

Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

PROPUESTA PARA UN NUEVO ENFOQUE DE APLICACIÓN ÁGIL CON SCRUM, LSD Y KANBAN

JOSÉ IGNACIO GAETE RODRIGUEZ-PEÑA

Profesor Guía: **Rodolfo Villarroel Acevedo**

INFORME FINAL DEL PROYECTO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO CIVIL EN INFORMÁTICA

Julio 2016

Agradecimientos

Primero que todo agradezco a mi familia, que siempre estuvo ahí para apoyarme. A mi madre, que siempre llegaba a mi habitación con ese delicioso sándwich sorpresa en esos días en que el estudio me consumía a tal punto que me olvidaba hasta de comer, a mi hermana Bernardita, siempre preocupada por mí y preparada para darme un consejo cuando más lo necesitaba, y a mi hermana Francisca, que ante cualquier problema siempre estaba ahí con su mano extendida para ayudarme. También agradezco de manera muy especial a mi novia, que me soportó durante todos estos años y siempre estuvo ahí sin importar qué, pese a mis días de cansancios, pese a mis días de malas caras, pese a aquellas ocasiones en las que la universidad me absorbía completamente y no me comportaba como el buen novio que debía ser, siempre estuvo ahí, esperándome, y la amo profundamente por eso y mucho más. Igualmente agradezco a esos compañeros que estuvieron a mi lado a lo largo de toda esta travesía llamada Universidad, junto a quienes aprendí muchísimo, reí, sufrí, disfruté y a quienes hoy en día ya no reconozco como compañeros, sino como amigos. Finalmente, agradezco a mi profesor guía, ya que fue gracias a sus consejos que obtuve la inspiración que necesitaba para idear la temática de este estudio, y a mi profesor co-referente, por su preocupación constante, sus emails y porque, de no haber sido por él, no hubiese sido posible terminar este trabajo.

Índice

Resumen	v
Abstract.....	v
Lista de Figuras	vi
Lista de Tablas.....	ix
1. Introducción.....	1
2. Descripción del Problema.....	2
3. Objetivos.....	4
3.1. Objetivo General.....	4
3.2. Objetivos Específicos	4
4. SPEM.....	5
4.1. Características Generales.....	5
4.1.1. Representación de los Procesos.....	5
4.1.2. Importancia.....	6
5. Scrum.....	7
5.1. Teoría de Scrum.....	7
5.2. Características	7
5.3. Aplicación de Scrum.....	8
5.3.1. El equipo Scrum.....	8
5.3.1.1. Product Owner.....	8
5.3.1.2. Development Team.....	9
5.3.1.3. Scrum Master.....	9
5.3.2. Definición del Product Backlog	9
5.3.3. Definición y Planificación de Sprints	10
5.3.3.1. Sprint Planning	10
5.3.3.2. Daily Scrum.....	11
5.3.3.3. Development Work.....	11
5.3.3.4. Sprint Review	11
5.3.3.5. Sprint Retrospective	12
6. Lean Software Development (LSD).....	13
6.1. Principios de Lean	13

6.1.1. Eliminate Waste (Eliminar Residuos).....	13
6.1.2. Amplify Learning (Amplificar el Aprendizaje)	13
6.1.3. Decide as Late as Possible (Decide lo más Tarde Posible).....	13
6.1.4. Deliver as Fast as Possible (Entrega lo más Rápido Posible)	14
6.1.5. Empower the team (Potenciar al Equipo)	14
6.1.6. Build Integrity (Establecer Integridad)	14
6.1.7. See the Whole (Verlo todo).....	14
7. Kanban.....	15
7.1. Reglas Kanban	15
8. Ventajas y Desventajas Scrum, LSD y Kanban	17
8.1. Ventajas y Desventajas Scrum.....	17
8.2. Ventajas y Desventajas LSD.....	18
8.3. Ventajas y Desventajas Kanban.....	19
9. Comparación Scrum, LSD y Kanban	20
9.1. Análisis Scrum.....	20
9.2. Análisis Lean	20
9.3. Análisis Kanban.....	21
9.4. Similitudes	21
9.5. Tabla Comparativa.....	21
10. Propuesta	23
10.1. Aspectos a Destacar de Cada Metodología.....	23
10.2. Elaboración de la Propuesta.....	23
10.2.1. Modificaciones Lean	25
10.2.2. Modificaciones Kanban.....	25
10.3. Requisitos Básicos de Aplicación.....	27
11. Métricas	28
11.1. Cuantitativas	28
11.1.1. Lead Time	28
11.1.2. Touch Time	28
11.1.3. Velocidad	28
11.1.4. Requisitos no Completados.....	29
11.1.5. Velocidad de Estabilización	29
11.1.6. Eficiencia.....	29

11.2. Cualitativas	29
11.2.1. Satisfacción del Equipo de Trabajo.....	29
11.2.2. Facilidad de Adopción	29
11.2.3. Organización y Planificación	29
12. Estudio y Recabación de Datos	30
12.1. Método de Aplicación.....	30
12.2. Presentación de Datos Obtenidos	31
12.3. Análisis de Datos Obtenidos.....	31
12.3.1. Análisis de Datos Cuantitativos	31
12.3.1.1. Análisis Lead Time.....	31
12.3.1.2. Análisis Touch Time	32
12.3.1.3. Análisis Velocidad y Requisitos No Completados.....	32
12.3.1.4. Análisis Velocidad de Estabilización	33
12.3.1.5. Análisis Eficiencia.....	33
12.3.2. Análisis de Datos Cualitativos	34
12.3.2.1. Análisis Satisfacción del Equipo de Trabajo.....	35
12.3.2.2. Análisis Facilidad de Adopción.....	35
12.3.2.3. Análisis Organización y Planificación	35
13. Conclusiones.....	36
14. Referencias	39
Anexos	41
A: Sobre las Características de SPEM.....	41
A1. Jerarquía de Desglose del Trabajo	41
A2. Estructura de Paquetes	42
A3. Elementos de SPEM Utilizados	44
B: Modelamiento de Scrum con SPEM.....	46
B1. Definición de Actores	46
B2. Definición de Product Backlog	47
B3. Sprint Planning.....	48
B4. Daily Scrum	48
B5. Development Work	49
B6. Sprint Review.....	50
B7. Sprint Retrospective	50

C: Modelamiento de LSD con SPEM	51
C1. Definición de Actores	51
C2. Eliminate Waste	51
C3. Amplify Learning.....	52
C4. Decide as Late as Possible	53
C5. Deliver as Fast as Possible	53
C6. Empower the Team	54
C7. Build Integrity	54
C8. See the Whole	55
D: Resultados de Estudio.....	56
D1. Métricas Cuantitativas	56
D1.1. Datos Lead Time / Touch Time	56
D1.2. Datos de Incidencias por Iteración	64
D1.3. Datos de Incidencias No Completadas.....	68
D1.4. Datos Velocidad y Requisitos No Completados	72
D1.5. Datos Velocidad de Estabilización.....	73
D1.6. Datos Eficiencia	78
D2. Métricas Cualitativas	89
D2.1. Encuesta 1: Conocimientos previos	89
D2.2. Encuesta 2: Apreciaciones generales de la metodología utilizada	93

Resumen

El presente documento corresponde a un trabajo de investigación perteneciente al área de Ingeniería de Software y, específicamente, a las metodologías de desarrollo ágil. En él se muestra la revisión en detalle de tres metodologías ágiles existentes en la actualidad: Scrum, Lean Software Development y Kanban, las cuales servirán de base para la elaboración de una propuesta concerniente a un nuevo enfoque para el desarrollo ágil.

Para lograr esto, se comienza dando una descripción detallada de las tres metodologías en estudio, utilizando para ello un estándar de metamodelado de procesos denominado SPEM. Posteriormente, se analizan sus características y se establecen ventajas y desventajas de cada una de ellas, de modo de poder compararlas y así recopilar la información necesaria para establecer la nueva propuesta. Finalmente, se define un conjunto de métricas específicas que serán aplicadas en el curso Ingeniería Web de la Pontificia Universidad Católica de Valparaíso, con el fin de obtener datos cualitativos y cuantitativos que permitan corroborar los resultados de la investigación.

Palabras Clave: SPEM, Scrum, Lean Software Development, Kanban, Metodología Ágil.

Abstract

The present document corresponds to a research work belonging to the area of Software Engineering and, specifically, to the methodologies of agile development. It shows the detailed revision of three existing agile methodologies at present: Scrum, Lean Software Development and Kanban, which will serve as the basis for the development of a proposal concerning a new approach for the agile development.

To achieve this, it begins by giving a detailed description of the three methodologies under study, using a process metamodeling standard called SPEM. Subsequently, it analyses its characteristics and establishes advantages and disadvantages of each one of them, in order to being able to compared them and collect the necessary information to establish the new proposal. Finally, it defines a set of specific metrics that will be applies in the Ingeniería Web course of the Pontificia Universidad Católica de Valparaíso, in order to obtain qualitative and quantitative data to corroborate the results of the research.

Keywords: SPEM, Scrum, Lean Software Development, Kanban, Agile Methodology.

Lista de Figuras

Figura 2.1 Esquema de Metodología en Cascada.....	2
Figura 4.1 Marco de Trabajo de la Ingeniería de Procesos.....	5
Figura 4.2 Representación Básica de un Proceso en SPEM.....	6
Figura 7.1 Ejemplo Tablero Kanban.....	15
Figura 10.1 Visión General de la Propuesta.....	24
Figura 10.2 Propuesta Kanban - parte 1.....	25
Figura 10.3 Propuesta Kanban - parte 2.....	26
Figura 10.4 Propuesta Kanban - parte 3.....	26
Figura 10.5 Propuesta Kanban - parte 4.....	27
Figura A.1 Jerarquía de Desglose del Trabajo en SPEM.....	41
Figura A.2 Jerarquización de la Estructura de Paquetes.....	43
Figura A.3 Lista de Elementos SPEM Utilizados.....	44
Figura B.1 Definición de Roles Scrum.....	46
Figura B.2 Definición de Product Backlog.....	47
Figura B.3 Sprint Planning.....	48
Figura B.4 Daily Scrum.....	48
Figura B.5 Development Work.....	49
Figura B.6 Sprint Review.....	50
Figura B.7 Sprint Retrospective.....	50
Figura C.1 Definición de Actores en LSD.....	51
Figura C.2 Eliminate Waste.....	51
Figura C.3 Amplify Learning.....	52
Figura C.4 Decide as Late as Possible.....	53
Figura C.5 Deliver as Fast as Possible.....	53
Figura C.6 Empower the Team.....	54
Figura C.7 Build Integrity.....	54
Figura C.8 See the Whole.....	55
Figura D.1 Diagrama de Flujo Acumulado Grupo 1.....	57

Figura D.2 Diagrama de Flujo Acumulado Grupo 2.....	58
Figura D.3 Diagrama de Flujo Acumulado Grupo 3.....	59
Figura D.4 Diagrama de Flujo Acumulado Grupo 4.....	60
Figura D.5 Diagrama de Flujo Acumulado Grupo 5.....	61
Figura D.6 Diagrama de Flujo Acumulado Grupo 6.....	62
Figura D.7 Diagrama de Flujo Acumulado Grupo 7.....	63
Figura D.8 Gráfico Incidencias por Iteración Grupo 1.....	64
Figura D.9 Gráfico Incidencias por Iteración Grupo 2.....	64
Figura D.10 Gráfico Incidencias por Iteración Grupo 3.....	65
Figura D.11 Gráfico Incidencias por Iteración Grupo 4.....	65
Figura D.12 Gráfico Incidencias por Iteración Grupo 5.....	66
Figura D.13 Gráfico Incidencias por Iteración Grupo 6.....	66
Figura D.14 Gráfico Incidencias por Iteración Grupo 7.....	67
Figura D.15 Gráfico Incidencias No Completadas Grupo 1.	68
Figura D.16 Gráfico Incidencias No Completadas Grupo 2.	68
Figura D.17 Gráfico Incidencias No Completadas Grupo 3.	69
Figura D.18 Gráfico Incidencias No Completadas Grupo 4.	69
Figura D.19 Gráfico Incidencias No Completadas Grupo 5.	70
Figura D.20 Gráfico Incidencias No Completadas Grupo 6.	70
Figura D.21 Gráfico Incidencias No Completadas Grupo 7.	71
Figura D.22 Pizarra Estilo Kanban en JIRA.	73
Figura D.23 Gráfico Velocidad de Estabilización Grupo 1.....	74
Figura D.24 Gráfico Velocidad de Estabilización Grupo 2.....	74
Figura D.25 Gráfico Velocidad de Estabilización Grupo 3.....	75
Figura D.26 Gráfico Velocidad de Estabilización Grupo 4.....	75
Figura D.27 Gráfico Velocidad de Estabilización Grupo 5.....	76
Figura D.28 Gráfico Velocidad de Estabilización Grupo 6.....	76
Figura D.29 Gráfico Velocidad de Estabilización Grupo 7.....	77
Figura D.30 Información Proporcionada por JIRA de Cómo Leer el Gráfico.	78
Figura D.31 Gráfico Eficiencia Grupo 1.	79

Figura D.32 Gráfico Eficiencia Grupo 2.	80
Figura D.33 Gráfico Eficiencia Grupo 3.	81
Figura D.34 Gráfico Eficiencia Grupo 4.	82
Figura D.35 Gráfico Eficiencia Grupo 5.	83
Figura D.36 Gráfico Eficiencia Grupo 6.	84
Figura D.37 Gráfico Eficiencia Grupo 7.	85
Figura D.38 Pregunta 1: Abierta.	89
Figura D.39 Pregunta 2: Elección Única (Dicotómica).....	90
Figura D.40 Pregunta 3: Elección Múltiple.....	90
Figura D.41 Pregunta 4: Elección Única (Politémica).	91
Figura D.42 Pregunta 5: Elección Única (Politémica).	91
Figura D.43 Pregunta 6: Elección Única (Politémica).	92
Figura D.44 Pregunta 7: Escala Numérica.	92
Figura D.45 Pregunta 1: Escala Numérica.	93
Figura D.46 Pregunta 2: Elección Única (Dicotómica).....	93
Figura D.47 Pregunta 3: Escala Numérica.	94
Figura D.48 Pregunta 4: Escala Numérica.	94
Figura D.49 Pregunta 5: Escala Numérica.	95
Figura D.50 Pregunta 6: Escala Numérica.	96
Figura D.51 Pregunta 7: Escala Numérica.	96
Figura D.52 Pregunta 8: Escala Numérica.	97
Figura D.53 Pregunta 9: Elección Única (Dicotómica).....	97
Figura D.54 Pregunta 10: Escala Numérica.	98
Figura D.55 Pregunta 11: Escala Numérica.	98
Figura D.56 Pregunta 12: Escala Numérica.	99
Figura D.57 Pregunta 13: Abierta.	99
Figura D.58 Pregunta 14: Abierta.	100
Figura D.59 Pregunta 15: Abierta.	100
Figura D.60 Pregunta 16: Abierta.	100

Lista de Tablas

Tabla 8.1 Ventajas y Desventajas Scrum.	17
Tabla 8.2 Ventajas y Desventajas LSD.	18
Tabla 8.3 Ventajas y Desventajas Kanban.	19
Tabla 9.1 Tabla Comparativa de Scrum, Lean y Kanban.....	21
Tabla 12.1 Resumen Datos Eficiencia.....	33
Tabla 12.2 Análisis de Datos Cualitativos.....	34
Tabla D.1 Cálculo Lead Time / Touch Time Grupo 1.	57
Tabla D.2 Cálculo Lead Time / Touch Time Grupo 2.	58
Tabla D.3 Cálculo Lead Time / Touch Time Grupo 3.	59
Tabla D.4 Cálculo Lead Time / Touch Time Grupo 4.	60
Tabla D.5 Cálculo Lead Time / Touch Time Grupo 5.	61
Tabla D.6 Cálculo Lead Time / Touch Time Grupo 6.	62
Tabla D.7 Cálculo Lead Time / Touch Time Grupo 7.	63
Tabla D.8 Resumen Gráficos Incidencias por Iteración.....	67
Tabla D.9 Resumen Gráficos Incidencias No Completadas.....	71
Tabla D.10 Cálculo de Velocidad y Requerimientos No Completados.	72
Tabla D.11 Resumen Gráficos Velocidad de Estabilización.....	77
Tabla D.12 Análisis General Gráficos Eficiencia.	86

1. Introducción

En el ámbito de la Ingeniería de Software, la definición moderna de desarrollo ágil evolucionó a mediados de la década de los 90's a causa del descontento existente por las metodologías en cascada que dominaban el mercado, pues se consideraban demasiado estructuradas y estrictas. Este nuevo enfoque de desarrollo apuntaba a un desarrollo iterativo e incremental, donde los requisitos y soluciones evolucionan en el tiempo según las necesidades del proyecto y en donde la colaboración de los equipos de trabajo es esencial para el cumplimiento de los objetivos [1].

En la actualidad, pese a que ya han transcurrido más de 20 años desde su nacimiento, las metodologías ágiles están todavía en una etapa muy temprana de su desarrollo, y aun así han podido demostrar que su aplicación tanto en las áreas de producción como, más recientemente, en el área del desarrollo de software son inmensamente útiles para flexibilizar los procesos, unificar a los miembros de los equipos de trabajo y disminuir las deficiencias que pudiesen aumentar los costos de producción o desarrollo, tal como se puede apreciar en [2]. No obstante, las metodologías ágiles existentes actualmente, si bien ofrecen una gran cantidad de beneficios, también tienen muchas debilidades, y es justamente en este punto en el que se enfoca este trabajo.

Se han estudiado tres metodologías que están dentro de las más populares para la aplicación ágil actualmente: Scrum, Lean Software Development y Kanban, todas ellas con sus ventajas y desventajas, de modo de poder determinar de qué forma se pueden integrar sus ventajas y minimizar sus desventajas para obtener un mejor resultado a la hora de aplicarlas.

Para lograr esto, se comienza dando un vistazo al funcionamiento y características de SPEM, el cual proporciona un estándar de metamodelado que sirve para representar procesos en Ingeniería de Software y con el que posteriormente se modelarán las metodologías a estudiar. Luego, se realiza un análisis detallado de las tres metodologías con sus respectivos modelamientos y se establecen ventajas y desventajas de cada una de ellas, con el fin de determinar qué factores son los más útiles y esenciales entre las tres. A continuación, se realiza una comparación entre las metodologías, con el objetivo de extraer la información necesaria para establecer la nueva propuesta y, finalmente, se definen métricas enfocadas a la recopilación de datos cualitativos y cuantitativos que permitan poner a prueba los resultados de la investigación. Dichas métricas serán aplicadas en el curso de Ingeniería Web impartido por el profesor Ismael Figueroa Palet con quien se realizará un trabajo en conjunto para extraer los datos que sean necesarios a través de la información proporcionada por los alumnos.

2. Descripción del Problema

Durante los últimos 50 años, los diseñadores han estructurado el proceso de un proyecto distribuyendo cada una de sus partes en forma de cascada. Primero se definen, analizan y sintetizan los requerimientos del proyecto que se pretende realizar, se plantean supuestos y se valoran, se realiza la implementación de los requerimientos y finalmente, se entrega el producto final. Esta metodología proyectual define al diseño como un proceso secuencial que va en una sola dirección, algo que los diseñadores saben bien que no es posible. Lo cierto es que el desarrollo de proyectos nunca es secuencial puesto que la aparición de cambios a medida que se avanza supone ratificar pasos anteriores [3]. La Figura 2.1 muestra gráficamente el concepto de metodología en cascada.

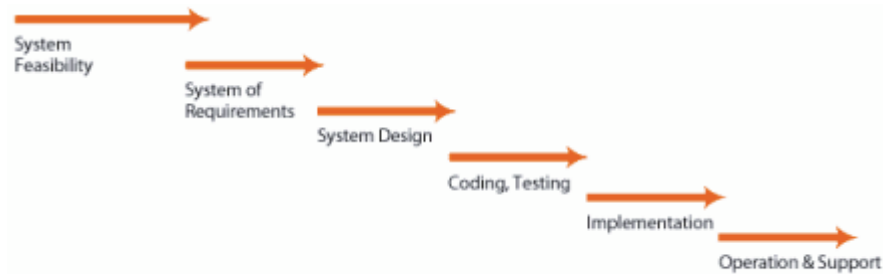


Figura 2.1 Esquema de Metodología en Cascada [3].

La planificación del proceso de diseño tiene que ser flexible ya que, si esperamos a los estados finales del proyecto para valorarlo, la introducción de cambios generará un gran impacto, tanto económico como temporal, puesto que a veces hay que redefinir conceptos de las fases más tempranas, lo que puede suponer empezar de nuevo. Es de ahí que nace lo que hoy se conoce como metodología ágil, que se define como un marco de trabajo cuyos principios son el valor del Work in Progress, la responsabilidad de cada individuo durante el desarrollo y una actitud positiva de respuesta frente al cambio [3].

Dentro de las metodologías ágiles, una de las más aceptadas y usadas actualmente se conoce con el nombre de Scrum [4]. En ella, el proyecto se planifica en diversos bloques temporales o iteraciones llamados Sprints, que se pueden entender como miniproyectos que deben proporcionar un resultado completo que el cliente ya pueda empezar a utilizar. La idea es que en cada iteración el equipo evolucione el proyecto de forma incremental a partir de los resultados obtenidos en iteraciones anteriores. Además, Scrum propone que durante el desarrollo de cada iteración, al comenzar cada nuevo día de trabajo, se haga una reunión de 15 minutos en la que cada integrante del equipo exponga en qué punto se encuentra su desarrollo, hacia dónde tiene pensado orientar los próximos pasos y las dificultades u oportunidades con las que se está encontrando [3].

El gran problema de Scrum, sin embargo, es que se presenta como un conjunto de reglas obligatorias, lo que puede limitar mucho la adaptabilidad del modelo a los distintos proyectos que los equipos de trabajo deben enfrentar. Muchas de sus reuniones diarias en realidad son innecesarias y varias de sus reglas, dependiendo de la situación particular de cada proyecto,

puede llegar a ser contraproducente para la eficiencia de los procesos que conforman su desarrollo [5].

Pero Scrum no es la única metodología ágil que existe, Lean Software Development, por ejemplo, es una metodología notoriamente más flexible que Scrum, la cual ofrece un conjunto de recomendaciones altamente adaptables enfocadas principalmente a la eliminación de residuos y el mejoramiento del flujo de trabajo [5]. Kanban por su parte, es otra metodología que se basa en una idea muy simple: el trabajo en curso debería limitarse y sólo se debería comenzar con algo nuevo cuando el bloque de trabajo actual haya sido entregado o haya pasado a la siguiente fase del flujo de trabajo [12].

Todas estas metodologías tienen sus pros y contras, pero hasta la fecha no existe ninguna opción que proporcione una forma de integrar lo mejor de cada una de ellas. Es por esto que la idea del presente proyecto es elaborar una propuesta para un nuevo enfoque de aplicación ágil a partir de las tres metodologías mencionadas anteriormente.

3. Objetivos

3.1. Objetivo General

Elaborar una propuesta que permita definir un nuevo enfoque para la aplicación de metodologías ágiles a través de la integración de tres metodologías seleccionadas: Scrum, Lean Software Development y Kanban.

3.2. Objetivos Específicos

- Estudiar en profundidad las metodologías seleccionadas para la investigación.
- Realizar una descripción detallada de las metodologías y modelarlas utilizando el estándar SPEM.
- Identificar las deficiencias y virtudes de cada metodología.
- Realizar una comparación detallada de las metodologías seleccionadas.
- Elaborar una propuesta a través de los resultados de la comparación.
- Definir métricas que permitan comprobar las ventajas que la nueva propuesta proporciona con respecto a las metodologías estudiadas.
- Poner a prueba los resultados de la investigación a través de la aplicación de las métricas definidas.

4. SPEM

SPEM (Software & Process Engineering Metamodel) es un estándar de metamodelado que sirve para representar procesos de la Ingeniería de Software, entendiendo proceso (de software) como un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto de Software [6]. SPEM ha sido desarrollado y aprobado por la OMG (Object Management Group), consorcio internacional dedicado al cuidado y el establecimiento de estándares para las tecnologías orientadas a objetos [7].

Es gracias al gran potencial que tienen estándares como SPEM y su antecesor, MOF (Meta Object Facility) que ha surgido lo que hoy conocemos como Ingeniería de Procesos (ver Figura 4.1), cuyo objetivo es la definición, implementación, medición y mejora de los procesos de la Ingeniería de Software [7].

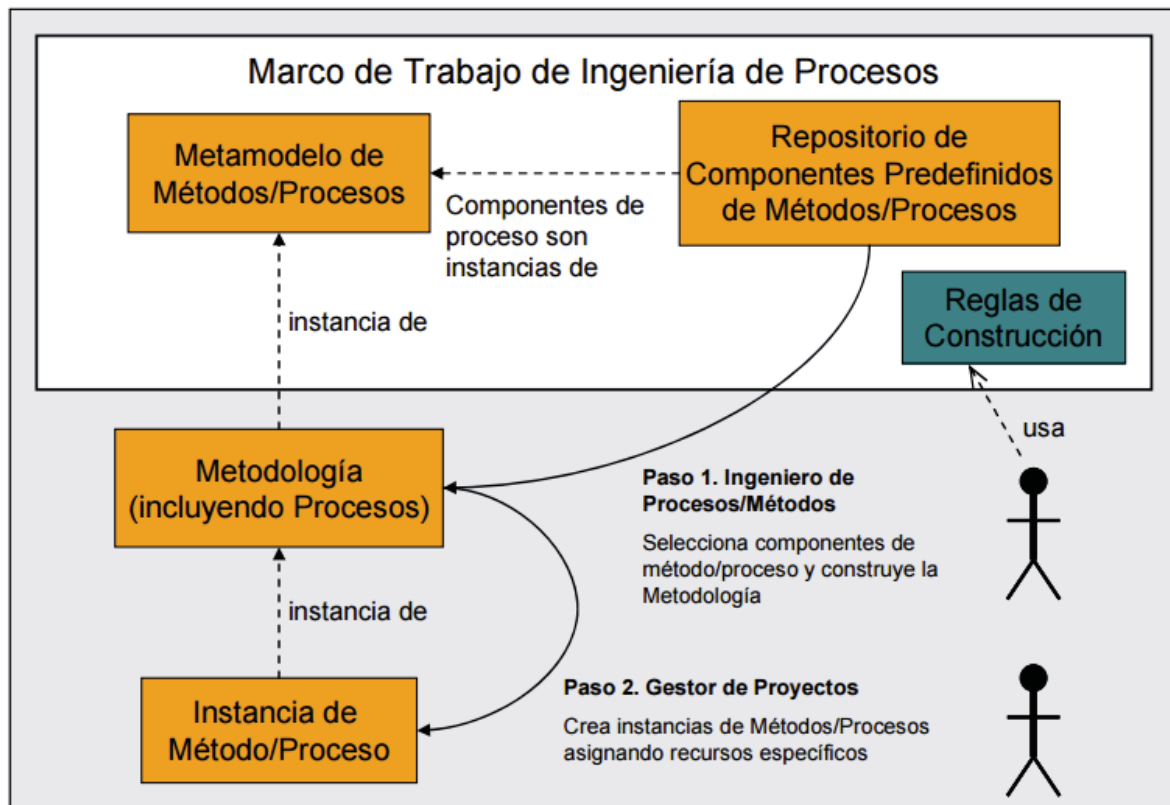


Figura 4.1 Marco de Trabajo de la Ingeniería de Procesos [6, 7].

4.1. Características Generales

4.1.1. Representación de los Procesos

Como ya se explicó anteriormente, SPEM sirve para definir procesos dentro de la Ingeniería de Software por lo que su alcance se limita a los elementos mínimos necesarios para definir dichos procesos sin añadir características específicas de un dominio o disciplina particular. La idea central de SPEM para representar procesos se basa en tres elementos básicos: tarea, rol y producto de trabajo (ver Figura 4.2) [7].

- **Tarea:** describe una unidad de trabajo asignable y gestionable, es decir, la unidad atómica de trabajo para definir procesos.
- **Rol:** representa a la persona o grupo de personas involucradas en el desarrollo de alguna tarea dentro de un proceso. Define el conjunto de habilidades y competencias y las responsabilidades asociadas al rol.
- **Producto de Trabajo:** representa las entradas que se utilizan en las tareas y las salidas que se producen. Existen tres tipos predefinidos de productos de trabajo: Artefacto (Artifact) que representan la naturaleza tangible del proceso (modelos, documentos, código, etc.); Entregable (Deliverable) que representa una salida de un proceso que tiene valor para un usuario, cliente u otro participante; y Resultado (Outcome) que representa la naturaleza intangible del proceso (estado de avance).

