



Cuadro de Mando para la gestión ágil, caso de implantación en departamento de B.I.

Aleixandre Maravilla Girbés

Máster universitario en Ingeniería de Telecomunicación

José Julio López Santos

Junio del 2016



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2016 Alexandre Maravilla Girbés.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

| | |
|---|--|
| Título del trabajo: | Cuadro de Mando para la gestión ágil, caso de implantación en departamento de B.I. |
| Nombre del autor: | Aleixandre Maravilla Girbés |
| Nombre del consultor: | José Julio López Santos |
| Fecha de entrega (mm/aaaa): | 06/2016 |
| Área del Trabajo Final: | Dirección y gestión de las TIC |
| Titulación: | <i>Máster universitario de Ingeniería de Telecomunicación</i> |
| Resumen del Trabajo (máximo 250 palabras): | |
| <p>Este Trabajo pretende aportar una solución práctica para la gestión ágil de proyectos de inteligencia de negocio (B.I).</p> <p>Como profesional del área de B.I y por mi propia experiencia, muchos de los proyectos terminan fallando en su aportación de valor. Tras investigar, pude comprobar que diversos reputados profesionales, coinciden en que los enfoques tradicionales para la gestión de proyectos de B.I, no siempre son los más oportunos.</p> <p>La solución planteada, consta de un proceso ágil pensado para las áreas de B.I (el contexto), y un Cuadro de Mando (la herramienta), dirigido a la implantación de dicho proceso.</p> <p>El principal producto obtenido es el Cuadro de Mando (CdM), pensado para realizar la planificación y el seguimiento de proyectos con enfoques ágiles. Además se ha probado la eficacia del CdM, a través de su implantación en un proyecto real de B.I.</p> <p>Para la implantación del CdM se ha diseñado un proceso ágil propio, pensado para las áreas de inteligencia de negocio. Este proceso creado, sirve como contexto para la implantación del CdM.</p> <p>Tras el análisis de los resultados obtenidos, se ha podido verificar la validez del CdM y del proceso ágil definido. Como líneas futuras de actuación, se propone continuar este Trabajo, validando la utilidad de la solución aportada, en otros proyectos de B.I.</p> | |

Abstract (in English, 250 words or less):

This project provides a practical solution for agile business intelligence (B.I) project management.

As a B.I professional and because of my experience, many projects end up failing in their value contribution. Researching I could check, that many reputable professionals, agree that traditional approaches, are not always the most appropriate when managing B.I projects.

The proposed solution consists of an agile process designed for B.I areas. Also it has been designed a scorecard aimed to the implementation of the agile process.

The main product obtained is the Scorecard. It has been designed for planning and monitoring projects with agile approaches. The effectiveness of the tool has been proven through the implementation of the Scorecard in a real B.I project.

The implementation of the Scorecard has followed an agile process designed for B.I areas. The B.I process created, serves as a context for the implementation of the Scorecard.

Once the results has been analyzed, it has been verified the Scorecard usefulness. It has also proven the usefulness of the agile process defined. Next step to continue this project could be the implementation of the agile process and the Scorecard, in other B.I projects, in order to validate their usefulness.

Palabras clave (entre 4 y 8):

Agilidad, Business Intelligence, Cuadro de Mando, Scrum

Índice

| | |
|---|----------|
| 1. Introducción..... | 1 |
| 1.1 Contexto y justificación del Trabajo..... | 1 |
| 1.2 Objetivos del Trabajo..... | 2 |
| 1.3 Enfoque y método seguido..... | 2 |
| 1.4 Planificación del Trabajo | 3 |
| 1.5 Breve resumen de productos obtenidos | 5 |
| 1.6 Breve descripción de los otros capítulos de la memoria | 7 |
| | |
| 2. Agile Business Intelligence | 8 |
| 2.1 Introducción..... | 8 |
| 2.1.1 Agilidad en la gestión de proyectos..... | 10 |
| 2.1.2 El Manifiesto Ágil..... | 11 |
| 2.1.3 Scrum..... | 14 |
| 2.1.3.1 Roles Scrum..... | 15 |
| 2.1.3.2 Las Historias de Usuario | 16 |
| 2.1.3.3 El ciclo de trabajo en Scrum..... | 19 |
| 2.1.3.4 Las reuniones..... | 21 |
| 2.1.4 Kanban..... | 23 |
| 2.2 Agilidad en proyectos de B.I..... | 24 |
| 2.2.1 Necesidades de las áreas de B.I..... | 25 |
| 2.2.2 Adaptación del método ágil a los proyectos de B.I..... | 26 |
| 2.3 Caso de implantación del método ágil en el departamento de B.I de una empresa española del sector de las TIC..... | 29 |
| 2.4 Cuadro de Mando para la gestión ágil..... | 32 |
| 2.4.1 Estructura | 33 |
| 2.4.2 Métricas..... | 37 |
| 2.4.3 Resultado de la ejecución del caso práctico en el cuadro de mando | 44 |
| 2.4.3.1 Priorización y alertas en el Product Backlog | 44 |
| 2.4.3.2 Seguimiento diario..... | 48 |
| 2.4.3.3 Seguimiento de los Sprints..... | 50 |
| 2.4.3.3.1 Planificado vs Completado | 50 |

| | |
|--|-----------|
| 2.4.3.3.2 Previsión de finalización | 51 |
| 2.4.3.4 Ampliación Product Backlog..... | 52 |
| 2.4.3.4 Ratios | 55 |
| 2.4.3.4.1 Productividad..... | 58 |
| 2.4.3.4.2 Velocidad..... | 58 |
| 2.4.3.4.3 Factor Foco | 60 |
| 2.4.3.4.4 Ratio Cycle Time | 60 |
| 2.4.3.5 Dashboard..... | 61 |
| 2.5 Conclusiones del método ágil y el CdM..... | 62 |
| 3. Conclusiones generales | 67 |
| 3.1 Lecciones aprendidas | 67 |
| 3.2 Consecución de objetivos..... | 67 |
| 3.3 Planificación y metodología seguida | 68 |
| 4. Glosario..... | 70 |
| 5. Bibliografía..... | 73 |

Lista de ilustraciones

| | |
|--|----|
| Ilustración 1: Planificación temporal de las distintas fases del proyecto | 4 |
| Ilustración 2: Esfuerzo estimado de las distintas fases planificadas en el proyecto | 5 |
| Ilustración 3: Ejemplo de ciclo de vida predictivo o en cascada..... | 9 |
| Ilustración 4: Ejemplo de ciclo de vida ágil (iterativo e incremental)..... | 11 |
| Ilustración 5: El Manifiesto Ágil..... | 13 |
| Ilustración 6: Los distintos roles en Scrum | 16 |
| Ilustración 7: Plantilla para recoger las Historias de Usuario (1) | 18 |
| Ilustración 8: Plantilla para recoger las Historias de Usuario (2) | 18 |
| Ilustración 9: El ciclo de trabajo de Scrum | 20 |
| Ilustración 10: Las reuniones en Scrum | 22 |
| Ilustración 11: Perspectiva completa del ciclo de trabajo de Scrum..... | 22 |
| Ilustración 12: Tablero Kanban..... | 23 |
| Ilustración 13: Tablero Kanban con elementos de Scrum | 27 |
| Ilustración 14: Tablero ágil implementado | 30 |
| Ilustración 15: Proceso ágil seguido..... | 32 |
| Ilustración 16: Portada del Cuadro de Mando para la gestión ágil | 32 |
| Ilustración 17: Bloque de Dashboard | 34 |
| Ilustración 18: Bloque de definición de las Historias de Usuario | 34 |
| Ilustración 19: Bloque de priorización del Producto Backlog..... | 35 |
| Ilustración 20: Bloque de seguimiento del Proyecto, apartado Iteraciones | 36 |
| Ilustración 21: Bloque de seguimiento del Proyecto, apartado Ratios..... | 36 |
| Ilustración 22: Bloque de seguimiento del Sprint, apartado Burn Down Valor. | 37 |
| Ilustración 23: Bloque de definición de las métricas..... | 37 |
| Ilustración 24: Bloque de definición de las métricas..... | 38 |
| Ilustración 25: Composición inicial del Product Backlog | 45 |
| Ilustración 26: Orden inicial del Product Backlog sin prioritzar..... | 45 |
| Ilustración 27: Product Backlog inicial priorizado | 46 |
| Ilustración 28: Detalle del Product Backlog inicial priorizado..... | 46 |

| | |
|---|----|
| Ilustración 29: Grafico auxiliar (1) del bloque Product Backlog al inicio del proyecto | 47 |
| Ilustración 30: Grafico auxiliar (1) del bloque Product Backlog al inicio del proyecto | 47 |
| Ilustración 31: Evolución real del proyecto. 2 primeras iteraciones (Sprints) . | 48 |
| Ilustración 32: Gráfico Burn Down Esfuerzo Sprint 1 | 48 |
| Ilustración 33: Gráfico Burn Down Valor Sprint 1 | 49 |
| Ilustración 34: Evolución del proyecto. Sprints 6 y 7 | 50 |
| Ilustración 35: Gráfico auxiliar (1) al final del Sprint 7 | 50 |
| Ilustración 36: Gráfico auxiliar (2) al final del Sprint 7 | 51 |
| Ilustración 37: Gráfico Burn Up Esfuerzo al final del Sprint 7 | 52 |
| Ilustración 38: Gráfico Burn Up Valor al final del Sprint 7..... | 52 |
| Ilustración 39: Composición del Product Backlog al inicio del Sprint 8..... | 53 |
| Ilustración 40: Grafico auxiliar (1) del bloque Product Backlog en el Sprint 8 | 53 |
| Ilustración 41: Grafico auxiliar (2) del bloque Product Backlog en el Sprint 8 | 53 |
| Ilustración 42: Gráfico Burn Up Esfuerzo después de la ampliación del Product Backlog | 54 |
| Ilustración 43: Gráfico Burn Up Valor después de la ampliación del Product Backlog | 55 |
| Ilustración 44: Evolución del proyecto. Sprints 8 y 9 | 56 |
| Ilustración 45: Gráfico Burn Up Esfuerzo Sprint 9..... | 56 |
| Ilustración 46: Gráfico Burn Up Valor Sprint 9..... | 57 |
| Ilustración 47: Gráfico Productividad al final del Sprint 9 | 58 |
| Ilustración 48: Gráfico Velocidad al final del Sprint 9 | 59 |
| Ilustración 49: Gráfico Factor Foco al final del Sprint..... | 60 |
| Ilustración 50: Gráfico Ratio Cycle Time al final del Sprint 9..... | 61 |
| Ilustración 51: Estado del Dashboard al final del Sprint 9 | 62 |
| Ilustración 52: Alerta de ampliación del tamaño del Product Backlog | 62 |

1. Introducción

1.1 Contexto y justificación del Trabajo

Las unidades técnicas de negocio vinculadas a empresas T.I.C (tecnologías de la información y la comunicación), a menudo trabajan en entornos altamente dinámicos, cambiantes e inciertos, donde difícilmente se puede realizar una gestión tradicional y estándar de los proyectos. En paralelo nuevos enfoques de gestión más ágiles han surgido pensados para cubrir este tipo de necesidades.

El 12 de febrero de 2001, los creadores de algunos de los métodos ágiles más conocidos en la actualidad: XP (Ron Jeffries), Scrum (Ken Schwaber), DSDM (Arie van Bennekum), Crystal (Alistair Cockburn), definieron el manifiesto ágil, el cual formaliza y agrupa los principios que rigen los principales métodos ágiles.

No obstante, adoptar una metodología ágil en un proyecto no es un trabajo sencillo, aspectos como; el tamaño, la criticidad, el nivel del equipo, el dinamismo y la cultura de los colaboradores, se presentan críticos en este empeño.

El problema más común al utilizar métodos ágiles es que no se aplican correctamente, es por tanto imprescindible estar preparado para abordar cualquier proyecto a través de estos enfoques.

Un proyecto ágil requiere de un equipo multidisciplinar, auto-organizado y auto-gestionado, con estos métodos, equipos bien motivados pueden obtener resultados notorios en plazos muy ajustados. No obstante la agilidad no está reñida con el control, los métodos ágiles nos animan a medir, analizar y mejorar de forma continua, y es en este punto donde está el foco de este proyecto.

Tras analizar las distintas herramientas de software libre de apoyo a la gestión ágil existentes en el mercado, se ha detectado la oportunidad de crear una nueva herramienta que complementa a las ya existentes, aportando una visión integrada del proceso de gestión y focalizándose en la gestión de la productividad.

Este trabajo tiene por finalidad aportar una solución práctica de apoyo en el uso de los métodos ágiles que sea de utilidad para la gestión de proyectos de inteligencia de negocio, pretende construir un elemento para el control de las principales métricas que aseguren la calidad del proyecto. El producto final es un Cuadro de Mando (CdM) mediante el cual realizar la planificación, seguimiento y control de proyectos ágiles.

Además el Cuadro de Mando se acompaña de la descripción del proceso seguido en el caso real de implantación de esta herramienta en una unidad de B.I. Mediante la implantación del proceso se pretende

validar tanto la eficacia del Cuadro de Mando, como la validez de los métodos ágiles (originalmente pensados para el desarrollo software), en el área de la inteligencia de negocio.

1.2 Objetivos del Trabajo

1. Aportar una solución práctica que sea de utilidad para la planificación, el seguimiento y el control en la gestión ágil de proyectos.
 1. Crear una visión única de las métricas de desempeño del proyecto que soporte las decisiones de gestión ágil
 2. Fomentar el diálogo diario entre los distintos actores involucrados en el proyecto
 3. Gestionar la productividad, maximizándola en las fases iniciales del proyecto
 4. Gestionar el riesgo, detectando con antelación las tareas críticas pendientes del Product Backlog.
 5. Facilitar la planificación, el seguimiento y el control
 6. Detectar desviaciones de la evolución respecto a la planificación
 7. Realizar una estimación orientativa de la fecha de finalización del proyecto
 8. Apoyar en la decisión de cese temprano del proyecto
 9. Identificar el exceso de tareas simultaneas en desarrollo
2. Definir y validar un proceso que permita implantar la gestión ágil en el área de planificación y B.I de una empresa española del sector de las TIC.
3. Conocer las bases de los métodos ágiles y su diferencia con los métodos tradicionales
4. Profundizar en el conocimiento de; Scrum y Kanban.
5. Aplicar el método ágil en la gestión de proyectos.

1.3 Enfoque y método seguido

Se ha optado por la construcción del Cuadro de Mando en formato compatible con el programa ofimático Microsoft Excel por su amplio y generalizado uso en entornos laborales dentro de las áreas de B.I.

Las limitaciones temporales del proyecto, sumadas a las limitaciones en recursos, supondrían un riesgo en caso de afrontar el desarrollo del Cuadro de Mando en un lenguaje de desarrollo software con mayor grado de especialización. No hay que olvidar que la finalidad del trabajo no es el desarrollo software en sí, sino la aportación de una solución que facilite y mejore la gestión ágil de proyectos de B.I.

Por tanto se ha optado por el desarrollo de un nuevo producto en formato compatible con MS Excel ya que todas las herramientas con licencias de uso abiertas analizadas, presentan un interface web o corresponden con software instalable en equipos locales. Además cabe destacar que el hecho de haber optado por construir la solución a través de una plataforma web hubiese podido crear rechazo en su uso por motivos de privacidad. La información almacenada en la herramienta puede resultar sensible en función del proyecto que se esté desarrollando y el trabajo en una unidad de equipo local puede aportar mayor grado de confianza.

El método seguido ha sido iterativo, revisando continuamente el producto en cada iteración. El enfoque ágil escogido ha supuesto lanzar en las primeras fases tempranas del proyecto un prototipo del producto, con la finalidad de ir depurándolo hasta presentar la propuesta final.

Utilizando el Cuadro de Mando como soporte físico, se ha diseñado además un proceso para la implantación del método ágil en un departamento de business intelligence. La implantación del proceso ha permitido comprobar la eficacia del Cuadro de Mando en un caso real de uso y ha propiciado un muy valioso feedback de cara a futuras mejoras de la herramienta.

Otra validación fruto de la implantación del proceso ha sido el testeo de la idoneidad de los métodos ágiles para proyectos de B.I, no tan centrados en desarrollo software y más orientados a la aportación directa de valor al negocio de una compañía.

1.4 Planificación del Trabajo

El Trabajo planificado se ha dividido en 6 fases, empezando el 29 de Febrero del 2016 y terminando el 24 de Junio del mismo año.

Fase 1. Mandato del proyecto.

- Elección del tema, descripción y planificación del trabajo.

Fase 2. Estudio / Análisis:

- Análisis del estado del arte de los métodos ágiles

- Realización del M.O.O.C; “Agilidad y Lean. Gestionando los proyectos y negocios del s. XXI”, profesor Javier Garzás, plataforma Miriada X, Universidad Rey Juan Carlos.

- Investigación y otra bibliografía.

Fase 3. Prototipado:

- Construcción del prototipo del Cuadro de Mando.

Fase 4. Cuadro de Mando:

- Construcción del Cuadro de Mando.

Fase 5 Diseño e implantación del método:

- Diseño del proceso para implantar el método ágil en un departamento de B.I.
- Caso real de implantación del método ágil apoyado en el Cuadro de Mando, en el departamento de B.I de una empresa española del sector TIC.

Fase 6. Entregables:

- Elaboración de los entregables finales; memoria, presentación y manual de usuario del CdM.

A continuación se muestran dos gráficos a modo de esquemas, el primero con la temporización de cada fase, y el segundo con el esfuerzo que supone cada fase en términos relativos sobre el total de las fases del proyecto.

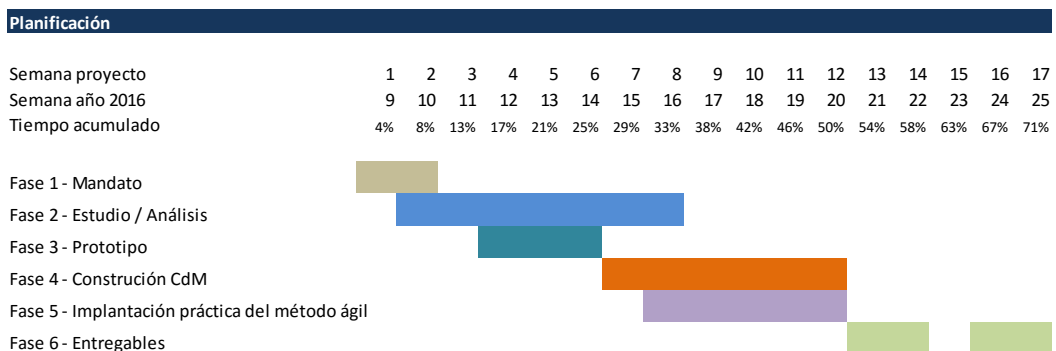


Ilustración 1: Planificación temporal de las distintas fases del proyecto

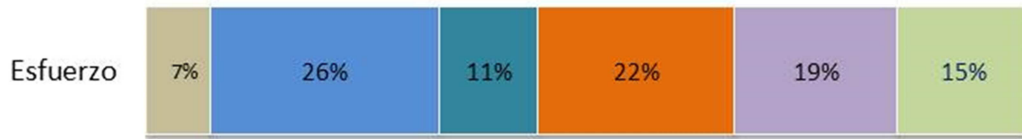


Ilustración 2: Esfuerzo estimado de las distintas fases planificadas en el proyecto

En la planificación se pueden observar distintos solapamientos temporales en la ejecución de las distintas fases. En algunos casos estos solapamientos son debidos a que las fases en desarrollo se pueden llevar a cabo en paralelo, ya que no guardan una relación directa entre sí, este caso lo encontramos en las fases 4 y 5 en dónde la construcción del Cuadro de Mando no interfiere en la implementación práctica del método ágil, ya que ésta no es llevada a cabo a través del Cuadro de Mando. En cambio por ejemplo el solapamiento de la fase de prototipado, sí que guarda relación directa con la fase de análisis, pues el prototipo del Cuadro de Mando se ha de elaborar en base al estudio de los principios que rigen los métodos ágiles. No obstante y dado el enfoque ágil e iterativo del que se ha decidido dotar al desarrollo de este proyecto, el solapamiento entre fases no es un riesgo ni siquiera en el caso en que las fases sí guarden una estrecha relación, es más, el paralelismo se percibe como una ventaja de cara a poder entregar un mejor producto final.

Cabe destacar el momentáneo parón en el desarrollo de la Fase 6, en concreto en la semana del año número 23 (15 del proyecto). Éste hueco en la planificación ha sido debido a la preparación de las distintas pruebas finales del máster.

En cuanto a la relación de la planificación con las distintas pruebas de evaluación continua (PEC), la primera prueba corresponde con la Fase 1 y describe el mandato del proyecto y la planificación. En la segunda PEC se ha entregado el prototipo del Cuadro de Mando que corresponde con la Fase 3. La tercera PEC planificada se ha correspondido con la Fase 4 y en ella se ha hecho entrega del Cuadro de Mando previamente prototipado. Por último la PEC final hace referencia a la Fase 7 en la que se han preparado los entregables.

1.5 Breve resumen de productos obtenidos

Los productos obtenidos han sido:

1. Cuadro de Mando para la gestión ágil de proyectos.

Acomete el primer objetivo del Trabajo proporcionando un interfaz para la planificación, el seguimiento y el control, en la gestión ágil de proyectos. Su formato es compatible con el software MS Excel

y para ejecutarlo es necesario disponer de una licencia activa del programa Microsoft Excel y un equipo que soporte su ejecución.

2. Manual de usuario del Cuadro de Mando.

Compuesto por dos videos en donde se explica el modo de uso del Cuadro de Mando. A través de un ejemplo desarrollado desde cero, el manual permite la comprensión del orden de ejecución de las tareas necesarias para insertar datos, y entender su representación o salida en las distintas partes del Cuadro de Mando.

3. Ficheros auxiliares. Datasets y CdMs de pruebas, junto con un fichero resumen de la evolución del proyecto.

Formado por 8 ficheros compatibles con el software MS Excel; Dataset 1.xlsx, Dataset 2.xlsx, CdM ejemplo 1.xlsx, CdM ejemplo 2.xlsx, CdM ejemplo 3.xlsx, CdM ejemplo 4.xlsx, CdM ejemplo 5.xlsx y Evolución proyecto.xlsx.

La finalidad de los Datasets es proporcionar datos de entrada para la comprobación de la funcionalidad del propio Cuadro de Mando. Cada fichero recoge un conjunto de Historias de Usuario correspondientes a las Historias de Usuario reales implementadas en caso práctico que posteriormente se detallará. Las Historias de Usuario recogidas en un mismo fichero tienen la misma fecha de entrada en el Product Backlog, las fechas de ambos ficheros son distintas motivo por el cual se ha decidido hacer entrega de dos ficheros separados.

Los 5 ejemplos del Cuadro Mando muestran la evolución del caso real de implantación del CdM. Cada uno de los ficheros está parcialmente completado emulando la evolución real del proyecto. El objetivo de estos documentos es seguir con más detalle la explicación del caso expuesta en la memoria, a la vez que permitir (junto con los Datasets) realizar las pertinentes pruebas para la comprobación del funcionamiento del CdM.

El fichero; Evolución proyecto.xlsx, muestra las evolución de los Sprints y las Historias de Usuario que se han ido trabajando en cada uno de ellos. La utilidad de este fichero es acompañar la explicación del caso real de implementación del método ágil sobre el CdM, detallado en la memoria.

4. Memoria y presentación del Trabajo.

Memoria en documento de texto con el detalle del Trabajo realizado y presentación compuesta por; un documento en formato Microsoft Power Point y un video en formato "wmv" compatible con la mayoría de reproductores multimedia. De forma detallada la memoria y de forma resumida la presentación, ambos documentos abordan los puntos detallados a continuación en el

apartado 1.6, “Breve descripción de los otros capítulos de la memoria”.

