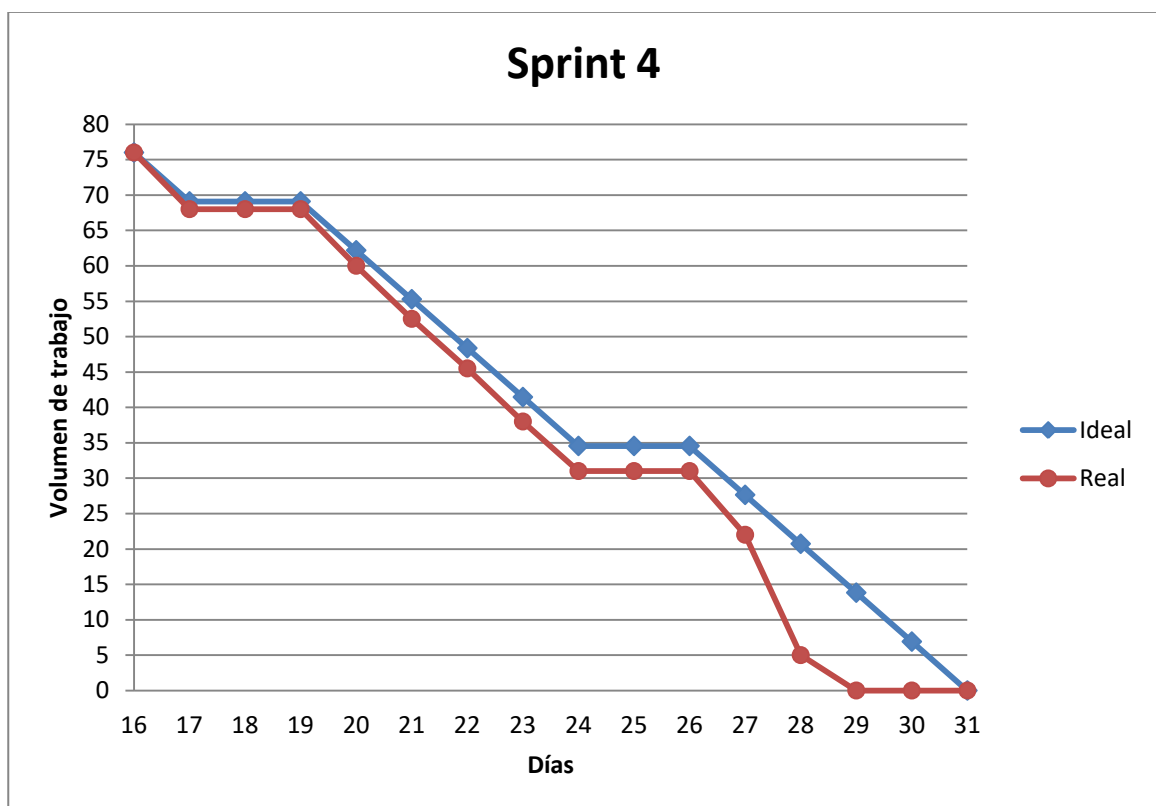


Figura 6-20. Diagrama Sprint Burndown del Sprint 4 (Fuente: Elaboración propia)

Entregable:

Último entregable correspondiente a la documentación reglamentaria. En la imagen, carta de empalme y acta de replanteo de una CTO ya aprobada para su instalación.

The image shows a screenshot of a fiber optic connection report from MASMÓVIL and a corresponding building replanning act form.