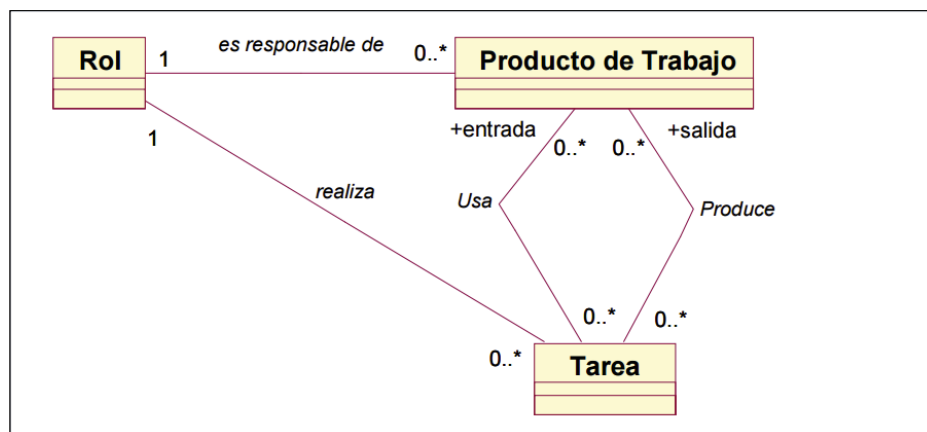


Figura 4.2 Representación Básica de un Proceso en SPEM [6, 7].

4.1.2. Importancia

SPEM provee un marco de trabajo conceptual que proporciona los elementos necesarios para modelar, documentar, presentar, publicar, intercambiar y gestionar los procesos de software. Además, puede ser un instrumento muy eficaz para resolver problemas relacionados con estos procesos, problemas como [7]:

- No se tiene un acceso fácil y centralizado a la información de los procesos que se necesitan.
- Diferentes desarrolladores manejan diferentes versiones de la misma información.
- Dificultad a la hora de combinar e integrar informaciones y procesos que están en diferentes formatos.
- La creación, revisión, reutilización, adaptación y generación de documentación para las metodologías suele ser un proceso manual.

Hasta aquí, se le ha entregado al lector una idea muy general del funcionamiento y características de SPEM, pero teniendo en consideración que SPEM no forma parte de los objetivos de la investigación y sólo se utiliza con el fin de modelar las metodologías en estudio, no se entregarán mayores detalles en esta sección. Si el lector está interesado en este tema y desea seguir leyendo acerca de él, puede dirigirse al Anexo A.

5. Scrum

Scrum es una metodología ágil de desarrollo de proyectos que toma su nombre y principios de los estudios realizados sobre nuevas prácticas de producción por Hirotaka Takeuchi e Ikujijo Nonaka a mediados de los 80 [8].

Jeff Sutherland aplicó el modelo al desarrollo de software en 1993 en Easel Corporation. En 1996 lo presentó junto con Ken Schwaber como proceso formal, también para gestión del desarrollo de software en la conferencia anual de investigación OOPSLA 96 (Object-Oriented Programming, Systems, Languages & Applications). Posteriormente, en 2001, serían dos de los promulgadores del Manifiesto Ágil y creadores de la famosa “Scrum Guide” [8].

Scrum es un framework de procesos que ha sido utilizado para gestionar el desarrollo de productos complejos desde principios de los años 90. No es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. Scrum muestra la eficacia de las prácticas de gestión de producto y las prácticas de desarrollo [9].

El marco de trabajo Scrum está conformado principalmente por los Equipos Scrum y sus roles, los eventos, los artefactos y sus reglas asociadas. Cada componente dentro del marco de trabajo sirve para un propósito específico y es esencial para el éxito de Scrum y para su uso [9].

5.1. Teoría de Scrum

Scrum se basa en la teoría de control empírica de procesos o empirismo, el cual asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en los que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control de riesgo. Tres pilares soportan esta teoría [9]:

- **Transparencia:** los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado.
- **Inspección:** los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo para determinar variaciones.
- **Adaptación:** si un inspector determina que uno o más aspectos de un proceso se desvían de los límites aceptables, y que el producto resultante será inaceptable, el proceso o el material que está siendo procesado deben ser ajustados.

5.2. Características

Scrum es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto y para controlar esta evolución emplea las siguientes prácticas de la gestión ágil [8]:

- **Revisión de iteraciones:** al finalizar cada iteración (normalmente 30 días) se lleva a cabo una revisión con todas las personas implicadas en el proyecto.

- **Desarrollo incremental:** al final de cada iteración se dispone de una parte operativa del producto que se puede inspeccionar y evaluar.
- **Desarrollo evolutivo:** los modelos de gestión ágil se emplean para trabajar en entornos de incertidumbre e inestabilidad. En Scrum se toma la inestabilidad como una premisa y se adoptan técnicas de trabajo para permitir esa evolución sin degradar la calidad de la arquitectura que se irá generando durante el desarrollo. El desarrollo Scrum va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto.
- **Auto-organización:** los equipos tienen un margen de decisión suficiente como para tomar las decisiones que consideren oportunas.
- **Colaboración:** las prácticas y el entorno de trabajo ágiles facilitan el trabajo en equipo. Éste es necesario para que funcione la auto-organización ya que cada miembro del equipo debe colaborar de forma abierta con los demás, según sus capacidades y no según su rol o su puesto.

5.3. Aplicación de Scrum

A continuación, se describirá en detalle el funcionamiento de la metodología Scrum a la hora de realizar proyectos de Software. En esta sección, sólo se presentará la descripción paso a paso de la aplicación de Scrum, pero en el Anexo B el lector podrá encontrar el modelamiento en SPEM de cada uno de los pasos que lo componen.

5.3.1. El equipo Scrum

El primer paso a la hora de implementar una metodología Scrum es seleccionar al equipo de Scrum. Este equipo está conformado por 3 Roles principales, el Product Owner (Propietario del Producto), el Development Team (Equipo de Desarrollo) y el Scrum Master. Este modelo de equipo está diseñado para optimizar la flexibilidad, la creatividad y la productividad durante el desarrollo de los proyectos, por lo que sus miembros deben ser autoorganizados y multifuncionales [9].

5.3.1.1. Product Owner

Es el responsable de maximizar el valor del producto y del trabajo desarrollado por el equipo de desarrollo. Es “la voz del cliente” y la única persona responsable de gestionar el Product Backlog (Lista de Objetivos). La gestión del Product Backlog incluye [9]:

- Expresar claramente los ítems incluidos en Product Backlog.
- Ordenar los elementos del Product Backlog de modo de poder alcanzar los objetivos de la mejor manera posible.
- Optimizar el valor del trabajo desempeñado por el Development Team.
- Asegurarse que el Product Backlog sea visible, transparente y claro para todos.
- Asegurarse que el Development Team entiende adecuadamente los elementos del Product Backlog.

El Product Owner es una única persona, no un comité, sin embargo, este podría representar los deseos de un comité en el Product Backlog. Para que el Product Owner pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones, por lo que debe ser elegido con cuidado [9].

5.3.1.2. Development Team

Son los profesionales encargados de entregar un incremento del producto “terminado”, es decir, que potencialmente se pueda poner en producción al final de cada Sprint. Sólo los miembros del Development Team participan en la creación de estos incrementos [9].

Los miembros de un Development Team deben ser estructurados, autoorganizados y multifuncionales y la organización en la cual se desenvuelven debe permitirles organizar y administrar su propio trabajo. Algunas de las características de estos equipos son [9]:

- Scrum no reconoce títulos para los miembros de un Development Team, todos son desarrolladores.
- Scrum no reconoce sub-equipos en el Development Team, no importan los dominios particulares que se deseen tomar en cuenta.
- La responsabilidad de las acciones de cualquier miembro del Development Team recae siempre en todo el equipo.

El tamaño óptimo de un Development Team se calcula considerando que sea lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad importante de trabajo. Por un lado, al tener menos de tres personas se reduce significativamente la interacción entre los miembros, lo que se traduce en una disminución en las ganancias de productividad, además, se podrían encontrar limitaciones en cuanto a las habilidades requeridas para llevar a cabo el Sprint. Por otro lado, tener más de nueve miembros, requiere demasiada coordinación y la complejidad de la gestión aumenta enormemente [9].

5.3.1.3. Scrum Master

Es el responsable de asegurar que Scrum sea entendido y adoptado, esto lo hacen asegurándose de que el Development Team se ajusta a las teorías, prácticas y reglas de Scrum. Es un líder que está al servicio del Equipo Scrum y ayuda a las personas externas al equipo a entender qué interacciones podrían ser de ayuda y cuáles no. [9]

El Scrum Master debe adoptar distintos roles dependiendo de con quien se relacione. Con el Product Owner debe ayudarlo a encontrar técnicas que le permitan gestionar adecuadamente el Product Backlog, ayudarlo a entender las bases de las metodologías ágiles y tener la capacidad de explicar la planificación del producto en un entorno empírico. Con el Development Team debe ser capaz de supervisarlos de tal modo que puedan generar productos de gran valor para el cliente, explicarles en caso de que no comprendan algo de la metodología y eliminar cualquier obstáculo que impida su progreso. Finalmente, si la relación es directamente con la organización, debe guiarla en todos los pasos necesarios para la correcta adopción de Scrum [9].

5.3.2. Definición del Product Backlog

Luego de identificar todos los roles que estarán involucrados en la elaboración del proyecto se debe realizar una reunión inicial con el cliente, en la cual deberán participar el Product Owner, el ScrumMaster y, en caso de ser necesario, algún analista o especialista en la toma de requerimientos. En esta reunión el cliente deberá explicar detalladamente todos los aspectos y características que espera obtener del producto que se desarrollará y el Product Owner en conjunto con los analistas y el Scrum Master deberán hacer uso de su experiencia y sus

capacidades de comunicación para consolidar esa información en un solo documento que deberá ser aceptado por todos los miembros de la reunión y que servirá como respaldo y guía durante todo el proceso. El Product Backlog es un documento que nunca está completo, el desarrollo más temprano del mismo solo refleja los requisitos conocidos y mejor entendidos al principio. El Product Backlog evoluciona a medida que el producto y el entorno en el que se usará también lo hacen, por lo tanto, debe ser dinámico y cambiar constantemente para identificar lo que el producto necesita [9].

5.3.3. Definición y Planificación de Sprints

El corazón de Scrum es el Sprint, el cual es un bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto “terminado”, es decir, utilizable y potencialmente desplegable [9].

Durante un Sprint [9]:

- No se deben realizar cambios que puedan afectar al objetivo del Sprint.
- Los objetivos de calidad no disminuyen.
- El alcance puede ser clarificado y renegociado entre el Product Owner y el Development Team a medida que se va adquiriendo mayor información sobre el proyecto.

Cada Sprint puede considerarse como un proyecto en sí mismo y su duración no puede exceder las cuatro semanas puesto que si el horizonte del Sprint es demasiado grande la definición de lo que se está construyendo podría cambiar, la complejidad podría aumentar y con ello, el riesgo de cambio sería mayor [9].

Los Sprints están divididos en Scrum Event o Eventos Scrum, y estos son: Sprint Planning (Planificación del Sprint), Daily Scrum (Scrum Diario), Development Work (Trabajo de Desarrollo), Sprint Review (Revisión del Sprint) y Sprint Retrospective (Retrospectiva del Sprint) [9].

5.3.3.1. Sprint Planning

La reunión de planificación de Sprint es aquella en donde se reúnen todos los miembros del Scrum Team con el objetivo de planificar el Sprint. Durante este evento se decide qué requerimientos o tareas del Product Backlog se le asignará a cada uno de los elementos del equipo y cada integrante deberá establecer el tiempo que crea prudente para llevar a cabo los requerimientos que le fueron asignados. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito, además les enseña a mantenerse dentro del bloque de tiempo [9].

La reunión de planificación de Sprint responde a las siguientes preguntas [9]:

- ¿Qué puede ser entregado en el incremento resultante del Sprint que comienza?
- ¿Cómo se logrará realizar el trabajo necesario para entregar el incremento propuesto?

Al finalizar esta reunión se obtendrá un Documento Sprint que respaldará todo lo dicho en ella, de modo de poder comprobar los logros obtenidos al finalizar el Sprint [9].

5.3.3.2. Daily Scrum

Es una reunión que debe realizarse diariamente en un tiempo máximo de 15 minutos y que tiene la finalidad de que los miembros del Development Team puedan sincronizar sus actividades y se cree un plan para las próximas 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Daily Scrum y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente [9].

El Daily Scrum se debe realizar todos los días a la misma hora y en el mismo lugar y durante la reunión cada miembro del Development Team deberá explicar [9]:

- ¿Qué hice ayer para ayudar al Development Team a cumplir los objetivos del Sprint?
- ¿Qué haré hoy para ayudar?
- ¿Veo algún impedimento que evite que el Development Team o yo logremos el objetivo del Sprint?

En un Daily Scrum, el Scrum Master se asegura de que el Development Team tenga la reunión, pero son estos los que la dirigen, nadie fuera de los miembros del Development Team puede participar en un Daily Scrum [9].

5.3.3.3. Development Work

Corresponde a todo el proceso de desarrollo necesario para finalizar todos los objetivos fijados en el Documento Sprint. Una vez finalizado el desarrollo se obtiene el Producto Sprint, el cual es un incremento totalmente funcional del producto final [9].

5.3.3.4. Sprint Review

Al finalizar un Sprint se debe realizar una inspección del Sprint, con el fin de analizar el incremento alcanzado y adaptar el Product Backlog si es que fuese necesario. Durante el Sprint Review el Scrum Team y sus interesados colaboran para determinar qué se hizo durante el Sprint, qué cambios o agregaciones se podrían realizar al Product Backlog y cuáles serán las siguientes tareas a realizar para poder optimizar el valor del proyecto para el cliente [9].

En el Sprint Review se incluyen las siguientes actividades [9]:

- El Product Owner explica qué elementos del Product Backlog se han terminado y qué elementos quedan pendientes y proyecta fechas probables de finalización del proyecto en base a los avances obtenidos.
- El Development Team analiza y determina qué cosas salieron bien durante el Sprint, qué cosas salieron mal, qué problemas aparecieron y cómo fueron resueltos esos problemas.
- El Development Team muestra a los demás participantes el trabajo que se pudo finalizar en el Sprint y responde preguntas acerca del incremento.
- El grupo completo da ideas y recomendaciones acerca de lo que se deberá hacer en el próximo Sprint.

El resultado de la Sprint Review es un Product Backlog revisado, que describe todos los elementos del Product Backlog que son posibles de realizar para el siguiente Sprint [9].

5.3.3.5. Sprint Retrospective

Sprint Retrospective es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismos y crear un plan de mejoras que sea abordado durante el siguiente Sprint. Este evento tiene lugar después del Sprint Review y antes del siguiente Sprint. El Scrum Master se asegura que el evento se lleve a cabo y que los asistentes entiendan el propósito del mismo, además debe participar activamente de él como un miembro del equipo puesto que la responsabilidad del proceso Scrum recae sobre él [9].

El propósito de la Sprint Retrospective es [9]:

- Inspeccionar cómo fue el último Sprint en cuanto a las personas, las relaciones, los procesos y las herramientas utilizadas.
- Identificar y ordenar los elementos que pueden tener alguna ineficiencia o se puedan mejorar de alguna forma.
- Crear un plan de implementación para las mejoras encontradas.

6. Lean Software Development (LSD)

LSD o simplemente Lean, es un conjunto de principios definidos por la industria de fabricación de automóviles de Japón en la década de los 80. El ingeniero de calidad de Toyota, John Krafcik, acuñó el término mientras observaba los procesos y herramientas utilizados para la eliminación de residuos en la producción de los automóviles. No fue sino hasta 2003 que María y Tom Poppendieck introdujeron Lean como un proceso de desarrollo de Software en su libro “Lean Software Development: An Agile Toolkit” [5].

6.1. Principios de Lean

A continuación, se describe los principios fundamentales de LSD, si el lector desea una visión más gráfica de estos principios, puede dirigirse al Anexo C, donde podrá encontrar una versión modelada con SPEM de los mismos.

6.1.1. Eliminate Waste (Eliminar Residuos)

En un sentido pragmático, todo lo que no es un valor directo para el cliente es considerado un residuo. Para ofrecer algunos ejemplos, un residuo puede ser: un error, un comentario en el código, una función sin usar, una llamada personal, etc. Cualquier cosa que un individuo o un equipo de trabajo hagan y no represente un valor para el cliente es un residuo, y se deben tomar las acciones apropiadas para eliminarlo [5].

6.1.2. Amplify Learning (Amplificar el Aprendizaje)

Si se han detectado residuos, lo más natural es esforzarse para que no se vuelvan a repetir las mismas ineficiencias. Pero ¿por qué hay residuos? Un factor muy recurrente es debido a que algún miembro del equipo no sabe muy bien cómo abordar un problema en particular. Eso está bien; nadie lo sabe todo, pero para solucionarlo se debe establecer un alto valor a la educación. Se deben identificar las áreas que necesitan la mayor mejoría y comenzar a capacitar. Cuanto más capacitado esté un equipo de trabajo, más fácil es reducir la cantidad de residuos producidos [5].

Otro medio importante para aprender y mejorar el conocimiento sobre el sistema es a través del feedback del usuario, el cual permite identificar cómo los usuarios utilizan el producto. Este conocimiento es inmensamente valioso ya que permite detectar errores que normalmente pasarían desapercibidos y con esto, minimizar los residuos del sistema [5].

6.1.3. Decide as Late as Possible (Decide lo más Tarde Posible)

Cada decisión tiene un costo, puede que no sea inmediato, pero existe. Aplazar decisiones ayuda a prepararse plenamente para los problemas a los que deba enfrentarse durante el desarrollo del proyecto. Probablemente el ejemplo más común de retraso en las decisiones está en el diseño de base de datos.

Por otro lado, los usuarios no siempre saben lo que quieren. Al aplazar las decisiones se puede esperar hasta que el usuario realmente necesite una nueva funcionalidad o cambio en el sistema y mientras tanto seguir adquiriendo nuevos conocimientos y datos sobre el problema, de modo que a la hora de necesitarse una modificación se dispondrá de mayor información y se podrá proporcionar la funcionalidad necesaria sin mucha dificultad [5].

6.1.4. Deliver as Fast as Possible (Entrega lo más Rápido Posible)

Vivimos en un mundo en constante cambio. Los programas que escribimos hoy son para los equipos que serán obsoletos en dos años. La ley de Moore sigue vigente, y seguirá siendo así.

La entrega de un producto en un plazo de tres años es demasiado tiempo para este mundo de creciente demanda, por lo que es muy importante dar valor a los clientes tan pronto como sea posible. La historia ha demostrado que un producto incompleto con una cantidad aceptable de errores es mejor que nada. Además, se tienen comentarios de los usuarios, que representan un valor incalculable para el desarrollo [5].

6.1.5. Empower the team (Potenciar al Equipo)

Lean identificó que los programadores desmotivados causan grandes residuos a la hora de desarrollar un producto. La razón es simple, la falta de motivación genera una disminución en el rendimiento de los trabajadores lo cual disminuye enormemente la calidad del producto final y aumentan los tiempos de desarrollo. Lean alienta a los gerentes a que escuchen a sus desarrolladores, y anima a los desarrolladores a que enseñen a sus directivos los procesos de producción de software [5].

6.1.6. Build Integrity (Establecer Integridad)

Se percibe que un sistema tiene integridad cuando el usuario piensa, “¡Sí! Eso es exactamente lo que quiero. La cuota de mercado es una medida aproximada de la integridad percibida de los productos, ya que mide la percepción del cliente a través del tiempo. La integridad conceptual significa que los conceptos centrales del sistema, trabajan en conjunto como un todo sin problemas y de forma cohesionada, y este es un factor crítico en la creación de la integridad percibida. El software necesita un nivel adicional de integridad que debe mantener su utilidad a lo largo del tiempo. Un software con integridad tiene una arquitectura coherente, puntuaciones altas en cuanto a la usabilidad y la aptitud para el uso, y es mantenible, adaptable y extensible [10].

6.1.7. See the Whole (Verlo todo)

La integridad en los sistemas complejos requiere de una amplia experiencia en diversas áreas. Uno de los problemas más difíciles con el desarrollo de productos es que los expertos en cualquier área (por ejemplo, bases de datos o GUI) tienen una tendencia a maximizar el rendimiento de la parte del producto que representa su propia especialidad en lugar de centrarse en el rendimiento general del sistema. Muy a menudo, el bien común se ve afectado si la gente solo se preocupa por sus intereses propios. Cuando los individuos u organizaciones se miden por su contribución especializada en lugar de rendimiento global, es probable que surja algo llamado suboptimización. Este problema es aún más pronunciado cuando dos organizaciones hacen una alianza de trabajo, porque la gente naturalmente va a querer maximizar el rendimiento de su propia empresa. Es difícil implementar prácticas que eviten la suboptimización en una organización grande, y es un orden de magnitud más difícil cuando hay alianzas involucradas [10].

7. Kanban

El término japonés Kanban, fue empleado por Taiichi Onho (Toyota), para referirse al sistema de visualización empleado en los procesos de producción que coordinan en una cadena de montaje la entrega a tiempo de cada parte en el momento que se necesita, evitando sobreproducción y almacenamiento innecesario de producto [11].

El término aplicado a la gestión ágil de proyectos se refiere a técnicas de representación visual de información para mejorar la eficiencia en la ejecución de las tareas de un proyecto. Las principales reglas de Kanban son las siguientes: visualizar el flujo de trabajo, determinar el límite de trabajo en curso y medir el tiempo en terminar una tarea [11].

7.1. Reglas Kanban

Visualizar el flujo de trabajo: Parece algo básico, pero no siempre se ven realmente las fases por las que pasa un proyecto ni qué personas trabajan en qué. Esto es muy habitual en grandes empresas, donde el desconocimiento entre el trabajo de otros equipos está a la orden del día. El método Kanban recomienda usar un panel con tarjetas (que dan nombre al método) denominado Tablero Kanban en el cual se define cada tarea dividiéndola en columnas que indican cada fase del proyecto. Este tablero es visible en todo momento para todos los miembros del equipo de desarrollo. En la Figura 7.1 se puede observar un ejemplo del Tablero Kanban [12].

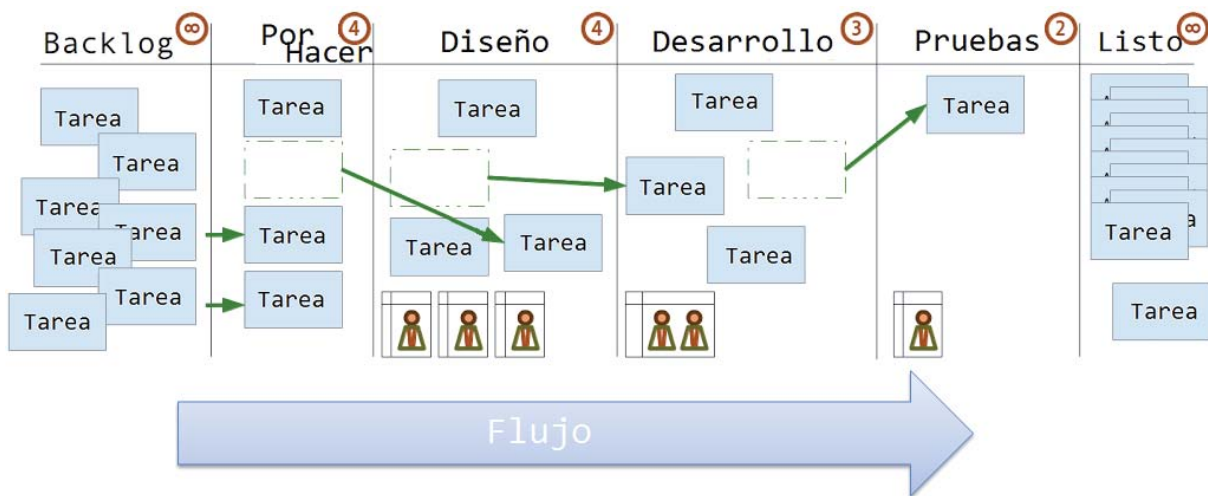


Figura 7.1 Ejemplo Tablero Kanban.

En primer lugar, se pueden observar las distintas columnas que conforman el tablero, las cuales reciben el nombre de lanes. Estas columnas o lanes representan las fases del proceso de desarrollo y son completamente adaptables según las necesidades de cada equipo de desarrollo. No hay límite en la cantidad de columnas, depende netamente de la situación particular de cada proyecto.

En segundo lugar, se aprecian las distintas tarjetas que están distribuidas a lo largo del tablero, estas representan las tareas o requerimientos que están siendo desarrollados, y

dependiendo del lane en el que se encuentran, es el nivel de avance de la tarea. Para distribuir y ordenar estas tarjetas es muy importante mantener siempre un orden y una visibilidad que sea entendible para todos, ya sea a través de la utilización de colores para categorizar los distintos tipos de tareas o a través de la priorización [12].

En cada lane además se pueden apreciar dos elementos característicos: por un lado, está el equipo de trabajo –representado por los símbolos con forma humana de color naranja que están en la parte inferior– y por el otro, el un número de color rojo en la parte superior que indica el límite de trabajo por lane. Finalmente, la flecha de la parte inferior representa el flujo de trabajo.

Limitar el trabajo en curso: Hacer muchas cosas, pero dejarlas todas a medias no sirve de nada. Si se empieza algo hay que terminarlo antes de empezar otra cosa, ése es un principio básico del método Kanban y uno de los pilares para que un proyecto funcione. Dentro del tablero Kanban, este límite está determinado por un número que es colocado en cada uno de los lanes, el cual indica el número máximo de tareas que pueden realizarse simultáneamente en cada lane. Este número además se debe fijar con mucho cuidado, ya que debe ser lo suficientemente grande para mantener a todos los miembros del equipo ocupados y a la vez lo suficientemente pequeño para evitar la priorización prematura [12].

Hacer Políticas Explícitas: Permite que todos los miembros del equipo tengan un entendimiento claro de cómo se trabaja. Puede definirse como una especie de criterio de aceptación y cada lane debe tener su propio criterio de aceptación, esto quiere decir que deben cumplirse condiciones específicas para que las tareas puedan pasar de un lane a otro [12].

Medir y Administrar el Flujo: Existen varios números o conceptos que nos permiten medir el ritmo al que el equipo está trabajando, entre ellos destacan el Lead Time (tiempo que transcurre desde que una tarea entra al Backlog hasta que se entrega), Cycle Time (lo que se demora efectivamente en desarrollarse una tarea, desde que entra a desarrollo), estadísticas, Cuadro de Flujo Acumulativo (permite conocer el estado en el que se encuentra el proyecto), entre otros [12].

Mejora Continua: Se debe mantener el tablero Kanban permanentemente actualizado de modo que cualquiera pueda mirarlo y saber en qué situación está el proyecto. De esta manera se podría analizar y determinar acciones para evitar que haya acumulaciones en los lanes o retrasos. Kanban sugiere la realización de reuniones para revisar y repriorizar las tareas [12].

8. Ventajas y Desventajas Scrum, LSD y Kanban

A continuación, se presentan todas las ventajas y desventajas que se han podido detectar de cada una de las metodologías en estudio. Cada una de estas características se ha obtenido a través del análisis en profundidad de las metodologías en cuestión y las Tablas 8.1, 8.2 y 8.3 lo muestran de manera ordenada.

8.1. Ventajas y Desventajas Scrum

Tabla 8.1 Ventajas y Desventajas Scrum.

Scrum	
Ventajas	Desventajas
Transparencia: El equipo de trabajo sabe lo que tiene que hacer cada día y el cliente sabe lo que se le entregará en cada Sprint.	Para aplicar Scrum se necesita un equipo muy organizado y disciplinado, de no ser así la metodología nunca podrá aplicarse correctamente.
Rápida detección y corrección de errores.	Si bien en un equipo de Scrum no es necesario que todos sus miembros posean una gran experiencia, tampoco puede ocurrir que ninguno de sus miembros sea experimentado. Siempre debe haber personas con experiencia en el equipo.
Se obtiene software ejecutable lo más rápido posible y éste cumple con los requerimientos más importantes.	Debido a las grandes exigencias a las que se ve sometido el equipo de Scrum para poder cumplir con los objetivos de los Sprint, sus miembros son muy propensos a estresarse.
Se trabaja en iteraciones cortas, lo que permite tener un alcance acotado y viable.	Para aplicar Scrum se recomiendan equipos de trabajo pequeños (entre 3 a 9 miembros para el equipo de desarrollo según la Scrum Guide [9]).
Se acepta la incertidumbre de los proyectos como una premisa y se adoptan estrategias para adaptarse a los cambios.	La evidencia o documentación que se genera al aplicar esta metodología es mucho menor que en otras, puesto que no se exige hacerlo (queda a criterio de cada equipo de Scrum).
Se incentiva la creatividad de los desarrolladores (auto-organización).	Scrum se presenta como un conjunto de reglas obligatorias, lo que a veces acarrea que se deje de lado la adaptabilidad que el manifiesto ágil entrega por seguirlas al pie de la letra.
Fácil visualización del avance de los proyectos.	Las reuniones diarias no siempre son eficientes.
Fácil de aprender e implementar.	