1.6 Breve descripción de los otros capítulos de la memoria

Los capítulos a desarrollar en la memoria son:

1. Introducción a los métodos ágiles en la gestión de proyectos
En el primero de los capítulos realiza una breve aproximación a los conceptos clave que definen la agilidad, necesarios para entender el resto del trabajo.
2. Agilidad en proyectos de B.I
El segundo capítulo se centra en el diseño de un proceso ágil adaptado a las necesidades de las áreas de business intelligence, dado que estas áreas no están únicamente centradas en desarrollo software sino que además también están orientadas a la aportación directa de valor al negocio vía análisis, reporting, y otras tareas no directamente relacionadas con el desarrollo software.
3. Presentación del caso de implantación del método ágil en el departamento de B.I de una empresa española de las TIC
El tercer capítulo describe el caso real de implantación del método ágil, en él se presenta el proyecto real que se ha implementado bajo el enfoque de la agilidad.
4. Cuadro de Mando para la gestión ágil
El cuarto capítulo presenta la herramienta construida y se muestran sobre ella los resultados obtenidos en el caso real de implantación del método ágil.
5. Conclusiones método ágil y CdM.
Por último el quinto capítulo corresponde con las conclusiones de los productos entregados, se evalúa el valor que aporta el Cuadro de Mando construido, y se valida la idoneidad de los métodos ágiles para labores de inteligencia de negocio.

2. Agile Business Intelligence

2.1 Introducción

Durante mucho tiempo la gestión de proyectos software ha abrazado la predictibilidad acotándola bajo el término “ciclo de vida en cascada”.

La predictibilidad se basa en la planificación, en la división de un proyecto en fases dotándolas de una cierta rigidez, en la secuenciación de las distintas etapas de un proyecto haciendo necesario terminar exitosamente una etapa antes de iniciar la siguiente.

Las etapas son predictivas porque predicen lo que sucederá en la siguiente fase, por ejemplo los planos que un arquitecto diseña, intentan predecir lo que sucederá en la etapa de construcción, y es de vital importancia que esos planos sean lo suficientemente detallados como para que los jefes de obra no tengan la más mínima duda sobre cómo interpretarlos. Es evidente que hasta que los planos no estén terminados los obreros no podrán empezar a construir ni siquiera los cimientos (pongamos por ejemplo de un edificio). Cada una de estas fases es llevada a cabo una única vez (los planos se realizan únicamente al inicio), y los límites temporales (inicio y fin) están claramente acotados. Esta clara diferenciación de las fases o etapas da lugar a la aparición de profesionales especializados en cada una de ellas, arquitectos, aparejadores, jefes de obra u obreros, además cada etapa concluye con la entrega de documentación específica como pueden ser los planos o las distintas memorias.

La predictibilidad llevada al mundo del software se traduce en la aparición de los roles diferenciados (por ejemplo) de arquitecto software y programador, o en la separación de las fases (por ejemplo) de especificación de requisitos, diseño o construcción. Como vemos la gestión de proyectos predictiva es típica de disciplinas clásicas como la arquitectura, y el desarrollo software ha incorporado sus prácticas refiriéndose a ello como a la gestión basada en el ciclo de vida en cascada.

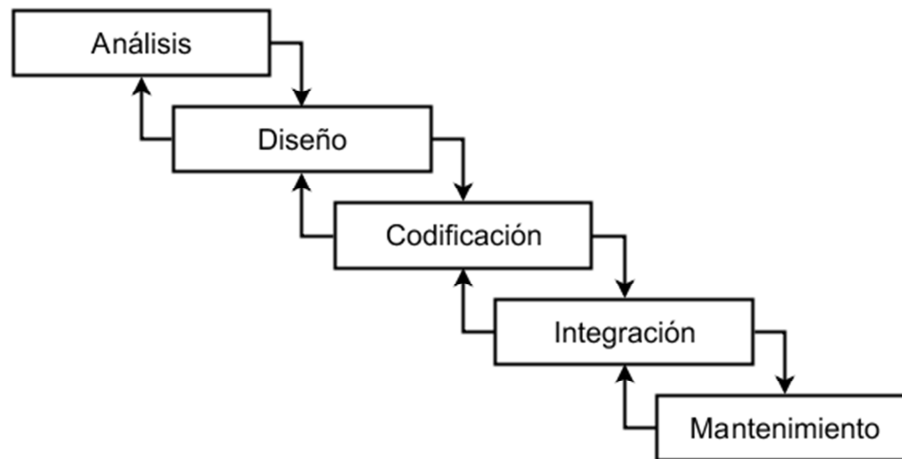


Ilustración 3: Ejemplo de ciclo de vida predictivo o en cascada

Observamos que la arquitectura (y otras ingenierías clásicas) necesita seguir este tipo de ciclos de vida en cascada o predictivos, porque precisa de un diseño detallado e inalterable previo a la construcción, pues el hecho de modificar los planos de un edificio una vez empezada la fase de construcción puede contraer un alto impacto en tiempo y costes.

No obstante lo que funciona para la arquitectura no funciona para el desarrollo software. Tal y como constata Javier Garzas en el material del curso; “Agilidad y Lean. Gestionando los proyectos y negocios del s. XXI” [11]:

“En software, la experiencia nos dice que es muy difícil especificar los requisitos en una única y primera fase. Por la complejidad de muchas de las reglas de negocio que automatizamos cuando construimos software, es muy difícil saber qué software se quiere hasta que se trabaja en su implementación y se ven las primeras versiones o prototipos. También es muy difícil documentar de una única vez, a la primera, antes de la codificación, un diseño que especifique de manera realista y sin variación todas las cuestiones a implementar en la programación.”

“En el desarrollo software es casi imposible cerrar un diseño a la primera para pasarlo a programación sin tener que modificarlo posteriormente.”

A todo esto hay que añadir el hecho que (y siguiendo el ejemplo de la construcción de un edificio), modificar un programa software en una fase avanzada del proyecto no tiene el mismo impacto en tiempo y coste en comparación con la modificación de la estructura de una de las vigas principales del edificio, en el caso del software basta con realizar cambios en el código fuente del programa o aplicación.

Vemos como en el desarrollo software las etapas de diseño y construcción no guardan una separación tan evidente como en otras disciplinas como la arquitectura, por tanto parece sensato cuestionarse

el hecho de que el uso de la predictibilidad como herramienta para la gestión de proyectos software sea la aproximación más idónea.

2.1.1 Agilidad en la gestión de proyectos

El término ágil como método de desarrollo de software fue acuñado en el Agile Manifiesto (del que luego hablaremos) en el año 2001. Este enfoque se caracteriza por la creatividad, la flexibilidad, la adaptabilidad, la capacidad de respuesta y el foco en las personas. El actual entorno complejo, incierto y cambiante está presionando a los desarrolladores a adoptar métodos ágiles en lugar de desarrollar software de manera tradicional.

Los métodos ágiles, de hecho, llegaron como respuesta a los fracasos constantes de los proyectos software gestionados de manera tradicional. Los métodos ágiles se postularon como alternativa después de décadas de aplicación de metodologías de desarrollo de software tradicionales, basadas en procesos que se caracterizan por la abundante documentación y el énfasis en los procesos y no tanto en la comunicación con el cliente [4].

Los enfoques ágiles se basan en un método de control empírico - un proceso de toma de decisiones basadas en las realidades observadas en el proyecto actual [9]. Para dar cabida a una inspección y adaptación frecuente, los proyectos ágiles trabajan en iteraciones (segmentos más pequeños de la totalidad del proyecto). En los proyectos ágiles, se sigue teniendo el mismo tipo de trabajo que en un proyecto tradicional en cascada, sigue siendo necesario crear requisitos y diseños, desarrollar el producto e integrarlos y testarlo. Sin embargo, en lugar de completar estos pasos para todas las funciones del producto a la vez, se rompe el proyecto en iteraciones, también llamados Sprints (de los que posteriormente en el capítulo “2.1.3 Scrum”).

Así pues, la agilidad, frente al ciclo de vida en cascada promueve un ciclo de vida incremental e iterativo, incremental porque en cada iteración se añade valor al producto, iterativo porque en cada ciclo o iteración se revisa y mejora el mismo.

Por tanto el ciclo de vida iterativo e incremental es la base del método ágil, en él, en cada iteración se van liberando partes del producto (prototipos) periódicamente, y cada nueva versión aumenta la funcionalidad y mejora en calidad respecto a la anterior, se persigue conseguir una mínima versión funcional del producto en fases tempranas del proyecto y no esperar hasta las últimas etapas, en donde el tiempo de reacción es un hándicap negativo.

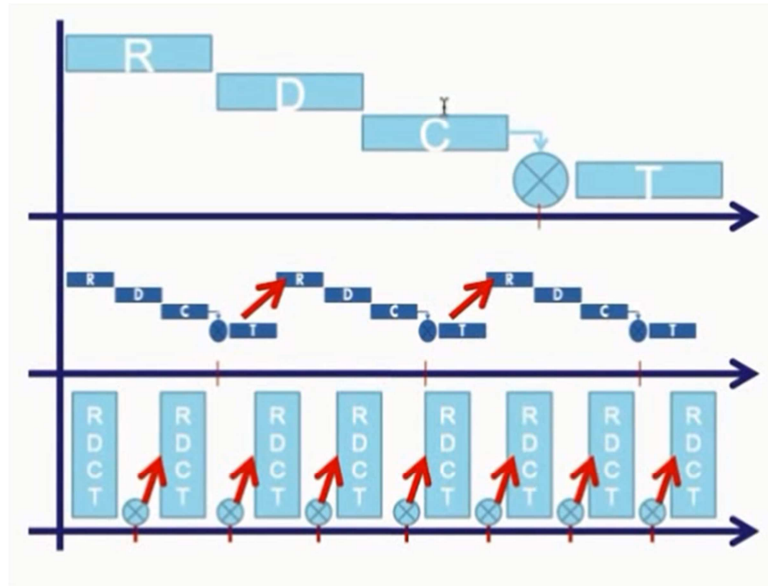


Ilustración 4: Ejemplo de ciclo de vida ágil (iterativo e incremental)

En la figura 4 observamos como el ciclo de vida ágil no rehúye de las distintas actividades que forman un proyecto; requisitos, diseño, construcción y testado. No obstante vemos como estas actividades no se producen de manera lineal, sino que se desarrollan en paralelo y no tienen el porqué de seguir un orden lógico, podemos diseñar a partir de unos mínimos requisitos, luego construir el producto y volver a diseñar para posteriormente modificar o ampliar los requisitos.. etc. Además cabe destacar que con el ciclo de vida ágil se multiplican los puntos de entrega del producto, avanzando al máximo la entrega del primer prototipo y siendo éste una mínima unidad funcional del producto global, que finalmente esperamos conseguir.

Para concluir mencionaremos a Scott Ambler [22], quien afirma que un proyecto ágil seguiría la estructura de un proyecto con un ciclo de vida iterativo e incremental, en el que los colaboradores trabajarían de manera colaborativa formando equipos autónomos y autoorganizados.

2.1.2 El Manifiesto Ágil

A mediados de la década de 1990, y en contexto de la burbuja de las puntocom, los desarrolladores software estaban bajo constante presión para ganar la partida tecnológica frente a sus competidores, la industria de la tecnología de la información (IT) fue completamente reinventada en unos pocos años.

Teniendo en cuenta el ritmo de cambio en ese momento, surgieron los primeros problemas a los que los desarrolladores se enfrentaron por la aplicación de los métodos de tradicional de gestión de proyectos (ciclo de vida en cascada). Estos métodos no permitían a los desarrolladores ser lo suficientemente sensibles a la naturaleza dinámica del mercado y a la aparición de nuevas aproximaciones a los negocios. Los equipos de

desarrollo comenzaron a explorar alternativas a estos enfoques anticuados para la gestión de proyectos.

El 12 de febrero del 2001, 17 destacados y conocidos profesionales de la ingeniería del software se reunieron en Snowbird, Utah, para compartir sus experiencias, ideas y prácticas, y sugerir formas de mejorar el mundo del desarrollo de software.

Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por su rigidez y dirigidos por la documentación que se genera en cada una de las etapas de los proyectos. Su objetivo fue establecer los valores y principios que permitirían a los equipos desarrollar software rápidamente y responder a los cambios producidos a lo largo del ciclo de vida de un proyecto. El trabajo de este grupo de profesionales estaba destinado a hacer que la industria del software fuese más productiva, más humana y más sostenible.

El manifiesto ágil [10] se compone de 4 principios:

- Individuos e interacciones en lugar de herramientas y procesos:

Valorar las buenas prácticas de desarrollo y gestión de los participantes del proyecto, esto facilita el trabajo en equipo y disminuirá los impedimentos para que realicen su trabajo. Asimismo compromete al equipo de desarrollo y a los individuos que lo componen.

Una simple conversación puede resolver muchos problemas en un tiempo relativamente corto. Tratar de emular el poder de una conversación directa con el correo electrónico, hojas de cálculo o documentos puede requerir de una gran cantidad esfuerzo innecesario, en lugar de añadir claridad, este tipo de comunicaciones no directas son a menudo ambiguas y requieren de mucho más tiempo a la vez que distraen al equipo de desarrollo del principal propósito de crear un producto.

- Trabajar en el desarrollo de software en lugar de trabajar en la elaboración de extensa documentación:

El problema no es la documentación sino su utilidad. No se considera útil producir extensos documentos a menos que se vayan a utilizar de forma inminente para tomar una decisión de alto impacto. Los documentos deben ser cortos y centrarse en lo fundamental lo que no significa que la documentación válida sea el propio código fuente, este exceso de flexibilidad puede terminar siendo improductivo. La variación de la cantidad y tipo de documentación puede ser amplia dependiendo el tipo de cliente o de proyecto

- Colaboración con el cliente en lugar de la negociación de contratos:

Históricamente los tradicionales enfoques de gestión de proyectos implicaban al cliente en el proyecto en 3 momentos puntuales; al inicio, cuando alguna modificación relevante sucedía y al final. Este enfoque desalienta la entrada del cliente en el proceso de gestión del proyecto y elimina la valiosa aportación que puede realizar, además incluso puede crear una relación de confrontación con los equipos de proyecto. Los pioneros de la agilidad entendieron que la colaboración, en lugar de la confrontación, produce mejores resultados generando productos más útiles.

Por tanto se entiende que el éxito del proyecto depende de la directa interacción entre el cliente y el equipo de desarrollo, el problema es que muchas veces el cliente no está disponible con lo que será necesario designar a un interlocutor del cliente que forme parte del proceso ágil.

- Responder al cambio en lugar de seguir un plan:

El cambio es una valiosa herramienta para la creación de productos. Los equipos de proyecto que son capaces de responder rápidamente a los clientes, los usuarios de los productos y el mercado en general, son capaces de desarrollar productos relevantes y útiles que la gente quiere usar. Por tanto pasamos de la anticipación y la planificación estricta sin poder volver hacia atrás a la adaptación, la flexibilidad no es total, pero existen muchos puntos donde se pueden adaptar las actividades.



Ilustración 5: El Manifiesto Ágil

Como vemos el manifiesto ágil se centra en 4 aspectos; personas, comunicación, producto y flexibilidad.

Los creadores del manifiesto ágil se centraron originalmente en el desarrollo de software ya que trabajaban en la industria de TI. Sin embargo, desde 2001, las técnicas de gestión de proyectos ágiles se han extendido más allá del desarrollo de software e incluso fuera de los productos relacionados con la informática. Hoy en día, las personas utilizan enfoques ágiles para crear productos en una variedad de industrias, incluyendo la medicina, la ingeniería, la comercialización, el trabajo sin fines de lucro, e incluso la construcción de edificios.

2.1.3 Scrum

Scrum es un conjunto de buenas prácticas (framework) que proporciona un marco para la gestión de proyectos, posiblemente sea el método ágil con mayor popularidad.

Fue un modelo planteado originalmente por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, quienes compararon la nueva forma de trabajo en equipo que proponían algunas empresas de manufactura tecnológica cuando desarrollaban sus productos, con el avance en formación de melé (scrum en inglés) de los jugadores de Rugby.

Posteriormente en 1995, Ken Schwaber y Jeff Sutherland presentaron “Scrum Development Process”, un marco de reglas para desarrollo de software, basado en los principios de Scrum.

En palabras de sus precursores (Ken Schwaber y Jeff Sutherland) [8]:

“Scrum es un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible, productiva y creativamente”

Como se indica en la “Scrum Guide” [8], existen tres pilares en los que se basa:

- Transparencia

Todos los aspectos del proceso que afectan al resultado son visibles para todos aquellos que administran dicho resultado. Por ejemplo, se utilizan pizarras y otros mecanismos o técnicas colaborativas para mejorar la comunicación.

- Inspección

Se debe controlar con la frecuencia suficiente los diversos aspectos del proceso para que puedan detectarse variaciones inaceptables en el mismo.

- Revisión

El producto debe estar dentro de los límites aceptables. En caso de desviación se procederá a una adaptación del proceso y el material procesado.

Scrum propone:

- La creación de pequeños equipos interdisciplinarios y autoorganizados en contraposición a trabajar con equipos cuyos integrantes sean en su totalidad especialistas en una tarea definida.
- La división del trabajo en una lista de entregables pequeños y concretos, que puedan ser ordenados según prioridad y a su vez estimados según el esfuerzo relativo que suponen.
- La división del tiempo en iteraciones cortas de longitud fija (generalmente de 1 a 4 semanas), siendo el resultado de estas iteraciones una mínima versión funcional del producto o una nueva versión incremental del mismo
- La optimización del plan de entregas y la actualización de las prioridades en colaboración con el cliente, basándose en los conocimientos adquiridos mediante la inspección del entregable después de cada iteración.
- La Optimiza el proceso teniendo una retrospectiva después de cada iteración.

2.1.3.1 Roles Scrum

Scrum identifica distintos actores participantes del el proceso ágil, asignándoles distintas funciones y responsabilidades:

- Product Owner

El Dueño de Producto, representa a una única persona con una visión muy clara de aquello que se quiere desarrollar. Es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. Es la única persona habilitada para modificar el Product Baklog.

- Equipo Scrum

Los profesionales que desempeñan el trabajo de entregar un incremento de producto al final de cada Sprint. Como se indica en La Guía Scrum [8]:

“Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento”

Por definición los Equipos en Scrum son autoorganizados y multifuncionales. Esto se traduce en que nadie indica al Equipo Scrum (ni siquiera el Scrum Master) como debe organizarse para acometer los objetivos planteados. Además un Equipo Scrum (sus integrantes), debe de contar con todas las habilidades necesarias para conseguir el resultado esperado.

- Scrum Master

Líder al servicio del Equipo Scrum. Por una parte interactúa con el Equipo de Desarrollo para asegurarse que Scrum es implementado de manera correcta, por otra, ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no. Además también da servicio al Dueño de Producto ayudándole a cumplir con sus funciones.

Aunque Scrum directamente no los define, sí que permite además que todas aquellas personas que, ya sean inversores (stakeholders) o interesados en el producto como usuarios, formen también parte del proyecto, respetando los roles formales y participando como audiencia.

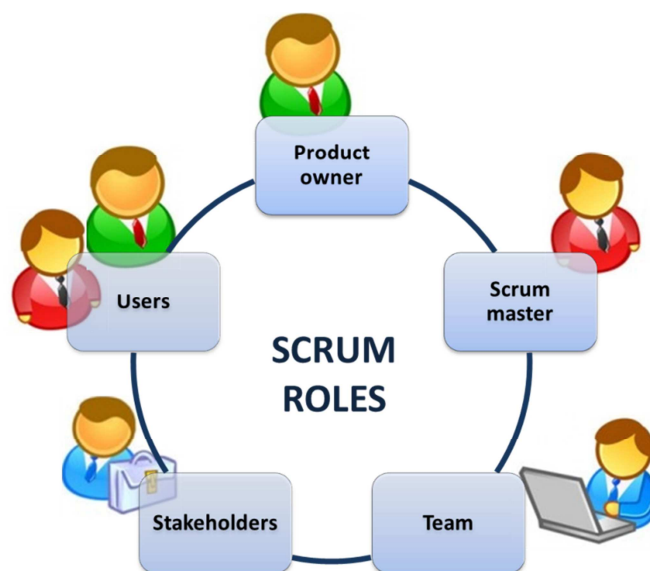


Ilustración 6: Los distintos roles en Scrum

2.1.3.2 Las Historias de Usuario

El concepto de historia de usuario tiene sus orígenes en la metodología “eXtremeProgramming” (programación extrema). Esta metodología fue

creada por Kent Beck y descrita por primera vez en 1999 en su libro “eXtreme Programming Explained” [11].

En las metodologías ágiles la descripción de las necesidades se realiza a partir de las historias de usuario que son principalmente lo que el cliente o el usuario quiere que se implemente; es decir, son una descripción breve, de una funcionalidad del producto tal y como la percibe el usuario. Es por ello que son un elemento clave para asegurar el éxito de todo proyecto ágil.

Cabe destacar que no son una especificación de requisitos del producto final. Para implementar una historia de usuario se necesita mucha más información que la recogida en la propia historia, por lo que el objetivo de una historia de usuario no es contener toda la información necesaria para poder implementarla, eso sería una especificación de requisitos, el objetivo de la historia de usuario es apuntar a alto nivel lo que se quiere hacer.

Una historia de usuario:

- Representa una funcionalidad
- Aporta un claro valor al usuario
- No es una especificación de requisitos
- Es pequeña y memorizable

Una historia de usuario se compone de 3 partes:

- Medio físico (plantilla)
- Conversación (discusión que provoca)
- Confirmación (pruebas y verificación)

A continuación se muestra una propuesta para la implementación de la Plantilla para recoger las Historias de Usuario.

Proyecto: XXXX **ID: XXX** **FECHA: dd/mm/yyyy**

Funcionalidad:

La funcionalidad que quiero añadir al producto finales; _____.

Descripción:

La funcionalidad va dirigida a; _____
y aporta los siguientes beneficios; _____.

Dependencias:

Proyecto: _____
ID: _____

Valor: 1 a 5 **Esfuerzo:** 1 a 5

Ilustración 7: Plantilla para recoger las Historias de Usuario (1)

Proyecto: XXXX **ID: XXX**

Tarea 1:

_____ **% Esfuerzo:** 0 a 100

Tarea 2:

_____ **% Esfuerzo:** 0 a 100

Tarea 3:

_____ **% Esfuerzo:** 0 a 100

Tarea 4:

_____ **% Esfuerzo:** 0 a 100

Ilustración 8: Plantilla para recoger las Historias de Usuario (2)

La plantilla va dirigida tanto para el Product Owner (campos sombreados en azul) como para el Equipo Scrum (campos sombreados en naranja).

En la plantilla se recogen dos importantes métricas que son el Valor y el Esfuerzo (ambas métricas serán comentadas posteriormente).

En la Figura 6 el Valor hace referencia a la importancia que el Product Owner otorga a la Historia de Usuario. El Esfuerzo por su parte referencia a la carga de trabajo que supone la Historia de Usuario para el Equipo Scrum. Por tanto observamos como el Valor es asignado por el Product Owner y en cambio el Esfuerzo es asignado por el Equipo Scrum.

En la Figura 7 se detallan las tareas de la Historia de Usuario y el % de Esfuerzo, este porcentaje reparte la puntuación del Esfuerzo de la Historia de Usuario otorgada en la primera hoja entre las Tareas, así la suma de estas casillas debe de sumar 100.

2.1.3.3 El ciclo de trabajo en Scrum

Tal y como hemos comentado en apartados anteriores una de las bases de los proyectos ágiles es el desarrollo mediante las iteraciones incrementales. En Scrum a cada iteración se le denomina Sprint.

Un Sprint es un bloque de tiempo normalmente definido entre 1 y 4 semanas durante el cual se crea un incremento del producto. Es conveniente que los Sprints tengan toda la misma duración durante el transcurso de un mismo proyecto.

Un Sprint es un contenedor en dónde se producen una serie de eventos. Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective), todos estos eventos serán detallados en el siguiente apartado (2.1.3.4 Las reuniones).

Los Sprints se usan para lograr algo, cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante, además cabe destacar que según la definición de Scrum, una vez que comienza un Sprint, su duración es fija y no puede acortarse o alargarse.

Tal y como se indica en La Guía Scrum [8]:

“Los Sprints están limitados a un mes calendario. Cuando el horizonte de un Sprint es demasiado grande la definición de lo que se está construyendo podría cambiar, la complejidad podría elevarse y el riesgo podría aumentar”.

Un Sprint se alimenta del Sprint Backlog que a su vez depende del Product Backlog:

- Product Backlog

La Pila de Producto o Product Backlog representa el conjunto de funcionalidades descritas por el Product Owner a través de las Historias de Usuario y que deben ser implementadas a lo largo de los diferentes Sprints.