Thread connections report:

- Created by: Luisa Noguera
- Created: 24.05.2019 13:59
- Telecom premises: 489-04-A02092-CTO-A0

From	Cable / Device	Fibers / Pairs	Cable / Device	Fibers / Pairs	Outgoing
CTO EXT HUAVEI 8 puertos	CTO EXT HUAVEI 8 puertos-489-04-A02-20092-1-SP	0 (IN)-8	MMPROK-000016 489-04-A02-KP-016-12-12	12	(12) 489-04-A02015-2CS 489-04-A02-100015.5-SP 4
CTO EXT HUAVEI 8 puertos-IN		1-1	D0027/02 53.4m	12	
CTO EXT HUAVEI 8 puertos-IN		2-2	MMPROK-000016 489-04-A02-KP-016-11-11	11	
CTO EXT HUAVEI 8 puertos-IN		3-4	D0027/02 53.4m	11	
(8) 489-04-A02015-2CS 489-04-A02-100015.4-SP 5	MMPROK-000016 489-04-A02-KP-016-1-0 (PASO -)		MMPROK-000016 489-04-A02-KP-016-1-0	1-0	(7) 489-04-A02097-CTO-A0 IN 2 (8) 489-04-A02097-CTO-A0 IN 1
		1	D0027/03 127.4m	1	
		2		2	
		3		3	
		4		4	
		5		5	

ACTA DE REPLANTEO DE EDIFICIOS - RED FTTH MM!

MASMÓVIL

Código INE	04A02	Población	CAMPOHERMOSO		Fecha	24/05/2019	
Tipo de Vía	CALLE	Nombre Vía	MURCIA		Nº	27	
GESCAL 17	04000770008800027-----						
Tipo de Bloque	Casa	Tipo de Finca	Residencial		DUMMY	NO	
Contacto Acceso Comunidad:						Teléfono:	
Nº Total Uis + Locales	1	Nº Total viviendas	1	Nº total Locales	0	¿Necesario permiso finca anterior?	NO
¿Hay salida lateral de TESA?	NO	¿Hay canalizaciones de Telecomunicaciones?	NO	¿Está libre la canalización?	N/A		
¿Hay otro operador FTTH? ¿Cuál?	NO	¿Tipo de instalación ?	Pedestal	¿Necesario Upgrade de TESA?	NO		
¿Instalación compatible?	NO	¿CTO modular?	NO	¿Necesario Upgrade de TESA?	NO		
Ubicación posible CTO MM!	Pedestal	CODIGO CTO otro operador:					
Otras observaciones (falso techo, grapeado, garaje común a varias comunidades)							
¿En caso de acometidas a demanda, tiene capacidad para el 100%?	SI	¿Se debe adecuar el interior para ubicar equipos?	NO				
¿CDP en registro en planta?	NO	Geotipo	TIPO 1 - UI DESDE CTO DE EXTERIOR				
Ubicación posible CTO en OC MM							
Técnico Replanteo	DAVID DOMINGUEZ		Teléfono	640 146 743	Empresa	DABLERED	
			E-mail				

Figura 6-21. Carta de empalme y acta de replanteo (Fuente: Elaboración propia)

Retrospectiva del Sprint:

Nuevamente se obtiene un balance favorable para esta última iteración y, por tanto, el equipo Scrum considera acertada la planificación tomada para este Sprint. Los resultados de los entregables fueron satisfactorios y se mantuvo a lo largo del proceso un ritmo de desarrollo mejor al ideal, dando lugar a escasez de trabajo en los últimos días. Este exceso de productividad no esperado va ligado a la falta de prioridades entre las tareas del Product Backlog, lo cual facilitó al equipo de desarrollo su organización y gestión del proceso, aportando total libertad a la hora de elegir el ítem a realizar.

Para un mejor aprovechamiento del tiempo de trabajo, se acuerda una reducción de días de producción para futuros Sprints de documentación. Con ésto se pretende disponer de más días para otros Sprints que requieran más dedicación y tiempo, como sería el caso del Sprint 2.

Con la entrega y aprobación de este último entregable se daría por concluido el proyecto “diseño de red FTTH, Campohermoso de Níjar”. El cliente MásMóvil tiene a su disposición todos los incrementos necesarios para comenzar con la siguiente fase del ejercicio y con la garantía de que todos ellos cumplen con los requisitos acordados.

7 CONCLUSIONES

“La vida es el arte de sacar conclusiones suficientes a partir de datos insuficientes.”.