8.2. Ventajas y Desventajas LSD

Tabla 8.2 Ventajas y Desventajas LSD.

Lean Software Development	
Ventajas	Desventajas
La eliminación de los residuos conduce a la eficiencia global del proceso de desarrollo, y por ende, a la reducción de tiempos y costos.	El éxito del proyecto depende en gran medida de la cohesión del equipo de desarrollo y los compromisos individuales de sus miembros.
La entrega temprana del producto permite que se genere valor al cliente lo antes posible.	Ya que LSD se enfoca en la generación de valor, es fundamental trabajar rápido y en conjunto con el cliente, por lo que este deberá estar dispuesto a tomar muchas decisiones difíciles en poco tiempo.
El empoderamiento del equipo de desarrollo ayuda a mejorar la capacidad de toma de decisiones, lo cual a su vez crea un equipo más motivado.	Al ser una metodología cuyo objetivo es entregar rápidamente valor al cliente, esa misma rapidez trae como consecuencia que muchas veces no haya tiempo para realizar muchas pruebas o verificaciones.
Fomenta la capacitación de los miembros de los equipos de trabajo.	
Permite la creación de productos integrales y enfocados al cliente.	
Se promueve la buena relación con el cliente.	
En general los productos resultantes de un desarrollo bajo una metodología LSD son de mejor calidad (debido a la eliminación de residuos).	
Permite la integración de sistemas complejos y de ámbitos diferentes.	
Facilita el monitoreo de los procesos de desarrollo.	
Altamente adaptable. A diferencia de Scrum, Lean es una metodología que proporciona recomendaciones, no reglas.	

8.3. Ventajas y Desventajas Kanban

Tabla 8.3 Ventajas y Desventajas Kanban.

Kanban	
Ventajas	Desventajas
Otorga información siempre actualizada sobre el estado actual del proyecto.	Falta de reglas. Aunque la ausencia de reglas facilita bastante la implementación de la metodología, también representa un gran problema cuando existe algún desarrollador inexperto en el equipo, pues este no tendrá ninguna guía de referencia para trabajar.
Permite saber en todo momento quién está trabajando en qué cosa.	Dificultad a la hora de prever posibles problemas. Aunque la localización y solución de problemas en Kanban es sencilla, preverlos no lo es tanto, puesto que la metodología no ofrece ninguna herramienta para hacerlo.
Gran capacidad de detección de errores.	Dificultad de entregar a tiempo (en especial en grandes proyectos). Esto debido a que no hay un control específico del tiempo máximo que se debe emplear en cada actividad.
Permite controlar y limitar el trabajo en progreso.	Debido a que Kanban permite limitar el trabajo en progreso, a veces se pueden producir cuellos de botella entre los lanes de las tablas Kanban (por ejemplo, que un lane alcance su máxima cantidad de actividades simultáneas y una actividad finalizada del lane anterior deba quedar en espera).
Facilita la visualización del trabajo en curso durante todo el proceso de desarrollo.	
Casi no tiene reglas, por lo que es muy fácil de implementar.	
Promueve el trabajo en equipo.	
Reduce los tiempos muertos o de inactividad en el desarrollo.	
Gracias a la visualización que Kanban proporciona del flujo de trabajo, todos los miembros del equipo de desarrollo saben en todo momento qué actividades se están desarrollando, por lo que facilita bastante la detección e implementación de mejoras.	
Kanban además de una metodología, es también una herramienta, por lo que es muy fácil integrarla con otras metodologías.	

9. Comparación Scrum, LSD y Kanban

En base a todo lo ya estudiado y habiendo analizando las ventajas y desventajas de cada metodología, se puede apreciar que entre ellas hay una serie de características que podrían ser bastante útiles a la hora de integrarlas y elaborar la propuesta. Dichas características se explican a continuación:

9.1. Análisis Scrum

Scrum es una metodología que proporciona una serie de reglas y tareas específicas que se deben realizar en cada una de las iteraciones de un proyecto de software para asegurar la correcta elaboración del mismo. Esto da como resultado una metodología bastante sencilla de aplicar, puesto que sus reglas son claras y no se prestan a la subjetividad, pero también puede suponer una menor capacidad de adaptación a la hora de aplicarla.

La gran debilidad de Scrum radica principalmente en su rigidez, es decir, su poca flexibilidad en relación a otras metodologías ágiles. Esto se debe a que sus reglas en general son bastante estrictas y muchos amantes de esta metodología tienden a seguirlas al pie de la letra, ya que asumen que es la mejor forma de obtener el máximo provecho a un enfoque de desarrollo ágil, además, Scrum espera que lo hagan, sólo basta con pensar en el Scrum Master que, como se explicó en la sección 6, su principal función radica en asegurar que las reglas de Scrum se estén aplicando correctamente. Sin embargo, en ocasiones puede ocurrir que alguna de las reglas o tareas exigidas por Scrum no sea realmente necesaria para el desarrollo óptimo de un proyecto específico, lo que podría acarrear deficiencias que retrasen innecesariamente la elaboración del mismo, justamente lo que se desea evitar. Un ejemplo claro de esto son las Daily Scrum que, si bien es indudable su utilidad, en algunas circunstancias y dependiendo de la situación particular de cada proyecto, podrían no ser realmente necesarias o realizarse con menor frecuencia.

9.2. Análisis Lean

Lean ofrece una opción mucho más flexible a la proporcionada por Scrum, en la cual se entrega una serie de recomendaciones altamente adaptables cuyo objetivo radica en la generación de valor al cliente a través de la eliminación de residuos y entregas rápidas. Al ser recomendaciones en vez de reglas, presentan un alto grado de subjetividad y cada equipo de trabajo deberá decidir cómo aplicarlas. Esto supone una mayor dificultad en la aplicación de la metodología, pero al mismo tiempo, una mayor adaptabilidad y versatilidad en los desarrollos.

El mayor problema que presenta Lean está muy relacionado con su incesante afán por generar valor al cliente, ya que, muchas veces para lograrlo será necesario acelerar ciertas entregas o prescindir de ciertos elementos o tareas importantes con tal de proporcionarle un valor al cliente lo antes posible. Esto no es del todo malo, ya que gracias a las entregas rápidas el cliente puede estar más involucrado en los procesos de desarrollo y el equipo de trabajo podrá recibir un mayor feedback de él, pero a la vez, podría resultar en una disminución en la calidad del producto final.

9.3. Análisis Kanban

Como ya se ha explicado con anterioridad, Kanban cumple el rol tanto de metodología como de herramienta y su objetivo principal es la visualización del flujo de trabajo y la organización de las tareas. Al poseer características propias de una herramienta, es una metodología bastante fácil de implementar, sólo basta con entender su funcionamiento y asegurarse que todos los miembros del equipo de trabajo la apliquen correctamente. Además, gracias a sus cualidades de herramienta, es muy sencillo de integrar a otras metodologías.

El principal problema que presenta Kanban tiene que ver con su propiedad “limitar el trabajo en curso” ya que al ser aplicada de la forma que lo plantea Kanban, pueden generar cuellos de botella en el desarrollo. Lo anterior se explica de la siguiente forma: al trabajar con un tablero de Kanban, en cada una de las columnas (lanes) se asigna un número que representa la máxima cantidad de tareas que puede haber en dicho lane, y si en algún determinado momento la columna se llena, no se podrá recibir nuevas tareas hasta que se finalice alguna de las tareas en ejecución. Lo anterior es un gran problema puesto que, si alguno de los lanes de la tabla se llena, el lane anterior no podrá entregarle sus tareas finalizadas, lo que podría resultar en tareas inactivas, que son una gran ineficiencia para el proyecto.

9.4. Similitudes

Como punto de similitud entre las tres metodologías, se puede destacar lo siguiente:

- Se requiere un equipo de trabajo muy cohesionado.
- Es esencial que haya al menos un grupo de miembros experimentados en los equipos de trabajo, de modo que puedan ayudar y capacitar a los miembros con menos experiencia.
- Los miembros de los equipos de trabajo deben ser muy disciplinados, orientados al trabajo en equipo y con una gran capacidad de auto-organización.
- Se requiere una constante comunicación con el cliente.
- El cliente debe estar dispuesto a tener un rol participativo en los proyectos.
- Las tres son metodologías ágiles por lo que comparten los principios básicos del desarrollo ágil.

9.5. Tabla Comparativa

Tabla 9.1 Tabla Comparativa de Scrum, Lean y Kanban.

Factores Comparativos	Scrum	Lean	Kanban
Nivel de Interacción con el Cliente	Alto.	Alto.	Medio.
Complejidad de Uso	Media.	Alta.	Media.
Adaptabilidad	Media-Baja.	Alta.	Alta.
Características del Equipo de Trabajo	Altamente Cohesionado.	Altamente Cohesionado.	Altamente Cohesionado.

Cualidades necesarias para los miembros del Equipo de Trabajo	Disciplinados, autoorganizados, orientados al trabajo en equipo, altamente motivados.	Disciplinados, autoorganizados, orientados al trabajo en equipo, altamente motivados.	Disciplinados, autoorganizados, orientados al trabajo en equipo, altamente motivados.
Objetivo Principal	Brindar un marco de trabajo que ayude a agilizar los procesos que conforman un proyecto de software, utilizando una serie de reglas y tareas específicas divididas en iteraciones.	Generar la mayor cantidad posible de valor al cliente a través de sugerencias enfocadas a la eliminación de residuos y entregas rápidas.	Proporcionar un conjunto de normas que permitan mejorar la visualización de las tareas y el flujo de trabajo a través de la utilización del tablero Kanban.
Método de Aplicación	El proyecto se divide en Sprints. Los Sprints representan las iteración del proyecto y están conformados por un conjunto de tareas y reglas específicas.	El proyecto se divide en un conjunto de iteraciones cortas enfocadas a entregar rápidamente avances beneficiosos para el cliente y procurando eliminar todo lo que no sea útil para este.	Las tareas del proyecto se exponen en un tablero visible para todos los miembros del equipo de trabajo. El tablero se deberá mantener actualizado y tendrá reglas específicas.
Principal Ventaja	Se trabaja en iteraciones cortas denominadas Sprint, lo que permite tener un alcance acotado y viable. Además, al finalizar cada Sprint, el cliente recibirá un incremento funcional del producto final.	La eliminación de los residuos conduce a la eficiencia global del proyecto, lo que se traduce en una reducción de costos y tiempo. Además, la entrega temprana permite que se genere valor al cliente lo antes posible.	Otorga información siempre actualizada sobre el estado actual del proyecto y permite saber en todo momento quién está trabajando en qué, lo que favorece la organización y distribución de las tareas a realizar.
Principal Desventaja	Comparativamente a otras metodologías ágiles, Scrum presenta una gran rigidez relacionada principalmente con sus reglas, las cuales suelen ser bastante estrictas.	Para poder generar rápidamente valor al cliente, a veces será necesario prescindir de alguna tarea o adelantar alguna entrega, lo que puede resultar en una disminución en la calidad.	Debido a la propiedad de limitación del trabajo en curso, pueden producirse cuellos de botella durante las etapas de desarrollo.

10. Propuesta

10.1. Aspectos a Destacar de Cada Metodología

En base a la comparación realizada y a los análisis efectuados se han detectado los siguientes puntos:

- Scrum es una excelente metodología desde el punto de vista de la facilidad de uso, ya que proporciona un marco de trabajo completo y específico que no se presta a ambigüedades ni a errores de interpretación. Los equipos de trabajo saben exactamente cómo deben realizar cada iteración, cuáles son los pasos a seguir, qué reglas deben respetar, cual es el rango de tiempo que deberían asignar para la realización de cada actividad, etc. Sin embargo, el gran problema de esta metodología es que al ser tan estructurada pierde un poco de flexibilidad.
- Lean por su parte es una metodología mucho más compleja de implementar que Scrum, ya que sólo entrega recomendaciones, no instrucciones específicas. Esto da como resultado que se preste bastante a la subjetividad y a la interpretación de cada equipo de trabajo, pero lo compensa con una gran versatilidad, la posibilidad de crear software más económico y en menor tiempo y la opción de tener una relación mucho más cercana con el cliente gracias a las constantes entregas y generación de valor.
- Finalmente, Kanban se presenta como una metodología que además de establecer una serie de propiedades, proporciona una herramienta con la que se puede mejorar notablemente la visualización del flujo de trabajo dentro de los proyectos. Esto permite que Kanban, al igual que Scrum, sea bastante sencillo de implementar y, además, bastante sencillo de integrar con otras metodologías. El problema sin embargo radica en que Kanban puede producir cuellos de botella a la hora de ser aplicado, lo cual representa una gran deficiencia para el desarrollo.

10.2. Elaboración de la Propuesta

En base a lo explicado anteriormente se ha decidido elaborar una propuesta con las siguientes características:

- Se tomará una metodología Scrum clásica como base y se integrarán a ellas las virtudes ofrecidas por Lean y Kanban con el objetivo de mejorarla.
- De Lean se utilizarán sus siete recomendaciones básicas, con lo cual se pretende mejorar la flexibilidad ofrecida por Scrum. De esta forma, a la hora de aplicar el nuevo enfoque, no se ejecutarán todas las reglas de Scrum de manera tan estricta, sino que se flexibilizará su aplicación utilizando los principios de Lean. Para que esto funcione, es esencial estar siempre al tanto de qué cosas son las que realmente le generan valor al cliente y cuáles pueden considerarse como residuo, de modo de descartar todo lo que no sea estrictamente necesario para cumplir los objetivos de los Sprints. Además, se adherirán nuevas recomendaciones para complementar las siete recomendaciones básicas de Lean y así poder mejorar la integración de las dos metodologías.

- Kanban por su parte aportará a la propuesta en su rol de herramienta, por lo cual se utilizará un tablero Kanban para manejar todas las actividades que se estén ejecutando en cada Sprint. Esto último, sin embargo, no resuelve el gran inconveniente de cuello de botella de Kanban, por lo que, para solucionar este problema, se propone una estrategia de manejo dinámico de recursos, la cual se explicará más adelante.
- La figura 10.1 detalla la visión general de la propuesta.

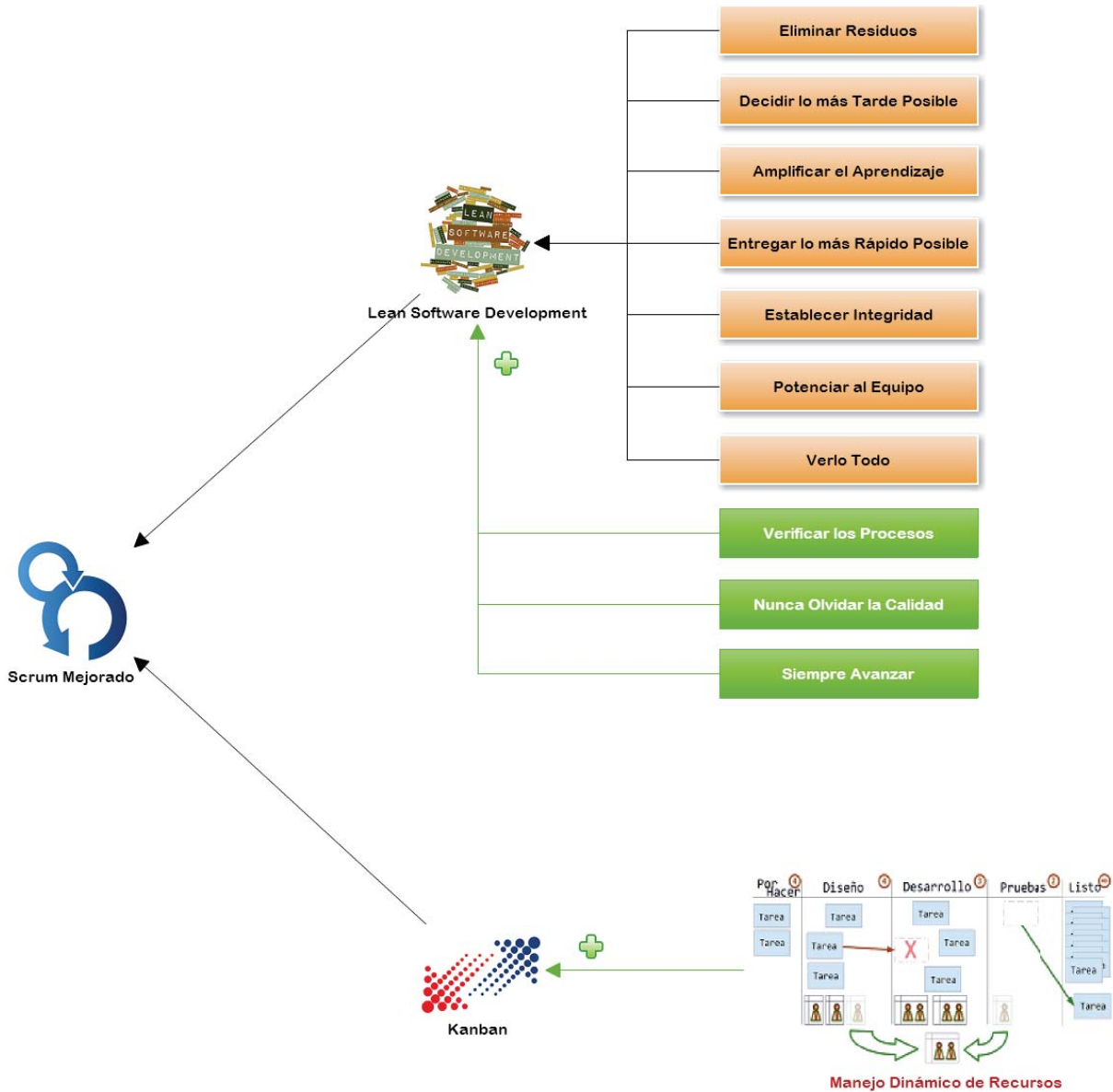


Figura 10.1 Visión General de la Propuesta.

10.2.1. Modificaciones Lean

Con el objetivo de mejorar la integración de Lean dentro del nuevo enfoque propuesto, se plantean 3 nuevas recomendaciones que complementarán a las recomendaciones básicas de Lean:

Verificar los Procesos: Para respetar el principio más importante de Lean, es decir, eliminar los residuos, se recomienda que ninguno de los procesos establecidos en la metodología sea considerado de carácter obligatorio. Siempre se debe priorizar la generación de valor, y si se detecta que alguna de las actividades o tareas que conforman la metodología interfiere con este objetivo, se recomienda eliminarla.

Nunca Olvidar la Calidad: Siempre que se decida eliminar algún elemento que a simple vista no esté generando valor al cliente, antes de llevar a cabo la eliminación se recomienda verificar cómo repercutirá esta acción en la calidad final del producto y decidir si realmente vale la pena hacerlo.

Siempre Avanzar: No se recomienda bajo ninguna circunstancia detener las actividades o tareas que se están realizando. Cualquier actividad ociosa o cuello de botella que se produzca, puede suponer una notable disminución en la eficiencia del desarrollo, lo cual implica un enorme residuo para el proyecto.

10.2.2. Modificaciones Kanban

Como se dijo anteriormente, para combatir el problema de la generación de cuellos de botella que tiene Kanban, se ha propuesto una estrategia de manejo dinámico de recursos. La explicación completa de esta estrategia se expone a continuación:

Para explicar la estrategia propuesta, se ha diseñado un escenario correspondiente a un proyecto de software cualquiera, en el que se expondrá un ejemplo explicativo que permita al lector entender de la mejor forma posible cómo se generan los cuellos de botella en Kanban y de qué forma se pretende solucionar:

En la Figura 10.2, se muestra el tablero Kanban de un proyecto en el que hay 3 personas que trabajan en diseño, 2 pares que trabajan en desarrollo y un tester. Las tareas sólo pueden pasar a la siguiente columna cuando se hayan completado. Si un diseñador termina una tarea, esta se mueve hacia desarrollo y el diseñador coge una nueva tarea para seguir trabajando.

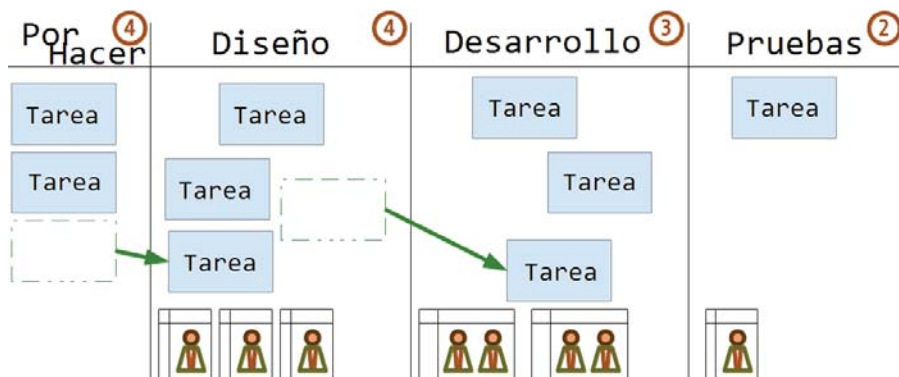


Figura 10.2 Propuesta Kanban - parte 1.

Este es el proceso normal, pero ¿qué pasa si se llega al máximo de tareas por columna? Si un diseñador termina una tarea, pero en desarrollo todavía no se han terminado, se tendrían 4 tareas en el lane de desarrollo: una situación no deseada. La figura 10.3 muestra la situación expuesta.

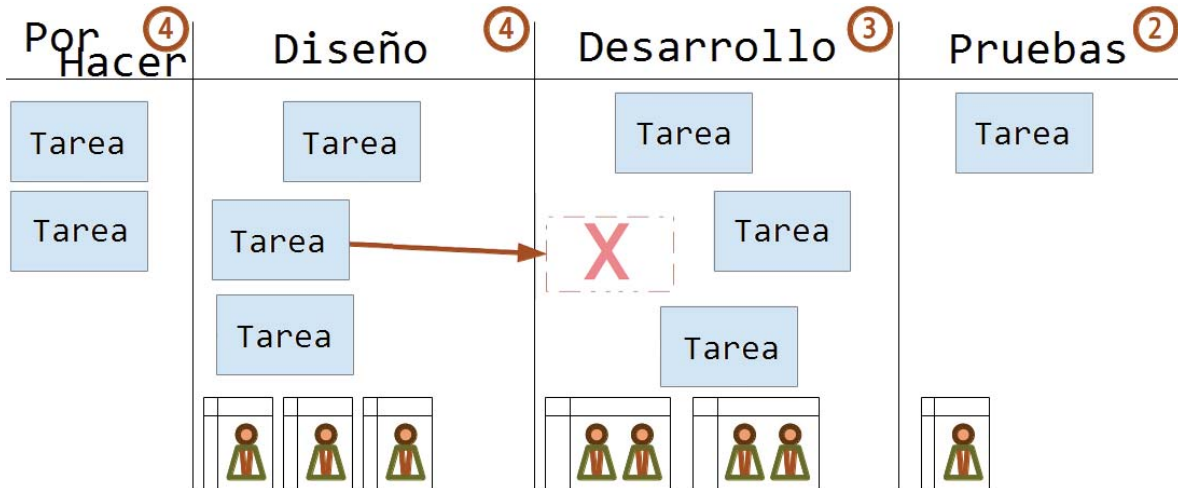


Figura 10.3 Propuesta Kanban - parte 2.

En este caso lo que se debe hacer es reasignar los recursos. Lo primero que intentaría realizar el diseñador es entregar la tarea que acaba de terminar a Desarrollo, pero no puede hacerlo porque el lane está al máximo de su capacidad. Su única opción es trabajar en alguna de las tareas de desarrollo. Puede que no sea el mejor desarrollador, pero sus esfuerzos ayudarán a mantener el flujo del proceso. Bajo la misma idea, si el tester, que sólo tenía una única tarea, la termina, podría también ayudar a los desarrolladores en sus tareas. Cabe recalcar que para que esta solución dé resultado, hay que tener mucha precaución con respecto a las tareas que se le asignará a cada persona a la hora de ser cambiado de un lane a otro. La idea es que la persona que realice el cambio, lo haga sólo en carácter de apoyo, no es recomendable que se le asignen tareas demasiado complejas en los otros lanes, sobre todo si la persona no es experta en la actividad en cuestión. La Figura 10.4 muestra el escenario descrito.

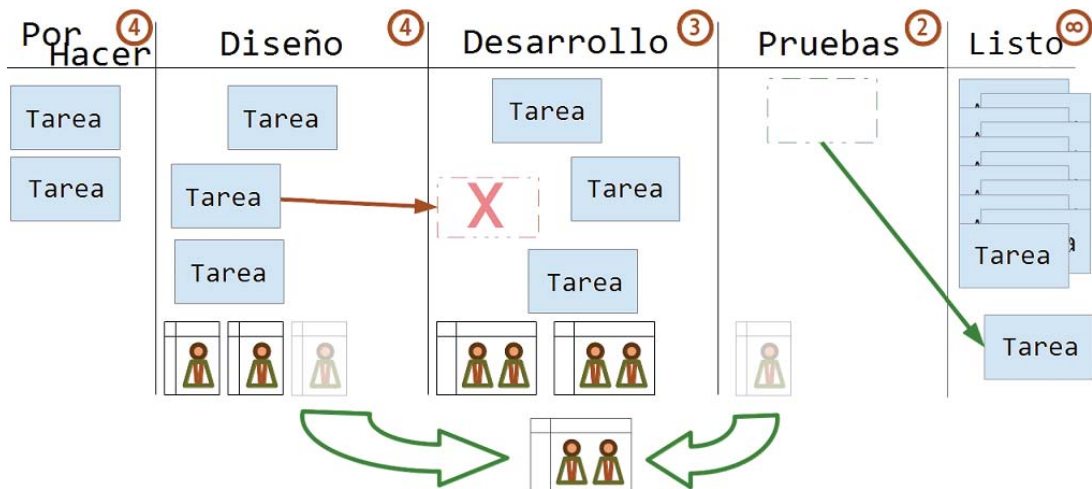


Figura 10.4 Propuesta Kanban - parte 3.

Finalmente, una vez que alguna tarea en la columna de desarrollo se haya completado, se puede regresar a la situación de normalidad (el tester volverá a su columna para poder probar la nueva tarea desarrollada y el diseñador comenzará a trabajar en una nueva tarea de diseño, como se observa en la Figura 10.5). Se debe tener en cuenta que esto no es una receta exacta, cada uno puede elegir qué hacer en estos casos.

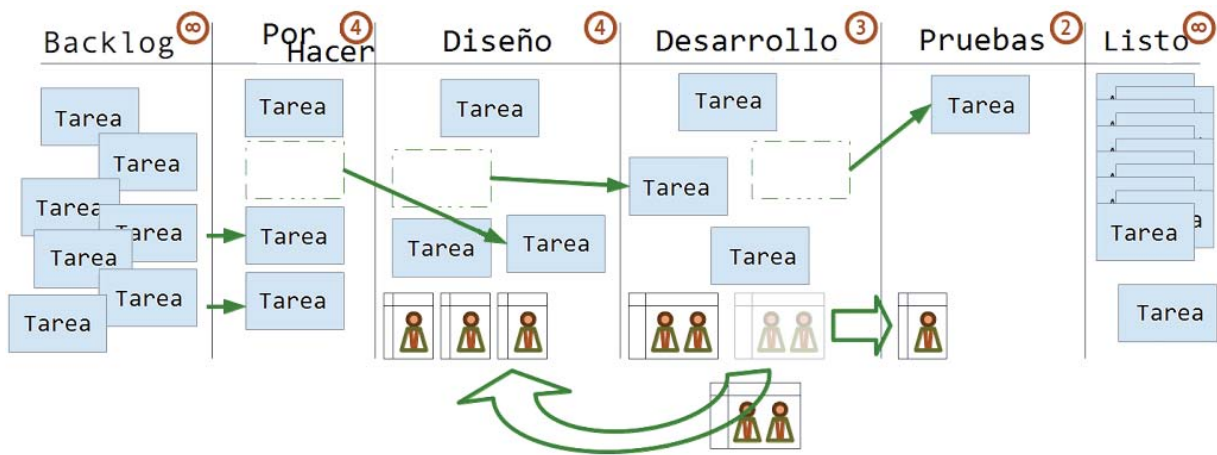


Figura 10.5 Propuesta Kanban - parte 4.

10.3. Requisitos Básicos de Aplicación

Para que la propuesta pueda ser aplicada correctamente, deben cumplirse una serie de requisitos, los cuales se listan a continuación:

- Al igual que las metodologías que la conforman, la propuesta deberá cumplir con las mismas características explicadas en la sección 9.4.
- Para que la estrategia de manejo dinámico de recursos de Kanban funcione correctamente, es necesario que todos los miembros del equipo de trabajo tengan al menos un conocimiento básico de cada una de las fases (lanes) que conforman el flujo de trabajo de un proyecto de software.
- Dentro del enfoque propuesto deberá existir un nuevo rol que se encargue de velar que las recomendaciones de Lean no se dejen de tomar en cuenta en ningún momento. Este nuevo rol será denominado Lean Master y deberá ser una persona de confianza para los miembros del equipo de trabajo, con un gran liderazgo y una gran capacidad para entender lo que el cliente necesita. Este nuevo rol deberá participar activamente en todos los procesos que conforman la elaboración de los proyectos, desde la definición del Product Backlog hasta la entrega final del producto.