El Product Backlog contiene el valor y el esfuerzo que supone el proyecto, no existe un criterio preestablecido en Scrum para ordenar las Historias de Usuario que forman el Product Backlog aunque el más aceptado es partir del valor que aporta al negocio la implementación de la historia de usuario. El responsable de ordenar el Product Backlog es el Product Owner, aunque también puede ser ayudado o recibir asesoramiento de otros roles como, por ejemplo, el Scrum Master y el equipo de desarrollo. Es muy importante entender el Product Backlog como un elemento el cual siempre está abierto a recibir nuevas Historias de Usuario, por tanto cada vez que entren nuevas Historias de Usuario deberán ser priorizadas junto con el conjunto de las Historias de Usuarios pendientes en ese momento.

- Sprint Backlog

Es la pila de elementos pendientes del Sprint. Formado por los elementos del Product Backlog que se entregarán una vez finalice el Sprint. Dicho de otra manera, el Sprint Backlog corresponde con los elementos del Product Backlog que se planifican para el Sprint, más el plan para terminarlos.

En la siguiente figura (Figura 9), se muestra el ciclo completo de trabajo en Scrum. Se puede apreciar como partiendo del Product Backlog, (formado por un conjunto de Historias de Usuario), y pasando por los diferentes Sprints (a través de los Sprints Backlogs), se consigue ir incrementado la funcionalidad y el valor del producto deseado.

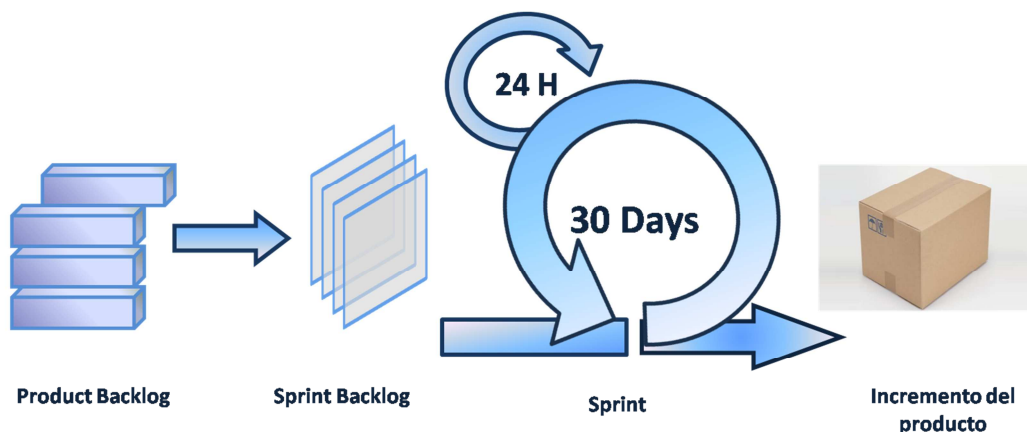


Ilustración 9: El ciclo de trabajo de Scrum

2.1.3.4 Las reuniones

Tal y como hemos comentado en el anterior capítulo los Sprints están formados por una serie de eventos, en concreto por 4 eventos:

- Sprint Planning Meeting

El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación del Sprint. Tal y como indica La Guía Scrum [8], el Sprint Planning Meeting debe servir para responder a las siguientes preguntas:

- ¿Qué puede entregarse en el Incremento resultante del Sprint que comienza?
- ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?

Esta reunión da lugar al Sprint Backlog. En la reunión participan todos los roles. El Product Owner presenta el conjunto de historias de usuario del Product Backlog y el equipo de desarrollo selecciona las historias de usuario sobre las que se trabajará.

Además en esta reunión también se crea el Sprint Goal (Objetivo del Sprint), que representa una meta establecida para el Sprint. El Sprint Goal proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento.

- Daily Scrum Meeting

El Scrum Diario es una reunión de no más de 15 minutos para sincronizar las actividades del Equipo de y para crear un plan para las siguientes 24 horas. Participan el Equipo de Desarrollo y el Scrum Master. En esta reunión cada miembro del equipo presenta lo que hizo el día anterior, lo que va a hacer hoy y los impedimentos que se ha encontrado.

La finalidad de la reunión es encontrar y solucionar posibles amenazas que impidan la consecución del objetivo del Sprint por lo que el Sprint Daily Meeting optimiza las posibilidades de que el Equipo de Desarrollo cumpla el Objetivo del Sprint.

- Sprint Review

La reunión de revisión del Sprint se realiza al final del Sprint y como su nombre indica sirve para inspeccionar el Incremento producido. Participan el equipo de desarrollo, el Scrum Master y el Product Owner. El objetivo de la reunión es presentarle al Product Owner el trabajo realizado para poder validar con él la utilidad del mismo.

- Sprint Retrospective

No forma parte de todos los Sprints aunque en caso que se produzca se realiza al final del Sprint. La reunión sirve para que tanto el Equipo como el Scrum Master intercambien impresiones sobre el trabajo que se ha realizado con el objetivo de mejorar los aspectos que no han ido del todo bien de cara a las siguientes iteraciones.

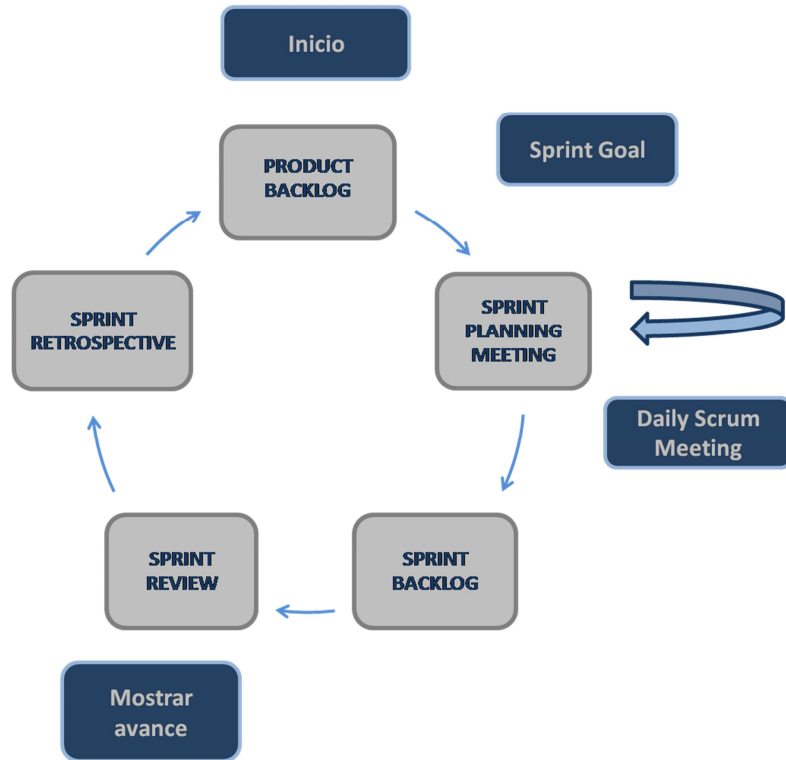


Ilustración 10: Las reuniones en Scrum

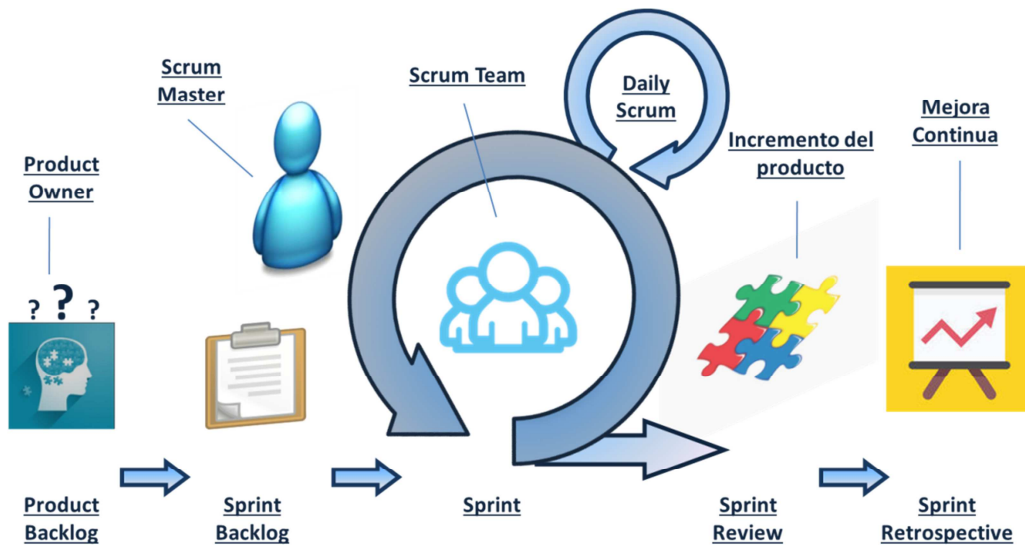


Ilustración 11: Perspectiva completa del ciclo de trabajo de Scrum

2.1.4 Kanban

Kanban en japonés significa tarjeta o tablero, es una técnica creada en Toyota para el control del avance de la producción. Actualmente y mediante los Tableros Kanban (que son la representación visual de la técnica) se está aplicando al desarrollo software.

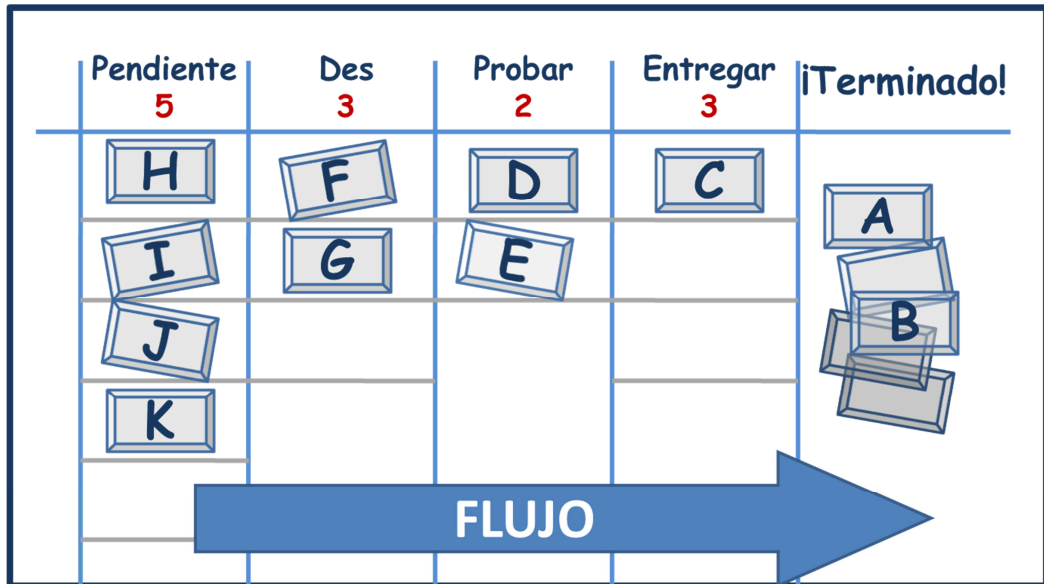


Ilustración 12: Tablero Kanban

Las tres reglas de este método son las siguientes:

- Visualizar el flujo de trabajo

Dividir el trabajo en bloques pequeños representándolos en el tablero a través de pegatinas o tarjetas. Hay que situar las tarjetas en las columnas correspondientes para visualizar el flujo de trabajo.

- Limitar el WIP

El WIP (Work In Progress) o trabajo en curso, debe de tener asignado límites concretos referentes a cuántos elementos pueden estar en progreso en cada estado (columna del tablero) del flujo de trabajo.

- Medir el Lead Time

El Lead Time es tiempo medio transcurrido para completar un elemento o tarjeta. Kanban trata de optimizar el proceso para que el Lead Time sea tan pequeño y predecible como sea posible.

2.2 Agilidad en proyectos de B.I.

La Inteligencia de Negocio se está convirtiendo en un importante marco de referencia dentro de las T.I que puede ayudar a las organizaciones a gestionar, desarrollar y comunicar sus activos intangibles tales como la información y el conocimiento. Por lo tanto, se puede considerar como un marco imprescindible en la era actual de la economía basada en el conocimiento.

El objetivo de cualquier solución de BI es tener acceso a datos de múltiples fuentes, transformar estos datos en información y luego en conocimiento. El objetivo principal de cualquier solución de BI es mejorar las capacidades de toma de decisiones de la organización.

Las aplicaciones de business intelligence se caracterizan principalmente por la flexibilidad y capacidad de adaptación, factores con los que las aplicaciones tradicionales no son capaces de tratar [4]. El modelado de procesos tradicional requiere de una gran cantidad de documentos e informes, y esto hace que la metodología tradicional no pueda cumplir con los requisitos dinámicos de cambio a la alta velocidad que el entorno demanda.

La principal confusión acerca de los proyectos de B.I se presenta cuando el business intelligence sólo se considera un producto. La inteligencia de negocio es a la vez un proceso y un producto. Como proceso, el B.I es un conjunto de métodos y actividades que las organizaciones deben realizar para desarrollar la información y conocimiento útil (o "inteligencia"), para sobrevivir y prosperar en una economía basada en TI global. Como producto, B.I es el sistema de información que permite a las organizaciones predecir su comportamiento y tomar decisiones sobre su futuro [4].

Según el estudio sobre los diferentes enfoques metodológicos utilizados en proyectos de inteligencia de negocios, llevado a cabo por los investigadores; J. Fernández (Technical University of Catalonia, Spain), E. Mayol (Technical University of Catalonia, Spain) y J.A. Pastor (Universitat Oberta de Catalunya, Spain) [4], muchos de estos enfoques presentan debilidades ya que no se definen específicamente para proyectos de B.I, y por lo tanto no se ajustan a las características de los proyectos o a las necesidades de los usuarios. Según sugieren, esto puede ser la causa principal que explica el por qué no existe una metodología de BI ampliamente aceptada. Tal y como explican, a pesar que encontrar la metodología idónea para los proyectos de B.I puede no ser una tarea sencilla, el enfoque ágil para abarcar este tipo de proyectos parece la mejor alternativa, y finalmente concluyen afirmando que:

“En base a nuestro análisis, consideramos que las metodologías de desarrollo de proyectos de BI deben seguir un enfoque ágil”.

Ken Collier [1] consultor y profesional de D.W / B.I apunta tres ideas obtenidas fruto de sus experiencias; la construcción de sistemas de D.W / B.I es una tarea difícil, los proyectos de desarrollo de D.W / B.I fallan muy a menudo, y es mejor fallar rápido y adaptarse antes que fallar tarde después de haber gastado gran parte del presupuesto.

Collier [1] prosigue afirmando que existen evidencias de que la agilidad aumenta las tasas de éxito en los proyectos de B.I, para ello se apoya en los estudios realizados por Scott Ambler, líder en el desarrollo de bases de datos y modelado ágil.

Ambler ha llevado a cabo numerosos estudios sobre el desarrollo ágil para cuantificar el impacto y la eficacia de estos métodos. Comenzando en 2007, Ambler ha llevado a cabo tres encuestas específicamente relacionados con la tasa de éxito de los proyectos de B.I enfocados con metodologías ágiles y metodologías tradicionales. La encuesta realizada en 2007 concluyó que sólo el 63 por ciento de los proyectos tradicionales B.I tuvieron éxito, mientras que los proyectos ágiles experimentaron una tasa de éxito del 72 por ciento. La encuesta de 2008 se centró en cuatro criterios de éxito: calidad, retorno de la inversión, funcionalidad y la programación. En las cuatro áreas los métodos ágiles superaron significativamente los enfoques de desarrollo tradicionales. La encuesta de 2010 continuó mostrando que los métodos ágiles en B.I producen mejores resultados.

Cabe destacar la distinción que Collier [1] realiza entre las métricas que tradicionalmente validaban el éxito de un proyecto como eran; la duración temporal, el alcance del presupuesto y el cumplimiento de las especificaciones, frente a las métricas que según él deberían de validar el éxito de un proyecto gestionado de manera ágil que son; el valor aportado al cliente y su nivel de satisfacción.

El desarrollo ágil también en B.I está dirigido principalmente a la entrega de valor a los clientes, ésta es la máxima prioridad. Las medidas de progreso y éxito deben centrarse en la entrega de valor en lugar de pretender seguir evaluando las métricas tradicionales.

2.2.1 Necesidades de las áreas de B.I.

Cualquier solución de B.I se puede dividir en 3 capas [4]:

1. Capa de datos
2. Capa de analítica
3. Capa de visualización

Tal y como hemos comentado en el apartado anterior las áreas de B.I trabajan con el objetivo de añadir valor al proceso de toma de decisiones que el Negocio necesita, de hecho las 3 capas presentadas están orientadas a tal finalidad. También anteriormente se ha confirmado

(según la opinión de distintos profesionales y expertos) la idoneidad de las técnicas ágiles aplicadas a esta área de conocimiento.

Cabría preguntarse si los procesos ágiles presentados en este Trabajo son útiles y tienen aplicabilidad directa a los procesos de B.I o si por el contrario se podrían modificar orientándolos a las necesidades de esta disciplina. Una de las principales motivaciones de este Trabajo nace de la creencia que existe una oportunidad de mejorar los métodos ágiles para ajustarlos a las necesidades de los procesos del business intelligence. .

Las necesidades de adaptabilidad, flexibilidad y dinamismo son una característica propia de los proyectos de B.I, estas características impactan sobre cada una de las 3 capas mencionadas; datos, análisis y visualización.

Podemos encontrar proyectos de B.I situados en la primera capa (capa de datos), cuyo objetivo por ejemplo sea la construcción de una arquitectura para el almacenamiento de información no estructurada, con conexión al datawarehouse sí estructurado de una compañía. Posiblemente el alcance de este proyecto sea muy amplio y el método a aplicar sea muy similar a los métodos analizados en los apartados 2.1.3 Scrum y 2.1.4 Kanban.

Por el contrario podemos encontrar proyectos situados en la capa 3 (capa de visualización), cuyo objetivo sea conseguir más presupuesto en el comité de dirección para ampliar la red comercial en una determinada provincia. Posiblemente en este caso el alcance aunque igual de importante, no sea tan ambicioso en tiempo o en recursos. Quizás éste proyecto además tengas mayores necesidades de adaptabilidad, flexibilidad y dinamismo, pues el comité de dirección posiblemente se adelante, o el enfoque del discurso que queremos presentarle al CEO no esté definido hasta la víspera anterior a la reunión.

Por tanto vemos como los proyectos de B.I tienen una amplia elasticidad en cuanto a necesidades, y la propuesta de este Trabajo es dotar de mayor flexibilidad y capacidad de respuesta si cabe a los métodos ágiles.

2.2.2 Adaptación del método ágil a los proyectos de B.I.

Se propone la utilización conjunta de Scrum y Kanban, juntando la visión de los componentes del ciclo de Scrum; Product Backlog y Sprint Backlog, con los flujos de Kanban; To Do, Doing, Done y su elemento de control WIP (work in progress). En este caso el estado "To Do" corresponde con el Sprint Backlog.

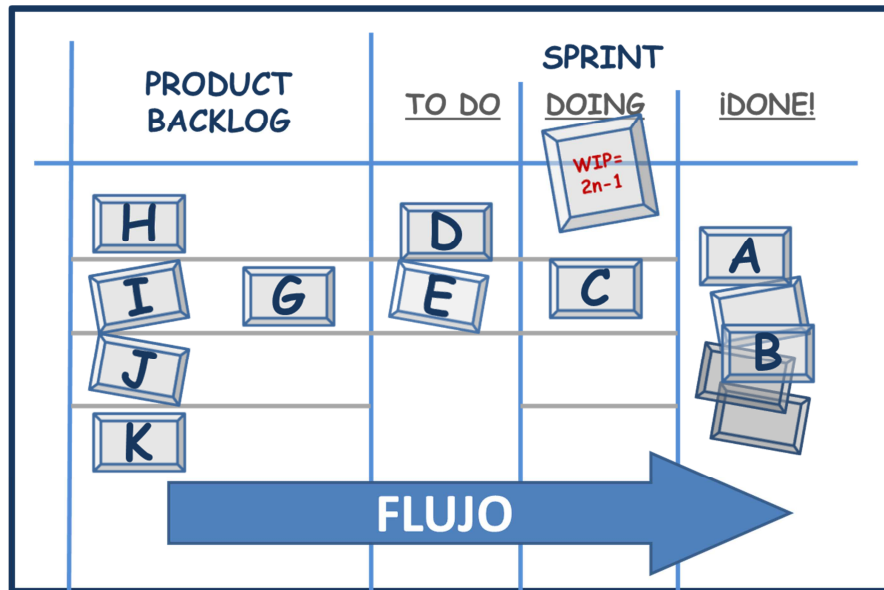


Ilustración 13: Tablero Kanban con elementos de Scrum

Modificaciones a Scrum:

- Disminución de la duración de los Sprints

Scrum define los Sprints como ventanas temporales de entre 1 y 4 semanas (5 i 20 días), la aportación que se realiza propone permitir Sprints de 2 días.

Un Sprint de 2 días permite realizar una reunión de planificación del Sprint en el día 1, una Daily Meeting el día 2 en su inicio, y un Sprint Review al finalizar también el segundo día. La utilidad de esta breve ventana temporal es ajustarse a proyectos con un grado de urgencia muy elevado y un margen de incertidumbre también muy importante. Serian proyectos similares a los mencionados en el apartado anterior (ejemplo proyecto capa 3). Como veremos a continuación la eliminación del Sprint Retrospective es de utilidad para este tipo de Sprints con cortas ventanas temporales en las que esta reunión pudiese alargar en exceso la jornada laboral del Equipo, sobre todo en el segundo día del Sprint.

- Habilitar la modificación del Sprint Backlog aun cuando el Sprint esté en curso

En Scrum una vez empezado el Sprint no se puede modificar la pila de trabajo pendiente (Sprint Backlog), la propuesta es eliminar esta restricción habilitando la modificación de la pila de pendientes, permitiendo añadir y quitar Historias de Usuario siempre que no se haya empezado a trabajar en las mismas.

Se ponderan más si caben las necesidades del Negocio frente a la planificación del Equipo de Desarrollo. Pongamos por ejemplo

que el Equipo ha decidido implementar las Historias de Usuario 1,2 y 3. Empieza por la primera y la termina, luego sigue con la segunda y estando ésta en proceso, el Product Owner manifiesta la necesidad de priorizar la Historia de Usuario 4 (que no estaba en el Sprint Backlog) frente a la Historia de Usuario 3. Según la definición de Scrum, no sería posible incorporar la Historia de Usuario 4 al actual Sprint, ya que se habría planificado los recursos necesarios para abordar las Historias de Usuario 1,2 y 3 únicamente. El hecho de extraer del Sprint Backlog la Historia de Usuario 3 e incorporar la 4, puede suponer la no culminación de los objetivos iniciales del Sprint, no obstante, de cara a la aportación de valor al Negocio, empezar cuanto antes la Historia de Usuario 4 (y terminarla en el siguiente Sprint), es mucho más interesante.

- Posibilitar el llevar a cabo el Sprint Review de manera offline

La reunión de Sprint Review se planifica en Scrum como un punto de contacto presencial entre el Product Owner, el Scrum Master y el Equipo de Desarrollo, se propone habilitar la posibilidad de llevar a cabo esta reunión de manera indirecta a través del e-mail.

Pese a la voluntad de Scrum de crear un diálogo directo entre los distintos actores del proceso ágil, en los proyectos de B.I las necesidades y urgencias del Negocio impiden que incluso un Product Owner representante del Cliente, pueda asistir presencialmente u online a todas las reuniones de Sprint Review, además esta situación se agrava cuando acortamos la duración de los Sprints. Por tanto la alternativa planteada pretende acordar desde el inicio del proyecto la manera en que las reuniones de revisión se llevaran a cabo (presenciales, online u offline), y así agilizar el tiempo efectivo de trabajo en los Sprints.

- Eliminación del Sprint Retrospective

Scrum define una reunión para hacer balance sobre los aciertos y desaciertos, puntos positivos e impedimentos surgidos durante el transcurso del Sprint, la propuesta es eliminar esta reunión y tratar sus contenidos en los Daily Meetings, ampliando la duración de éstos de 15 a 30 minutos.