- Samuel Butler -

El proyecto “diseño de red FTTH, Campohermoso de Níjar” concluyó el 31 de mayo de 2019 con la entrega y aprobación de todos los incrementos acordados con el cliente MásMóvil. El ejercicio se efectuó de manera exitosa y muy eficiente, cumpliendo con todos los plazos de entregas y requisitos que se negociaron en la planificación del proyecto.

Mediante las entregas parciales efectuadas a la conclusión de cada Sprint, MásMóvil tiene a su disposición todos los incrementos que conforman en su totalidad el diseño final de la red de fibra óptica con el que podrá acometer la segunda fase de despliegue y servicio de red FTTH en Campohermoso, consistente en la instalación en campo de la red diseñada y sus equipos correspondientes. Por ello, era crucial que todos los entregables quedaran exentos de errores e incongruencias que perjudicasen al proceso de instalación, produciéndose contratiempos y fallos irreversibles que supusiesen gastos extras a la compañía de telefonía. Scrum ha resultado ser una herramienta muy útil para la gestión del proyecto y ha conseguido dotar a Dable Red S.L. de un sistema de trabajo potencialmente funcional que permitiese garantizar unos resultados sólidos y efectivos, capaces de cumplir con los objetivos exigidos.

La metodología Scrum ha conseguido implantar en la empresa un entorno de trabajo más cooperativo y flexible, lo cual ha fomentado un clima idóneo de compañerismo y colaboración. A esto hay que añadirle otra de las principales características de un equipo ágil, el continuo seguimiento y revisión del producto en todo momento, el cual siempre estuvo bajo estudio y evaluación tanto por el equipo de desarrollo, como por el cliente. La comunicación mantenida con este último también resultó ser de vital importancia, sobretodo en aquellos momentos donde el equipo Scrum se encontraba superado por la carga de trabajo y necesitaba de la ayuda del cliente para solventar los rechazos y errores que impedían el transcurso normal del ejercicio, además de toda la transparencia que esto otorgaba al proyecto.

Todo lo anterior se traduce en una mayor productividad y una producción de mejor calidad, atributos de los que carecía Dable Red S.L. en anteriores ejercicios. La empresa usaba un método de trabajo totalmente diferente al estudiado en este apartado. No se trataba de un método incremental, por lo que no existían entregas parciales del producto. En su lugar, a cada trabajador se le asignaba un cluster a realizar (prediseño, diseño y documentación), el cual se trabajaba de manera individual y se entregaba a la conclusión de todas las tareas. Este método presentaba un gran inconveniente. Al realizarse una sola entrega del producto y no contar con asesoramiento o supervisión, se producían numerosos rechazos por parte del cliente, la mayoría de ellos suponían rediseños completos y cambios de documentación. Esto suponía un gran contratiempo a los trabajadores, los cuales tenían que paralizar sus diseños actuales para corregir y solventar los reparos que iban llegando. Además, el continuo cambio entre clusters y diseños provocaban equivocaciones y despistes por parte de los trabajadores, los cuales no conseguían centrarse en una tarea concreta. Debido a estos desbarajustes, muchos proyectos se alargaban más de lo previsto y no se terminaban de cerrar por completo, llegando incluso a solaparse con otros proyectos sucesores.

A continuación se muestra una tabla comparativa de ambos métodos de trabajo, donde se reflejan las principales diferencias entre ellos:

Tabla 7-1. Comparativa entre los métodos de trabajo (Fuente: Elaboración propia)

Método convencional	Método Scrum
Método no incremental y secuencial.	Método incremental e iterativo.
Desarrollo de tareas de forma secuencial.	Desarrollo de tareas de forma simultánea.
Una única entrega por cluster.	Varias entregas parciales por cluster (prediseño, diseño y documentación).
Trabajo individual.	Trabajo cooperativo (equipos de 2 a 3 componentes).
Tiempos de entrega largos.	Tiempos de entrega cortos.
Equipo sin distinción de funciones.	Equipo auto-organizado y con roles definidos.
Evaluación final antes de la entrega del producto.	Seguimiento y supervisión continua del producto.
Poca comunicación con el cliente.	Contacto directo con el cliente en todo el desarrollo.
Baja capacidad de reacción ante contratiempos y modificaciones.	Alta capacidad de respuesta ante alteraciones.
Productividad reducida y baja calidad de los productos.	Alta productividad y buena calidad de producción.
Poca información sobre el estado del producto y las tareas realizadas.	Alta transparencia de las operaciones y estado del producto.