11. Métricas

Luego de todos los análisis efectuados y las deducciones obtenidas a partir de ellos, es esencial que se reúnan todos estos datos empíricos y se lleven a la práctica, con el fin de comprobar si la propuesta planteada representa realmente una contribución para el campo del desarrollo ágil. Para esto, se han formulado una serie de métricas que permitirán recabar datos específicos enfocados al análisis de las principales características de la propuesta, de modo de verificar su comportamiento al momento de ser implementada. Las métricas seleccionadas se utilizaron en el curso de Ingeniería Web, impartido por el profesor Dr. Ismael Figueroa Palet, con quien se trabajó activamente para realizar el estudio. A continuación, se describen las métricas que se utilizaron para la medición y comprobación final de los resultados de esta investigación.

11.1. Cuantitativas

Para seleccionar las métricas a utilizar, primero se efectuó un estudio en profundidad sobre las métricas más utilizadas en contextos de desarrollo ágil, luego, se analizó cuáles de ellas ofrecen un beneficio real para los objetivos que se persiguen en este trabajo y, finalmente, se seleccionaron las más adecuadas para la investigación. Además, se ha diseñado una nueva métrica creada específicamente para la medición del manejo dinámico de recursos que plantea el nuevo enfoque.

11.1.1. Lead Time

Registra el tiempo que transcurre entre el momento en que se solicita el desarrollo de algún ítem de trabajo del Product Backlog hasta el momento en que finalmente dicho ítem de trabajo es entregado. El objetivo de esta métrica será evaluar qué tanto debe esperar un cliente para poder obtener avances reales durante la ejecución de un proyecto (obtener valor) y para implementarla se utilizarán los días de trabajo como unidad de medida [13].

11.1.2. Touch Time

Registra el tiempo en el cual un ítem de trabajo es realmente trabajado (o tocado) por el equipo, es decir, mide el tiempo real que tarda el equipo de trabajo para finalizar algún ítem, descontando todos los posibles períodos de inactividad que puedan producirse en él a lo largo del proceso. La unidad de medida utilizada para esta métrica serán las horas de trabajo [13].

11.1.3. Velocidad

Permite visualizar la velocidad de avance del equipo de trabajo a través de la cantidad de requerimientos o ítems de trabajo completados en cada iteración. Esta métrica permite medir la productividad del equipo de trabajo, ya que mientras mayor sea la cantidad de ítems de trabajo que sean capaces de terminar por iteración, menor será el tiempo necesario para finalizar el proyecto [14].

11.1.4. Requisitos no Completados

Registra la cantidad de requerimientos que no se pudieron completar en cada una de las iteraciones del proyecto pese a haber sido establecidos como objetivo de la iteración. Esta métrica permitirá evaluar qué tan bien se están cumpliendo los objetivos establecidos al inicio de cada Sprint y, además, evaluar el desempeño del equipo de trabajo [14].

11.1.5. Velocidad de Estabilización

Registra los eventos de sobrecarga de trabajo que se hayan producido a lo largo de cada iteración. Esta métrica está diseñada específicamente para comprobar la funcionalidad del manejo dinámico de recursos que se plantea en la propuesta y su objetivo principal es medir la eficiencia del equipo de trabajo para sobreponerse a estos cuellos de botella.

11.1.6. Eficiencia

La eficiencia está relacionada con el cumplimiento de las tareas con el mínimo gasto de recursos que sean posible. Para el caso en estudio, resulta principalmente relevante verificar el gasto de tiempo en relación a las tareas realizadas, para lo cual se pondrá especial énfasis a aquellas tareas que presentan características atípicas, ya que a través de ellas se podría llegar a obtener información interesante incluso fuera del ámbito de la eficiencia.

11.2. Cualitativas

Análogo a lo explicado en las métricas cuantitativas, para estas métricas también se debió realizar un estudio previo con el cual se definió cuáles eran las mejores opciones para los objetivos que se persiguen en esta investigación.

11.2.1. Satisfacción del Equipo de Trabajo

Esta métrica está ligada principalmente a los desarrolladores. Se espera obtener información importante respecto a lo útiles que les puedan resultar las herramientas que la metodología les proporciona, el ambiente de trabajo que se genera, entre otros. Para recabar estos datos, se realizarán una serie de encuestas y/o entrevistas que deberán ser respondidas por los miembros de los equipos de trabajo.

11.2.2. Facilidad de Adopción

Esta métrica tiene como objetivo evaluar la facilidad con que la metodología puede ser implementada, así como también la facilidad de aprendizaje y la facilidad de uso. Para evaluar estos conceptos, al igual que para la métrica anterior se utilizarán encuestas y/o entrevistas que permitan recabar los datos necesarios.

11.2.3. Organización y Planificación

Permite evaluar qué tan bien la metodología ayuda a organizar las distintas tareas que se deben realizar en cada iteración de un proyecto y qué tanto favorece la metodología a la planificación. Como ya es costumbre, esta métrica también deberá ser evaluada a través de encuestas y/o entrevistas.

12. Estudio y Recabación de Datos

12.1. Método de Aplicación

Como ya se explicó al comienzo de la sección 11, las métricas anteriores han sido evaluadas en el curso Ingeniería Web impartido por la Escuela de Ingeniería Informática de la Pontificia Universidad Católica de Valparaíso. En este curso, se les pidió a los alumnos desarrollar un proyecto semestral en grupos de 3 a 6 integrantes, en el cual debían utilizar una metodología de desarrollo ágil para su elaboración. Los alumnos comenzaron utilizando una metodología Scrum clásica con el objetivo de que pudieran interiorizarse con las características generales de un desarrollo ágil, puesto que muchos de ellos jamás habían trabajado con metodología de este tipo. Posteriormente, se agregaron las características adicionales correspondientes a la propuesta explicada en este documento, de modo que los alumnos pudiesen tener una experiencia real al utilizarla.

Para reunir los datos necesarios para las métricas cuantitativas, se trabajó con una herramienta basada en web denominada JIRA, la cual se enfoca en el seguimiento de errores, incidencias y gestión operativa de proyectos. El objetivo era que los estudiantes utilizaran esta herramienta para organizar el flujo de trabajo de sus proyectos, de manera de poder utilizar los datos ingresados por ellos para construir cada una de las métricas requeridas. El motivo por el cual se escogió JIRA es porque, entre sus funciones, existe la posibilidad de generar diversos informes ágiles que permiten visualizar el estado de los proyectos, la eficiencia de los equipos de trabajo, la cantidad de tareas ejecutadas en cada iteración, entre otros; lo cual facilita enormemente la creación de las métricas. Además, para la organización del flujo de trabajo, JIRA ofrece una interfaz completa y amigable que denomina pizarras, las cuales son altamente configurables y adaptables, gracias a lo cual fue posible crear pizarras con características Kanban para que los estudiantes pudieran utilizarlas a lo largo del desarrollo de sus proyectos, permitiendo así la posterior comprobación del manejo dinámico de recursos.

Pero JIRA no fue el único elemento utilizado para la captación de los datos cuantitativos, ya que, si bien es una herramienta muy útil, en algunas ocasiones la información proporcionada por esta puede resultar algo confusa, especialmente si los usuarios (en este caso, los alumnos) no la utilizan adecuadamente. Por lo anterior, se tomó la precaución de solicitarle a los estudiantes que a lo largo del semestre realizaran presentaciones orales (Sprint Retrospective) al finalizar cada iteración, en las cuales debían explicar la experiencia adquirida hasta ese punto, las dificultades que debieron enfrentar y los beneficios que les aportó la metodología utilizada, así como los objetivos que perseguirán en la siguiente iteración. Además, dicha presentación debía estar acompañada de un breve video complementario en el que se expliquen detalles respecto a las historias de usuario que hayan colocado en JIRA a lo largo de la iteración.

Para el caso de las métricas cualitativas, se ha decidido utilizar encuestas creadas a través de la aplicación web Google Form, ya que brinda una mayor libertad a los estudiantes a la hora

de responder cada encuesta gracias a que pueden hacerlo desde cualquier dispositivo con acceso a internet. Además, Google Form permite generar automáticamente gráficos porcentuales con las respuestas de los encuestados, lo cual facilita bastante la posterior evaluación de los resultados. Se han diseñado un total de dos encuestas, una para evaluar los conocimientos generales de los encuestados respecto a las metodologías ágiles, la cual se aplicó al inicio del semestre, y otra para evaluar sus experiencias y apreciaciones respecto a la metodología propuesta en este documento, la cual se aplicó una vez finalizados los proyectos.

Finalmente, para llevar a cabo este estudio, el primer paso consistió en realizar una reunión con los alumnos del curso Ingeniería Web, en la que se les explicó en detalle todos los aspectos de la propuesta diseñada en este documento, de modo que ellos pudieran implementarlo en sus respectivos proyectos cuando llegara el momento de hacerlo. Además, se les explicó el funcionamiento de la herramienta JIRA, de modo que fueran capaces de utilizarla sin problemas a la hora de trabajar con ella. Por último, se les entregó un documento complementario con todos los detalles expuestos en dicha reunión plasmados de forma resumida.

12.2. Presentación de Datos Obtenidos

Con el afán de mantener el orden y no contaminar demasiado el texto principal del documento, se ha decidido trasladar los datos obtenidos a través de este estudio al Anexo D, si el lector desea conocer más detalles, puede dirigirse a dicho apartado.

12.3. Análisis de Datos Obtenidos

12.3.1. Análisis de Datos Cuantitativos

A continuación, se analizarán, desde el punto de vista de las métricas cuantitativas, los valores obtenidos durante de la recolección de datos mostrada en el Anexo D.

12.3.1.1. Análisis Lead Time

Al observar los valores de Lead Time en las incidencias de los diferentes grupos, se pueden apreciar las siguientes características:

- Algunos grupos ingresaron la mayor parte de las incidencias de su proyecto al inicio de este y las fueron resolviendo poco a poco en cada iteración, lo que demuestra una buena planificación inicial.
- Otros grupos hicieron justo lo contrario, al principio sólo crearon unas pocas incidencias y a medida que avanzaron en el proyecto fueron planteando nuevas, lo que demuestra que su estilo de planificación fue más progresivo.
- Gran parte de los grupos planificaron incidencias que no pudieron resolver en el tiempo asignado, lo que es un claro error de planificación.
- El Lead Time promedio entre todos los grupos es de 45.7 días, lo cual corresponde aproximadamente al 68.2% del ciclo de vida de los proyectos, por

lo tanto, desde el punto de vista del cliente, los requerimientos tardaron mucho en resolverse.

12.3.1.2. Análisis Touch Time

Los valores correspondientes al Touch Time de las incidencias presentan las siguientes características a destacar:

- Se aprecia una discrepancia bastante grande entre los valores de Lead Time y Touch Time, lo cual indica que, en muchas ocasiones, las incidencias permanecieron en el Backlog por largos períodos, lo cual es una clara ineficiencia en el desarrollo.
- Se aprecia también una gran diferencia entre los valores del Touch Time de un grupo a otro, habiendo grupos que llegaron a ocupar más de 100 horas en la resolución de una incidencia y otros que sólo tardaron 5 horas como máximo. Esto es un claro indicio de que, a la hora de calcular las horas trabajadas, algunos grupos utilizaron sus propios criterios subjetivos, ya que es prácticamente imposible que se presente una desigualdad tan abismal entre grupos de trabajo con un nivel de conocimiento relativamente similar.
- En promedio, los grupos presentaron un Touch Time de 15.6 horas, es decir, menos de 1 día por incidencia, y considerando que en promedio los grupos registraron unas 13 incidencias, significa que en unos 8 días y medio podrían haberlas terminado todas, lo que al ser comparado con los 45.7 días de promedio del Lead Time, demuestran una gran cantidad de tiempo no aprovechado.

12.3.1.3. Análisis Velocidad y Requisitos No Completados

Al observar los datos correspondientes a las métricas de velocidad y requisitos no completados, se observa lo siguiente:

- Pese a la capacitación, los alumnos no fueron capaces de utilizar adecuadamente la herramienta web JIRA, dificultando bastante el análisis de los datos. En este caso, se tuvo que recurrir a la información obtenida a través de los Sprint Retrospective para determinar los verdaderos valores correspondientes a las métricas de velocidad y requisitos no completados.
- A nivel general, los grupos demuestran una gran eficacia a la hora de resolver sus incidencias, sólo dos equipos tuvieron incidencias no resueltas, pero de estos, hubo uno que presentó una mayor cantidad de incidencias no resueltas en comparación a las que sí se resolvieron, lo que indica que dicho equipo no está cumpliendo los objetivos que tenían planificados.
- Como se explicó en la sección 11.1.3., a través de la velocidad es posible determinar la productividad del equipo de trabajo, ya que, en situaciones normales, a medida que el tiempo avanza el equipo va adquiriendo una mayor experiencia y entendimiento del proyecto, lo que les permite resolver un mayor

número de incidencias por iteración. En este caso, ninguno de los grupos de trabajo cumplió con esta premisa, lo que refleja una baja productividad.

12.3.1.4. Análisis Velocidad de Estabilización

Según los informes extraídos de JIRA, salvo por dos equipos, la mayor parte de los grupos presentaron sobreesfuerzo durante el transcurso de sus proyectos, algunos incluso más de una vez. Al observar los tiempos de estabilización de estos sobreesfuerzos, se observa que algunos equipos lo solventaron rápidamente, mientras que otros mantuvieron esta situación durante varios días, 9 días en el peor de los casos. Concretamente, la situación de los grupos fue la siguiente:

- 2 grupos no presentaron sobreesfuerzo.
- 3 grupos presentaron una o más situaciones de sobreesfuerzo, pero las resolvieron en un día o menos.
- 2 grupos presentaron una o más situaciones de sobreesfuerzo y las mantuvieron por varios días.

Con esto, se puede concluir que la mayor parte de los grupos en estudio tuvo un comportamiento bastante positivo con respecto a los sobreesfuerzos, pudiendo solventarlos con rapidez y exhibiendo así una alta velocidad de estabilización.

12.3.1.5. Análisis Eficiencia

La Tabla 12.1 presenta un resumen con la información más relevante obtenida a través del análisis efectuado en la Tabla D.12

Tabla 12.1 Resumen Datos Eficiencia.

Grupo	Eficiencia General	Predecibilidad General	Valores atípicos	Nº Clústeres
1	Alta	Alta	14	4
2	Baja	Alta	10	3
3	-	-	2	0
4	Alta	Alta	11	3
5	Baja	Baja	10	3
6	-	-	1	0
7	Baja	Alta	26	6

Al observar esta tabla se aprecia claramente que, en términos de eficiencia, hubo grupos con un alto nivel de eficiencia, y grupos que resultaron ser bastante ineficientes, de modo que es difícil encontrar un patrón claro al respecto. La predecibilidad por su parte, demuestra que, en casi todos los casos, al observar el comportamiento pasado, se podría llegar a predecir las tendencias que se presentarían en el futuro, lo cual sería de gran utilidad a la hora de realizar una planificación. Finalmente, los valores atípicos y clústeres, por la cantidad y las características de estos, son un claro indicio de que los estudiantes presentaron diversas

complicaciones durante su desarrollo, lo cual se relaciona en gran medida con su inexperiencia con la metodología y su falta de conocimientos respecto a las herramientas utilizadas, eso, sumado a que presumiblemente no utilizaron la aplicación JIRA como debían hacerlo, dio como resultado estos valores.

12.3.2. Análisis de Datos Cualitativos

A continuación, en la Tabla 12.2 se presentarán todos los datos relevantes que se han podido extraer a partir de los resultados obtenidos de las encuestas realizadas.

Tabla 12.2 Análisis de Datos Cualitativos.

Aspecto analizado	Resultado
Características de los sujetos en estudio.	Corresponden a estudiantes universitarios con poca o nula experiencia en metodologías de trabajo ágil. Algunos de ellos poseen conocimientos teóricos, pero de un nivel muy básico.
Nivel de conocimiento de Scrum, LSD y Kanban.	Sólo un pequeño porcentaje de los estudiantes cuentan con un cierto nivel de conocimiento de Scrum. Kanban y LSD son desconocidos para ellos.
Dificultad de aprendizaje y adopción de la metodología.	Moderada. Los estudiantes coinciden en que la metodología ofrece las herramientas necesarias para facilitar esta tarea.
Capacidad de la metodología de impulsar el trabajo en equipo.	Muy alta.
Relación entre la cohesión de un grupo de trabajo y su desempeño al momento de utilizar la metodología.	Muy importante. Se percibe una notoria diferencia en las respuestas de aquellos alumnos que dicen haber pertenecido a equipos de trabajo sólidos con aquellos que no.
Utilidad de las Daily Scrum.	Para los sujetos en estudio, estas reuniones resultaron ser bastante útiles, aunque no se ha podido corroborar que efectivamente las hayan realizado todos los días.
Capacidad de resolver problemas para un equipo de trabajo regido por esta metodología.	Alta.
Importancia de los líderes dentro de los grupos de trabajo.	Muy importantes. Incluso aquellos alumnos que mostraban un claro descontento con el grupo de trabajo al que pertenecían, destacaban la labor realizada por su líder.
Capacidad de la metodología de favorecer la visualización del flujo de trabajo.	Muy alta.

Capacidad de la metodología de incentivar la planificación.	Muy alta.
Dificultad en la división de proyectos en iteraciones para grupos de trabajo sin experiencia previa.	Baja.
Dificultad en la asignación y distribución de tareas para un grupo de trabajo sin experiencia previa.	Moderada.
Dificultad en la planificación de tiempos para un grupo de trabajo sin experiencia previa.	Alta.
Satisfacción de los sujetos en estudio con la metodología utilizada.	Muy satisfechos.
Mayores dificultades presentadas por los sujetos en estudio.	Organización del trabajo en equipo, asignación de las tareas y distribución de los tiempos.

12.3.2.1. Análisis Satisfacción del Equipo de Trabajo

En base a la información mostrada en la tabla anterior, se observa que los estudiantes demostraron una gran satisfacción con la metodología utilizada y las herramientas que esta les ofrece. Sus principales disgustos se relacionan con aspectos como la planificación de las tareas y el trabajo en equipo, lo cual, considerando que era su primer encuentro con una metodología de este tipo, puede relacionarse perfectamente con su falta de experiencia.

12.3.2.2. Análisis Facilidad de Adopción

Los estudiantes coinciden en que la metodología proporciona las herramientas y la información necesarias para facilitar su adopción, permitiendo que incluso un grupo inexperto pueda implementarla con relativa facilidad.

12.3.2.3. Análisis Organización y Planificación

Sin duda este aspecto fue el más complejo para todos los estudiantes. Muchos tuvieron problemas a la hora de planificar, demorándose muchas más horas de las presupuestadas para realizar ciertas tareas o en algunos casos, muchas menos. También les resultó bastante difícil planificar el número de tareas a realizar por iteración, lo que daba como resultado tareas que no se realizaban o que debían trasladarse a iteraciones siguientes. Respecto a la organización, muchos tuvieron problemas para coordinarse y distribuir las tareas, además, varios estudiantes coincidieron en que su grupo de trabajo no era el mejor, lo que repercutía directamente en su rendimiento y en el ambiente de trabajo.

13. Conclusiones

Las metodologías ágiles hoy en día son una alternativa muy efectiva para el desarrollo de software debido a la gran versatilidad que ofrecen, que es ideal para el mercado cambiante de la era actual. Los proyectos en general poseen una gran incertidumbre y los requerimientos en muchos casos tienden a cambiar con el tiempo, lo que implica la realización de cambios que pueden significar grandes costos para las metodologías tradicionales. Lo anterior, sin embargo, no implica que las metodologías ágiles sean perfectas o utilizables para cualquier caso, depende mucho del contexto y de la situación particular de cada proyecto, además, en la actualidad existe una gran variedad de metodologías ágiles y todas ellas presentan ventajas y desventajas frente a las otras.

A través de la presente investigación, se estudiaron en detalle tres de las metodologías ágiles existentes, y en todas ellas se han detectado deficiencias que deberían corregirse y también muchos beneficios. Esa fue la idea que dio origen a esta investigación, en la cual se pensó en la posibilidad de agrupar todos estos beneficios e intentar unificarlos a través de una propuesta que permitiera minimizar las deficiencias y ofrecer una nueva forma de aplicar estas metodologías.

Para lograr esto, el primer paso fue elaborar una extensa investigación que se dividió en 3 pasos fundamentales: seleccionar y estudiar las metodologías que conformarían la propuesta, elaborar la propuesta y, una vez planteada, probarla a través de un estudio. Para este último paso, se pensó en dos posibilidades, la primera consistía en poner a prueba la propuesta en una empresa real con la cual se pudiese trabajar en conjunto y obtener resultados al respecto, pero esa alternativa no fue factible de aplicar ya que no se pudo encontrar ninguna empresa que estuviera dispuesta a cambiar todo su paradigma de trabajo para colaborar con esta investigación. La segunda alternativa y la que finalmente se llevó a cabo, consistió en la implementación de la metodología en un curso de la Pontificia Universidad Católica de Valparaíso impartido por el profesor Dr. Ismael Figueroa Palet, en donde los estudiantes debían desarrollar un proyecto semestral en grupos, lo que brindaba una ocasión perfecta para implementar la metodología en sus proyectos.

Una vez finalizado el estudio y analizado los resultados obtenidos, las conclusiones que se pueden formular a partir de estos resultan bastante interesantes. En primer lugar, es importante identificar a los sujetos de estudio, es decir, a los estudiantes, y corroborar el nivel de conocimientos que estos poseían. El resultado es que los sujetos sometidos a este estudio corresponden a personas jóvenes con ciertos conocimientos de programación, pero sin ninguna experiencia en el campo de las metodologías ágiles, además, muchos de ellos desconocían varias de las herramientas que debieron utilizar durante el semestre, entre las que destaca JIRA, que cumplió un rol fundamental en la obtención de los datos cuantitativos.

Los proyectos fueron realizados en un plazo de unos 2 meses y medio aproximadamente, los cuales no estuvieron exentos de problemas. Por un lado, el semestre coincidió con un extenso período de movilizaciones en Chile, lo que claramente alteró los datos obtenidos, y por el otro, varios estudiantes internamente presentaron conflictos de grupo y en algunos casos, ya sea por falta de tiempo, por falta de conocimiento o cualquier otro factor, no utilizaron correctamente la aplicación JIRA, lo que claramente repercutió en los resultados.

Pese a lo anterior, el estudio se realizó de la mejor manera posible y efectivamente se pudieron comprobar la mayor parte de las características de la propuesta planteada. Según la información proporcionada por los alumnos, la metodología les pareció amigable, fácil de entender pese a su poca experiencia y muy útil a la hora de planificar, organizar, distribuir las tareas y resolver los problemas. Además, todos coincidieron en que la metodología lograba fomentar la visibilidad del flujo de trabajo gracias a su pizarra Kanban y, pese a que algunos grupos no lograron una buena relación de trabajo, la mayoría concordó que la metodología lograba fomentar el trabajo en equipo. Otro aspecto a destacar es la importancia de un buen equipo de trabajo al utilizar esta metodología, las opiniones de los estudiantes lo dejaron de manifiesto. Es de suma importancia que para cualquier equipo que decida incursionar en esta metodología, haya la suficiente confianza y cohesión entre sus miembros, ya que de lo contrario la metodología podría resultar incluso perjudicial, además, la labor del líder es vital, por lo que se debe escoger con cuidado a quien desempeñará este rol.

Respecto a los datos cuantitativos recabados, se observó que los grupos, a nivel general, desempeñaron un trabajo bastante bueno a lo largo de todo el proyecto. Las principales ineficiencias se concentraron en las primeras semanas de trabajo, lo cual es comprensible considerando que los alumnos recién estaban en proceso de aprendizaje. En el ámbito de la planificación, fue bastante notoria la falta de experiencia de los equipos de trabajo, habiendo incidencias que nunca se resolvieron y otras que permanecieron por demasiado tiempo en el Backlog, lo cual es un claro ejemplo de “residuo” para el proyecto. Por otra parte, desde el punto de vista del tiempo, los alumnos tuvieron grandes lagunas de tiempos muertos, habiendo períodos muy concretos de resolución de incidencias y otros en los que no se apreciaba ninguna actividad, aun así, los alumnos demostraron, en su mayoría, ser bastante eficaces en la resolución de sus incidencias, siendo capaces de resolver la mayor parte de estas pese a todo el tiempo desaprovechado. De lo anterior, se pueden deducir que los equipos de trabajo lograron cumplir con los objetivos que se habían planteado, pero a la vez, su productividad fue bastante baja, ya que, al concentrar todo su trabajo en períodos cortos, no les daba tiempo de ir más allá e intentar mejorar la calidad de su trabajo o cuando menos, resolver una mayor cantidad de incidencias por iteración.

En referencia a los sobreesfuerzos o cuellos de botella de Kanban, los grupos en general tuvieron un buen comportamiento al respecto, pudiendo resolverlos con rapidez y logrando mantener un trabajo bien distribuidos entre todos los miembros. Asimismo, los grupos demostraron un nivel de eficiencia bastante normal durante el desarrollo, habiendo algunos equipos más eficientes y otros no tanto, lo cual se relaciona en gran medida con la cohesión que estos hayan tenido para organizarse de mejor o peor manera.

Como conclusión final, es claro que quizá el método utilizado para la evaluación de la metodología no fue el mejor, los sujetos de estudio no eran los más idóneos y las circunstancias en que fue aplicada no fue la óptima, pero por sobre todo eso, la metodología demostró ser capaz de ayudar a los estudiantes en la elaboración de sus proyectos, mejorando la visibilidad de sus flujos de trabajo y brindándoles herramientas para mejorar su planificación, su organización y su trabajo en equipo. Además, a través de Kanban, fueron capaces mantener un adecuado orden en sus tareas y lograron sobreponerse sin dificultades a los sobreesfuerzos, evitando así que algún miembro del equipo se viera abrumado por una sobrecarga de trabajo. Por todo lo anterior y pese a todas las dificultades que tuvieron los grupos durante el desarrollo de los proyectos, la

metodología brindó resultados bastante positivos y sin duda podría llegar a ser un gran aporte para los desarrollos ágiles.

Finalmente, para trabajos futuros que se puedan realizar en base a este trabajo, sería importante que se realizara una mejor y más exhaustiva evaluación de la propuesta, en mejores circunstancias e idealmente en una empresa real del rubro de la informática, en la que se pueda evaluar con más tiempo, más recursos y asegurándose de comprobar todos y cada uno de los aspectos que involucran la propuesta, cosa que en este caso, lamentablemente no se pudo hacer.

14. Referencias

- [1] Wikipedia, La enciclopedia libre, *Desarrollo ágil de software*, Diciembre 2015. Disponible vía web en: https://es.wikipedia.org/wiki/Desarrollo_ágil_de_software.
- [2] The Standish Group, *Chaos Manifesto 2013*, larlet david web, 2013 [fecha de consulta: 18 Octubre 2015]. Disponible vía web en: <https://larlet.fr/static/david/stream/ChaosManifesto2013.pdf>.
- [3] rosallop, *Agile, un nuevo marco de trabajo para el desarrollo de proyectos*, Julio 2012 [fecha de consulta: 16 Agosto 2015]. Disponible vía web en: <http://www.rosallop.com/blog/agile-un-nuevo-marco-de-trabajo-para-el-desarrollo-proyectual/#sthash.pBhJS8iX.aKcbqA5v.dpbs>.
- [4] Garzías Javier, *Cuál es la herramienta más usada, la metodología más aplicada, las razones para ser ágil, causas de fracaso...*, Marzo 2013 [fecha de consulta: 21 Agosto 2015]. Disponible vía web en: <http://www.javiergarzas.com/2013/03/versionone-encuesta-agil.html>.
- [5] Csaba Patkos, *From Scrum to Lean*, envatotuts+, Enero 2013 [fecha de consulta: 18 Agosto 2015]. Disponible vía web en: <http://code.tutsplus.com/articles/from-scrum-to-lean--net-29420>.
- [6] Ruiz Francisco, *Ingeniería de Procesos de Software*, Castilla-La Mancha: Universidad de Castilla-La Mancha, 2008 [fecha de consulta: 10 Septiembre 2015]. Disponible vía web en: http://alarcos.esi.uclm.es/per/fruiz/cur/ips/trans/ips_spem.pdf.
- [7] Ruiz Francisco y Verdugo Javier, *Guía de Uso de SPEM2 con EPF composer*, Castilla-La Mancha: Universidad de Castilla-La Mancha, Abril 2008 [fecha de consulta: 11 Septiembre 2015]. Disponible vía web en: http://alarcos.esi.uclm.es/ipsw/doc/guia-spem2&epf_v30.pdf.
- [8] Palacio Juan, *El modelo Scrum*, Navegapolis, 2006 [fecha de consulta: 24 Septiembre 2015]. Disponible vía web en: http://www.navegapolis.net/files/s/NST-010_01.pdf.
- [9] Sutherland Jeff y Schwaber Ken, *The Scrum Guide*, Julio 2013 [fecha de consulta: 26 Septiembre 2015]. Disponible vía web en: <http://www.scrumguides.org/scrum-guide.html>.
- [10] Poppendieck Tom y Poppendieck Mary, *Lean Software Development*, 2001 [fecha de consulta: 10 Diciembre 2015]. Disponible vía web en: <https://leansoftwaredevelopment.wordpress.com/los-7-principios-de-lean>.
- [11] Scrum Manager Body of Knowledge, *Kanban, origen y definición*, Mayo 2014 [fecha de consulta: 9 Octubre 2015]. Disponible vía web en: http://www.scrummanager.net/bok/index.php?title=Kanban:_origen_y_definici%C3%B3n.