Al acortar las ventanas temporales de los Sprints (hasta 2 días), la reunión de retrospectiva aumenta su peso relativo frente al total del tiempo que dura un Sprint, esto reduce el tiempo efectivo de trabajo en cada iteración. Además los Daily Meetings ya actúan como correctores de evolución de los Sprints. El planteamiento es incorporar la retrospectiva diaria y eliminar la retrospectiva global puesto que la visión retrospectiva global se puede conseguir también a partir de las diarias.

Modificaciones a Kanban:

- Medición del WIP por tareas en lugar de por funcionalidades

En Kanban las tarjetas del tablero contienen las funcionalidades a implementar en el producto final, se propone que el WIP en lugar de referenciar a las funcionalidades del producto, referencie a las tareas que son necesarias acometer para conseguir las funcionalidades.

De este modo se persigue aumentar el control sobre el trabajo en progreso y aumentar la implicación del Equipo en todas las partes del proyecto, a la vez que se fomenta un mayor conocimiento colectivo de las funcionalidades a implementar.

2.3 Caso de implantación del método ágil en el departamento de B.I de una empresa española del sector de las TIC

A continuación se describirá el caso real de implantación del método ágil propuesto en este Trabajo, sobre un departamento de B.I.

El contexto nos sitúa en una unidad territorial (Comunidad Valenciana, Murcia y Baleares) de Inteligencia de Negocio de la empresa Telefónica de España. El experimento se ha llevado a cabo durante los meses de abril, mayo y junio de 2016. Esta unidad de B.I está centrada en proyectos de capa 2 y 3 (análisis y visualización), y se caracteriza por estar muy pegada a la Dirección.

El proyecto sobre el cual se va a implementar el proceso ágil es:

- Diseño e implementación del Informe de Seguimiento de Actividad Comercial, de la red de Ventas de Telefónica de España, en la Comunidad Valenciana, Murcia y Baleares.

La red de Ventas se compone de diversos Canales; Tiendas, Atención Telefónica y Canal Online. El objetivo del proyecto es proveer de información diaria sobre el estado de las Ventas por provincia y Canal a la Dirección.

Es un proyecto importante a la vez que urgente, pues por necesidades del Negocio se ha decidido rediseñar de nuevo el actual informe vigente. La repercusión sobre el Negocio es alta, ya que de este informe y de su seguimiento diario se toman decisiones con impacto directo en la cuenta de resultados, un ejemplo son las decisiones sobre la prórroga o eliminación de las promociones comerciales, las cuales están influenciadas por la evolución diaria de la actividad comercial. Por tanto y hasta que no esté operativa una mínima versión funcional del informe, el seguimiento diario de la actividad tendrá que ser llevado a cabo por otras

vías, mermando la eficiencia y productividad de los Equipos y de los Directivos.

El proyecto implica trabajar en las 3 capas que forman los procesos de B.I; datos, análisis y visualización. Aunque no es necesario montar una arquitectura de almacenamiento de datos dedicada para proyecto (se utilizará el datawarehouse corporativo), si será necesario trabajar en el modelado de las pertinentes reglas de Negocio, que transformen el dato del datawarehouse en información preparada para ser analizada, por tanto tendremos una parte de nuestro trabajo en la primera capa del proceso de B.I (capa de datos). También nos tocará trabajar en la segunda capa del proceso (capa de análisis) ya que la información deberá ser analizada, procesada y preparada para atender las demandas del Product Owner. Por último el trabajo en la tercera capa (visualización) se centrará en encontrar la representación más oportuna de la información solicitada.

En este contexto se decide implantar el método ágil propuesto en el capítulo anterior con las siguientes características:

- Sprints de 5 días
- WIP de $2n-1$ tareas, siendo n el número de colaboradores trabajando en el Sprint

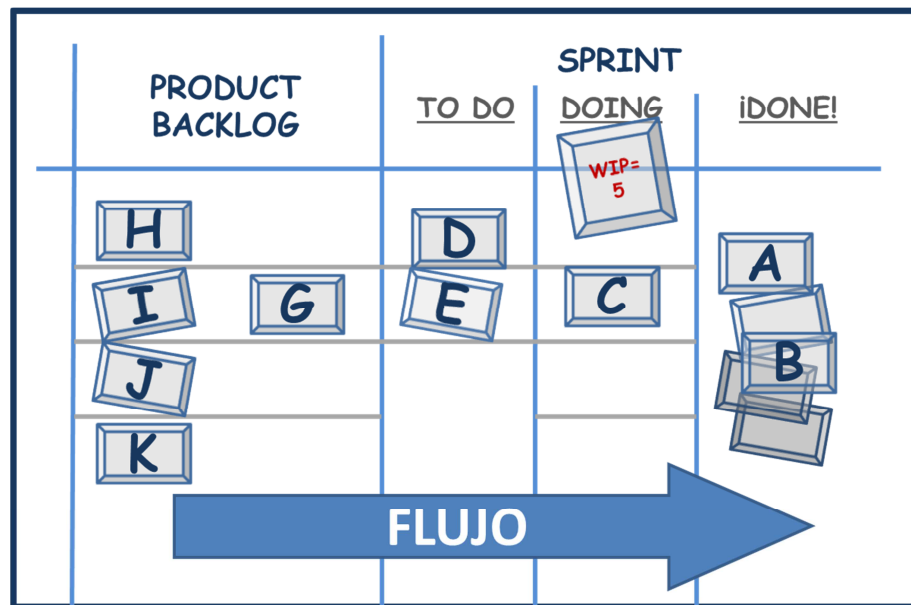


Ilustración 14: Tablero ágil implementado

Se ha decidido fijar la ventana de los Sprints a 5 días laborables, debido a que el proyecto toca las 3 capas mencionadas, y se pretende comenzar desde cero sin ninguna referencia previa considerada como válida, por tanto el Scrum Master y el Equipo han decidido que esta duración de 5 días será la más adecuada para asegurar que en cada

Sprint se consigue aportar un incremento de la funcionalidad del producto.

El WIP como vemos afecta al estado de "DOING" y limitará en la mayor parte del proyecto a 5 las tareas que pueden estar desarrollándose en paralelo. Un WIP de 5 ($WIP=2n-1=5$) implica que el Equipo de Desarrollo estará formado por 3 integrantes ($n=3$). La elección del WIP de tareas a $2n-1$ integrantes fomenta que al menos siempre exista un colaborador con una tarea menos en progreso en comparación con los demás integrantes del Equipo, y por tanto pueda ayudar a los demás a terminar sus tareas, esto contribuye a que el Equipo sea consciente que antes de empezar nuevas tareas o funcionalidades es necesario cerrar las que están en proceso.

Las reuniones planificadas para los Sprints son las siguientes:

1. Sprint Planning Meeting; planificada para el día 1 de cada Sprint y de duración aproximada de 1 hora.
2. Sprint Daily Meeting; planificada para todos los días del Sprint y de duración aproximada de no más de 30 minutos.
3. Sprint Review; planificada para el último día del Sprint aunque puede variar en función de si el Incremento se consigue con anterioridad a la finalización del Sprint. No se estima duración ya que esta reunión puede no producirse presencialmente ni de manera simultánea (modo offline).

Las roles definidos son los siguientes:

4. Product Owner; mando intermedio responsable de la interlocución con la Dirección. Conoce perfectamente el Negocio.
5. Scrum Master; en este caso he ejercido personalmente como responsable de esta figura. Mi principal aportación es fomentar que el proceso ágil sea implementado tal y como se ha definido.
6. Equipo Scrum; formado en la mayor parte del proyecto por 3 personas entre las que también formo parte.
7. Stakeholders y Usuarios; la Dirección en este caso ejerce ambos roles, por una parte son los patrocinadores del proyecto y también serán los futuros usuarios del producto.

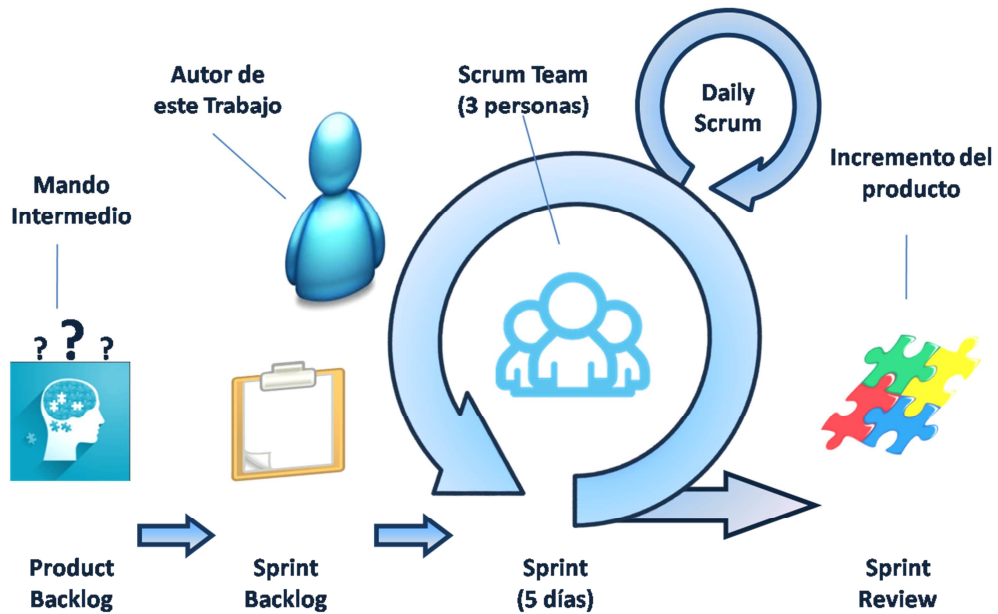


Ilustración 15: Proceso ágil seguido

2.4 Cuadro de Mando para la gestión ágil

Además de la adaptación de los métodos ágiles a las necesidades de las unidades de B.I, otro de los objetivos de este Trabajo es la aportación de una herramienta útil para la gestión de proyectos dentro del contexto de la agilidad.

Por ello se ha creado un cuadro de mando (CdM) que pretende ser una herramienta de utilidad para la planificación y el seguimiento de diferentes métricas del proyecto ágil. Pensado para el Product Owner, el Equipo Scrum, y el Scrum Máster, será éste último el poseedor y responsable de su actualización.

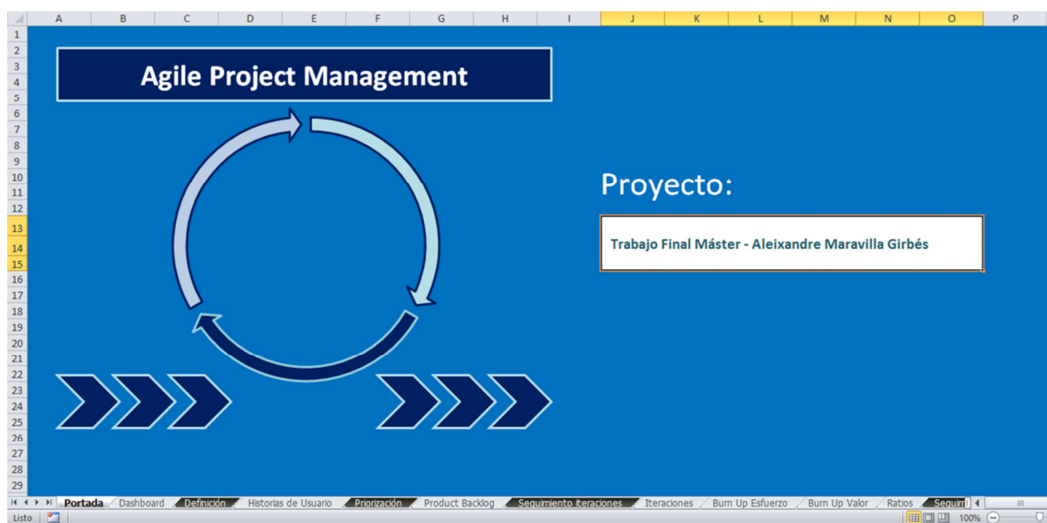


Ilustración 16: Portada del Cuadro de Mando para la gestión ágil

Los objetivos que persigue el Cuadro de Mando son:

1. Maximizar la productividad (Valor / Esfuerzo) en las primeras iteraciones (Sprints) del proyecto
2. Alertar del riesgo a posibles desviaciones sobre lo planificado
3. Estimar la fecha prevista de finalización del proyecto
4. Apoyar a la toma de decisión de cese temprano del proyecto

Para alcanzar los objetivos definidos el CdM propone:

- Un mecanismo de priorización de las Historias de Usuario que forman el Product Backlog
- Un sistema que alerta de las tareas críticas que requieren de mayor atención en su desarrollo
- Un bloque dedicado al seguimiento de la evolución del proyecto (Sprints)
- Un bloque dedicado al seguimiento de la evolución del Sprint en curso

2.4.1 Estructura

El Cuadro de Mando admite un total de 1000 Historias de Usuario permitiendo que cada Historia de Usuario se pueda descomponer en un máximo de hasta 10 tareas a realizar. Las celdas sombreadas en blanco son celdas reservadas para la entrada de datos (inputs), el resto de celdas son celdas o bien destinadas a la salida de datos (outputs), o bien no tienen ninguna función asignada.

El cuadro de mando se divide en 6 bloques:

- Bloque 1; Dashboard

Bloque pensado para la rápida comprensión del estado en el que se encuentra el proyecto. Contiene información sobre la fecha estimada de finalización del proyecto, la evolución actual y prevista de la productividad y sobre el estado del Product Backlog.

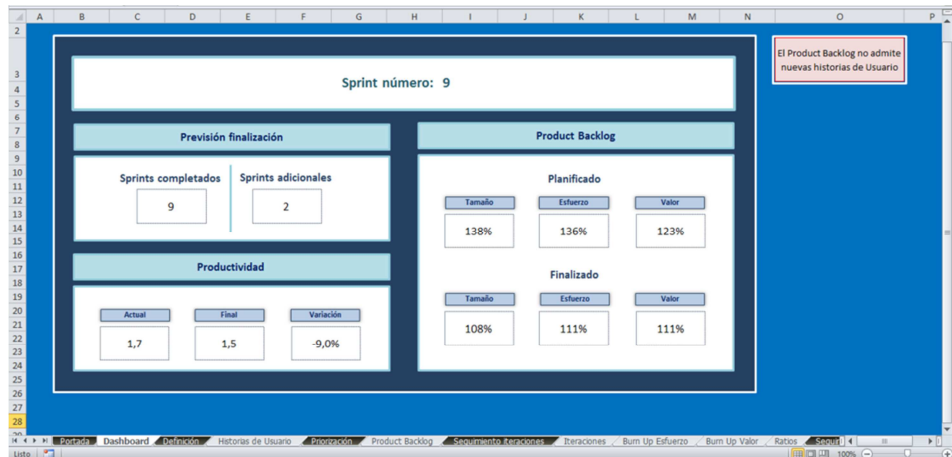


Ilustración 17: Bloque de Dashboard

- **Bloque 2; Definición de Historias de Usuario**

En este bloque se definen las Historias de Usuario y las tareas necesarias para desarrollarlas. Cada Historia de Usuario debe llevar asociada una ponderación del Valor y del Esfuerzo que suponen (estas métricas serán definidas en el siguiente apartado “2.4.2 Métricas”). Además cada una de las tareas en las que se desglosa la Historia de Usuario debe llevar asociado un valor porcentual % que represente la parte del Esfuerzo relativa sobre el total asignado.

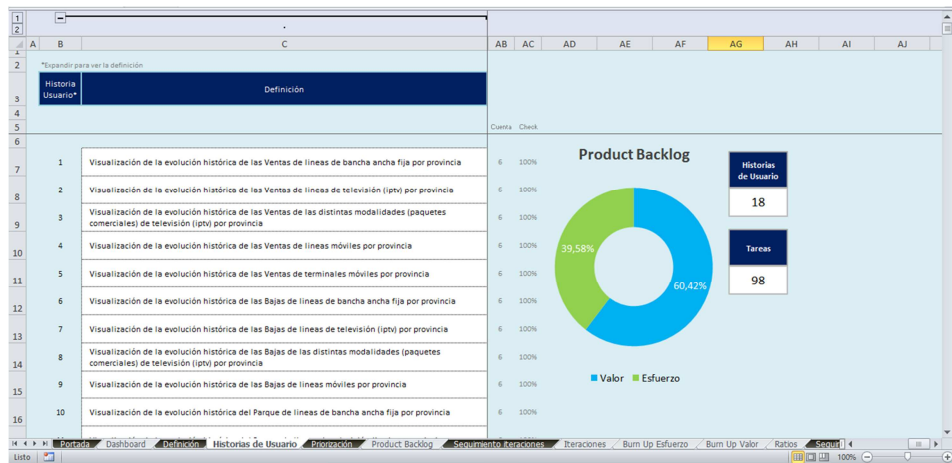


Ilustración 18: Bloque de definición de las Historias de Usuario

- **Bloque 3; Priorización y alerta del Product Backlog**

El bloque de priorización del Product Backlog acomete dos de los 4 principales objetivos del Cuadro de Mando, la priorización de las Historias de Usuario y la alerta de las tareas críticas.

El objetivo de la priorización de las Historias de Usuario es la maximización de la productividad (Valor / Esfuerzo) en fases tempranas del proyecto. Con ello se consigue que en caso de tener que abandonar el proyecto por motivos por ejemplos

presupuestarios, el producto creado tenga el máximo valor posible para el Product Owner.

Este bloque además alerta sobre la criticidad de determinadas tareas todavía por ejecutar. La finalidad de las alertas es la detección de los posibles riesgos que amenacen la ejecución del proyecto, para de este modo poder reaccionar con antelación, por ejemplo, dedicando más recursos (colaboradores) en el siguiente Sprint.

| Product Backlog | | | | | | | | | | | | | | Resumen | | | | | | | | | | |
|------------------|-------|----------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|-------------------|--------|--------------|------------------|-------|-----------|-------------|-----------------|----------|--------------|----------------|--------------------|------------|
| Historia Usuario | Valor | Esfuerzo | Esfuerzo Tarea 1 | Esfuerzo Tarea 2 | Esfuerzo Tarea 3 | Esfuerzo Tarea 4 | Esfuerzo Tarea 5 | Esfuerzo Tarea 6 | Esfuerzo Tarea 7 | Esfuerzo Tarea 8 | Esfuerzo Tarea 9 | Esfuerzo Tarea 10 | Total | Priorización | Historia Usuario | Valor | Valor (%) | Valor Acum. | Valor Acum. (%) | Esfuerzo | Esfuerzo (%) | Esfuerzo Acum. | Esfuerzo Acum. (%) | Criticidad |
| 1 | 300 | 200 | 30.0% | 30.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 1 | 12 | 500 | 8.6% | 500 | 8.6% | 100 | 2.6% | 100 | 2.6% | No |
| 2 | 400 | 200 | 30.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 2 | 10 | 300 | 5.2% | 800 | 13.8% | 100 | 2.6% | 200 | 5.3% | No |
| 3 | 500 | 400 | 50.0% | 10.0% | 5.0% | 10.0% | 5.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 3 | 2 | 400 | 6.9% | 1,200 | 20.7% | 200 | 5.3% | 400 | 10.3% | No |
| 4 | 200 | 200 | 30.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 4 | 11 | 400 | 6.9% | 1,600 | 27.6% | 200 | 5.3% | 600 | 15.8% | No |
| 5 | 100 | 200 | 40.0% | 15.0% | 5.0% | 10.0% | 5.0% | 25.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 5 | 13 | 200 | 3.4% | 1,800 | 31.0% | 100 | 2.6% | 700 | 18.4% | No |
| 6 | 300 | 300 | 30.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 6 | 1 | 300 | 5.2% | 2,100 | 36.2% | 200 | 5.3% | 900 | 23.7% | No |
| 7 | 400 | 300 | 30.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 7 | 7 | 400 | 6.9% | 2,500 | 43.1% | 300 | 7.9% | 1,200 | 31.6% | No |
| 8 | 500 | 500 | 50.0% | 10.0% | 5.0% | 10.0% | 5.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 8 | 3 | 500 | 8.6% | 3,000 | 51.7% | 400 | 10.5% | 1,600 | 42.1% | No |
| 9 | 200 | 300 | 30.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 9 | 4 | 200 | 3.4% | 3,200 | 55.2% | 200 | 5.3% | 1,800 | 47.4% | No |
| 10 | 300 | 100 | 30.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 10 | 6 | 300 | 5.2% | 3,500 | 60.3% | 300 | 7.9% | 2,100 | 55.3% | No |
| 11 | 400 | 200 | 30.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 11 | 8 | 500 | 8.6% | 4,000 | 69.0% | 500 | 13.2% | 2,500 | 68.4% | No |
| 12 | 500 | 100 | 50.0% | 10.0% | 5.0% | 10.0% | 5.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 12 | 15 | 400 | 6.9% | 4,400 | 75.9% | 100 | 2.6% | 2,700 | 71.1% | No |
| 13 | 200 | 100 | 30.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 13 | 14 | 300 | 5.2% | 4,700 | 81.0% | 100 | 2.6% | 2,800 | 73.7% | No |
| 14 | 300 | 100 | 15.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 14 | 16 | 300 | 5.2% | 5,000 | 86.2% | 300 | 7.9% | 3,100 | 81.6% | Si |
| 15 | 400 | 100 | 15.0% | 25.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 15 | 17 | 200 | 3.4% | 5,400 | 93.1% | 100 | 2.6% | 3,200 | 84.2% | No |
| 16 | 500 | 300 | 25.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 16 | 18 | 100 | 1.7% | 5,500 | 94.8% | 100 | 2.6% | 3,300 | 86.8% | No |
| 17 | 200 | 100 | 15.0% | 25.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 17 | 9 | 200 | 3.4% | 5,700 | 98.3% | 300 | 7.9% | 3,500 | 94.7% | Si |
| 18 | 100 | 100 | 15.0% | 20.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 10.0% | 100.0% | 18 | 5 | 100 | 1.7% | 5,800 | 100.0% | 200 | 5.3% | 3,800 | 100.0% | No |

Ilustración 19: Bloque de priorización del Producto Backlog

- Bloque 4; Seguimiento del Proyecto

El cuarto bloque sirve para planificar y realizar un seguimiento de la evolución de los Sprints (iteraciones) del proyecto. Cumple 2 de las 4 principales funciones del CdM: estimar la fecha prevista de finalización del proyecto y apoyar a la toma de decisión de cese temprano del proyecto.