En conclusión, la aplicación del método Scrum supuso una notable mejora en todos los aspectos con respecto al método anterior, optimizando en gran medida el proceso de desarrollo y gestión del proyecto. La mejoría de servicio quedó patente y con margen de mejora para futuros ejercicios. Como ya se comentó anteriormente, explotar al máximo el método Scrum no es tarea sencilla. Exprimir el método para beneficiarse de todas las ventajas que puede llegar a aportar es una ardua tarea que conlleva preparación y experiencia. Para Dable Red S.L. ha sido una primera toma de contacto muy satisfactoria y que, a través de las retrospectivas llevadas a cabo en cada Sprint, pretende ampliar sus conocimientos sobre la metodología ágil y progesar funcionalmente en futuros proyectos.

La innovación es el motor del progreso y las metodologías ágiles son las mejores herramientas para ello.

REFERENCIAS

Kent Beck y resto de críticos (17 de Febrero de 2001) *Manifiesto Ágil, los cuatro postulados de los métodos ágiles.* / Available at: <http://www.caminoagil.com>

Kent Beck y resto de críticos (17 de Febrero de 2001) *Los 12 principios recogidos en el Manifiesto Ágil.* / Available at: <http://www.caminoagil.com>

Taiichi Ohno (1912-1990) 'Toyota Production System: Beyond Large-Scale Production'. *Ideas sobre producción industrial.*

David J Anderson (Nov 2016) 'Kanban: Cambio evolutivo exitoso oara su negocio de tecnología'. *Definición de Kanban*

David J Anderson (Nov 2016) 'Kanban: Cambio evolutivo exitoso oara su negocio de tecnología'. *Principios fundamentales definidos*

Google (2010) 'DesignsSprintKit'. *Definición de Sprint* / Available at: <https://designSprintkit.withgoogle.com/>

Ikujiro Nonaka y Takeuchi (1986). *Analogía del rugby y Scrum.*

David Matthew (no date). Scrum Master Certificado de la empresa Incentive Technology Group. *Reflexión sobre la complejidad del método Scrum.* / Available at: <https://blog.trello.com/es/metodologia-scrum>

“Los objetivos no son un destino, son una dirección. No son órdenes, son compromisos. No determinan el futuro; son medios para movilizar los recursos y la energía capaces de crear el futuro”.

- Peter Druker, -

“La clave del éxito en los negocios es detectar hacia dónde va el mundo y llegar ahí primero”.

- Bill Gates, -

“Toda compañía debería trabajar duro para hacer que su propia línea de productos quede obsoleta, antes de que lo haga competencia”.

- Philip Kotler, -

“La agilidad puede tomar diferentes formas, pero básicamente consiste en la habilidad de adaptarse rápidamente o incluso anticiparse al contexto y liderar un cambio. La agilidad, en el sentido más amplio de su definición, afecta al diseño estratégico, las operaciones, la tecnología e innovación, y la habilidad que tiene una organización de innovar en la incorporación de productos, procesos y modelos de negocios”.

- Michael Cusmano, profesor de la MIT Sloan School of Management, -

“Scrum es como tu suegra, resalta todos tus fallos”.

-Ken Schwaber -

“El mundo cambia rápidamente. Los grandes ya no serán los que venzan a los pequeños; serán los ágiles quienes superen a los lentos.”.

- Rupert Murdoch -

“La vida es el arte de sacar conclusiones suficientes a partir de datos insuficientes.”.

- Samuel Butler -