- [12] Rabanal Rosana, *Metodo Kanban en Ingenieria de Software*, SG Virtual Conference, Febrero 2013 [fecha de consulta: 13 Octubre 2015]. Disponible vía web en: <http://sg.com.mx/sgvirtual/7/sessions/metodo-kanban-ingenieria-software#.VnQAGPnhDIV>.
- [13] Universidad de Valencia, *Métricas en Kanban, algo más de metodologías ágiles*, [fecha de consulta: 28 de Marzo de 2016]. Disponible vía web en: <http://www.uv-mdap.com/blog/metricas-en-el-kamban-algo-mas-de-metodologias-agiles>.
- [14] Xavier Albaladejo, *Métricas ágiles y cuadro de mandos integral para Scrum*, Diciembre 2008 [fecha de consulta: 28 de Marzo de 2016]. Disponible vía web en: <https://proyectosagiles.org/2008/12/07/metricas-agiles-cuadro-mandos-balanceado-scrum>

Anexos

A: Sobre las Características de SPEM

A1. Jerarquía de Desglose del Trabajo

En SPEM se han definido una serie de conceptos que permiten representar los distintos paquetes de trabajo necesarios para la elaboración de los proyectos y sus procesos involucrados. Estos conceptos y su respectiva jerarquización (ver Figura A.1) son los siguientes [7]:

- **Delivery Process (Proceso de Despliegue):** sirve de base para representar el proyecto en su conjunto, con todos sus procesos.
- **Capability Pattern (Patrón de Capacidad):** representa un patrón de proceso, es decir, un fragmento de proceso que puede ser reutilizado más de una vez en un mismo Delivery Process.
- **Activity (Actividad):** es el elemento central para definir procesos ya que permite organizar sus elementos básicos (roles, productos de trabajo y tareas).
- **Task (Tarea):** es la porción de trabajo más pequeña.
- **Work Product (Producto de Trabajo):** representa las entradas y salidas de los procesos.

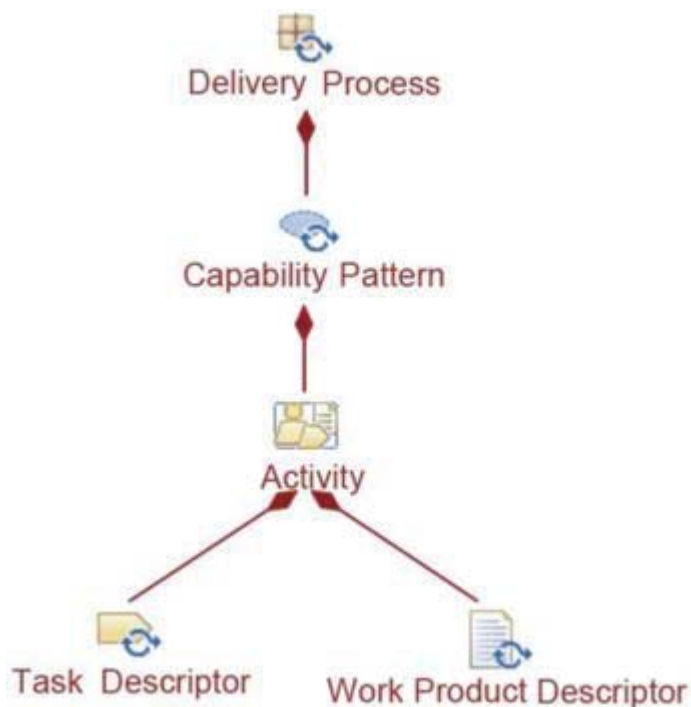


Figura A.1 Jerarquía de Desglose del Trabajo en SPEM [7].

A2. Estructura de Paquetes

Al igual que UML, SPEM está organizado en paquetes (packages) con la finalidad de lograr una mejor organización de todos los elementos y favorecer la reutilización. Cada paquete es una unidad lógica que extiende a sus paquetes padres, proveyéndose de estructuras y capacidades adicionales. De forma resumida el contenido de cada paquete es el siguiente [7]:

- **Core (Núcleo):** contiene las clases y abstracciones que sirven de base para las clases de los demás paquetes.
- **Process Structure (Estructura de Proceso):** define la base para la creación de modelos de proceso flexibles y sencillos, es decir, define la estructura de desglose de trabajo de manera estática mediante anidamiento de actividades y dependencias de precedencia entre ellas. Incluye referencias a la lista de Roles y a los Productos de Trabajo.
- **Process Behavior (Comportamiento de Procesos):** permite extender las estructuras del paquete anterior con modelos de comportamiento externos.
- **Managed Content (Contenido Administrado):** permite incorporar y gestionar descripciones en lenguaje natural, documentos y otras informaciones útiles. Hay libertad total para combinar modelos estructurales con procesos que contengan contenido en lenguaje natural. Así, un proceso puede estar formado sólo por una colección de guías de buenas prácticas; sólo por estructuras de actividades; o una combinación de ambas.
- **Method Content (Contenido del Método):** incluye los conceptos para crear elementos de método, que son los ítems que sirven de base para el ensamblado de procesos, metodologías, ciclos de vida, etc.
- **Process with Methods (Proceso con Métodos):** incorpora nuevas estructuras para poder integrar los procesos definidos con el paquete Process Structure con los elementos del método (instancias de Method Content).
- **Method Plugin (Plugin del Método):** incorpora conceptos para diseñar y gestionar bibliotecas o repositorios de contenido de métodos y procesos.

En la Figura A.2 se puede apreciar la estructura jerárquica de los paquetes y sus relaciones:

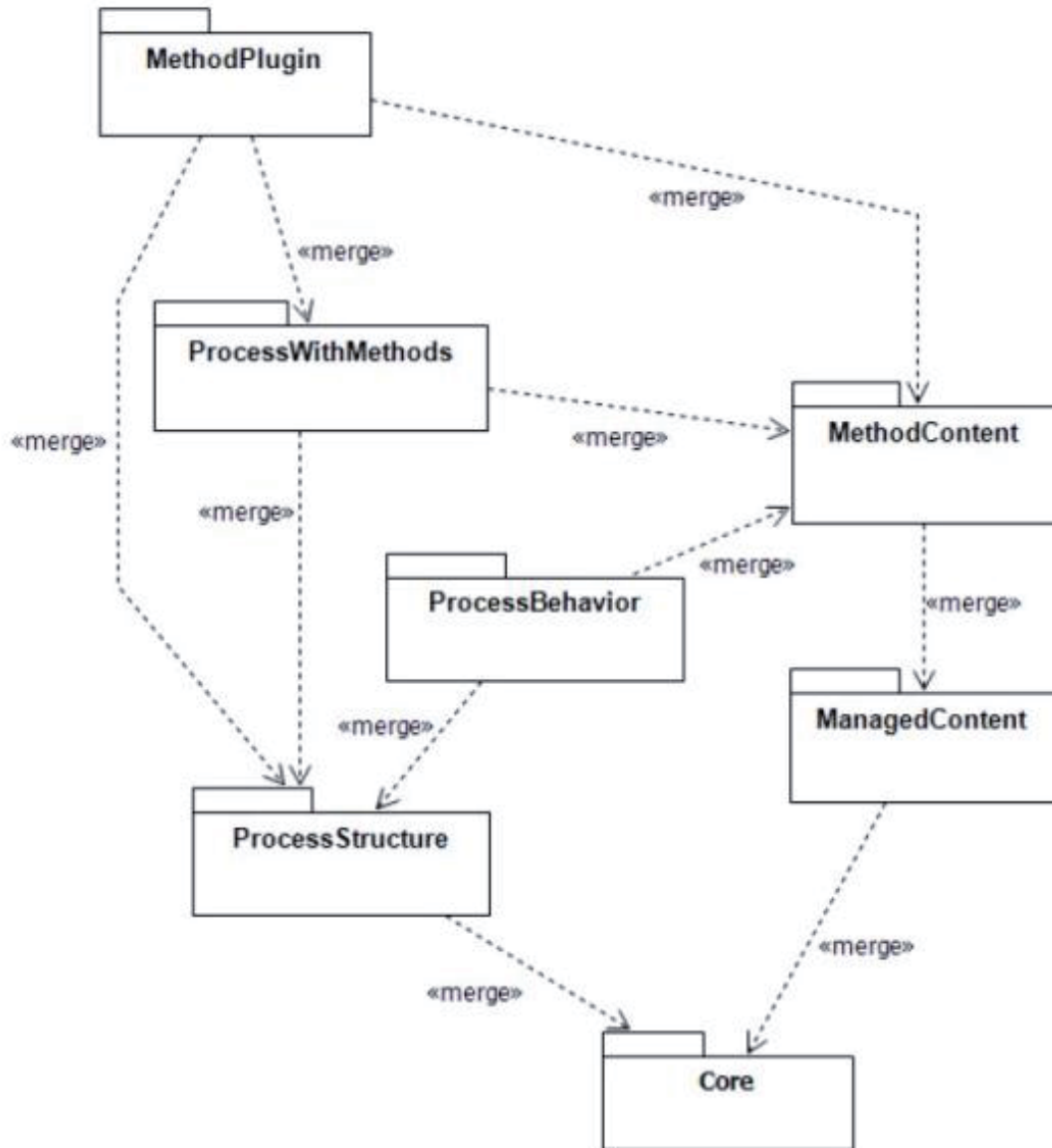


Figura A.2 Jerarquización de la Estructura de Paquetes [7].

A3. Elementos de SPEM Utilizados

A continuación, se describen todos los elementos que fueron utilizados en el presente documento para la elaboración de los diagramas SPEM que se emplearon en el modelamiento de cada una de las metodologías seleccionadas.

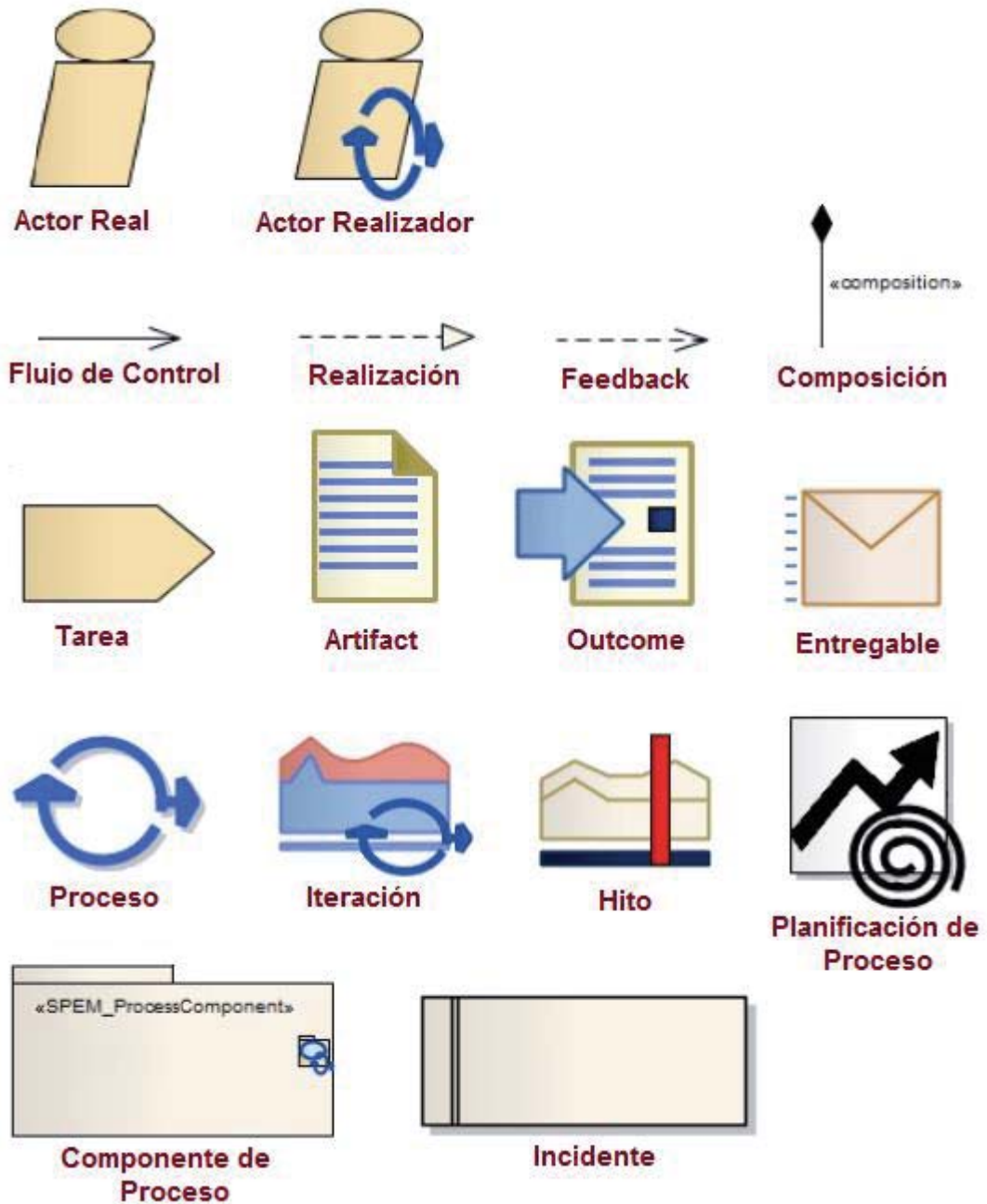


Figura A.3 Lista de Elementos SPEM Utilizados.

- **Actor Real:** representa a la persona, grupo o sistema real que debe ejecutar las acciones.
- **Actor Realizador:** representa la realización propiamente tal de la acción solicitada. A diferencia del elemento anterior, no tiene ninguna relación con su ejecutor, pero necesita de este para poder existir.
- **Flujo de Control:** unidad básica de flujo, sirve para transmitir datos en general.
- **Realización:** elemento de flujo que permite solicitar la realización de alguna acción determinada.
- **Feedback:** elemento de flujo que permite representar interacciones que no involucran al actor de manera directa en la tarea, pero sí como apoyo o supervisión.
- **Composición:** representación de que un elemento está compuesto por otro.
- **Tarea:** unidad básica de trabajo.
- **Artifact:** representan la naturaleza tangible del proceso (modelos, documentos, código, etc.).
- **Outcome:** representa la naturaleza intangible del proceso (feedback, estado de avance, etc.).
- **Entregable:** representa la salida de un proceso que implique un valor para el cliente.
- **Proceso:** conjunto coherente de tareas, actividades, procedimientos, artefactos, etc. que son necesarios para el cumplimiento de objetivos específicos.
- **Iteración:** repetición de un proceso o conjunto de procesos con el fin de cumplir algún objetivo no cumplido en la anterior ejecución de estos.
- **Hito:** Acontecimiento puntual y significativo que marca un momento importante en el desarrollo de un proceso.
- **Planificación de Proceso:** momento en el cual se decide la forma en que se desarrollará un proceso, los tiempos que se respetarán y los objetivos que se perseguirán.
- **Componente de Proceso:** representa un elemento o conjunto de elementos que forman parte de un proceso. Un proceso puede estar formado por múltiples componentes.
- **Incidente:** situación puntual y específica que genera ineficiencias o riesgos en el desarrollo de un proceso.

B: Modelamiento de Scrum con SPEM

B1. Definición de Actores

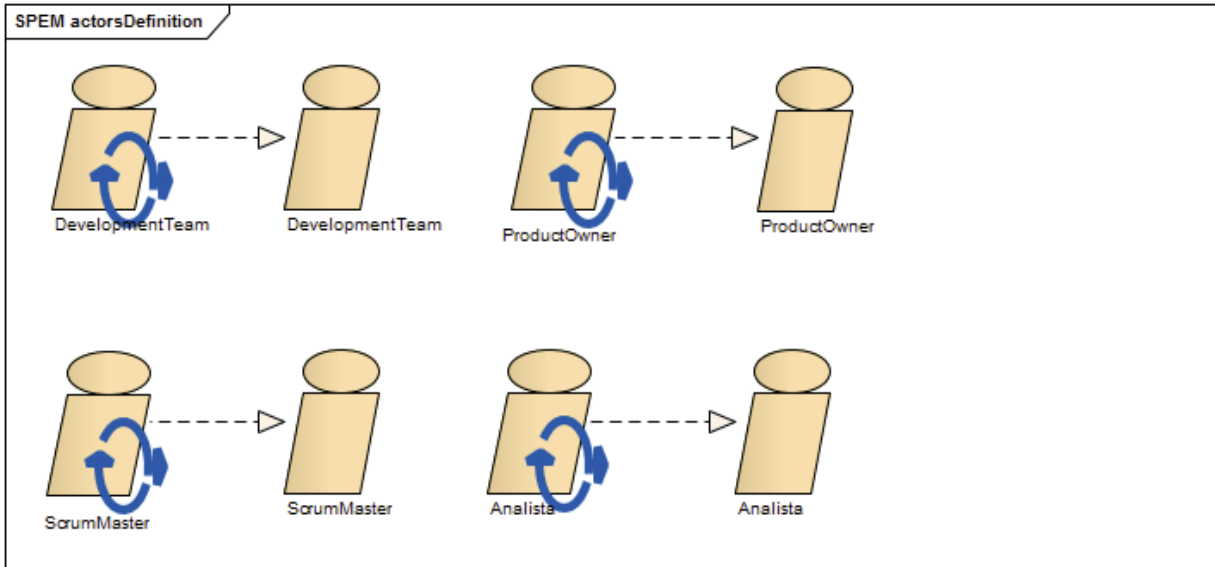


Figura B.1 Definición de Roles Scrum.

El primer paso para modelar cualquier proceso utilizando SPEM es definir a los actores involucrados. En la figura B.1 se puede observar que se han definido cuatro actores, tres de ellos corresponden a los miembros esenciales del equipo Scrum explicados en la sección 5.3.1 y adicionalmente se ha definido un analista. Se puede observar que se ha definido una estructura como la siguiente: Actor Realizador solicita a Actor Real que realice una acción; esta es la manera correcta de definir actores, pues, aunque parezca redundante, el Actor Realizador necesita a su versión real para ejecutar las acciones.

B2. Definición de Product Backlog

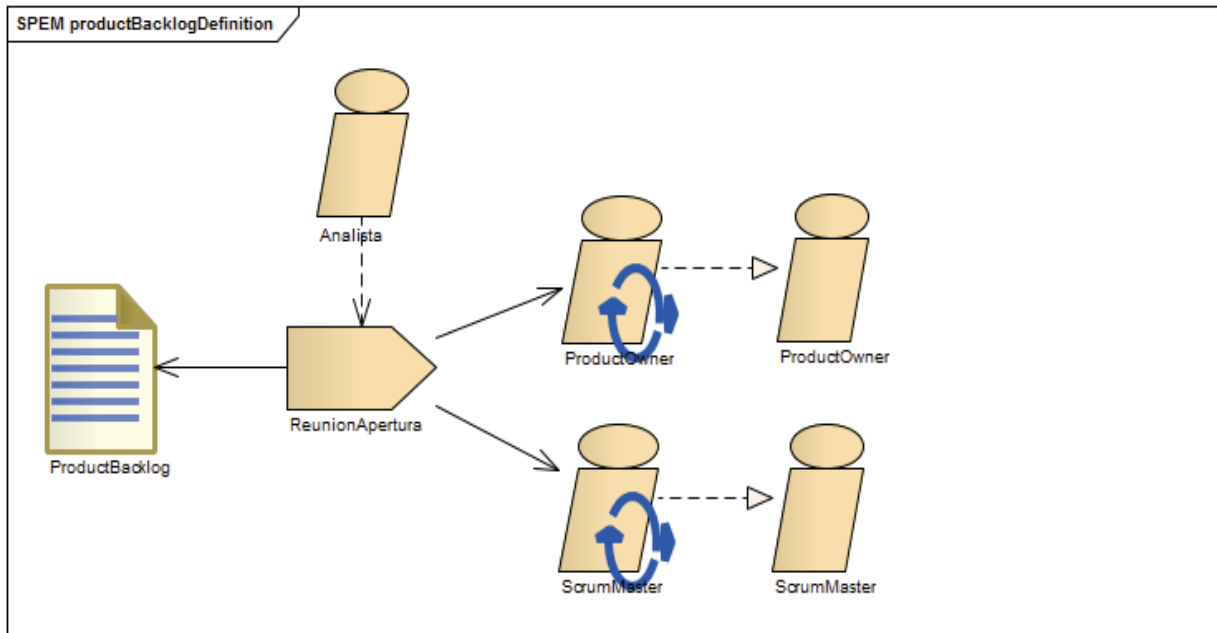


Figura B.2 Definición de Product Backlog.

En este modelo se describe el proceso de definición del Product Backlog, el cual es ejecutado tanto por el Product Owner como por el ScrumMaster a través de una reunión de apertura. El analista por su parte, también se involucra en la reunión de apertura, pero solamente como un apoyo, por lo que se le ha asignado una relación de Feedback con la respectiva tarea. Una vez finalizado el proceso, se deberá obtener el Artifact correspondiente al Product Backlog.

En este proceso, al igual que en la definición de los actores, se ha vuelto utilizar la misma estructura descrita, salvo en el caso del analista. La razón de esto es porque este último no realiza ninguna acción, sólo sirve de apoyo para los actores que realizan la tarea.

A partir de aquí y por motivos de orden y mejor entendimiento, se comenzará a obviar al Actor Real, es decir, a partir de ahora la aparición del Actor Realizador implica que hay una versión real de este realizando la acción, pero no aparecerá en el diagrama. Sólo se utilizará al Actor Real en situaciones muy específicas, las que serán explicadas en el momento que corresponda.

B3. Sprint Planning

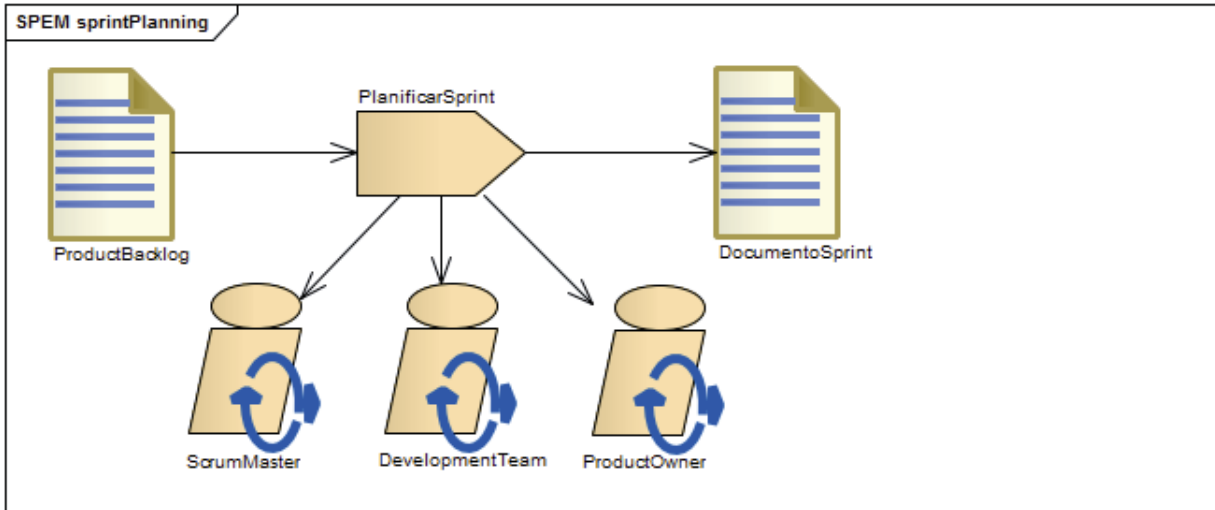


Figura B.3 Sprint Planning.

En este proceso deben participar todos los miembros del equipo Scrum. La entrada del proceso será el Product Backlog y su salida será el Documento Sprint el cual corresponde a un documento oficial que contiene todos los compromisos y decisiones respecto al Sprint actual.

B4. Daily Scrum

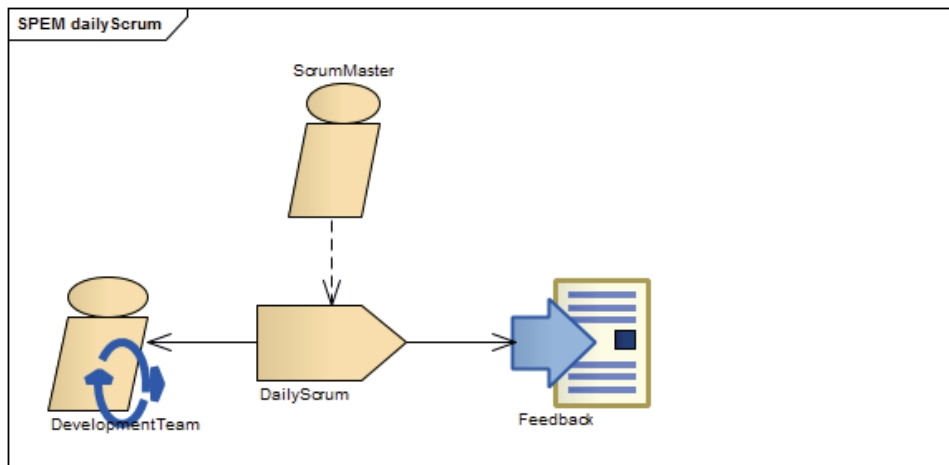


Figura B.4 Daily Scrum.

Este proceso es realizado únicamente por el Development Team pero, sin embargo, el Scrum Master también deberá supervisar muy de cerca todo lo conversado y decidido en esta instancia. El resultado de este proceso corresponde a un Outcome de Feedback que representa un gran aporte para el mejoramiento continuo que Scrum exige.

B5. Development Work

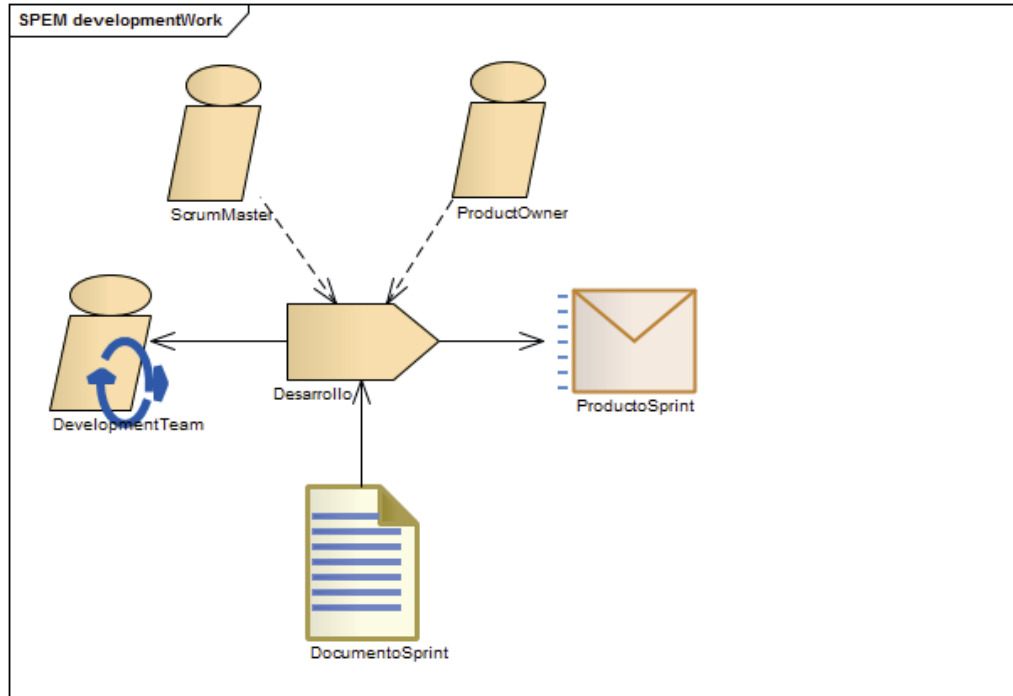


Figura B.5 Development Work.