Está compuesto de 4 secciones:

- Iteraciones; entrada de datos para la planificación de los Sprints.
- Burn Up Esfuerzo; evolución del Esfuerzo a lo largo del proyecto.
- Burn Up Valor; evolución del Valor a lo largo del proyecto.
- Ratios; Productividad, Velocidad, Factor Foco y Cycle Time.

| Planificación de los Sprints | | | | | | | | | | | | | | | | |
|------------------------------|--------------------------------|------------------------|------------------------|-----------------|--------------------------|---------------------------------------|-----------------------------|--------------------------------|---------------------------------------|-----------------------------|--|--------------------------------|-----------------|----------|--------------------------|-----------------------------|
| Sprint | Fecha Inicio Sprint (dd-mm-aa) | Tamaño Product Backlog | Días laborables Sprint | Personas Equipo | Jornada efectiva (horas) | Velocidad planificada (personas/hora) | Valor Total Product Backlog | Esfuerzo Total Product Backlog | Historia de Usuario finalizada (100%) | Prioridad finalizada (100%) | Historia Usuario NO finalizada (NO 100%) | Prioridad finalizada (NO 100%) | Unidades compl. | % compl. | Valor planificado Sprint | Esfuerzo planificado Sprint |
| 1 | 19-04-16 | 13 | 5 | 3 | 5 | | 4700 | 2800 | 2 | 3 | | | | | 1.200 | 400 |
| 2 | 25-04-16 | 13 | 5 | 3 | 5 | 5,33 | 4700 | 2800 | 13 | 5 | 1 | 6 | 100 | 49,9% | 600 | 400 |
| 3 | 02-05-16 | 13 | 5 | 3 | 5 | 4,00 | 4700 | 2800 | 1 | 6 | 7 | 7 | 100 | 33,3% | 300 | 300 |
| 4 | 09-05-16 | 13 | 5 | 3 | 5 | 4,00 | 4700 | 2800 | 7 | 7 | 3 | 8 | 100 | 25,0% | 400 | 300 |
| 5 | 16-05-16 | 13 | 5 | 4 | 5 | 4,00 | 4700 | 2800 | 3 | 8 | 4 | 9 | 100 | 50,0% | 500 | 400 |
| 6 | 23-05-16 | 13 | 5 | 4 | 5 | 5,00 | 4700 | 2800 | 6 | 10 | 8 | 11 | 200 | 40,0% | 300 | 500 |
| 7 | 30-05-16 | 13 | 5 | 3 | 5 | 4,00 | 4700 | 2800 | 6 | 10 | 8 | 11 | 300 | 60,0% | 0 | 300 |
| 8 | 06-06-16 | 18 | 5 | 3 | 5 | 4,00 | 5800 | 3800 | 15 | 12 | | | | | 900 | 300 |
| 9 | 13-06-16 | 18 | 5 | 4 | 5 | 4,00 | 5800 | 3800 | 16 | 14 | | | | | 800 | 400 |

Ilustración 20: Bloque de seguimiento del Proyecto, apartado Iteraciones

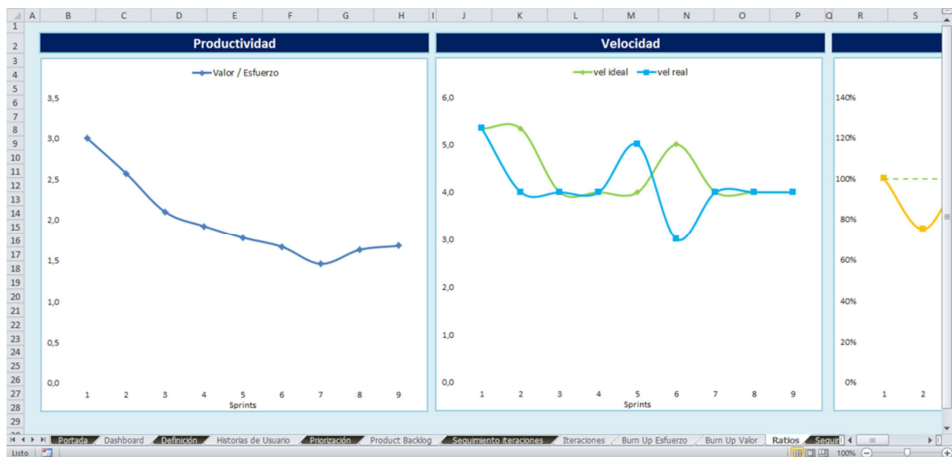


Ilustración 21: Bloque de seguimiento del Proyecto, apartado Ratios

- **Bloque 5; Seguimiento del Sprint**

En este bloque se realiza el seguimiento diario de la evolución del Sprint en curso. Permite detectar cualquier alteración que amenace el cumplimiento del Sprint.

Está compuesto de 3 secciones:

- Sprint; entrada de datos para la planificación del Sprint en curso.
- Burn Down Esfuerzo; evolución del Esfuerzo a lo largo del Sprint.
- Burn Down Valor; evolución del Valor a lo largo del Sprint.

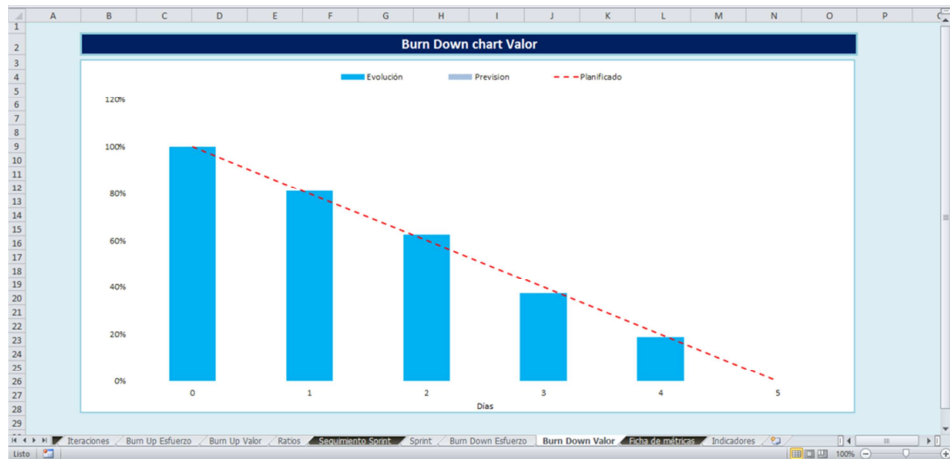


Ilustración 22: Bloque de seguimiento del Sprint, apartado Burn Down Valor.

- Bloque 6; Definición de las métricas

El último bloque presenta las distintas métricas evaluadas en el CdM, además detalla sus principales características; definición, utilidad, cálculo, rango de valores, implementación e inputs.

| Indicador | Valor |
|------------------|---|
| Indicador 1 | Valor |
| Definición | Métrica que pondera la utilidad de la Historia de Usuario |
| Utilidad | Relativizar la utilidad de las diferentes Historias de Usuario entre ellas mismas |
| Cálculo | |
| Rango de Valores | [100,200,300,400,500] |
| Implementación | Hoja: Historias de Usuario Columna: D |
| Inputs | Product Owner: Asignación junto con el Scrum Máster |
| Referencia | Elaboración propia |
| Indicador 2 | Esfuerzo |
| Indicador 3 | Factor de priorización |
| Indicador 4 | Factor de Criticidad |

Ilustración 23: Bloque de definición de las métricas

2.4.2 Métricas

El Cuadro de Mando agrupa las métricas según sean; para la planificación del proyecto (Valor, Esfuerzo, Factor de Priorización, Factor de Criticidad, Tareas Críticas y Velocidad Ideal) o para el seguimiento del estado del proyecto (Productividad Velocidad Real, Factor Foco, Cycle Time, Ratio Cycle Time).

Encontramos las distintas fichas de las métricas en el bloque Indicadores.

| | |
|------------------|---|
| Indicador 1 | Valor |
| Definición | Métrica que pondera la utilidad de la Historia de Usuario |
| Utilidad | Relativizar la utilidad de las diferentes Historias de Usuario entre ellas mismas |
| Cálculo | |
| Rango de Valores | [100,200,300,400,500] |
| Implementación | Hoja: Historias de Usuario Columna: D |
| Inputs | Product Owner: Asignación junto con el Scrum Máster |

Ilustración 24: Bloque de definición de las métricas

- Para la planificación del proyecto:

- **Valor**

Definición:

Métrica que pondera la utilidad de la Historia de Usuario.

Utilidad:

Relativizar la utilidad de las diferentes Historias de Usuario entre ellas mismas.

Cálculo:

Asignación directa.

Rango de valores:

[100, 200, 300, 400, 500]

Implementación:

Se asigna en el bloque de definición de las Historias de Usuario. Es implementada a lo largo del CdM.

Inputs:

Product Owner. Asignación junto con el Scrum Máster.

Referencia:

[1], [3], [5], [8], [11]

- **Esfuerzo**

Definición:

Métrica que pondera el coste en tiempo de implementar una Historia de Usuario.

Utilidad:

Relativizar el coste de implementación de las diferentes Historias de Usuario entre ellas mismas.

Cálculo:

Asignación directa.

Rango de valores:

[100, 200, 300, 400, 500]

Implementación:

Se asigna en el bloque de definición de las Historias de Usuario. Es implementada a lo largo del CdM.

Inputs:

Equipo Scrum. Asignación junto con el Scrum Máster.

Referencia:

[1], [3], [5], [8], [11]

- **Factor de priorización**

Definición:

Factor de ponderación de las Historias de Usuario en función de las métricas de Esfuerzo y Valor.

Utilidad:

Priorizar las Historias de Usuario no empezadas del Product Backlog.

Cálculo:

Factor priorización = $(V-E)+((V-E)/(V+E))$

V=Valor de la historia de usuario

E=Esfuerzo estimado de la historia de usuario

Rango de valores:

De -400,67 a +400,67 siendo +400,67 el valor que otorga mayor priorización.

Implementación:

Bloque Product Backlog.

Inputs:

No necesita ninguna entrada directa.

Referencia:

Elaboración propia.

- **Factor de criticidad**

Definición:

Factor de ponderación de las Tareas no iniciadas del Product Backlog en función de las métricas Esfuerzo y Valor.

Utilidad:

Identificar aquellas Tareas no iniciadas del Product Backlog que puedan amenazar la correcta evolución del proyecto.

Cálculo:

Factor criticidad=(V*E*%E_tarea)

V=Valor de la historia de usuario

E=Esfuerzo estimado de la historia de usuario

%E_tarea=El % de Esfuerzo de la Historia de Usuario que recae en la Tarea

Rango de valores:

De 0 a 250000 siendo 250000 el factor que otorga mayor criticidad.

Implementación:

Se implementa en bloques auxiliares no mostrados en el CdM, no obstante forma parte de la métrica de Tareas Críticas.

Inputs:

No necesita ninguna entrada directa.

Referencia:

Elaboración propia.

- **Tareas Críticas**

Definición:

Alerta de la criticidad de la Tarea con respecto al total del Product Backlog. Representa el 25% de las Tareas no iniciadas con mayor Factor de Criticidad.

Utilidad:

Detectar de manera visual el 25% de las tareas restantes del Product Backlog (no iniciadas), que requieren mayor atención en su implementación y seguimiento.

Cálculo;

Tercer cuartil del rango de valores que forma el Factor de Criticidad.

Rango de valores:

Sí o No (Sí=celda sombreada en naranja , No=celda no sombreada en naranja).

Implementación:

Bloque Product Backlog.

Inputs:

No necesita ninguna entrada directa.

Referencia:

Elaboración propia.

- **Velocidad ideal**

Definición:

Ritmo de trabajo estimado al planificar el Sprint y medido en unidades de persona/hora.

Utilidad:

Dimensionar el tamaño del Sprint Backlog y estimar las Historias de Usuario que estarán terminadas y las que quedarán por terminar cuando finalice el Sprint.

Cálculo:

$[(\text{Esfuerzo planificado en el Sprint}) / (\text{días Sprint} * \text{personas} * \text{horas diarias})]$.

Rango de valores:

De cero a infinito.

Implementación:

Bloque Iteraciones sección Ratios.

Inputs:

No necesita ninguna entrada directa.

Referencia:

[1], [3], [5], [8], [11]

- Para el seguimiento del estado del proyecto:

- **Productividad**

Definición:

Cociente entre el Valor entregado y el Esfuerzo que se ha realizado.

Utilidad:

Analizar el Valor que se obtiene por cada unidad de Esfuerzo consumida en cada iteración.

Cálculo:

Valor completado en el Sprint / Esfuerzo completado en el Sprint.

Rango de valores:

De infinito a cero.

Implementación:

Bloque Iteraciones sección Ratios.

Inputs:

No necesita ninguna entrada directa.

Referencia:

Elaboración propia.

- **Velocidad real**

Definición:

Ritmo de trabajo real obtenido al realizar el seguimiento del Sprint y medido en unidades de persona/hora.

Utilidad:

Dimensionar el tamaño del siguiente Sprint Backlog y estimar en que Sprint terminará el proyecto.

Cálculo:

$[(\text{Esfuerzo completado en el Sprint}) / (\text{días Sprint} * \text{personas} * \text{horas diarias})]$.

Rango de valores:

De cero a infinito.

Implementación:

Bloque Iteraciones sección Ratios.

Inputs:

No necesita ninguna entrada directa.

Referencia:

[1], [3], [5], [8], [11]

- **Facto Foco (%)**

Definición:

Cociente entre la velocidad real y la velocidad ideal de las iteraciones.

Utilidad:

Un Factor Foco menor a 100% indica peor desempeño del Equipo Scrum respecto a lo planificado.

Cálculo:

Velocidad real Sprint / Velocidad ideal Sprint.

Rango de valores:

De cero a infinito (%).

Implementación:

Bloque Iteraciones sección Ratios.

Inputs:

No necesita ninguna entrada directa.

Referencia:

[1], [5]

- **Cycle Time**

Definición:

Número de días laborables que transcurren desde que una Historia de Usuario es iniciada (Doing), hasta que es considerada como terminada (Done).

Utilidad:

Medir el ritmo de terminación de las Historias de Usuario desde el momento en que son iniciadas.

Cálculo:

Días laborables entre dos fechas; fecha inicio y fecha fin. La fecha inicio es la fecha en que se empieza a trabajar en la Historia de Usuario y la fecha fin es la fecha en que es considerada como terminada.

Rango de valores:

De cero a infinito.

Implementación:

Bloque Product Backlog.

Inputs:

No necesita ninguna entrada directa.

Referencia:
[1], [5]

- **Ratio Cycle Time**

Definición:
Cociente entre el Cycle Time y los días de Sprint.

Utilidad:
Si el ratio está por encima de la unidad significa que no se está poniendo suficiente foco en cada Tarea y es necesario que el Equipo se centre en terminar Tareas antes que en empezar otras.

Cálculo:
 $\text{Cycle Time} / \text{días Sprint}$.

Rango de valores:
De cero a infinito.

Implementación:
Bloque Iteraciones sección Ratios.

Inputs:
No necesita ninguna entrada directa.

Referencia:
Elaboración propia.

2.4.3 Resultado de la ejecución del caso práctico en el cuadro de mando

En este apartado se describirá la implementación real del proceso ágil propuesto, focalizándose en el resultado obtenido en el Cuadro de Mando.

El proyecto que se describe se ha llevado a cabo entre el 18/04/2016 y el 17/06/2016.

2.4.3.1 Priorización y alertas en el Product Backlog

Inicialmente el día 18/04/2016 el proyecto está formado por 13 Historias de Usuario, los detalles de las mismas se pueden encontrar en el documento: "Dataset 1.xlsx". Cada Historia de Usuario tiene asociadas 6 Tareas.

Observamos la composición del Product Backlog que podemos encontrar en el fichero: "CdM ejemplo 1.xlsx".

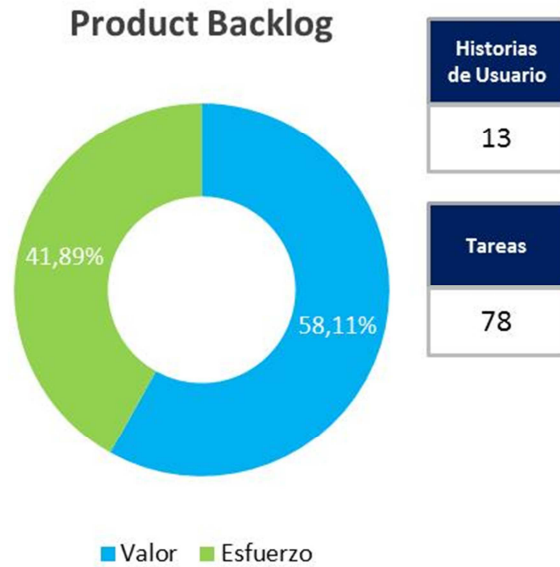


Ilustración 25: Composición inicial del Product Backlog

De inicio, el Valor que el Product Owner ha asignado al proyecto es mayor que el Esfuerzo que el Equipo ha considerado. En realidad esta comparativa puede resultar engañosa ya que ambas métricas son asignadas por distintos roles con criterios distintos. No obstante sí que puede dar una idea (sobre todo cuando ambos roles son más experimentados en este tipo de proyectos ágiles), de la rentabilidad del proyecto en términos de valor aportado.

Observamos a continuación el Product Backlog todavía sin priorizar:

| Product Backlog | | | | | | | | | | | | | |
|------------------|-------|----------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|------------------|-------------------|----------------|
| Historia Usuario | Valor | Esfuerzo | Esfuerzo Tarea 1 | Esfuerzo Tarea 2 | Esfuerzo Tarea 3 | Esfuerzo Tarea 4 | Esfuerzo Tarea 5 | Esfuerzo Tarea 6 | Esfuerzo Tarea 7 | Esfuerzo Tarea 8 | Esfuerzo Tarea 9 | Esfuerzo Tarea 10 | Total Esfuerzo |
| 1 | 300 | 200 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 2 | 400 | 200 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 3 | 500 | 400 | 50,0% | 10,0% | 5,0% | 10,0% | 5,0% | 20,0% | | | | | 100,0% |
| 4 | 200 | 200 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 5 | 100 | 200 | 40,0% | 15,0% | 5,0% | 10,0% | 5,0% | 25,0% | | | | | 100,0% |
| 6 | 300 | 300 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 7 | 400 | 300 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 8 | 500 | 500 | 50,0% | 10,0% | 5,0% | 10,0% | 5,0% | 20,0% | | | | | 100,0% |
| 9 | 200 | 300 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 10 | 300 | 100 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 11 | 400 | 200 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 12 | 500 | 100 | 50,0% | 10,0% | 5,0% | 10,0% | 5,0% | 20,0% | | | | | 100,0% |
| 13 | 200 | 100 | 30,0% | 20,0% | 10,0% | 10,0% | 10,0% | 20,0% | | | | | 100,0% |
| 14 | | | | | | | | | | | | | |

Ilustración 26: Orden inicial del Product Backlog sin priorizar

El Product Backlog sigue el orden en el que el Product Owner ha escrito las Historias de Usuario. Las alertas naranjas indican el 25% de las Tareas no iniciadas consideradas como críticas o importantes, debido a su grado de Valor y Esfuerzo. Observamos por ejemplo como la Tarea 1 de la Historia de Usuario 3 (HU3), esta sombreada en naranja, obvio ya

que la HU3 tiene asignadas 500 unidades de Valor y 400 de Esfuerzo y además la tarea 1 representa el 50% del Esfuerzo total.

Vemos como ya en esta representación del Product Backlog se implementa una de las principales funciones del CdM, la alerta de tareas críticas.

Seguidamente vemos el Product Backlog priorizado:

| Product Backlog Priorizado | | | | | | | | | | | | | | | | | | | | | |
|----------------------------|------------------|-------|-----------|-------------|-----------------|----------|--------------|----------------|--------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|--|
| Priorización | Historia Usuario | Valor | Valor (%) | Valor Acum. | Valor Acum. (%) | Esfuerzo | Esfuerzo (%) | Esfuerzo Acum. | Esfuerzo Acum. (%) | Crítico | Tarea 1 | Tarea 2 | Tarea 3 | Tarea 4 | Tarea 5 | Tarea 6 | Tarea 7 | Tarea 8 | Tarea 9 | Tarea 10 | |
| 1 | 12 | 500 | 11,6% | 500 | 11,6% | 100 | 3,2% | 100 | 3,2% | Sí | ■ | | | | | | | | | | |
| 2 | 10 | 300 | 7,0% | 800 | 18,6% | 100 | 3,2% | 200 | 6,5% | No | | ■ | | | | | | | | | |
| 3 | 2 | 400 | 9,3% | 1.200 | 27,9% | 200 | 6,5% | 400 | 12,9% | Sí | ■ | | | | | | | | | | |
| 4 | 11 | 400 | 9,3% | 1.600 | 37,2% | 200 | 6,5% | 600 | 19,4% | Sí | ■ | | | | | | | | | | |
| 5 | 13 | 200 | 4,7% | 1.800 | 41,9% | 100 | 3,2% | 700 | 22,6% | No | | | | | | | | | | | |
| 6 | 1 | 300 | 7,0% | 2.100 | 48,8% | 200 | 6,5% | 900 | 29,0% | Sí | ■ | ■ | | | | | | | | | |
| 7 | 7 | 400 | 9,3% | 2.500 | 58,1% | 300 | 9,7% | 1.200 | 38,7% | Sí | ■ | ■ | ■ | | | | | | | | |
| 8 | 3 | 500 | 11,6% | 3.000 | 69,8% | 400 | 12,9% | 1.600 | 51,6% | Sí | ■ | ■ | ■ | ■ | | | | | | | |
| 9 | 4 | 200 | 4,7% | 3.200 | 74,4% | 200 | 6,5% | 1.800 | 58,1% | No | | | | | | | | | | | |
| 10 | 6 | 300 | 7,0% | 3.500 | 81,4% | 300 | 9,7% | 2.100 | 67,7% | Sí | ■ | ■ | ■ | ■ | | | | | | | |
| 11 | 8 | 500 | 11,6% | 4.000 | 93,0% | 500 | 16,1% | 2.600 | 83,9% | Sí | ■ | ■ | ■ | ■ | ■ | | | | | | |
| 12 | 9 | 200 | 4,7% | 4.200 | 97,7% | 300 | 9,7% | 2.900 | 93,5% | Sí | ■ | ■ | ■ | ■ | ■ | ■ | | | | | |
| 13 | 5 | 100 | 2,3% | 4.300 | 100,0% | 200 | 6,5% | 3.100 | 100,0% | No | | | | | | | | | | | |

Ilustración 27: Product Backlog inicial priorizado

| Priorización | Historia Usuario | Valor | Valor (%) | Valor Acum. | Valor Acum. (%) | Esfuerzo | Esfuerzo (%) | Esfuerzo Acum. | Esfuerzo Acum. (%) |
|--------------|------------------|-------|-----------|-------------|-----------------|----------|--------------|----------------|--------------------|
| 1 | 12 | 500 | 11,6% | 500 | 11,6% | 100 | 3,2% | 100 | 3,2% |
| 2 | 10 | 300 | 7,0% | 800 | 18,6% | 100 | 3,2% | 200 | 6,5% |
| 3 | 2 | 400 | 9,3% | 1.200 | 27,9% | 200 | 6,5% | 400 | 12,9% |
| 4 | 11 | 400 | 9,3% | 1.600 | 37,2% | 200 | 6,5% | 600 | 19,4% |
| 5 | 13 | 200 | 4,7% | 1.800 | 41,9% | 100 | 3,2% | 700 | 22,6% |
| 6 | 1 | 300 | 7,0% | 2.100 | 48,8% | 200 | 6,5% | 900 | 29,0% |
| 7 | 7 | 400 | 9,3% | 2.500 | 58,1% | 300 | 9,7% | 1.200 | 38,7% |
| 8 | 3 | 500 | 11,6% | 3.000 | 69,8% | 400 | 12,9% | 1.600 | 51,6% |
| 9 | 4 | 200 | 4,7% | 3.200 | 74,4% | 200 | 6,5% | 1.800 | 58,1% |
| 10 | 6 | 300 | 7,0% | 3.500 | 81,4% | 300 | 9,7% | 2.100 | 67,7% |
| 11 | 8 | 500 | 11,6% | 4.000 | 93,0% | 500 | 16,1% | 2.600 | 83,9% |
| 12 | 9 | 200 | 4,7% | 4.200 | 97,7% | 300 | 9,7% | 2.900 | 93,5% |
| 13 | 5 | 100 | 2,3% | 4.300 | 100,0% | 200 | 6,5% | 3.100 | 100,0% |

Ilustración 28: Detalle del Product Backlog inicial priorizado

En la ilustración 27 vemos como ahora delante de la columna Historia de Usuario hay una columna titulada Priorización. La columna Priorización define el orden en que las Historias de Usuario deben de entrar a los Sprints Backlogs. La cantidad de Historias de Usuario de cada Sprint Backlog la definirá el Equipo Scrum.

El Product Backlog priorizado es otra de las principales aportaciones del CdM , en la ilustración 26 (celdas sombreadas en naranja), observamos como fruto de la correcta priorización, el CdM escoge inicialmente las Historias de Usuario con mayor Valor y menor Esfuerzo, y condensa en la parte central del proyecto las Historias de Usuario más críticas. De

este modo en caso de cese temprano del proyecto por motivos por ejemplo presupuestarios, el Product Owner habrá recibido el mayor valor posible en el menor tiempo disponible.

El resultado de la priorización del Product Backlog también lo podemos apreciar en los gráficos auxiliares disponible al final del bloque; Product Backlog.

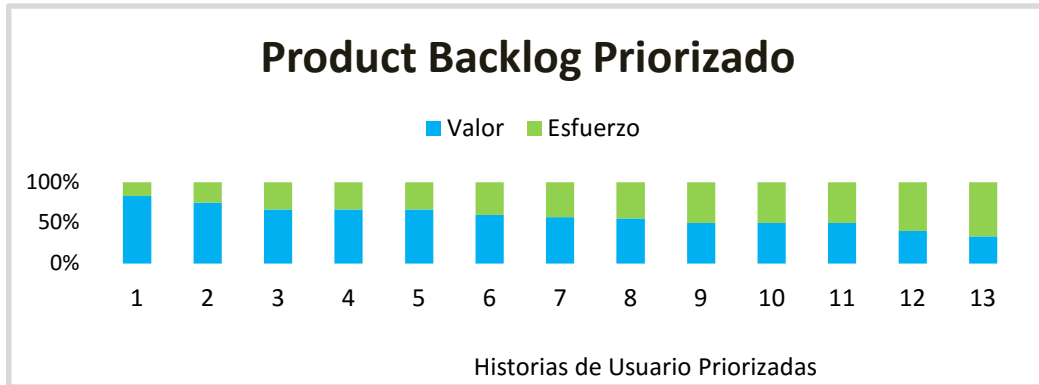


Ilustración 29: Grafico auxiliar (1) del bloque Product Backlog al inicio del proyecto

En la ilustración 28 se muestra en azul la evolución porcentual del Valor a lo largo del orden de ejecución que plantea el CdM. En verde se muestra la evolución del Esfuerzo. Observamos como las primeras ejecuciones planteadas, presentan porcentualmente mayor Valor que Esfuerzo, sucede lo contrario a medida que nos acercamos al final.

Veamos ahora el segundo gráfico auxiliar:

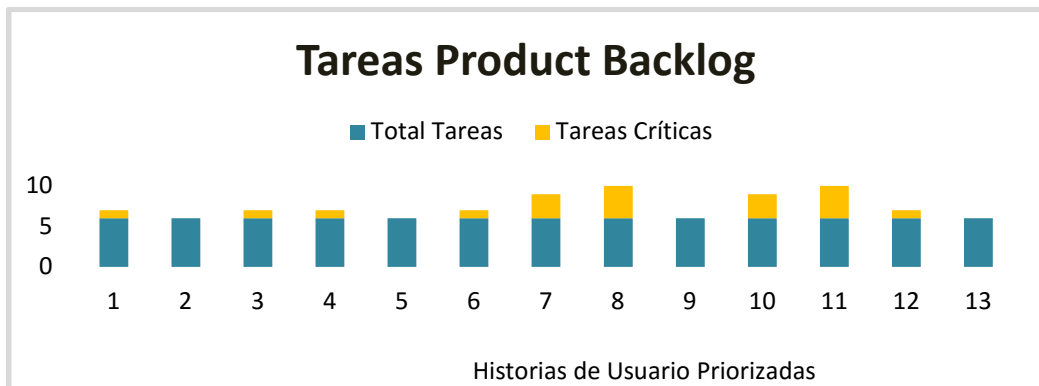


Ilustración 30: Grafico auxiliar (1) del bloque Product Backlog al inicio del proyecto

La ilustración 29 muestra en su eje horizontal el orden propuesto de ejecución de las Historias de Usuario. En el eje vertical se representan el total de Tareas y además el número de Tareas críticas sobre ese total. Observamos como el CdM concentra las Historias de Usuario con Tareas críticas pasada la mitad del orden de ejecución. No sería oportuno dejar Historias de Usuario críticas para el final de la ejecución, debido a que si la Historia de Usuario es crítica, significa que aporta suficiente Valor, y recordemos que el objetivo del CdM es maximizar el Valor aportado en las primeras iteraciones.

Una puntualización importante llegados a este punto es que no hay que confundir lo que hemos llamado ejecución, de lo que son las iteraciones. Por ejecución nos referimos a las Historias de Usuario priorizadas, por eso en los gráficos anteriores hay siempre 13 ejecuciones. Ahora bien, puede que las 3 primeras ejecuciones (ilustración 27; HU12, HU10, HU2) se implementen en la primera iteración del proyecto (Sprint 1).

2.4.3.2 Seguimiento diario

Siguiendo con el desarrollo de la explicación de la implementación del proyecto, mostramos la evolución real del proyecto en sus 2 primeras iteraciones (la evolución completa puede ser consultada en el documento adjunto “Evolución proyecto.xlsx”).

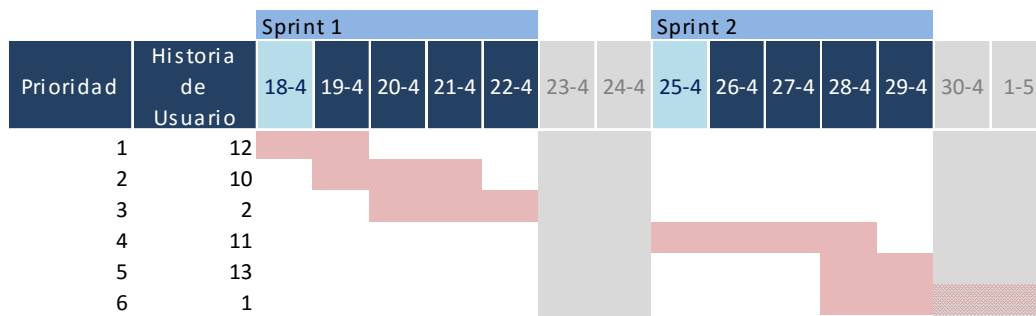


Ilustración 31: Evolución real del proyecto. 2 primeras iteraciones (Sprints)

En la ilustración 30 se observa como para la primera iteración, el Equipo ha decidido introducir 3 Historias de Usuario del Product Backlog, en el Sprint Backlog del Sprint 1. Concretamente las Historias de Usuario escogidas han sido; HU12, HU10 y HU2, se ha seguido el orden propuesto por el CdM.

Cada Sprint consta de 5 días laborables. Situándonos en el tercer día de la primera iteración (Sprint1), observamos los 2 siguientes gráficos relativos a la evolución diaria del proyecto.

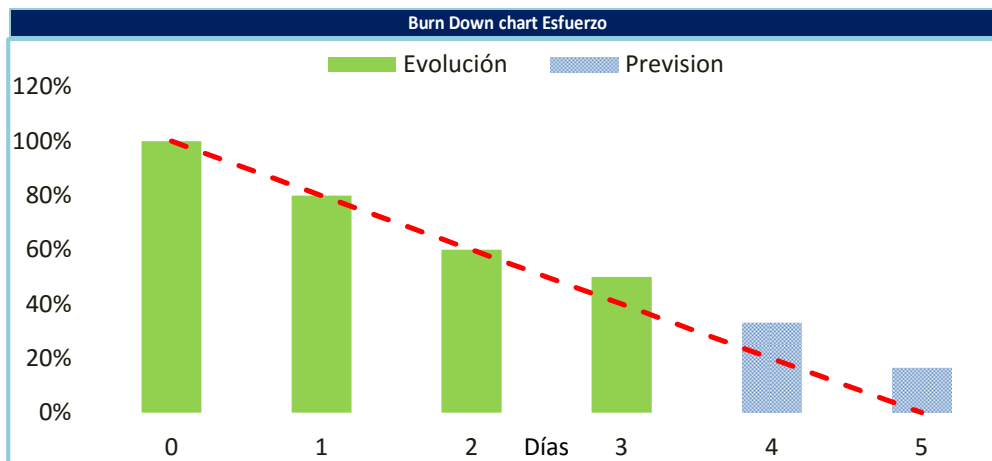


Ilustración 32: Gráfico Burn Down Esfuerzo Sprint 1

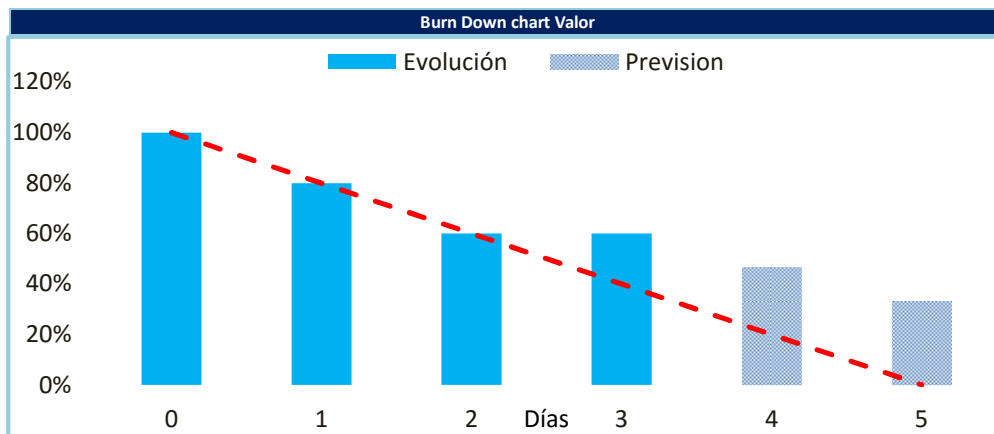


Ilustración 33: Gráfico Burn Down Valor Sprint 1

Ambos gráficos corresponden al bloque; Seguimiento Sprint. La ilustración 31 representa la evolución del Esfuerzo durante los 3 primeras días y su previsión para los días 4 y 5. Los gráficos pueden ser consultados en el fichero: “CdM ejemplo 2.xlsx”.

Se puede observar como los dos primeros días la evolución real (columna verde) se ha ajustado a la planificada (línea roja punteada). No obstante el tercer día el Equipo no ha conseguido aportar las unidades de Esfuerzo planificadas y así lo ha reflejado en el CdM el Scrum Master. Vemos como el CdM realiza una estimación del Esfuerzo que restará pendiente al finalizar el quinto día (final del Sprint). En este caso el CdM prevé que siguiendo con la velocidad real promedia que se tienen al final del tercer día, el Esfuerzo pendiente por completar al finalizar el Sprint será del 16.7%.

En el caso del Valor (ilustración 32), la evolución es similar pero la previsión para el quinto día es que quede pendiente un 33.3% del Valor por entregar. Si nos fijamos en el tercer día, el Equipo ha conseguido aportar más Esfuerzo que Valor sobre lo planificado, de ahí a que la previsión muestre que al final del Sprint restará pendiente mayor porcentaje de Valor que de Esfuerzo. Como podemos comprobar, el Esfuerzo y el Valor no guardan una relación lineal. El tercer día el Scrum Master ha considerado que aunque no se ha conseguido alcanzar el Esfuerzo que la planificación indica, su Equipo sí que ha adelantado bastante trabajo, con lo que decide valor el desempeño del Equipo a la mitad de lo planificado. En cambio el Scrum Máster es consiente que el Valor se mide según las funcionalidades entregadas al Product Owner, y al finalizar el tercer día el Equipo no ha conseguid terminar ninguna nueva Historia de Usuario (funcionalidad), por lo que no puede reducir el Valor remanente. De hecho si nos fijamos la columna azul (evolución real del Valor) de la ilustración 32 está en el mismo punto los días 2 y 3.

2.4.3.3 Seguimiento de los Sprints

Pasamos a continuación a mostrar el estado de la evolución global del proyecto al finalizar el Sprint número 7, concretamente el día 03/06/2016.

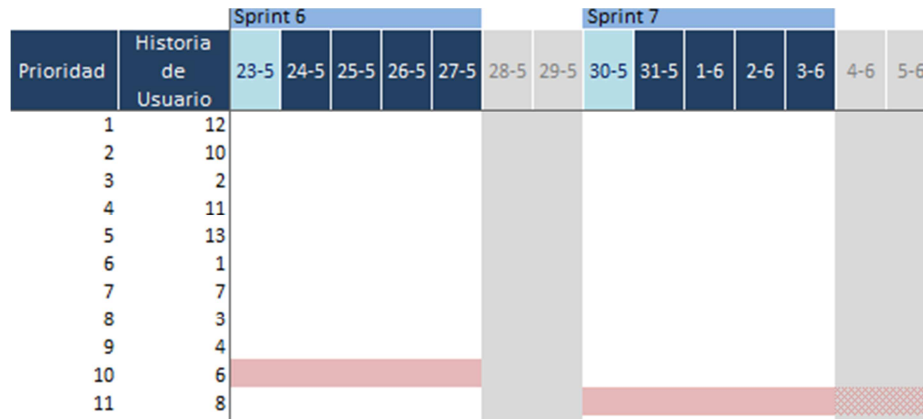


Ilustración 34: Evolución del proyecto. Sprints 6 y 7

El gráfico muestra que al finalizar el Sprint 7, la HU8 sigue pendiente de ser terminada (la barra roja del Sprint 7 sigue hacia el Sprint número 8). En cambio observamos como en el Sprint 6 sí que hemos conseguido finalizar la Historia de Usuario número 6.

2.4.3.3.1 Planificado vs Completado

A continuación se presentan 2 gráficos auxiliares que podemos encontrar en el bloque; Seguimiento Iteraciones, en la sección; Iteraciones. El documento ejemplo de referencia es: "CdM ejemplo 3.xlsx".

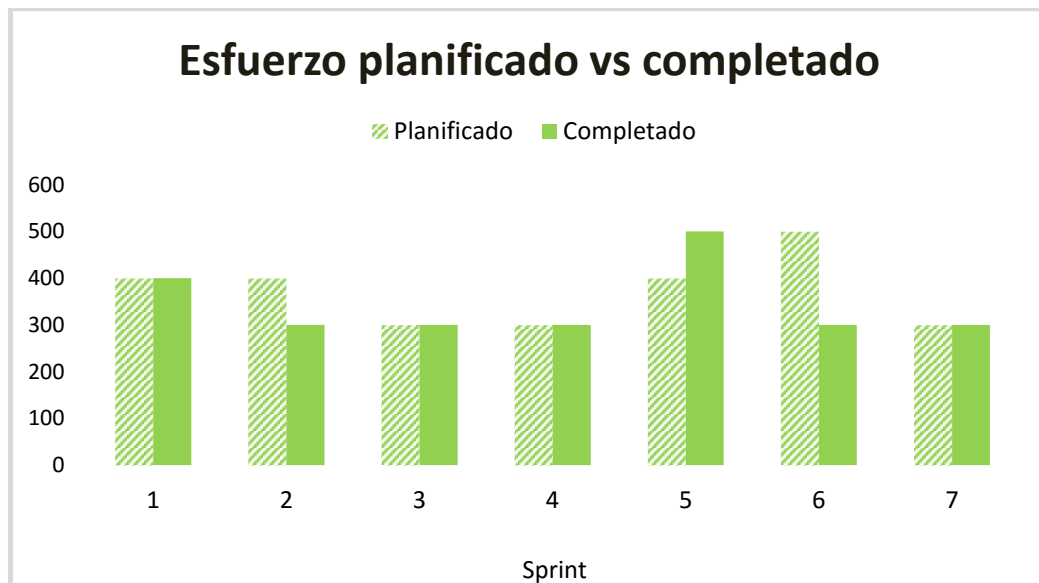


Ilustración 35: Gráfico auxiliar (1) al final del Sprint 7

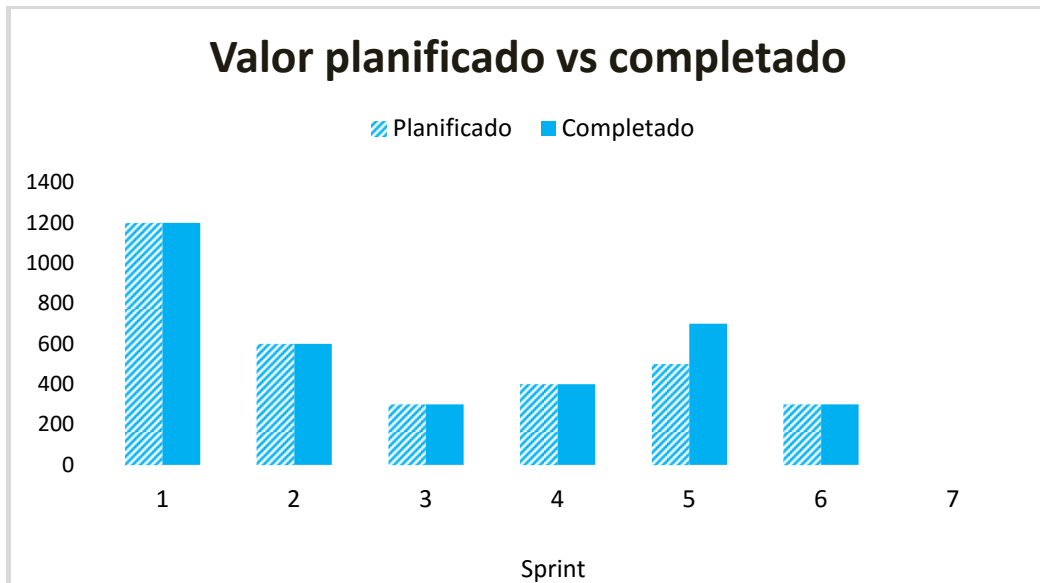


Ilustración 36: Gráfico auxiliar (2) al final del Sprint 7

Los gráficos muestran la evolución de cada Sprint y la cantidad de unidades absolutas de Esfuerzo y Valor planificadas y completadas. Si nos detenemos a analizar el Sprint 7, observamos como a pesar que se han planificado y completado 300 unidades de Esfuerzo, no se han planificado ni completado unidades de Valor.

El motivo es que el CdM tal y como hemos comentado para el ejemplo del seguimiento diario del Sprint, considera que el Valor se obtiene únicamente una vez se ha completado una Historia de Usuario en su totalidad. De la ilustración 33 vemos que no se ha conseguido terminal la Historia de Usuario número 8. No obstante, analizando la ilustración 35, entendemos que no se había planificado la culminación de esta Historia de Usuario (HU8) para el Sprint número 7.

Por tanto tenemos que por una parte el Scrum Master ha planificado trabajar en la HU8 durante el Sprint 7 y ello se refleja en la ilustración 34. Por otra parte el CdM relacionando el Esfuerzo con el Valor, ha previsto antes de empezar el Sprint 7 que la HU8 no será terminada y por ello no lo ha reflejado en la ilustración 35.

2.4.3.3.2 Previsión de finalización

En el mismo bloque de Seguimiento de las Iteraciones, encontramos los 2 siguientes gráficos relativos a la evolución real y la evolución prevista de los Sprints. El fichero ejemplo de referencia sigue siendo: "CdM ejemplo 3.xlsx".

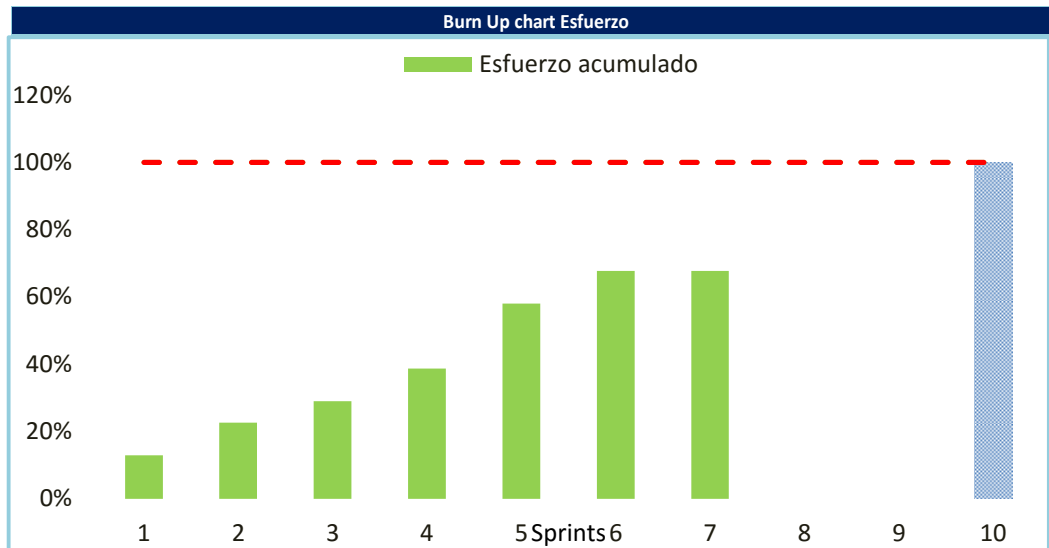


Ilustración 37: Gráfico Burn Up Esfuerzo al final del Sprint 7

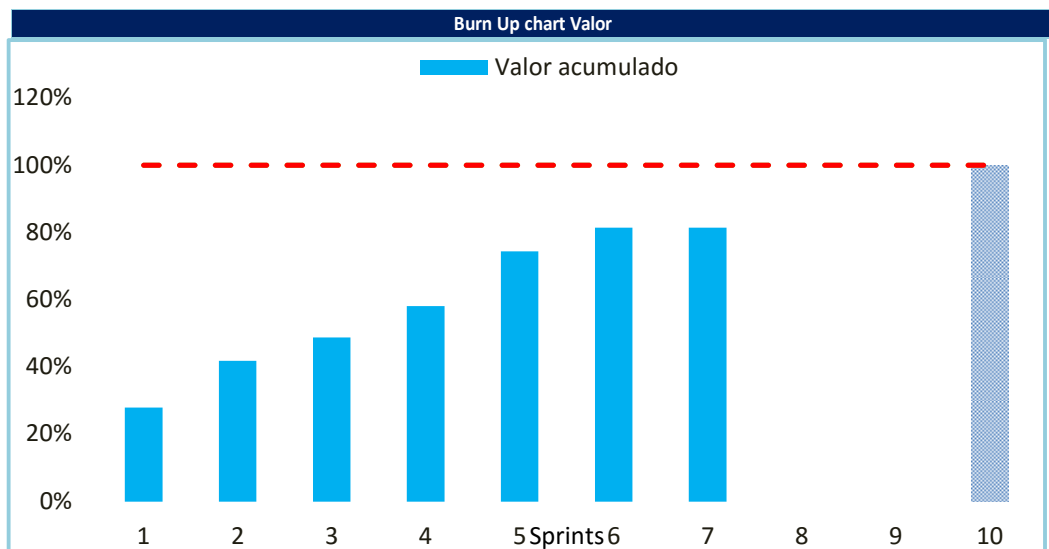


Ilustración 38: Gráfico Burn Up Valor al final del Sprint 7

Ambos gráficos representan el Valor y el Esfuerzo acumulados según las Historias de Usuario que se han logrado finalizar por completo. Por ese motivo en ambos gráficos las barras de los Sprints 6 y 7 son idénticas (no ha habido ningún avance).

Estas dos secciones del CdM implementan una de las características principales del Cuadro de Mando como es la estimación del Sprint de finalización. En nuestro caso la previsión es la finalización en el Sprint número 10, por lo que restarían 3 Sprints.

2.4.3.4 Ampliación Product Backlog

Previo al inicio del Sprint número 8, el Product Owner incorpora 5 nuevas Historias de Usuario al Product Backlog. Podemos ver el detalle

de las nuevas Historias de Usuario en el documento “Dataset 2.xlsx”. El fichero ejemplo de referencia es: “CdM ejemplo 4.xlsx”.