Para llevar a cabo este proceso es necesario utilizar el Documento Sprint obtenido en el Sprint Planning. El único actor involucrado en la realización del desarrollo es el Development Team, pero tanto el Scrum Master como el Product Owner deberán supervisar cada etapa de él. El resultado de este proceso será un Producto Sprint que representará un entregable para el cliente.

B6. Sprint Review

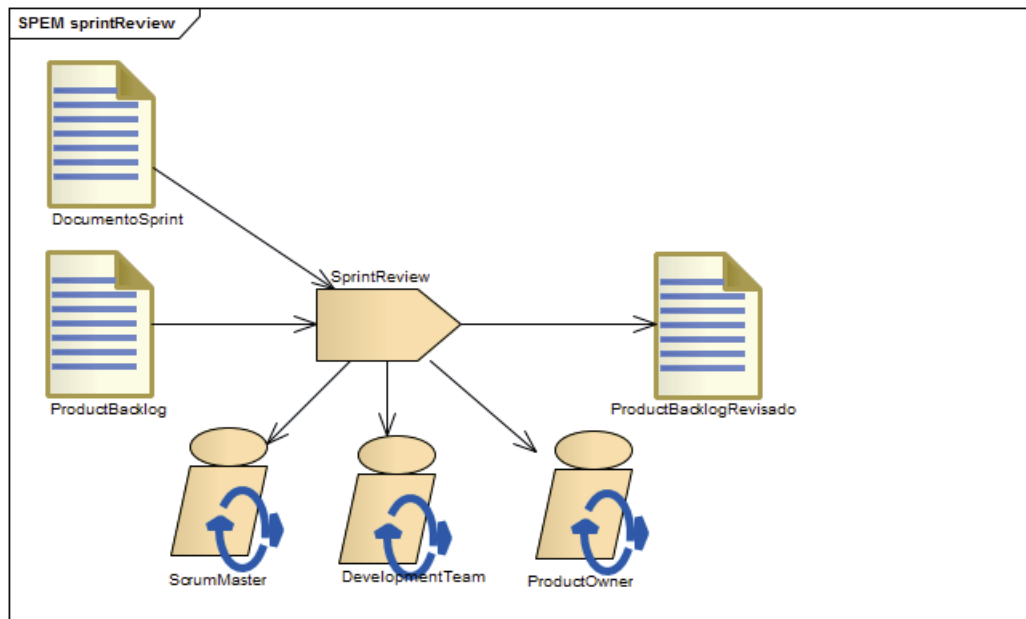


Figura B.6 Sprint Review.

En este proceso deberán participar todos los miembros del equipo Scrum y, además, serán necesarios tanto el Documento Sprint como el Product Backlog. Una vez realizado el análisis del Sprint, el resultado será un Product Backlog Revisado que servirá como entrada para el próximo Sprint.

B7. Sprint Retrospective

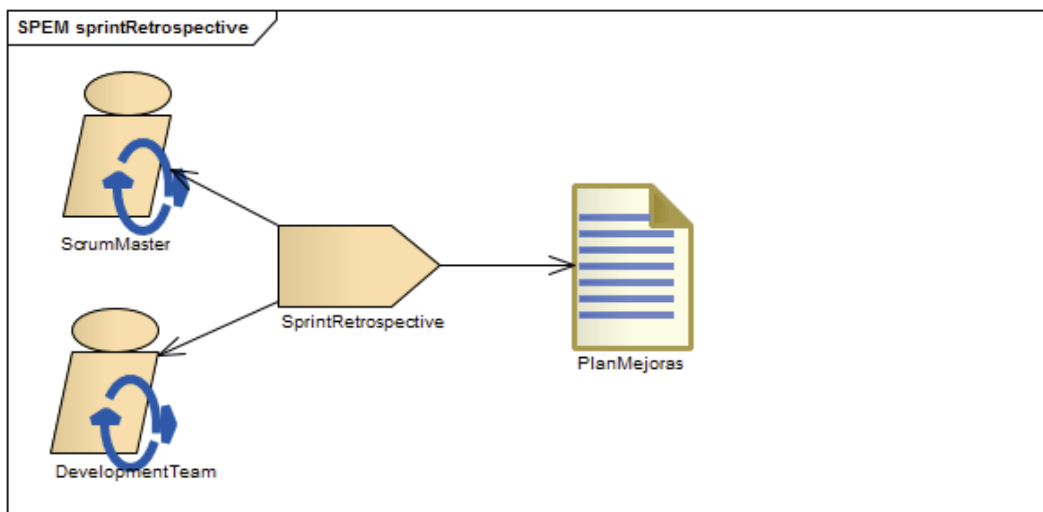


Figura B.7 Sprint Retrospective.

Para evaluar el rendimiento y los problemas ocurridos durante el Sprint, tanto el Scrum Master como el Development Team deberán realizar el análisis en conjunto, dando como resultado el plan de mejoras para el próximo Sprint.

C: Modelamiento de LSD con SPEM

Dado que LSD es una metodología mucho más ambigua que Scrum, para modelar cada uno de sus principios fue necesario diseñar una serie de escenarios en los cuales dichos principios fuesen utilizados.

C1. Definición de Actores

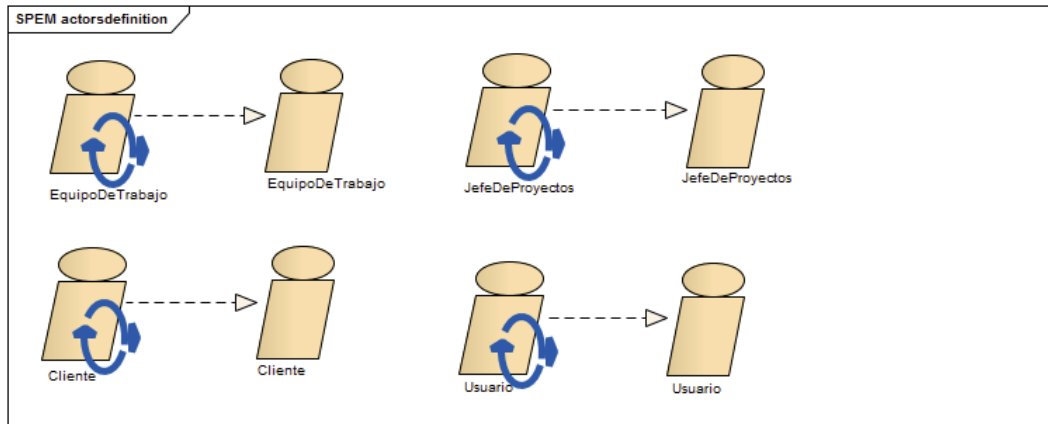


Figura C.1 Definición de Actores en LSD.

Para el modelamiento de LSD se han identificado cuatro actores: el equipo de trabajo, el jefe de proyectos, el cliente y el usuario, todos ellos con su respectivo Actor Real el cual, al igual que en el caso de los diagramas utilizados para modelar Scrum, se obviará a partir de ahora a menos que su utilización sea estrictamente necesaria.

C2. Eliminate Waste

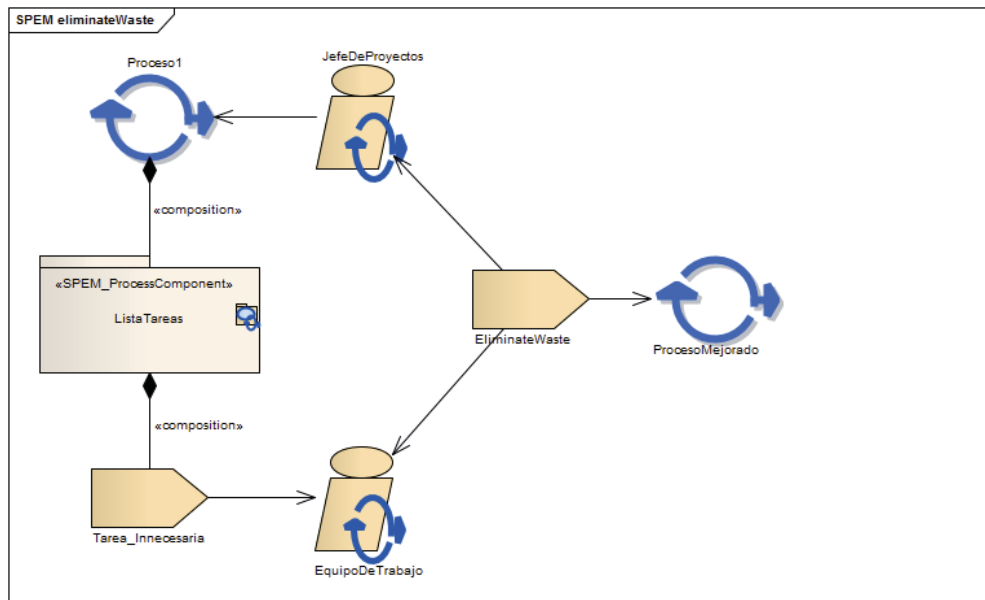


Figura C.2 Eliminate Waste.

Se pueden apreciar las siguientes situaciones:

- En la parte superior está el jefe de proyectos, el cual supervisa y gestiona las acciones y procedimientos que conforman el Proceso 1.
- El Proceso 1 está compuesto, entre otras cosas, por una Lista de Tareas, entre las que hay una que no aporta valor al cliente.
- En la parte inferior, el Equipo de Trabajo, encargado de realizar las tareas del Proceso 1, están realizando una tarea y notan que podría no ser necesaria.
- Al juntar los conocimientos del Equipo de Trabajo sobre la posible tarea innecesaria y la visión global del proceso que tiene el Jefe de Proyectos, se decide eliminar la tarea y con esto, obtener un proceso mejorado.

C3. Amplify Learning

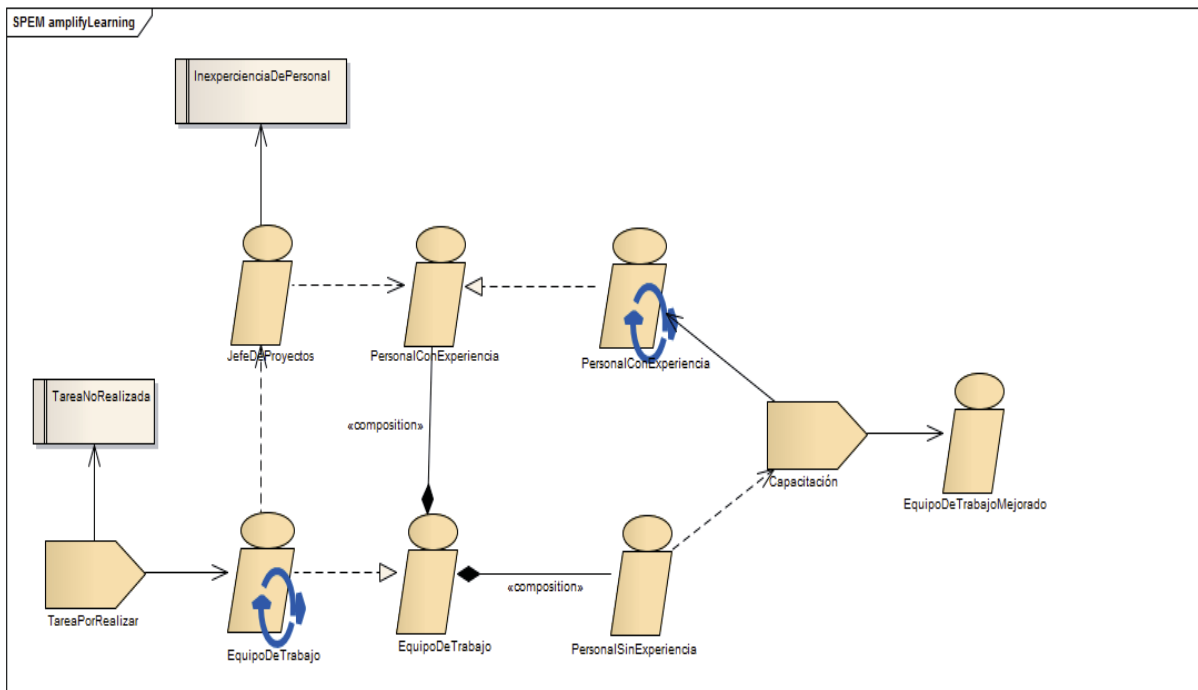


Figura C.3 Amplify Learning.

En este diagrama se observa que el Equipo de Trabajo intentó realizar una tarea la cual no pudo ser completada, por lo se desplegó un incidente avisando de este problema. El Jefe de proyectos por su parte recibe la información referente a esta falla por medio del análisis de la ejecución de la tarea y se da cuenta que el problema ocurrió por una falla del personal. Evidentemente en todo Equipo de trabajo siempre hay personas con más experiencia que otras, por lo que el Jefe de Proyecto decide separarlos en dos grupos y asignarles a sus trabajadores más experimentados la responsabilidad de capacitar a los menos experimentados. El resultado de todo esto es un Equipo de Trabajo más sólido y, por ende, más eficiente. Cabe recalcar que en este diagrama el Personal sin Experiencia no participa activamente en la ejecución de la Capacitación, sólo la recibe.

C4. Decide as Late as Possible

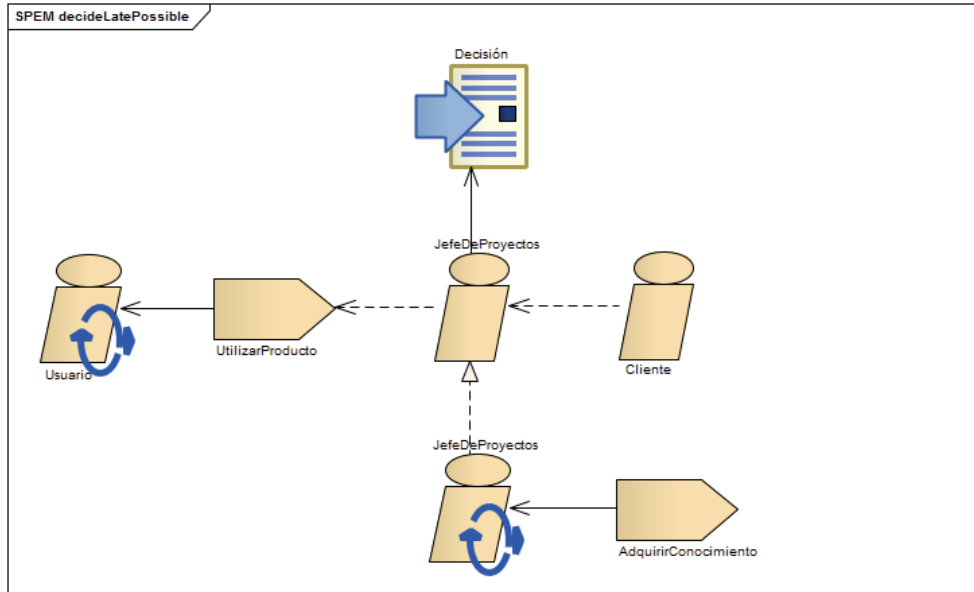


Figura C.4 Decide as Late as Possible.

En este diagrama el Jefe de Proyecto necesita tomar una decisión importante, pero para hacerlo primero necesita recibir el feedback del cliente respecto a sus necesidades más actualizadas, de modo de tomar la mejor decisión posible. Además, necesitará realizar pruebas de usabilidad con usuarios reales para comprobar si hay que hacer algún cambio significativo en el sistema y en un todo, deberá adquirir todo el conocimiento posible antes de decidir.

C5. Deliver as Fast as Possible

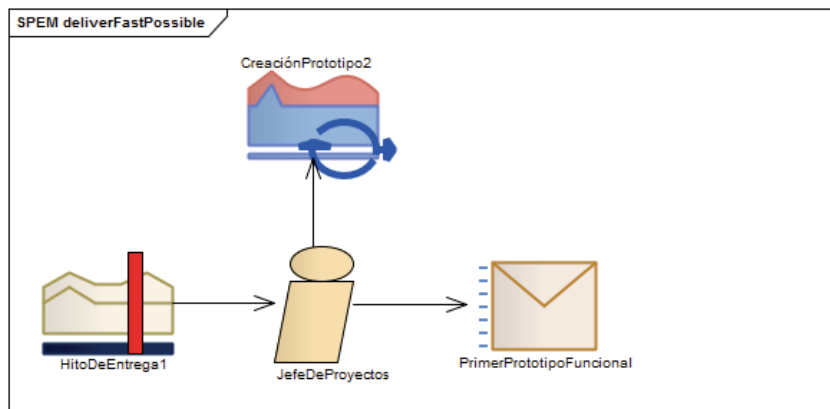


Figura C.5 Deliver as Fast as Possible.

Dentro de LSD se plantea la existencia de hitos de entrega a corto plazo, de modo de poder generarle valor al cliente lo antes posible. En este caso, se ha llegado al primer hito de entrega del proyecto y, por ende, el primer prototipo funcional está listo. Este contendrá sólo algunas de las características pedidas y, una vez entregado, se comenzará con la próxima iteración.

C6. Empower the Team

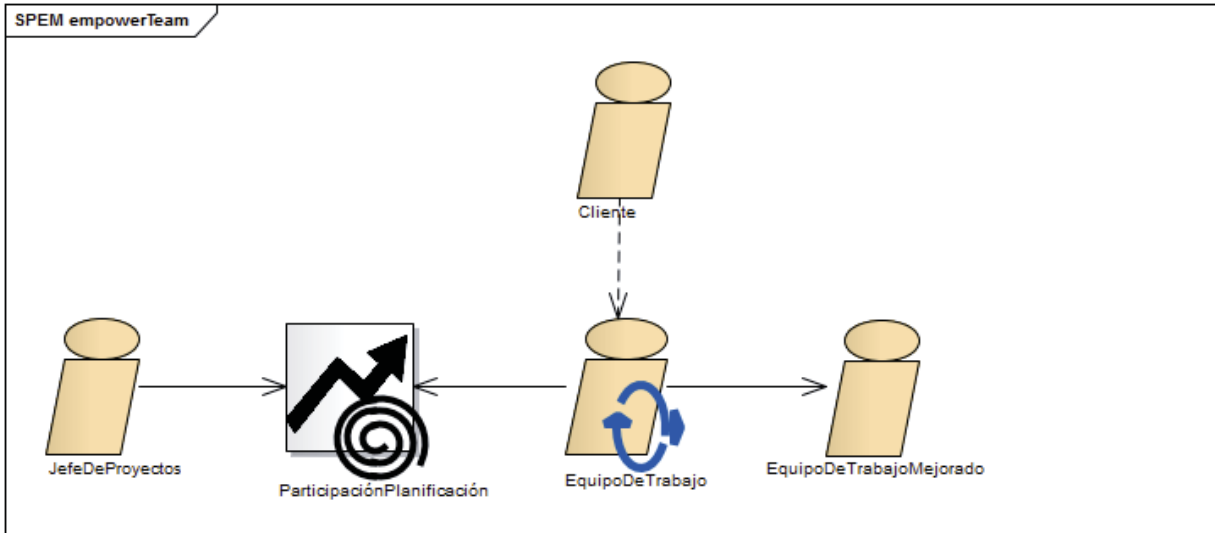


Figura C.6 Empower the Team.

En este diagrama se puede apreciar que el Jefe de Proyectos ha decidido otorgarle una mayor participación en el proceso de planificación de los procesos al Equipo de Trabajo. Ahora el Equipo de Trabajo deberá relacionarse con el cliente y tomar decisiones, lo que les genera una mayor motivación y, por ende, mejor rendimiento.

C7. Build Integrity

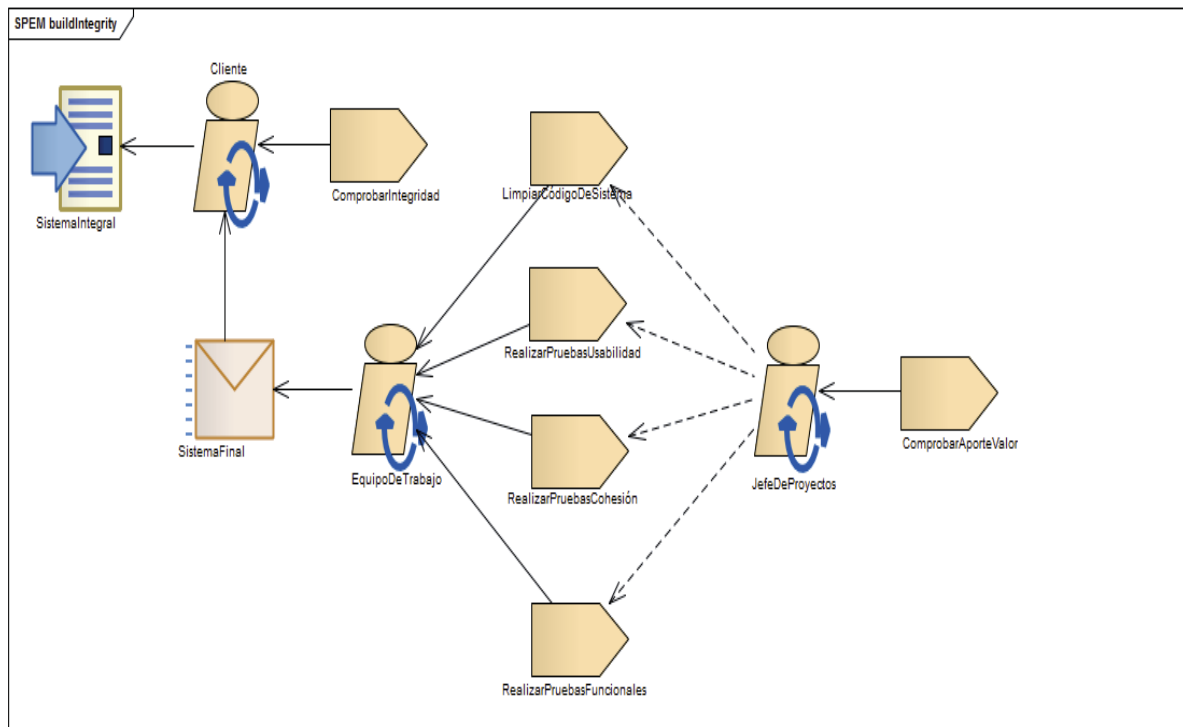


Figura C.7 Build Integrity.

Se producen las siguientes situaciones:

- El Equipo de Trabajo realiza un conjunto de pruebas de sistema para corroborar que este cumpla con todos los requerimientos, tenga la menor cantidad de errores y sea exactamente lo que el cliente pedía.
- El Jefe de Proyecto verifica estas pruebas para comprobar que ninguna de ellas sea innecesaria, de ser así se consideraría como basura y deberá ser eliminada del proceso.
- Una vez finalizadas las pruebas el Equipo de Trabajo le entrega el sistema terminado al cliente.
- Al recibir el sistema terminado, el cliente lo revisa exhaustivamente en busca de algún error, alguna inconsistencia o algo que simplemente no le agrade.
- Si el cliente no logra encontrar ningún problema en el sistema, informará al equipo involucrado en la realización del mismo que el sistema cumple con todas sus expectativas, esto quiere decir, el sistema es integral.

C8. See the Whole

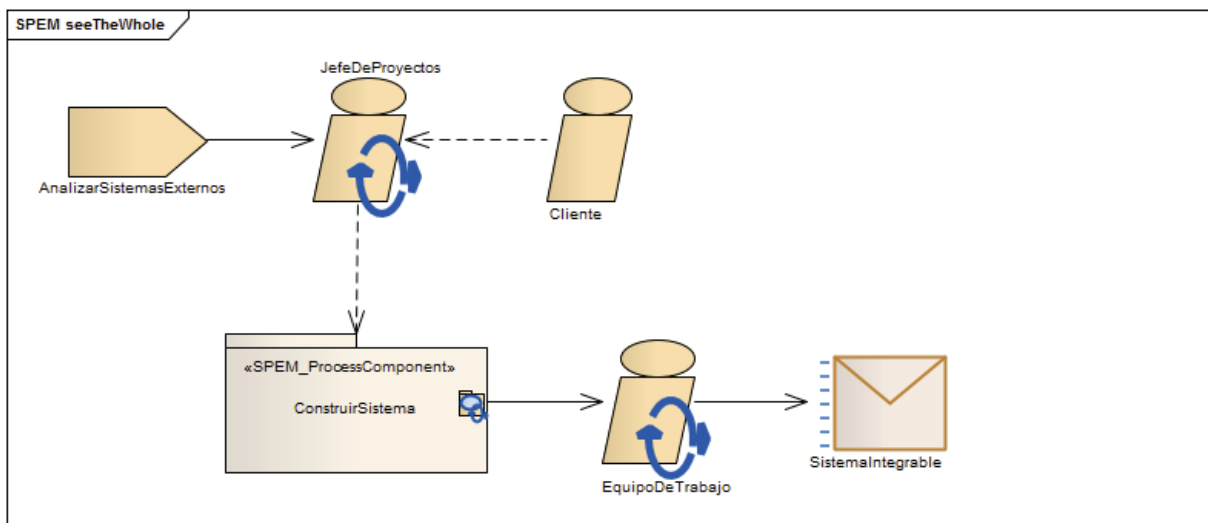


Figura C.8 See the Whole.

El Equipo de Trabajo debe construir el Sistema, pero para asegurarse de que el sistema sea integrable el Jefe de Proyectos debe entregarle toda la información necesaria respecto a los sistemas que interactuarán con el que se va a construir. Para lograr esto, el Jefe de Proyectos deberá recabar toda la información que pueda con el cliente y analizar lo máximo posible la composición y funcionamiento de los sistemas externos. Si todo lo anterior se realiza correctamente, el sistema podrá ser integrable.

D: Resultados de Estudio

D1. Métricas Cuantitativas

Como se explicó en la sección 12.1., las métricas cuantitativas se obtuvieron mediante la utilización de la aplicación web JIRA en conjunto con presentaciones orales que los grupos debieron realizar al término de cada iteración. Se formaron un total de 8 grupos de trabajo diferentes, cada uno con su propio proyecto particular, pero de estos, sólo serán evaluados 7 de ellos, ya que hubo un grupo que no utilizó adecuadamente la aplicación JIRA y por lo tanto la información ingresada por estos no es consistente con el resto de grupos. A continuación, se detallarán los resultados obtenidos mediante una serie de gráficos que permitirán extraer información relevante para la construcción de las métricas cuantitativas.

D1.1. Datos Lead Time / Touch Time

Los gráficos presentados a continuación muestran las incidencias creadas a lo largo del proyecto (representadas por puntos rojos) versus las resueltas (representadas por puntos verdes). En el contexto de JIRA, se entiende por incidencia todas las tareas, historias de usuario, épicas y errores pertenecientes a un proyecto. El gráfico viene, además, acompañado con una tabla de datos en la que se muestra de forma más detallada las fechas exactas en que las incidencias fueron agregadas a JIRA, las fechas en que fueron completadas y el tipo de incidencia para cada caso. En este estudio, con el fin de facilitar la interpretación de los datos, se les solicitó a los estudiantes que limitaran la creación de incidencias sólo a historias de usuario y tareas, siendo estas últimas, parte de las primeras (una historia de usuario está compuesta por una o más tareas).

Adicionalmente, se ha diseñado una tabla creada a partir de los datos proporcionados por los estudiantes en sus respectivas presentaciones, en la cual se muestra la cantidad exacta de horas que estos demoraron en realizar cada historia de usuario y en qué iteración fueron realizadas. Con estos datos sumados a los extraídos del gráfico, es posible calcular tanto el valor del Touch Time (que coincide con el valor de las horas efectivamente trabajadas) y el Lead Time (realizando un cálculo sencillo entre la fecha en que la incidencia fue creada y la fecha en que se dio por finalizada dentro de JIRA) para cada historia de usuario. Cabe recalcar que aquellos proyectos que poseen incidencias sin resolver, es decir, que no fueron ejecutadas durante el período de duración del semestre, no serán consideradas en este análisis, ya que no aportarían ninguna información relevante. La única excepción a esta regla se aplicará a aquellas incidencias que los mismos estudiantes señalaron como resueltas en sus respectivas presentaciones y que por el motivo que sea nunca llegaron a reflejarlo en JIRA, para estos casos, sí se considerará la incidencia, pero sólo para el cálculo del Touch Time, ya que el Lead Time tendría tiempos erróneos.

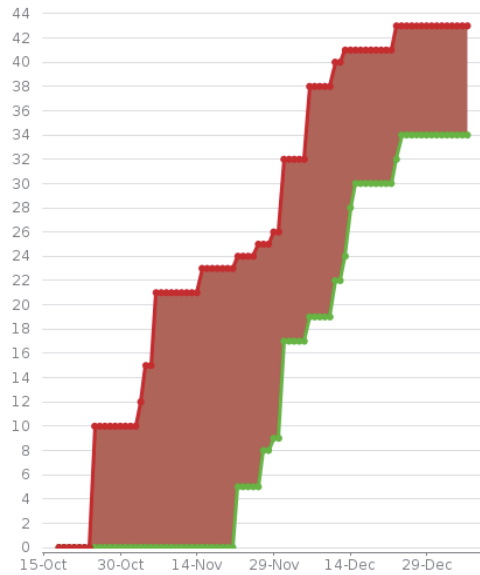


Tabla de datos

Periodo	Creada	Resuelta
25-October-2016	10 (Historia 7 Tarea 3)	0
3-November-2016	2 (Historia 2)	0
4-November-2016	3 (Historia 3)	0
6-November-2016	6 (Historia 6)	0
15-November-2016	2 (Historia 1 Tarea 1)	0
22-November-2016	1 (Tarea 1)	5 (Historia 3 Tarea 2)
26-November-2016	1 (Tarea 1)	0
27-November-2016	0	3 (Historia 1 Tarea 2)
29-November-2016	1 (Historia 1)	1 (Historia 1)
1-December-2016	6 (Tarea 6)	8 (Historia 2 Tarea 6)
6-December-2016	6 (Tarea 6)	2 (Tarea 2)
11-December-2016	2 (Tarea 2)	3 (Historia 1 Tarea 2)
13-December-2016	1 (Tarea 1)	2 (Tarea 2)
14-December-2016	0	4 (Historia 1 Tarea 3)
15-December-2016	0	2 (Historia 2)
23-December-2016	2 (Historia 1 Tarea 1)	2 (Historia 1 Tarea 1)
24-December-2016	0	2 (Historia 1 Tarea 1)

Figura D.1 Diagrama de Flujo Acumulado Grupo 1.

Tabla D.1 Cálculo Lead Time / Touch Time Grupo 1.

Nº Iteración	Nº Historia	Tiempo utilizado (Hrs.)	Lead Time (Días)	Touch Time (Hrs.)
1	1	40	38	40
	2	25	38	25
	3	50	29	50
	4	65	24	65
	5	30	17	30
	6	45	24	45
	7	20	8	20
2	8	20	51	20
	9	110	40	110
	10	70	48	70
	11	60	17	60
3	12	45	51	45
	13	45	1	45

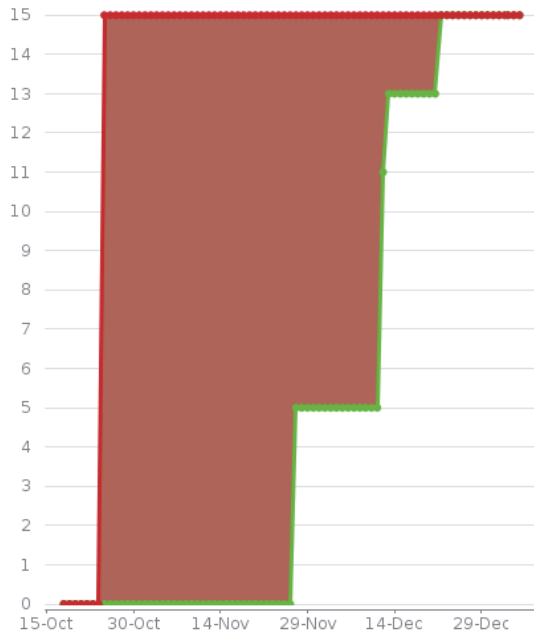


Tabla de datos

Periodo	Creada	Resuelta
25-October-2016	15 (Historia 15)	0
27-November-2016	0	5 (Historia 5)
12-December-2016	0	6 (Historia 6)
13-December-2016	0	2 (Historia 2)
22-December-2016	0	2 (Historia 2)

Figura D.2 Diagrama de Flujo Acumulado Grupo 2.

Tabla D.2 Cálculo Lead Time / Touch Time Grupo 2.

Nº Iteración	Nº Historia	Tiempo utilizado (Hrs.)	Lead Time (Días)	Touch Time (Hrs.)
1	1	27	34	27
	2	14	34	14
	3	32	34	32
	4	24	34	24
	5	45	34	45
2	6	18	49	18
	7	26	49	26
	8	34	49	34
	9	15	50	15
	10	15	50	15
	11	9	49	9
	12	12	49	12
	13	18	49	18
3	14	24	59	24
	15	6	59	6

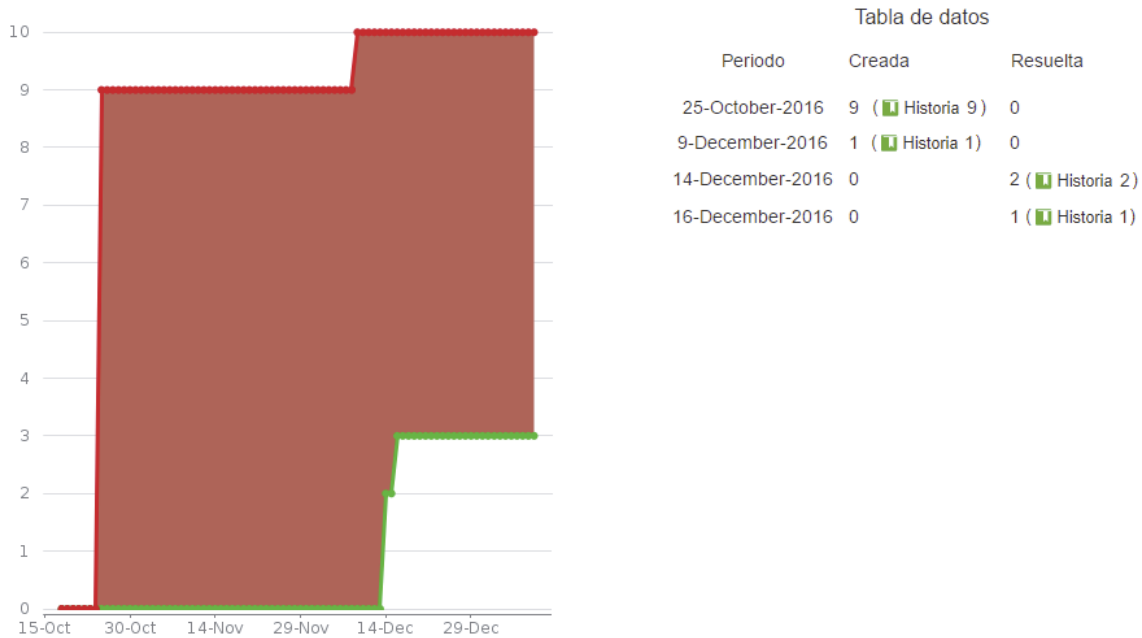
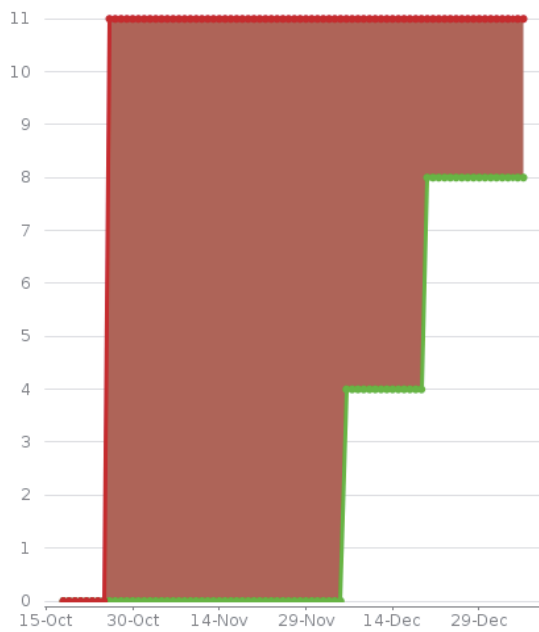


Figura D.3 Diagrama de Flujo Acumulado Grupo 3.

Tabla D.3 Cálculo Lead Time / Touch Time Grupo 3.

N° Iteración	N° Historia	Tiempo utilizado (Hrs.)	Lead Time (Días)	Touch Time (Hrs.)
1	1	15	51	15
	2	18	-	18
	3	18	53	18
	4	20	51	20
2	5	22	-	22
	6	32	-	32
3	7	12	-	12
	8	22	-	22
	9	19	-	19



Periodo	Creada	Resuelta
26-October-2016	11 (■ Historia 11)	0
6-December-2016	0	4 (■ Historia 4)
20-December-2016	0	4 (■ Historia 4)

Figura D.4 Diagrama de Flujo Acumulado Grupo 4.

Tabla D.4 Cálculo Lead Time / Touch Time Grupo 4.

N° Iteración	N° Historia	Tiempo utilizado (Hrs.)	Lead Time (Días)	Touch Time (Hrs.)
1	1	10	42	10
	2	22	42	22
	3	25	42	25
	4	30	42	30
2	5	3	56	3
	6	1	56	1
	7	4	56	4
	8	3	56	3
3	9	8	-	8
	10	4	-	4
	11	7	-	7

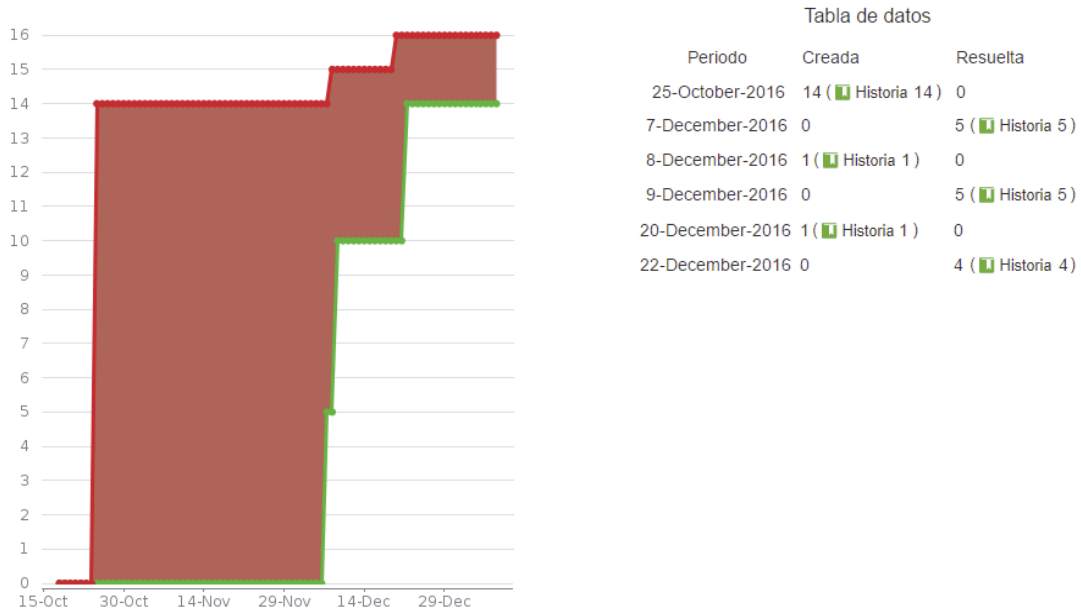


Figura D.5 Diagrama de Flujo Acumulado Grupo 5.

Tabla D.5 Cálculo Lead Time / Touch Time Grupo 5.

N° Iteración	N° Historia	Tiempo utilizado (Hrs.)	Lead Time (Días)	Touch Time (Hrs.)
1	1	2.5	44	2.5
	2	2.5	44	2.5
	3	5	44	5
	4	24	44	24
	5	2	44	2
2	6	3	46	3
	7	1.5	46	1.5
	8	2	46	2
	9	0.5	46	0.5
	10	1	46	1
3	11	1.5	59	1.5
	12	2	59	2
	13	4.5	59	4.5
	14	4	59	4

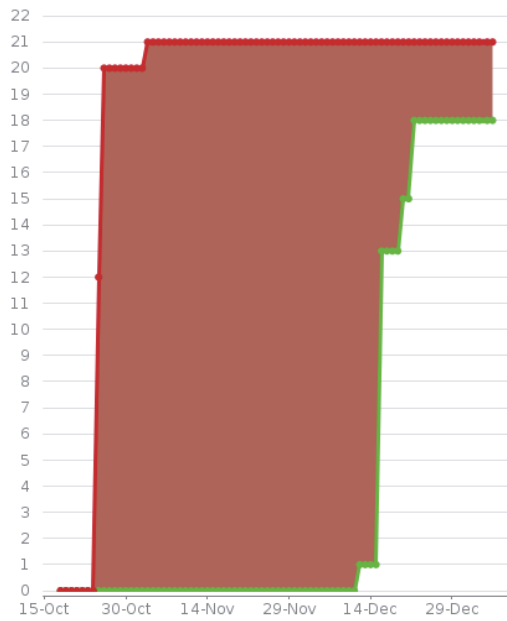


Tabla de datos

Periodo	Creada	Resuelta
25-October-2016	12 (Historia 12)	0
26-October-2016	8 (Historia 8)	0
3-November-2016	1 (Historia 1)	0
12-December-2016	0	1 (Historia 1)
16-December-2016	0	12 (Historia 12)
20-December-2016	0	2 (Historia 2)
22-December-2016	0	3 (Historia 3)

Figura D.6 Diagrama de Flujo Acumulado Grupo 6.

Tabla D.6 Cálculo Lead Time / Touch Time Grupo 6.

Nº Iteración	Nº Historia	Tiempo utilizado (Hrs.)	Lead Time (Días)	Touch Time (Hrs.)
1	1	4	49	4
	2	3	52	3
	3	4	52	4
	4	5	44	5
	5	6	53	6
2	6	2	52	2
	7	1.5	52	1.5
	8	2.5	53	2.5
	9	6	53	6
	10	3	53	3
	11	5	53	5
	12	4	53	4
	13	6	53	6
3	14	5	56	5
	15	9	56	9
	16	12	58	12
	17	15	59	15
	18	18	59	18

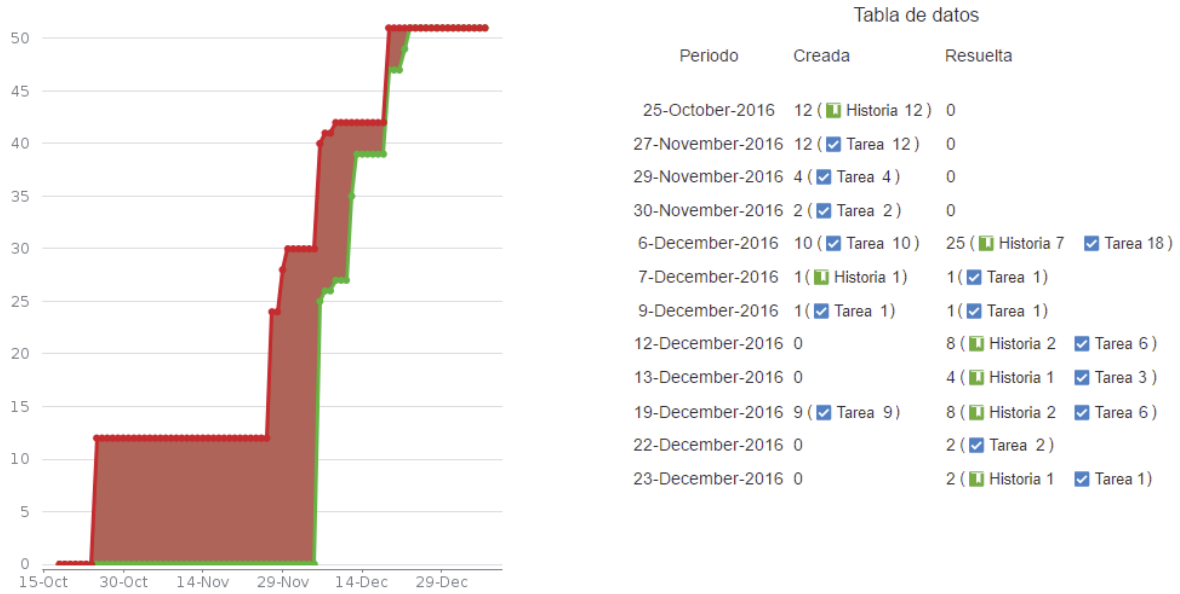


Figura D.7 Diagrama de Flujo Acumulado Grupo 7.

Tabla D.7 Cálculo Lead Time / Touch Time Grupo 7.

N° Iteración	N° Historia	Tiempo utilizado (Hrs.)	Lead Time (Días)	Touch Time (Hrs.)
1	1	5	43	5
	2	2	43	2
	3	2.5	43	2.5
	4	2.5	43	2.5
	5	3	43	3
	6	5	43	5
	7	8	43	8
2	8	6	49	6
	9	3	49	3
	10	4	50	4
3	11	2	60	2
	12	2	56	2
	13	3	13	3

D1.2. Datos de Incidencias por Iteración

Los gráficos presentados a continuación, muestran de manera clara y concisa la cantidad total de incidencias que los grupos establecieron para cada una de las iteraciones de sus respectivos proyectos dentro de JIRA. Se puede observar que algunos de estos gráficos poseen una sección denominada “Ninguno”, esta sección corresponde a aquellas incidencias que por algún motivo no fueron asignadas a ninguna iteración durante el proyecto. Cabe recalcar que estos gráficos sólo muestran la cantidad total de incidencias existentes en cada una de las secciones que conformar el gráfico, pero en ningún caso se entrega información referente a si la incidencia fue o no resuelta. También se puede observar cierta discrepancia en los nombres de cada sección entre un gráfico y otro, esto se debe principalmente a que fueron los propios estudiantes quienes decidieron con qué nombre identificarían a las iteraciones dentro de JIRA.

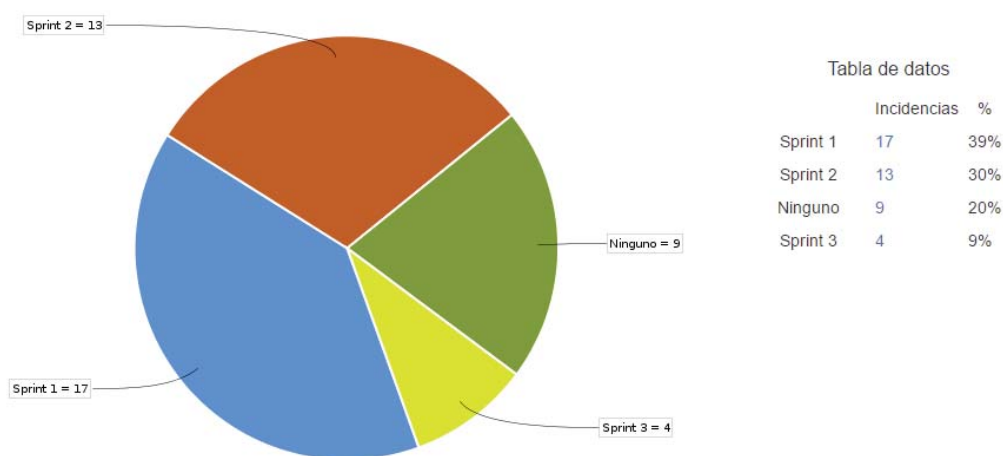


Figura D.8 Gráfico Incidencias por Iteración Grupo 1.

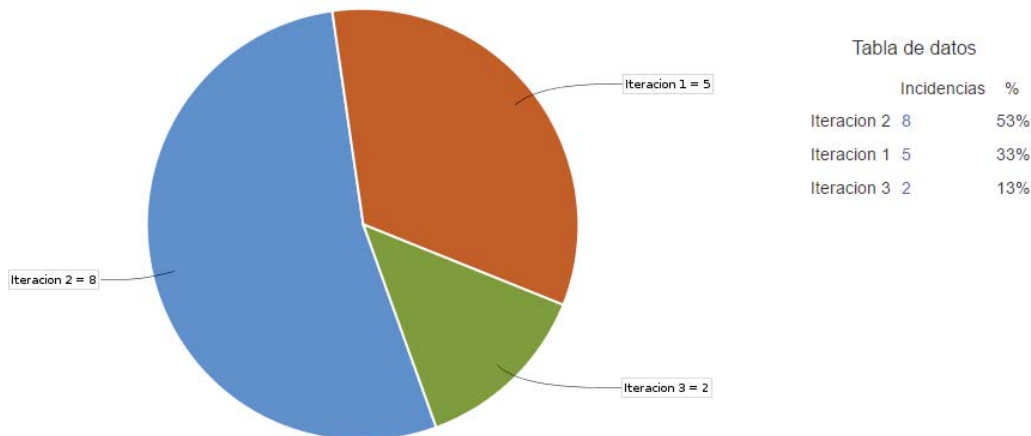


Figura D.9 Gráfico Incidencias por Iteración Grupo 2.

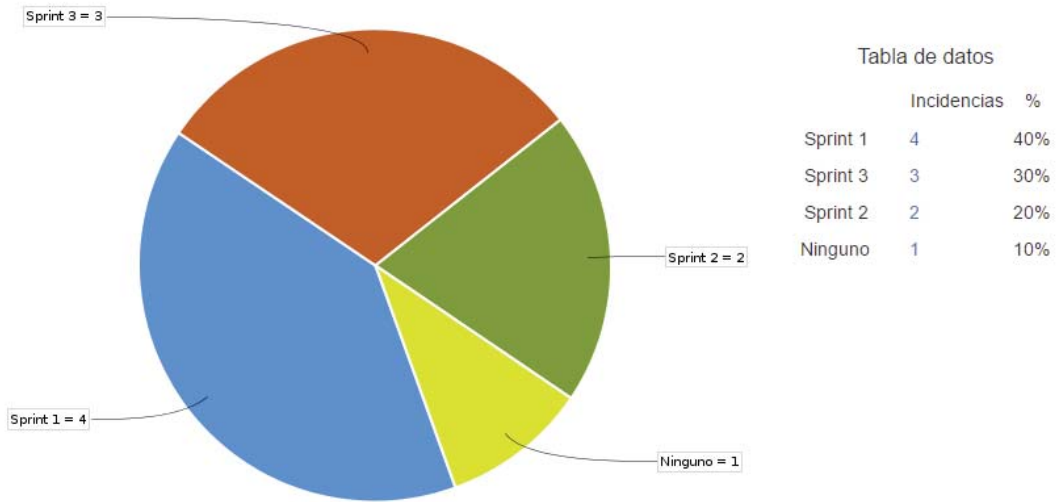


Figura D.10 Gráfico Incidencias por Iteración Grupo 3.

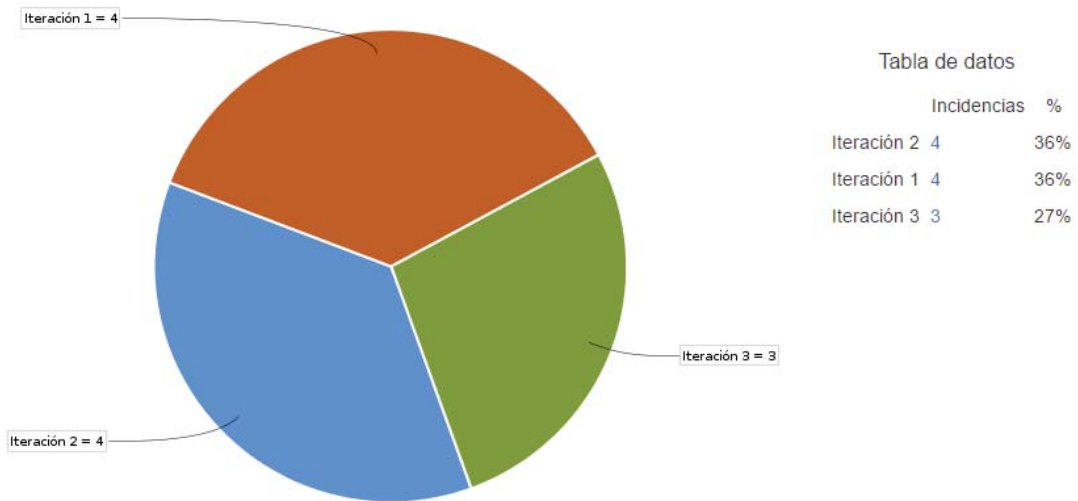


Figura D.11 Gráfico Incidencias por Iteración Grupo 4.

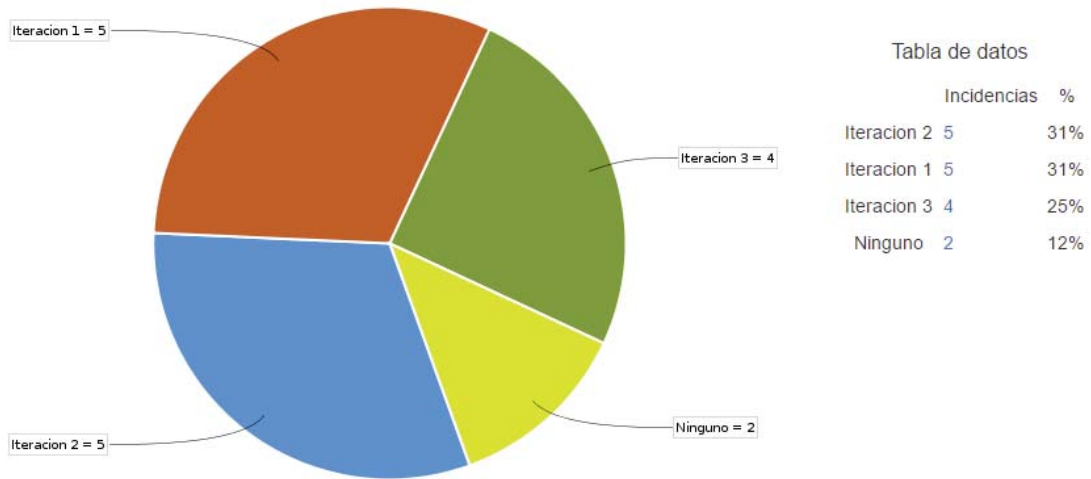


Figura D.12 Gráfico Incidencias por Iteración Grupo 5.

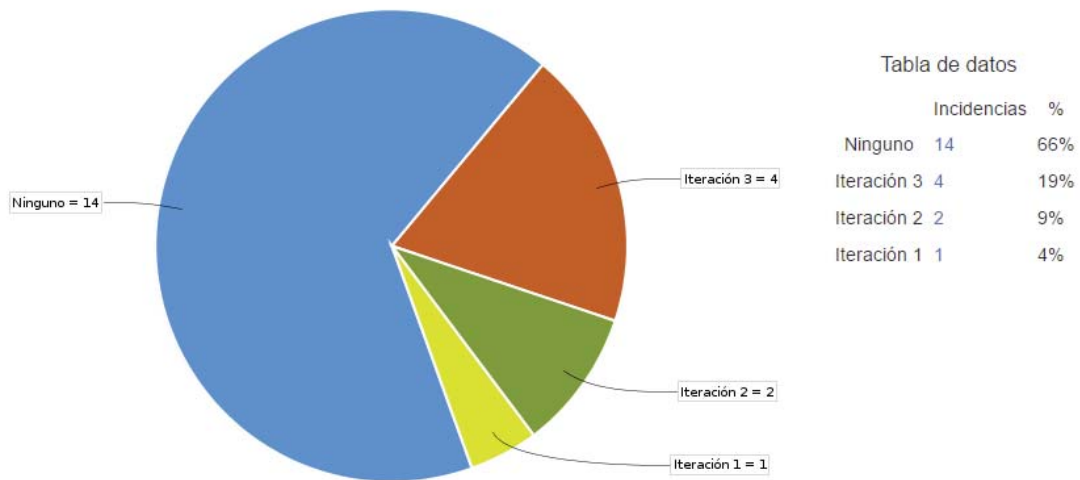


Figura D.13 Gráfico Incidencias por Iteración Grupo 6.

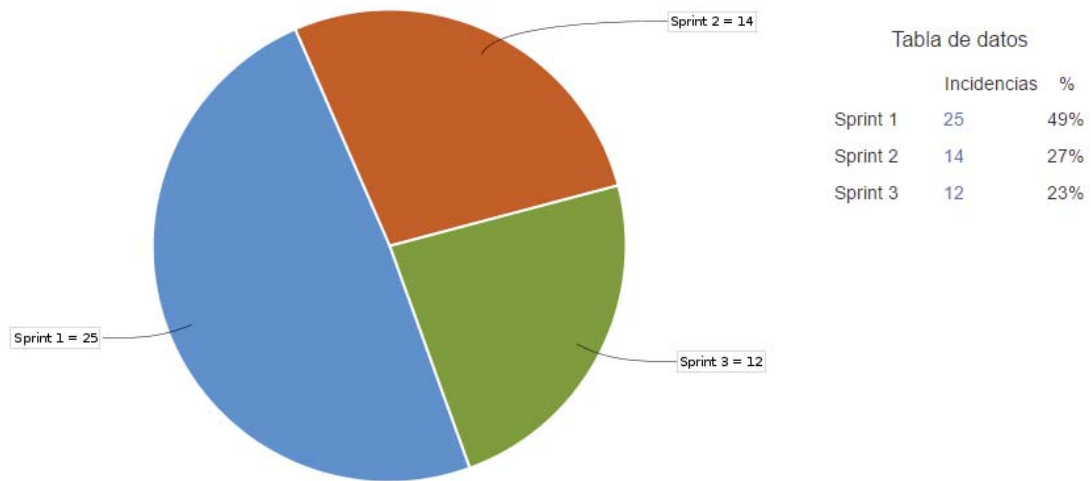


Figura D.14 Gráfico Incidencias por Iteración Grupo 7.