A continuación mostramos el estado del Product Backlog en este momento de la ejecución del proyecto:

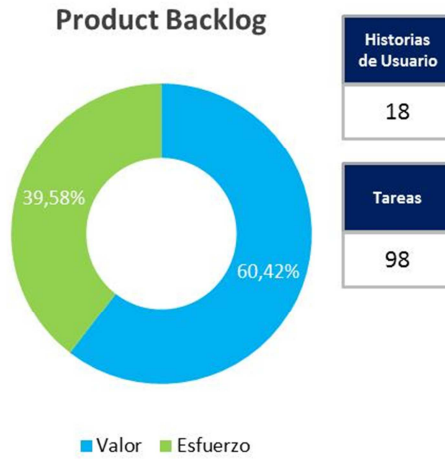


Ilustración 39: Composición del Product Backlog al inicio del Sprint 8

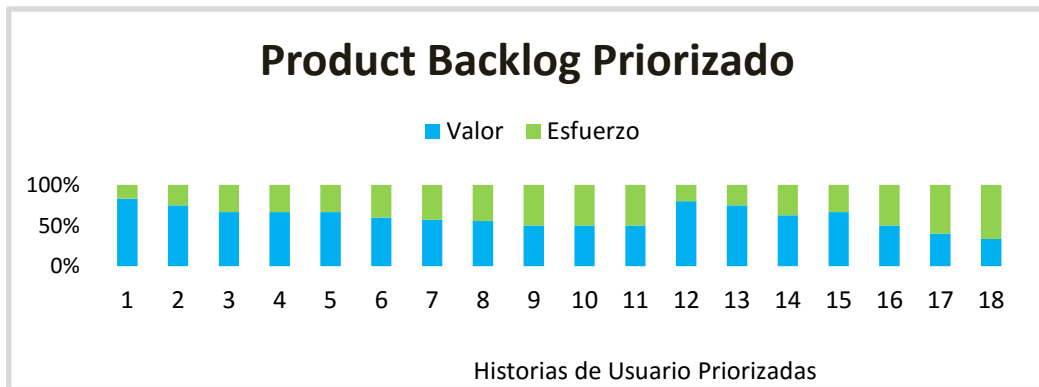


Ilustración 40: Grafico auxiliar (1) del bloque Product Backlog en el Sprint 8

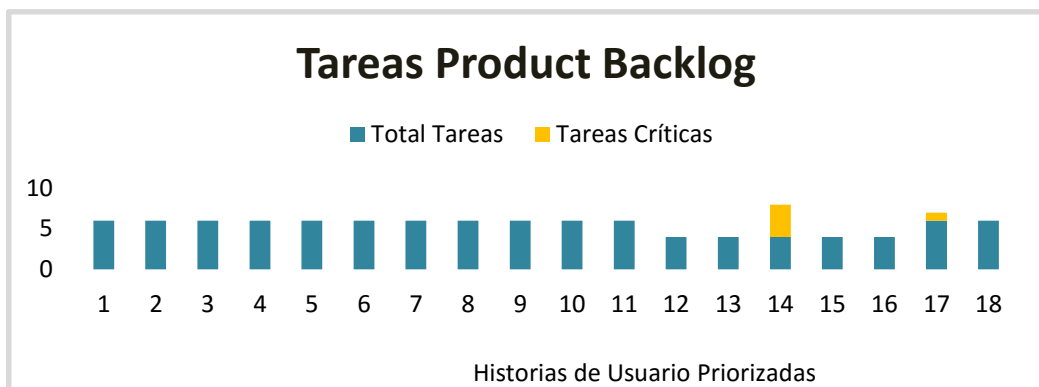


Ilustración 41: Grafico auxiliar (2) del bloque Product Backlog en el Sprint 8

Según la ilustración 33 al final del Sprint 7 seguimos trabajando en la HU8 que se corresponde con la prioridad o ejecución número 11. Recordamos que antes de empezar el Sprint 8 de amplia en 5 Historias

de Usuario el Product Backlog y que estamos visualizando sus gráficos auxiliares.

En la ilustración 39 se observa cómo a partir de la ejecución 12 el Valor vuelve a ser redistribuido y las ejecuciones inmediatas a la 12 presentan mayor porcentaje de Valor que de Esfuerzo. En efecto, cuando el Product Owner ha ampliado el Product Backlog, el CdM ha vuelto a priorizar las Historias de Usuario que todavía no habían sido empezadas. Con ello, y dado que posiblemente algunas de las 5 nuevas Historias de Usuario introducidas por el Producto Owner, aportaban más Valor que las Historias de Usuario que quedaban pendientes en el Product Backlog, la nueva priorización vuelve a maximizar el valor en las primeras ejecuciones pendientes.

La ilustración 40 vuelve a mostrar como el CdM sitúa la Tareas críticas que ha identificado en la mitad de las ejecuciones pendientes.

Analicemos ahora el efecto que la ampliación del Product Backlog ha tenido sobre la previsión de finalización. Para ello revisaremos los gráficos Burn Up en el momento previo al comienzo del Sprint 8. El fichero ejemplo de referencia sigue siendo: "CdM ejemplo 4.xlsx".

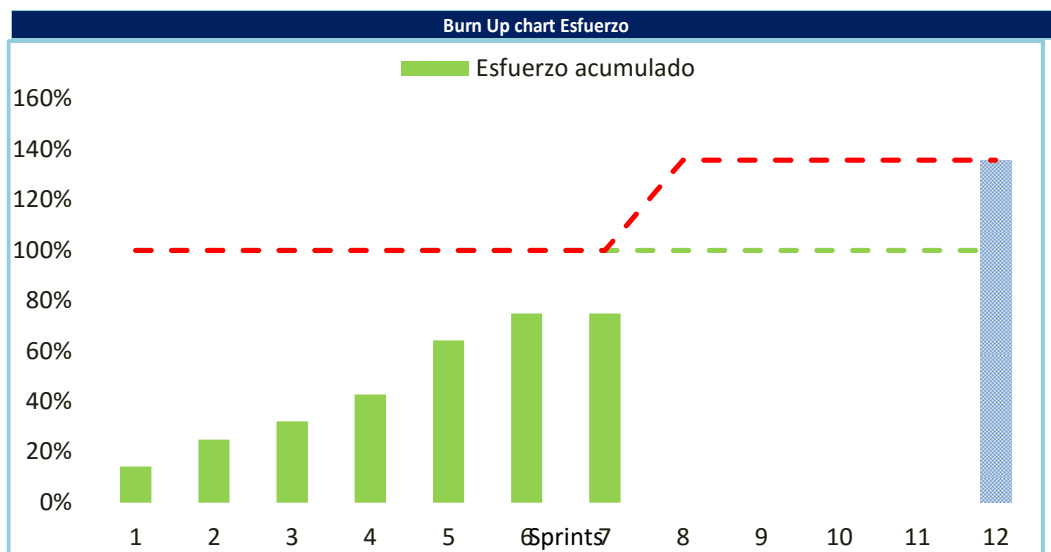


Ilustración 42: Gráfico Burn Up Esfuerzo después de la ampliación del Product Backlog

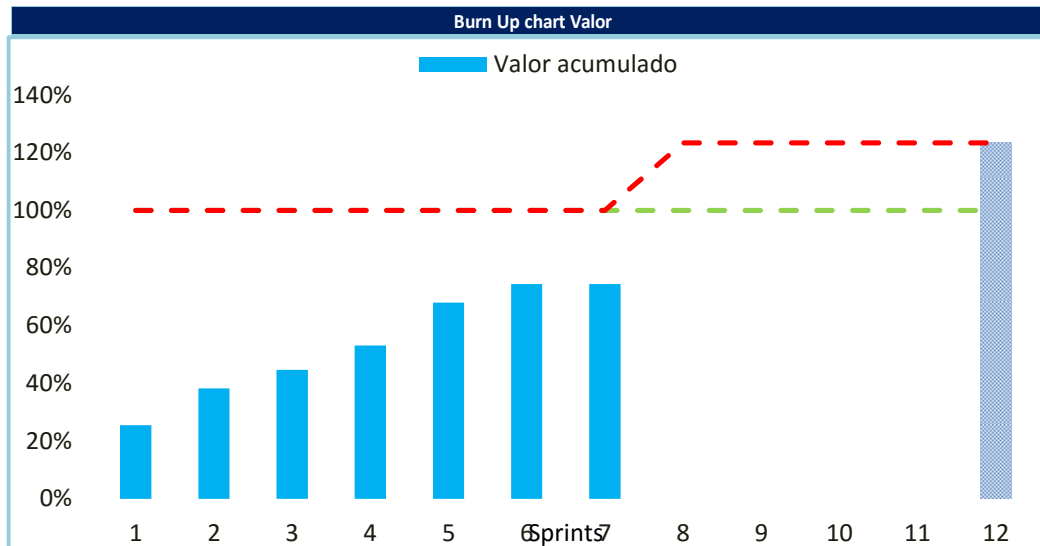


Ilustración 43: Gráfico Burn Up Valor después de la ampliación del Product Backlog

En los gráficos comprobamos que la previsión de finalización ha aumentado pasando al Sprint 12, con lo que en este momento restarían 5 Sprints adicionales.

Los gráficos también muestran el efecto de la ampliación del Product Backlog, vemos como en el caso del Esfuerzo la ampliación del Product Backlog supone un 40% de Esfuerzo adicional. Por el contrario en el caso del Valor la ampliación únicamente supone el 20%.

Antes de pasar al siguiente apartado es necesario mencionar una limitación del CdM relacionada con el tamaño del Product Backlog. El Cuadro de Mando admite únicamente una ampliación de su tamaño, en caso que el Product Backlog fuese ampliado más veces, la función de priorización no trabajaría correctamente. Posteriormente cuando comentemos el bloque Dashboard, veremos el indicador que alerta de esta condición.

2.4.3.4 Ratios

Pasaremos ahora a detallar los gráficos que podemos encontrar en el apartado de Ratios.

Nos situamos a continuación en el final del Sprint número 9, día 17/06/2016 y último día de implementación del caso. El fichero de referencia para todo este apartado y sus sub apartados es: "CdM ejemplo 5.xlsx".

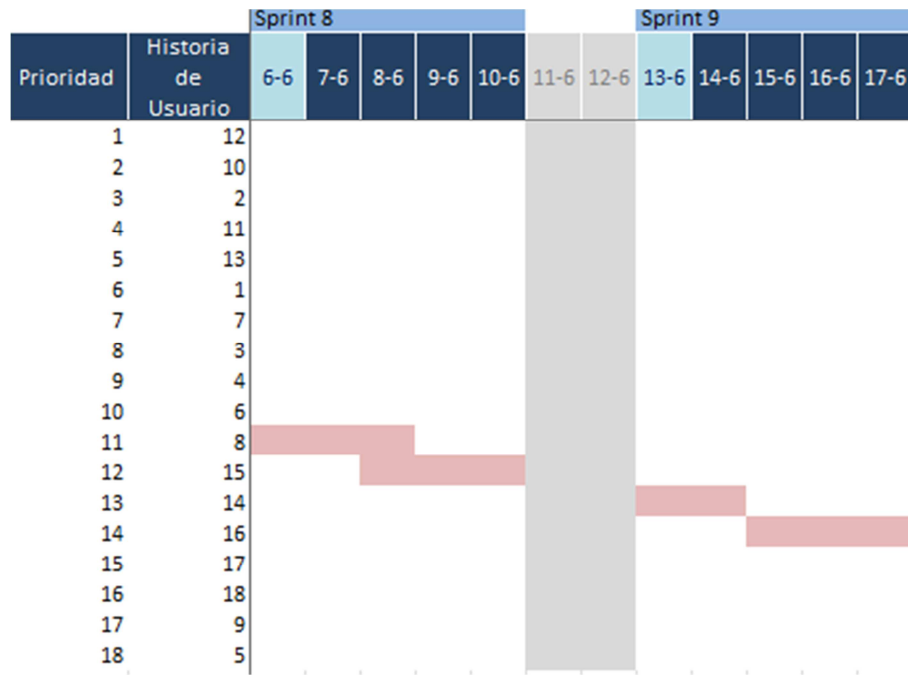


Ilustración 44: Evolución del proyecto. Sprints 8 y 9

A lo largo de los 9 Sprints implementados, se han conseguido terminar 14 prioridades o ejecuciones. La última Historia de Usuario totalmente completada es la HU16.

Mostramos el estado de los gráficos Burn Up:

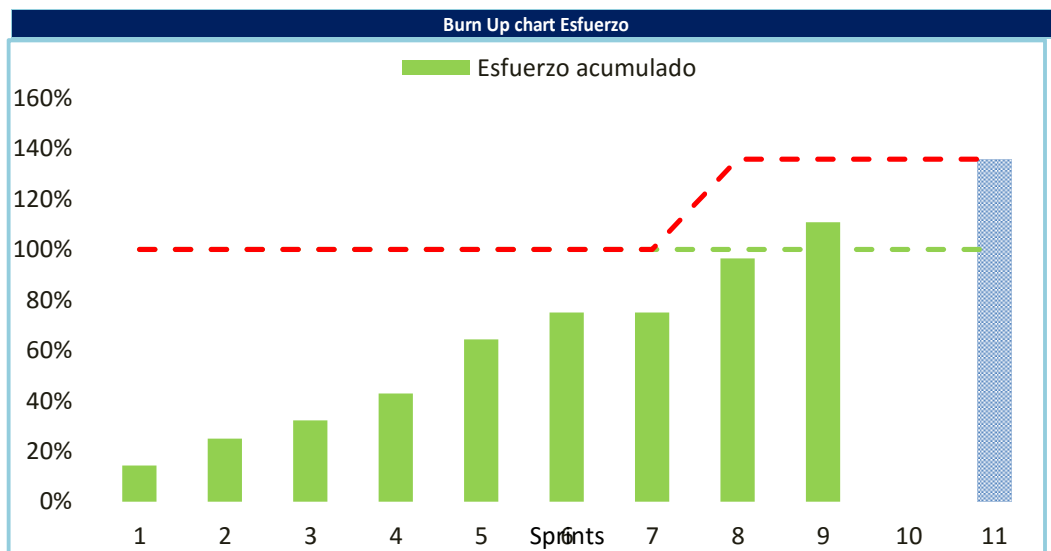


Ilustración 45: Gráfico Burn Up Esfuerzo Sprint 9

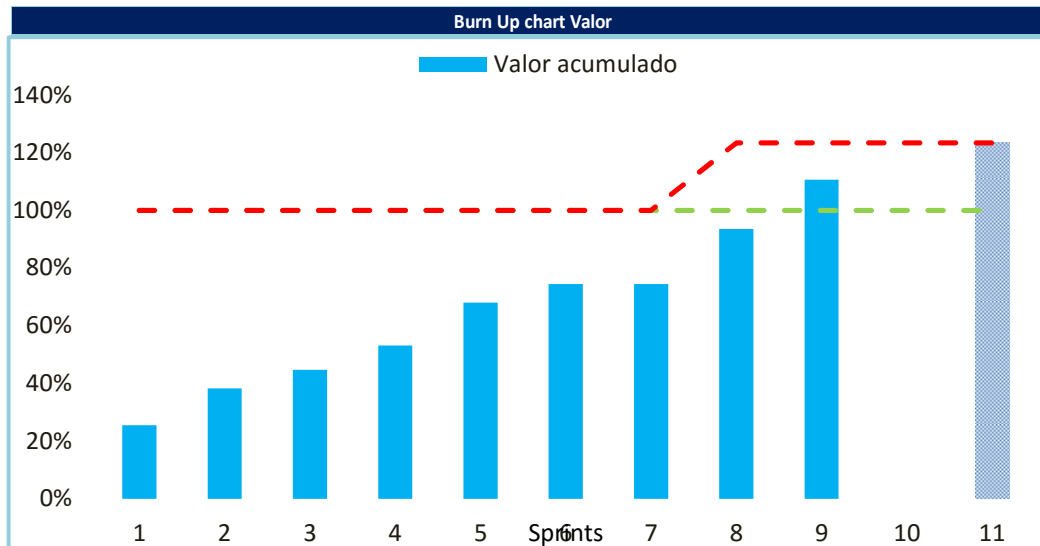


Ilustración 46: Gráfico Burn Up Valor Sprint 9

Observamos como en este punto del proyecto la previsión de finalización se ha adelantado al Sprint número 11.

Seguidamente analizaremos los distintos ratios obtenidos al final del Sprint 9.

2.4.3.4.1 Productividad

Empezamos analizando el gráfico de productividad. Observamos como la Productividad va en decremento desde el inicio del proyecto hasta la iteración 7, momento en el que vuelve a aumentar.

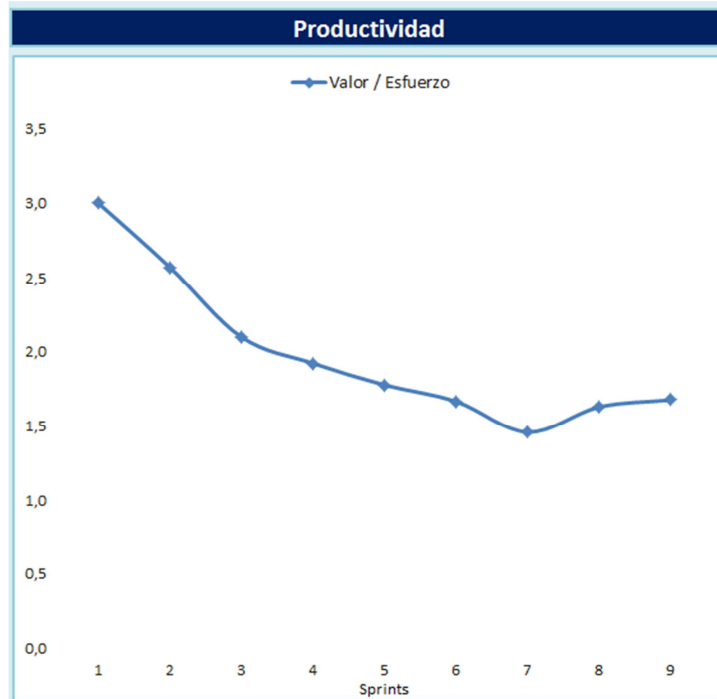


Ilustración 47: Gráfico Productividad al final del Sprint 9

El comportamiento de la Productividad demuestra cómo el CdM ha maximizado el Valor en las primeras iteraciones del proyecto. El repunte de esta métrica en la iteración 8 es debido al aumento del tamaño del Product Backlog. Como hemos comentado a partir del Sprint 8 el Product Owner ha aumentado el tamaño del Product Backlog en 5 Historias de Usuario. Recordamos que en ese momento el CdM ha vuelto a priorizar las Historias de Usuario pendientes de empezar, dando prioridad a las Historias con mayor Valor y menor Esfuerzo. Por tanto el repunte de la Productividad es correcto.

2.4.3.4.2 Velocidad

A continuación analizaremos el gráfico de Velocidad. Se muestra tanto la Velocidad Ideal como la Velocidad Real.

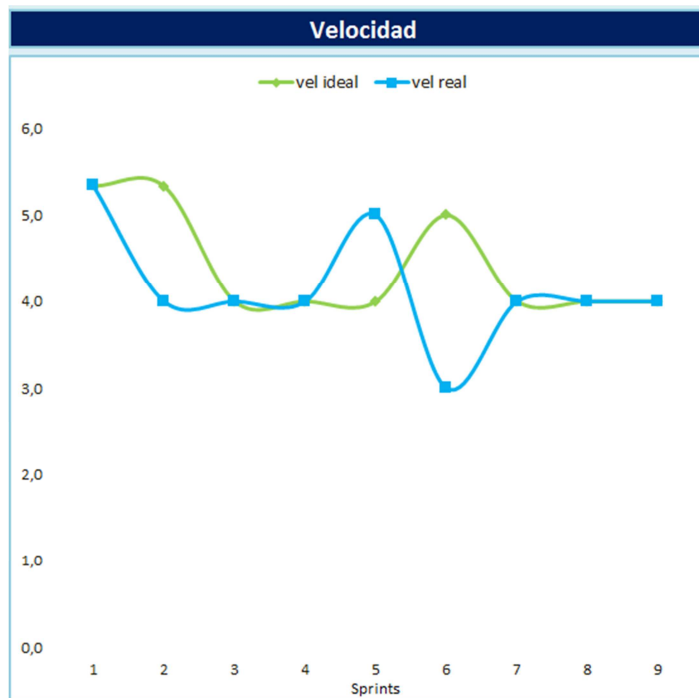


Ilustración 48: Gráfico Velocidad al final del Sprint 9

Como podemos comprobar en las 2 primeras iteraciones, ambas velocidades distan entre si. El motivo de tal diferencia es que de inicio el Equipo no conoce si el Esfuerzo que ha planificado para cada iteración es realizable o no.

Cabe tener en cuenta que las unidades de Esfuerzo que el Equipo a asignado a cada Historia de Usuario son relativas entre ellas mismas. Pudiese darse el caso que dos Historias de Usuario de distintos proyectos y valoradas ambas por ejemplo en 300 unidades, no fuesen realmente comparables en su tiempo real de ejecución. Esto es debido a como hemos comentado la relatividad en la asignación de las unidades de Esfuerzo.

El Equipo ha asignado las unidades de Esfuerzo de entre las Historias de Usuario de este proyecto, identificando primero las Historia de Usuario que suponen que deben de tener mayor y menor puntuación. A partir de esta identificación el Equipo otorga la puntuación a las demás Historias de Usuario relativizando cada una con respecto a la que tiene mayor y menor puntuación. Por este motivo, el Equipo no conoce realmente si la Velocidad planificada en la primera iteración es alcanzable o no.

La Velocidad Real y la Velocidad Ideal terminana por converger debido a que el Equipo poco a poco va conociendo su ritmo de entrega de trabajo en este proyecto. Observamos como incluso en el Sprint 5 la Velocidad Real superó a la Velocidad Ideal. Es curioso ver como en el Sprint 6 la situación es la inversa. El motivo de esta inversión es que el Equipo tras ver el resultado del Sprint 5 decide planificar para el Sprint 6 la misma Velocidad obtenida en el Sprint 5, si observamos el Historico de

velocidades vemos que la Velocidad Real del Sprint 5 podría considerarse un outlier.

2.4.3.4.3 Factor Foco

El Factor Foco ratifica lo que hemos comentado en el apartado anterior. Además clarifica el caso expuesto con anterioridad, permitiendo de una forma mucho más rápida analizar y entender la relación entre las velocidades.

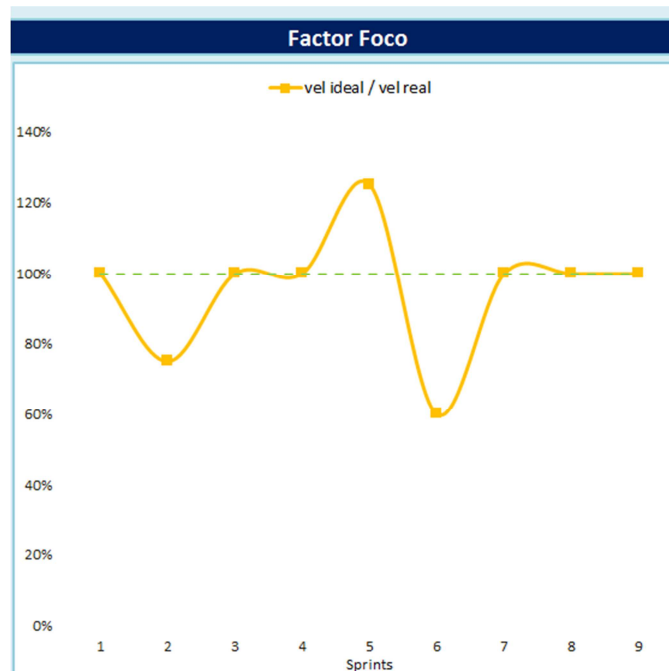


Ilustración 49: Gráfico Factor Foco al final del Sprint

2.4.3.4.4 Ratio Cycle Time

Según lo que se ha mencionado en apartado Métricas, el Ratio Cycle Time es el cociente entre el Cycle Time y los días de Sprint. Si el ratio está por encima de la unidad significa que no se está poniendo suficiente foco en cada Tarea y es necesario que el Equipo se centre en terminar Tareas antes que en empezar otras

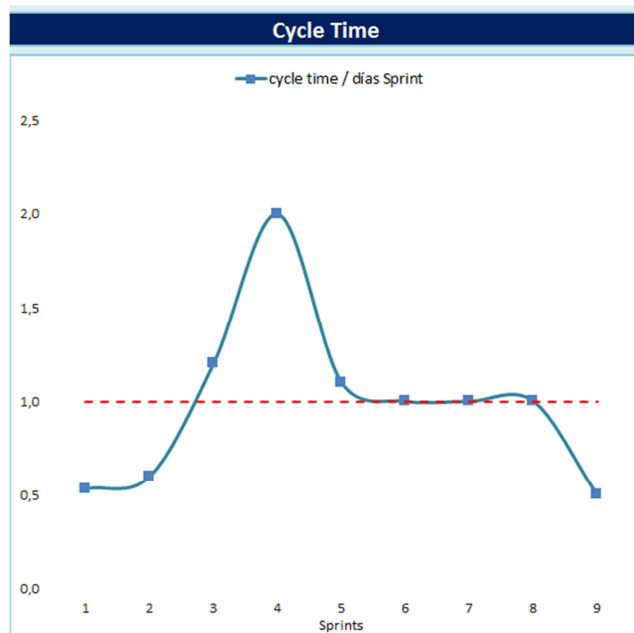


Ilustración 50: Gráfico Ratio Cycle Time al final del Sprint 9

Observamos cómo este ratio aumenta más allá de la unidad en el Sprint 4 para después y a partir del Sprint 5 mantenerse por debajo de 1. Si recordamos el gráfico del Factor Foco, en el Sprint 5 el Equipo ha terminado con una Velocidad Real superior a la Velocidad Ideal. Podemos entender que fruto del análisis de la gráfico del Ratio Cycle Time en el Sprint 4, el Equipo se percata de que debe focalizar más sus esfuerzos en menos Tareas. Fruto de esta focalización los miembros del Equipo se ayudan entre ellos y la colaboración desencadena un mayor rendimiento frente a lo esperado, por eso el Factor Foco aumenta en el Sprint 5.

2.4.3.5 Dashboard

Por último presentaremos el bloque con el resumen ejecutivo o Dashboard.

Como podemos observar en la siguiente ilustración (51), el Dashboard muestra información sobre; la previsión de finalización, la variación porcentual de la productividad hasta el final del proyecto y estado del Product Backlog. El fichero de referencia es: "CdM ejemplo 5.xlsx".

El Dashboard nos indica que nos encontramos en Sprint número 9 y que hemos completado 9 Sprints, por tanto en este momento no estamos en mitad de una iteración. Restan 2 Sprints para finalizar que impactaran reduciendo la productividad en un -9%. El descenso de la productividad también se aprecia en apartado relativo al Product Backlog, en él, la diferencia entre el Esfuerzo planificado y finalizado es de 25 puntos porcentuales mientras que la diferencia entre el Valor planificado y el finalizado es de 12 pp.

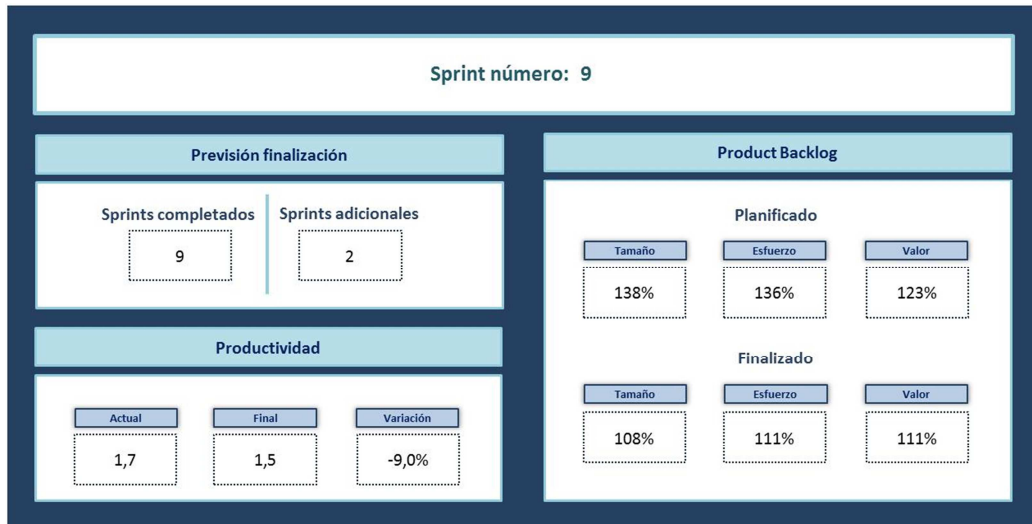


Ilustración 51: Estado del Dashboard al final del Sprint 9

Tal y como comentamos en el apartado Ampliación del Product Backlog (2.4.3.4) el CdM tiene una limitación en cuanto al número de veces que el Product Backlog puede ser ampliado. Esta condición se refleja en el Dashboard mediante una alerta que avisa si es posible o no ampliar el tamaño de la Pila de Producto.

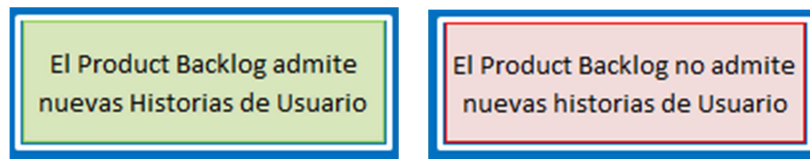


Ilustración 52: Alerta de ampliación del tamaño del Product Backlog

2.5 Conclusiones del método ágil y el CdM

En este apartado se evaluará el valor que aporta el Cuadro de Mando construido, y la idoneidad de los métodos ágiles para labores de inteligencia de negocio. Empezaremos analizando la validez de la metodología ágil planteada para unidades de B.I.

Tal y como hemos mostrado en el apartado; Agilidad en los proyectos de B.I (2.2), diversos autores y profesionales del sector referenciados, comparten la visión que la agilidad es sin duda un enfoque válido para los proyectos de inteligencia de negocio.

En el apartado; Adaptación del método ágil a los proyectos de B.I (2.2.2), se ha planteado la utilización conjunta de Scrum y Kanban, realizando algunas modificaciones a ambos.

Conclusiones de las modificaciones a Scrum:

1. Disminuir la duración de los Sprints:

Aunque en el proyecto implementado la duración de los Sprints ha sido de 5 días laborables, lo que no implica ninguna novedad frente a la teoría original de Scrum, sí se ha podido comprobar que en los proyectos de B.I, con una ventana de 5 días laborables, el contexto es lo suficientemente cambiante como para valorar la reducción de la ventana.

Tengamos en cuenta que el proyecto implementado ha abarcado las 3 capas que forman los procesos de B.I; datos, análisis y visualización. Incluso para un proyecto de estas características, y con una ventana de 5 días, el Equipo de Desarrollo ha tenido que modificar varias veces el Sprint Backlog. Este hecho no es negativo, pero sí puede mostrar que las ventanas de 5 días no son ninguna exageración y que incluso ventanas menores permitirían al Equipo trabajar sin la necesidad de modificar la Pila del Sprint.

Otro aspecto que se ha podido comprobar es que trabajar con ventanas inferiores a 5 días puede causar rechazo para Equipos poco experimentados. El horizonte semanal en los proyectos parece una referencia bastante aceptada, en cambio alteraciones rebajando este horizonte pueden presionar a los colaboradores, debido al alto nivel de control que estas cortas ventanas suponen.

2. Habilitar la modificación del Sprint Backlog aun cuando el Sprint esté en curso:

Se ha comprobado que esta modificación es necesaria. La alternativa sería reducir las ventanas de los Sprints y ya hemos comentado que no todos los Equipos pueden trabajar inmediatamente de este modo. El problema identificado ha sido que el Product Owner no ha sido capaz de acordar las funcionalidades con los Stakeholders de manera clara y concisa. Además también debido a la cambiante evolución del Negocio durante las semanas de desarrollo del CdM, ha sido necesario modificar las funcionalidades del producto.

En este contexto la lección aprendida es que es necesario que el Scrum Master trabaje con el Product Owner, para que éste último entienda la importancia de su rol, cuando se trata de proyectos de B.I.

3. Llevar a cabo el Sprint Review de manera offline:

Se ha demostrado su total validez. Únicamente el Product Owner ha podido asistir presencialmente u de manera online, a un 25% de las sesiones de revisión. El proyecto hubiese sufrido grandes retrasos si no se hubiese probado a eliminar esta restricción que Scrum define originalmente.

4. Eliminación del Sprint Retrospective:

Se ha comprobado que el Equipo ha ido realizando retrospectiva sobre la evolución del proyecto en los Daily Scrum de manera espontánea. De este modo el Scrum Master únicamente ha necesitado no más de 15 minutos de retrospectiva sobre la evolución global del Sprint, en la reunión de revisión, habiendo incluso Sprints en los que no ha sido necesario.

Por tanto podemos concluir que pese a que no se desaconseja el uso de la retrospectiva global del Sprint en la reunión de revisión, parece innecesaria una reunión dedicada a la retrospección.

Conclusiones de las modificaciones a Kanban:

5. Medición del WIP por tareas en lugar de por funcionalidades:

Al reducir las ventanas de los Sprints el número de funcionalidades por Sprint se reduce. Además y como es el caso de este proyecto, la implementación de una funcionalidad (se pueden consultar en los ficheros "Datsets" aportados) implica trabajar en las 3 capas de los procesos de B.I, con lo que las funcionalidades implican una alta carga de Esfuerzo. Por tanto el escenario en los Sprints es de pocas funcionalidades aunque muy exigentes.

En este contexto el hecho de haber limitado el WIP (work in progress) según tareas ha funcionado. La limitación ha ayudado al Equipo a avanzar eficientemente en el proyecto tal y como se ha contado en el apartado; 2.4.3.4.4 Ratio Cycle Time. En caso de haber optado por utilizar el WIP según funcionalidades los integrantes del Equipo no hubiesen colaborado para terminar las Historias de Usuario "atascadas", con la consiguiente pérdida de eficiencia.

A continuación pasaremos a analizar las conclusiones relativas al Cuadro de Mando.

Recordemos cuáles eran los objetivos del CdM:

1. Maximizar la productividad (Valor / Esfuerzo) en las primeras iteraciones (Sprints) del proyecto

Los dos primeros objetivos han sido alcanzados según hemos podido comprobar en el apartado; 2.4.3.1 Priorización y alertas en el Product Backlog.

En cuanto al primer objetivo, la Ilustración 40: Grafico auxiliar (1) del bloque Product Backlog en el Sprint 8, muestra como el

sistema de priorización de Historias de Usuario realmente sí maximiza el Valor en las primeras ejecuciones o prioridades que restan por acometer. Se ha podido corroborar con el Product Owner, que las Historias de Usuario han sido entregadas según el orden lógico de retorno de Valor (teniendo en cuenta las consideraciones de Esfuerzo del Equipo).

2. Alertar del riesgo a posibles desviaciones sobre lo planificado

En cuanto al segundo objetivo, la Ilustración 27: Product Backlog inicial priorizado, muestra como el sistema de alertas indica cuáles son las tareas que van a necesitar de una mayor atención. En la implementación real del proyecto, estos avisos han permitido aumentar el dimensionamientos del Equipo de Desarrollo en momentos puntuales, pivotando de 3 a 4 colaboradores implicados.

3. Estimar la fecha prevista de finalización del proyecto

El tercer objetivo ha sido alcanzado a través de los gráficos Burn Up, según hemos ido comprobando a lo largo de la exposición del punto; 2.4.3 Resultado de la ejecución del caso práctico en el cuadro de mando. Buena muestra de ello son las ilustraciones:

- 37: Gráfico Burn Up Esfuerzo al final del Sprint 7
- 42: Gráfico Burn Up Esfuerzo después de la ampliación del Product Backlog
- 45: Gráfico Burn Up Esfuerzo Sprint 9

En ellas la previsión de finalización ha ido variando desde 10 Sprints, pasando por 12, y finalmente hasta quedarse en 11.

4. Apoyar a la toma de decisión de cese temprano del proyecto

Por último, el cuarto objetivo se alcanza a través de un análisis conjunto de; gráficos Burn UP, Ratios (2.4.3.4 Ratios) y Dashboard (2.4.3.5 Dashboard).

En el caso de la implementación de este proyecto a través del CdM, el análisis conjunto de estos 3 bloques nos muestra que el Equipo ha alcanzado un nivel de madurez sobre el proyecto muy alto, pues los ratios reflejan una alta eficiencia en el trabajo; Ratio Cycle Time por debajo de 1 y Factor Foco igual a 100%.

La productividad tal y como se puede observar en el Dashboard va a caer un -9% desde el 1.7 hasta el 1.5 al final del proyecto. Además en la gráfica de Productividad observamos que ésta empezó valiendo 3. El Dashboard también nos informa que el

esfuerzo que supone terminar el proyecto es mayor al valor que aporta su culminación.

Con todo, podríamos concluir que cabría barajar la posibilidad de cese temprano de este proyecto, o cese momentáneo, en caso que surgiese la posibilidad de empezar otro nuevo proyecto con mayor potencial de aportación de valor. En tal caso, las funcionalidades del producto pendientes por implementar, podrían ser retrasadas hasta que el Equipo fuese liberado de su nuevo proyecto.

3. Conclusiones generales

3.1 Lecciones aprendidas

La principal lección aprendida es que el enfoque ágil realmente aporta valor en los procesos de B.I. Fruto del estudio y la aplicación de las metodologías ágiles a través del Cuadro de Mando, se ha podido comprobar la idoneidad de estos enfoques para proyectos de inteligencia de negocio.

Otra lección aprendida ha sido la derivada de la gestión de la agilidad en un caso real. Pensar en maximizar la aportación de valor al Product Owner en lugar de entregar productos “mediocres” aunque en tiempo y forma, simplemente por cumplir la planificación.

Por último, la necesidad de una buena planificación. Aunque parezca antagónico a la agilidad, una buena planificación es esencial para maximizar la consecución de los objetivos de cualquier proyecto. De hecho la agilidad fomenta la adaptación frente al seguimiento rígido de lo planificado, no obstante esto no significa que no deba de existir un plan.

3.2 Consecución de objetivos

El principal objetivo de este TFM era la aportación de un marco o framework ágil, dónde las áreas de B.I puedan desarrollar sus proyectos.

Para cumplir con el objetivo, se ha planteado una solución que va desde lo procedimental hasta lo práctico. Ambas partes han sido tratadas a partir del estudio de las metodologías ágiles y de mi propia experiencia como profesional del business intelligence.

El resultado ha sido el desarrollo de un proyecto real de B.I a partir del método ágil desarrollado, y apoyado en la principal herramienta o producto del TFM; el Cuadro de Mando. Si bien es cierto que el CdM no ha estado disponible con sus funcionalidades totales, durante buena parte del transcurso del proyecto, el enfoque ágil en su construcción, ha permitido trabajar con él desde fases muy tempranas del proyecto.

Tal y como se ha podido comprobar en el apartado; 2.5 Conclusiones del método ágil y el CdM, ambas propuestas han tenido un impacto positivo en el transcurso del proyecto utilizado como prueba. Por tanto podemos concluir que los dos principales productos de este TFM han cumplido con su finalidad.

Otros objetivos del proyecto eran; profundizar en las metodologías ágiles y en concreto en Scrum y Kanban, y también se pretendía aplicar el método ágil en la gestión de este proyecto en particular. Todos estos

objetivos también han sido conseguidos. La clave bajo mi punto de vista ha sido plantear una ambición mixta, entre lo teórico y lo práctico, lo cual ha reforzado el interés por profundizar en los conocimientos, a la vez que su análisis crítico para su adaptación y mejora.

3.3 Planificación y metodología seguida

La planificación ha sido seguida a pesar que en determinados momentos ha tenido que ser modificada.

Inicialmente se pretendía realizar una validación sobre la idoneidad del Cuadro de Mando y del método seguido con los profesionales del sector. El objetivo era realizar encuestas sobre el valor que aportaban los dos principales productos del Trabajo. No obstante ya en fases tempranas se detectó que a pesar de la voluntad por realizar las encuestas, el transcurso del proyecto no invitaba al optimismo y se decidió cancelar su puesta en marcha.

Se propuso dotar de una cierta flexibilidad al proyecto ya que los matices sobre lo que se quería conseguir no estaban 100% cerrados. Por tanto se optó por trabajar con un enfoque ágil. Se realizó un primer prototipo del CdM en fases tempranas del proyecto. De este modo y mientras se avanzaba en estudio de las metodologías ágiles, se fue capaz de entender el alcance de lo que se quería trabajar.

Finalmente cabe destacar que se cometió un error al no planificar bien las últimas semanas del proyecto. Tal y como se indica en el apartado; 1.4 Planificación del Trabajo, no se ha podido trabajar en el proyecto durante la semana 15, debido a la necesidad de preparar las pruebas finales del máster. La modificación en la planificación ha sido ya en fases muy avanzadas del proyecto y ha supuesto un riesgo para su consecución. En la reflexión se expone como error, debido a que este hecho era conocido con antelación, y podría haber sido corregido en las fases iniciales del proyecto.

3.4 Líneas de trabajo futuro

Queda pendiente el estudio en mayor profundidad de otras metodologías ágiles no tan populares como son; XP, DSDM o Crystal, con la finalidad de analizar sus propuestas y la idoneidad de su adaptación al método propuesto para B.I.

También sería interesante trabajar en ampliar la actual limitación del CdM, recordemos que actualmente tan solo admite una ampliación del Product Backlog.

Además, una vez comprobada la utilidad del CdM, sería también interesante adaptarlo a un formato web, ya que debido a las restricciones temporales, esto no ha sido posible en este proyecto.

Por último, la validación empírica del proceso ágil planteado, sería otra de las futuras líneas a desarrollar. Podría ser muy útil de cara a extraer conclusiones más allá de la experiencia personal en este Trabajo. La propuesta sería presentar el método y el CdM a distintos profesionales del sector, para que pudiesen probar a implementar la metodología en sus equipos. Junto con grupos de control la medición y comparación de las distintas métricas presentadas en este Trabajo, podrían concluir la validación de este método.

4. Glosario

Agilidad: El movimiento agilista trata de favorecer un cambio de mentalidad en el sector del desarrollo de software, basado fundamentalmente en los valores y principios que emanan del Manifiesto Ágil.

Business Intelligence (B.I): Conjunto de estrategias y aspectos relevantes enfocados a la administración y creación de conocimiento sobre el medio, a través del análisis de los datos existentes en una organización o empresa.

Burn Down chart: Representación gráfica del trabajo por hacer en un proyecto en el tiempo. Representa una serie temporal del trabajo pendiente.

Burn Up chart: Muestra el trabajo aportado por un equipo, o varios, a un proyecto ágil.

Cuadro de mando (CdM): Herramienta de medición del rendimiento para la gestión de empresas.

Cycle Time: Número de días laborables que transcurren desde que una Historia de Usuario es iniciada, hasta que es considerada como terminada.

Dashboard: Tablero de instrumentos o de indicadores.

Equipo Scrum: Los profesionales que desempeñan el trabajo de entregar un incremento de producto al final de cada Sprint.

Esfuerzo: Métrica que pondera el coste en tiempo de implementar una Historia de Usuario.

Factor de Priorización: Factor de ponderación de las Historias de Usuario en función de las métricas de Esfuerzo y Valor.

Factor Foco (%): Cociente entre la velocidad real y la velocidad ideal de las iteraciones.

Framework: Conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

Historia de Usuario: Descripción de las necesidades que el cliente o el usuario quiere que se implemente.

Kanban: Técnica para controlar el avance de la producción.

Manifiesto Ágil: Documento que resume los principios que rigen los métodos ágiles.

Predictibilidad: Técnica de gestión de proyectos que se basa en la planificación, en la división de un proyecto en fases dotándolas de una cierta rigidez, en la secuenciación de las distintas etapas de un proyecto haciendo necesario terminar exitosamente una etapa antes de iniciar la siguiente.

Product Backlog: Conjunto de funcionalidades descritas por el Product Owner a través de las Historias de Usuario y que deben ser implementadas a lo largo de los diferentes Sprints.

Product Owner: El dueño del producto. Representa a una única persona con una visión muy clara de aquello que se quiere desarrollar. Es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. Es la única persona habilitada para modificar el Product Backlog.

Productividad: Cociente entre el Valor entregado y el Esfuerzo que se ha realizado.

Ratio Cycle Time: Cociente entre el Cycle Time y los días de Sprint.

Scrum: Conjunto de buenas prácticas que proporciona un marco para la gestión de proyectos.

Scrum Master: Líder al servicio del Equipo Scrum. Por una parte interactúa con el Equipo de Desarrollo para asegurarse que Scrum es implementado de manera correcta, por otra, ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no.

Sprint: Bloque de tiempo normalmente definido entre 1 y 4 semanas durante el cual se crea un incremento del producto.

Sprint Backlog: Es la pila de elementos pendientes del Sprint. Formado por los elementos del Product Backlog que se entregarán una vez finalice el Sprint.

Sprint Daily Meeting: Reunión de no más de 15 minutos para sincronizar las actividades del Equipo y para crear un plan para las siguientes 24 horas.

Sprint Planning Meeting: Reunión de planificación del Sprint. Se planifica el trabajo a realizar durante el Sprint.

Sprint Retrospective: Se realiza al final del Sprint. La reunión sirve para que tanto el Equipo como el Scrum Master intercambien impresiones sobre el trabajo que se ha realizado con el objetivo de mejorar los

aspectos que no han ido del todo bien de cara a las siguientes iteraciones

Sprint Review: La reunión de revisión del Sprint se realiza al final del Sprint. Sirve para inspeccionar el incremento producido en el producto.

Stakeholders: Inversores o interesados en el producto.

Tares Críticas: Alerta de la criticidad de la Tarea con respecto al total del Product Backlog. Representa el 25% de las Tareas no iniciadas con mayor Factor de Criticidad.

Valor: Métrica que pondera la utilidad de la Historia de Usuario.

Velocidad Ideal: Ritmo de trabajo estimado al planificar el Sprint y medido en unidades de persona/hora.

Velocidad Real: Ritmo de trabajo real obtenido al realizar el seguimiento del Sprint y medido en unidades de persona/hora.

WIP (work in progress): Trabajo en curso, elementos que pueden estar en progreso en cada estado de Kanban.

5. Bibliografía

1. **Libro:** Ken Collier, Agile Analytics, A Value-Driven Approach to Business intelligence and Data Warehousing, Addison-Wesley, 2012
2. **Libro:** Jurgen Appelo, Management 3.0, Leading Agile Developers Developing Agile Leaders, Addison-Wesley, 2011
3. **Libro:** Mike Cohn, User Stories Applied For Agile Software Development, Addison-Wesley, 2004
4. **Libro:** Asim Abdel Rahman El Sheikh & Mouhib Alnoukari, Business intelligence and Agile Methodologies for Knowledge-Based Organizations: Cross-Disciplinary Applications, IGI Global Publishing, 2012
5. **Libro:** Henrik Kniberg & Mattias Skarin, Kanban y Scrum – obteniendo lo mejor de ambos, C4Media, 2010
6. **Libro:** Henrik Kniberg, Lean from the Trenches - Managing Large Scale Projects With Kanban, The Pramatic Programmers, 2011
7. **Libro:** Mike Cohn, Lean from the Trenches - Agile Estimating and Planning, Prentice Hall, 2006
8. **Libro / Guía:** Ken Schwaber & Jeff Sutherland, La Guía de Scrum, 2013
9. **Libro:** Mark C. Layton, Agile Project Management For Dummies, John Wiley & Sons, 2012
10. **Libro:** Kent Beck - Mike Beedle - Arie van Bennekum - Alistair Cockburn - Ward Cunningham - Martin Fowler - James Grenning - Jim Highsmith - Andrew Hunt - Ron Jeffries - Jon Kern - Brian Marick - Robert C. Martin - Steve Mellor - Ken Schwaber - Jeff Sutherland - Dave Thoma - Agile Manifesto, 2001
11. **Materiales del curso:** Agilidad y Lean. Gestionando los proyectos y negocios del s. XXI, Javier Garzas, MOOC Miriada X, Febrero 2016
12. **Web:** www.javiergarzas.com, Febrero 2016
13. **Web:** ocw.mit.edu, Febrero 2016
14. **Web:** it.gwu.edu, Marzo 2016
15. **Web:** proyectosagiles.org, Marzo 2016
16. **Web:** agilemethodology.org, Marzo 2016

17. **Web:** www.projectmanagement.com, Marzo 2016
18. **Web:** www.odu.edu, Marzo 2016
19. **Web:** scrummethodology.com, Marzo 2016
20. **Web:** www.sau.edu, Abril 2016
21. **Web:** www.gestiondeproyectosit.es, Abril 2016
22. **Web:** www.ibm.com/developerworks/community/blogs/ambler/entry/metric_acceleration?lang=en, Abril 2016
23. **Web:** www.crisp.se/scrum, Abril 2016
24. **Web:** <http://www.desarrolloweb.com/articulos/roles-scrum>
25. **Artículo de revista:** Marko Ikonen- Elena Pirinen- Fabian Fagerholm- Petri Kettunen- Pekka Abrahamsson, On the Impact of Kanban on Software Project Work An Empirical Case Study Investigation, 2011 16th IEEE International Conference on Engineering of Complex Computer Systems
26. **Artículo de revista:** Peter Middleton and David Joyce, Lean Software Management: BBC Worldwide Case Study, IEEE TRANSACTIONS ON ENGINEERING MANAGEMENT, VOL. 59, NO. 1, FEBRUARY 2012