Tabla D.8 Resumen Gráficos Incidencias por Iteración.

Grupo	1° Iteración	2° Iteración	3° Iteración	Ninguna	Total
1	17	13	4	9	43
2	5	8	2	0	15
3	4	2	3	1	10
4	4	4	3	0	11
5	5	5	4	2	16
6	1	2	4	14	21
7	25	14	12	0	51

D1.3. Datos de Incidencias No Completadas

Similar al caso anterior, los gráficos siguientes muestran la cantidad total de incidencias que los estudiantes pudieron completar durante el ciclo de vida del proyecto y la cantidad total de incidencias que se quedaron sin resolver.

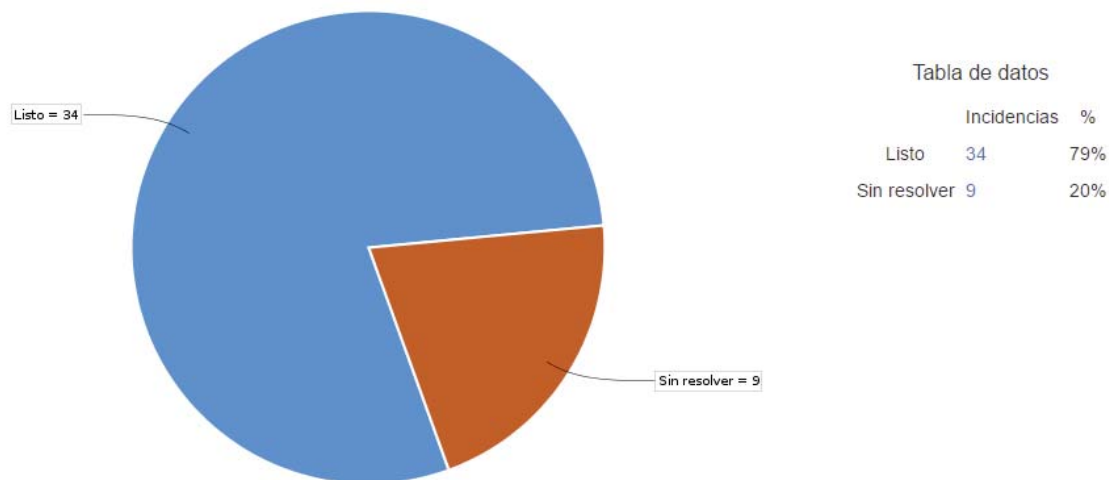


Figura D.15 Gráfico Incidencias No Completadas Grupo 1.

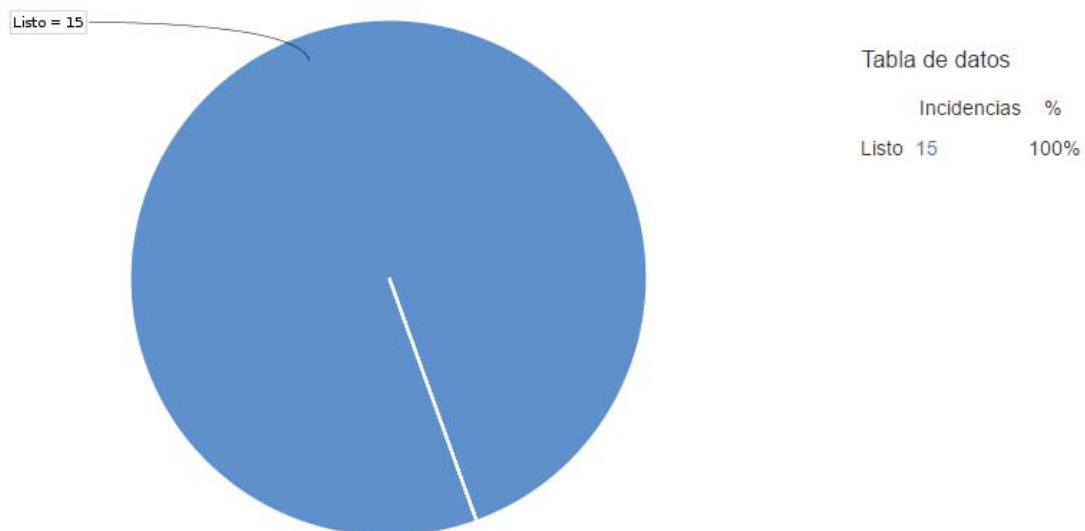


Figura D.16 Gráfico Incidencias No Completadas Grupo 2.

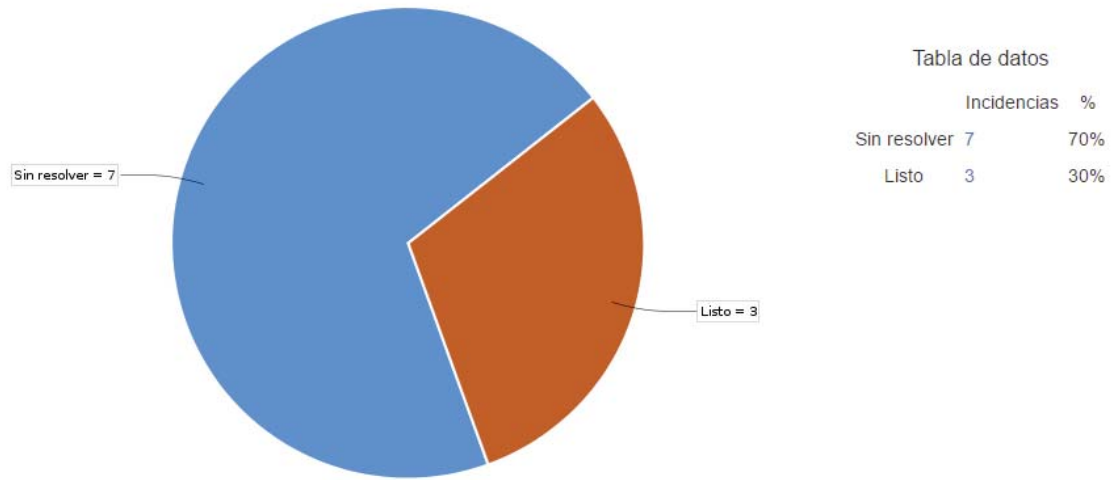


Figura D.17 Gráfico Incidencias No Completadas Grupo 3.

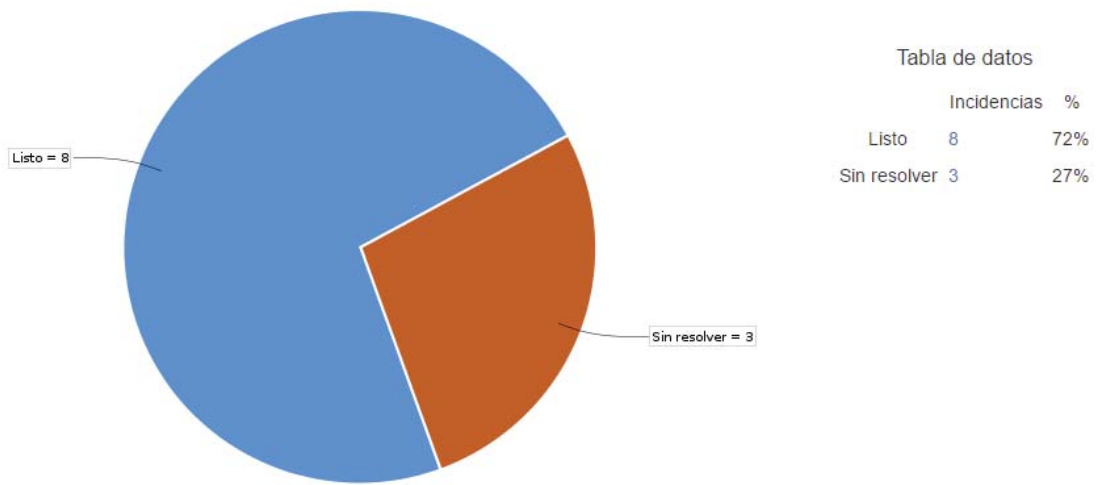


Figura D.18 Gráfico Incidencias No Completadas Grupo 4.

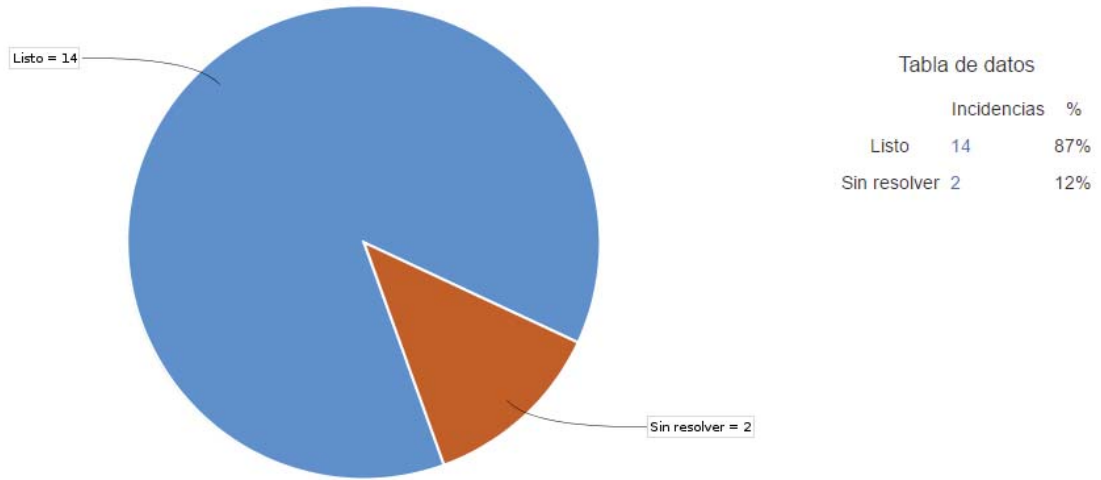


Figura D.19 Gráfico Incidencias No Completadas Grupo 5.

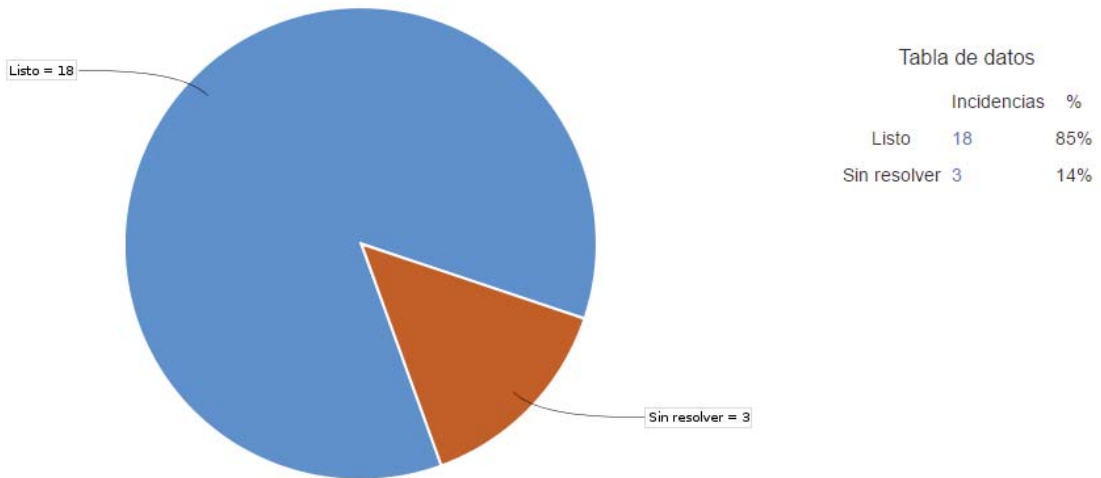


Figura D.20 Gráfico Incidencias No Completadas Grupo 6.

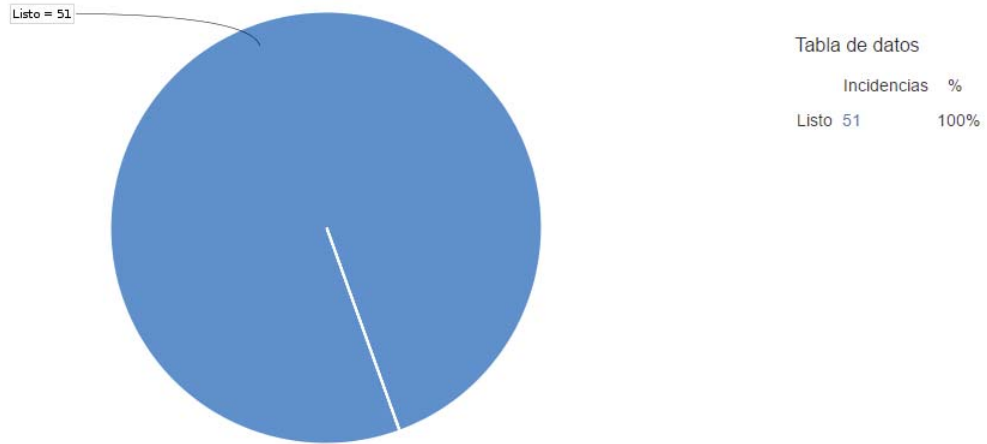


Figura D.21 Gráfico Incidencias No Completadas Grupo 7.

Tabla D.9 Resumen Gráficos Incidencias No Completadas.

Grupo	Completadas	No Completados	Total
1	34	9	43
2	15	0	15
3	3	7	10
4	8	3	11
5	14	2	16
6	18	3	21
7	51	0	51

D1.4. Datos Velocidad y Requisitos No Completados

A continuación, se muestra una tabla que contiene los valores correspondientes a las métricas de velocidad y requisitos no completados de cada equipo de trabajo. Como se explicó en las secciones 11.1.3. y 11.1.4. respectivamente, la velocidad está relacionada con la cantidad de incidencias que los grupos fueron capaces de resolver en cada iteración, mientras que los requisitos no completados corresponden a aquellas incidencias que fueron planificadas pero no resueltas durante dichas iteraciones, y para poder llegar a estos valores, se tuvo que realizar un análisis que involucró a los datos obtenidos en las secciones D1.1., D1.2. y D1.3. del presente anexo. De la sección D1.1., se verificó la información contenida en las tablas para determinar aquellas incidencias que, si bien en las presentaciones los estudiantes señalaron que las habían terminado, no lo reflejaron en JIRA. De las secciones D1.2. y D1.3. por su parte, se analizaron y compararon los datos obtenidos en las tablas resumen, de modo de comprobar la discrepancia existente entre ambas, lo cual, sumado al estudio de la sección D1.1., permitió finalmente construir la tabla siguiente.

Tabla D.10 Cálculo de Velocidad y Requerimientos No Completados.

Grupo	1° Iteración		2° Iteración		3° Iteración		Total	
	C	N	C	N	C	N	C	N
1	17	0	13	0	4	0	34	0
2	5	0	8	0	2	0	15	0
3	3	1	0	2	0	3	3	6
4	4	0	4	0	0	3	8	3
5	5	0	5	0	4	0	14	0
6	5	0	5	0	8	0	18	0
7	25	0	14	0	12	0	51	0

C = Incidencias completadas; N = incidencias no completadas.

D1.5. Datos Velocidad de Estabilización

Como se explicó en la sección 12.1., JIRA ofrece una interfaz para la gestión de incidencias denominada Pizarra, la cual se puede modificar y adaptar a las necesidades de cada proyecto. Para el caso de este estudio, las pizarras de los distintos proyectos fueron adaptadas para comportarse como un tablero Kanban, de modo de poder estudiar su comportamiento y determinar si los estudiantes han enfrentado algún cuello de botella durante el desarrollo de sus proyectos. La pizarra utilizada por los estudiantes es la siguiente.

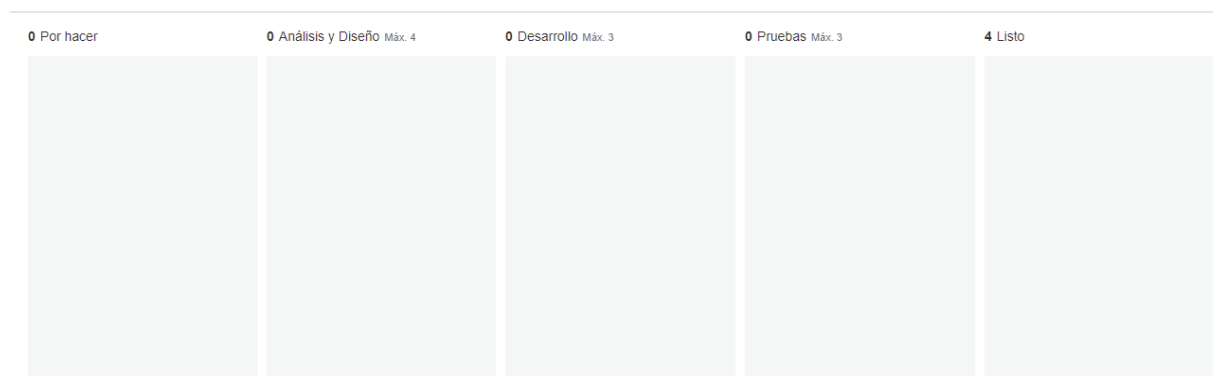


Figura D.22 Pizarra Estilo Kanban en JIRA.

En esta pizarra, cada una de sus columnas representan un lane dentro de Kanban, los cuales representan a su vez las distintas etapas que involucran la elaboración de una incidencia. En la cabecera de cada columna, aparecen dos números, uno a la izquierda que representa el número de incidencias que tiene el lane y otro a la derecha que representa la cantidad máxima de incidencias que están permitidas para dicho lane.

A continuación, se presentan una serie de gráficos basados en esta pizarra, con los cuales podremos determinar si los estudiantes tuvieron o no cuellos de botella durante su desarrollo y cuánto tiempo tardaron en corregirlos.

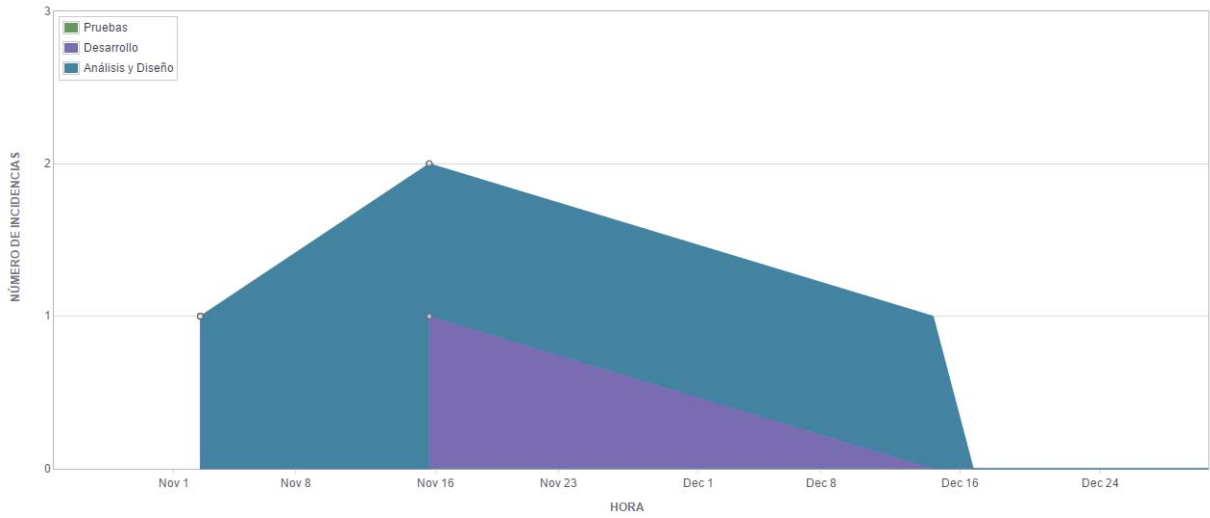


Figura D.25 Gráfico Velocidad de Estabilización Grupo 3.

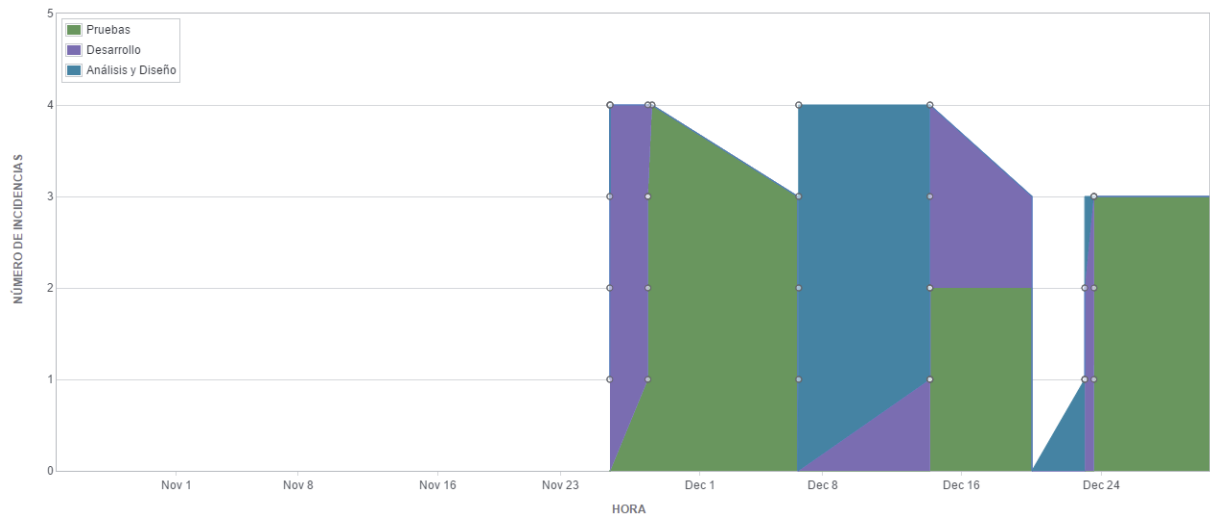


Figura D.26 Gráfico Velocidad de Estabilización Grupo 4.

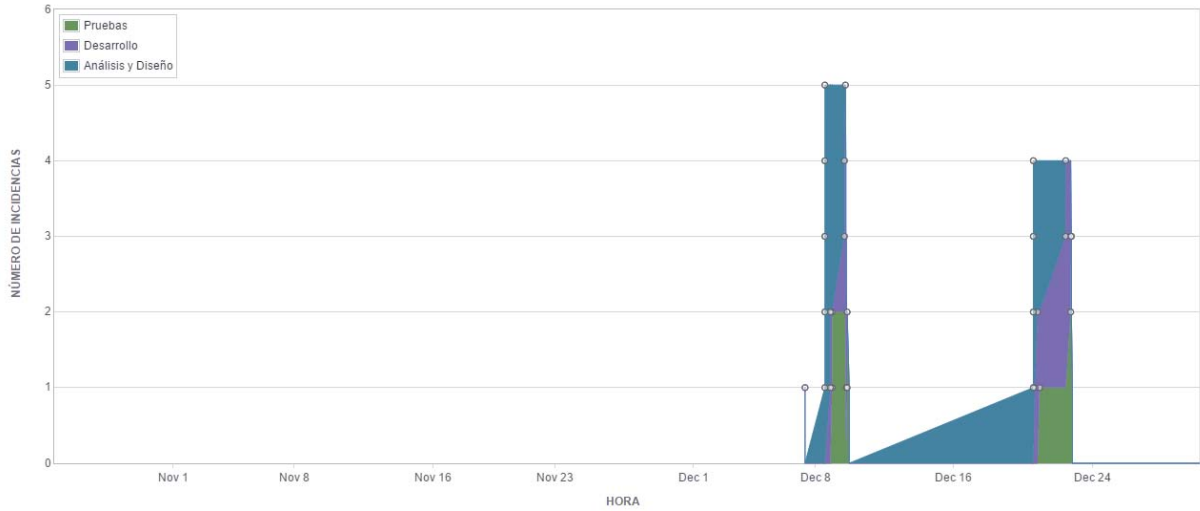


Figura D.27 Gráfico Velocidad de Estabilización Grupo 5.

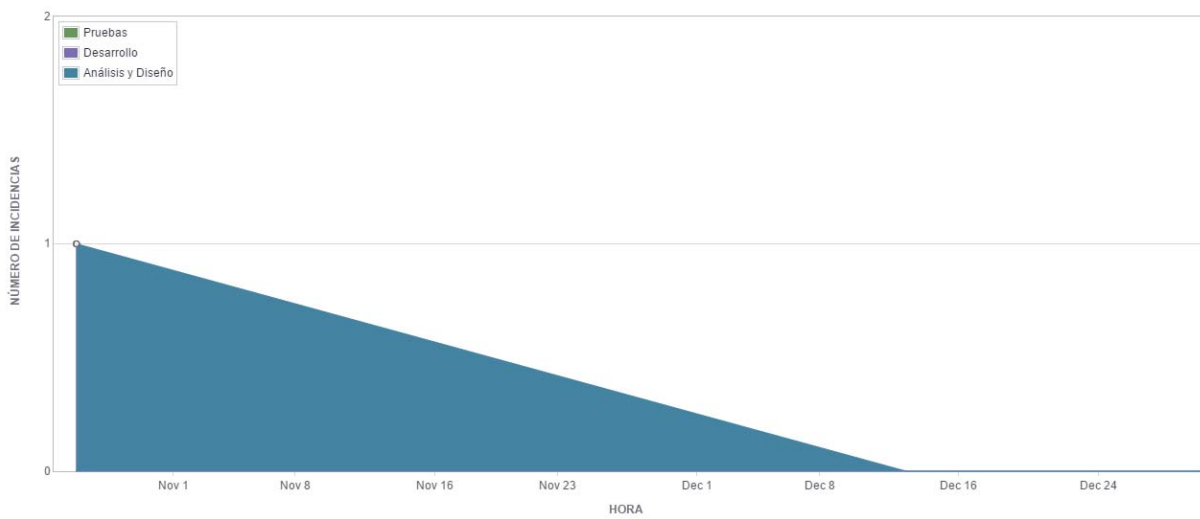


Figura D.28 Gráfico Velocidad de Estabilización Grupo 6.

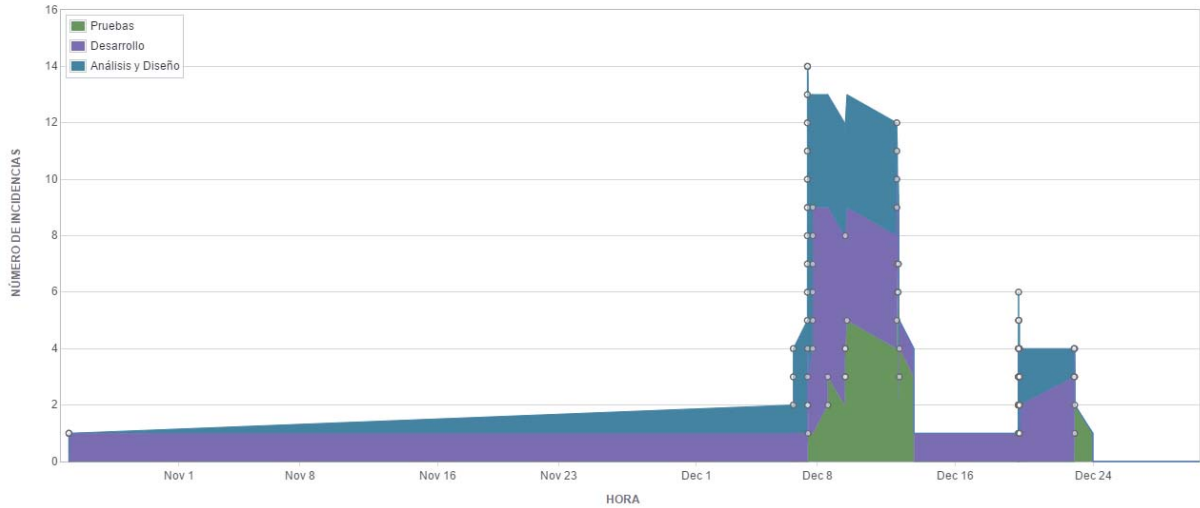


Figura D.29 Gráfico Velocidad de Estabilización Grupo 7.

Tabla D.11 Resumen Gráficos Velocidad de Estabilización.

Grupo	Sobreesfuerzo?	Lane	Tiempo de estabilización (Días)
1	Sí	Desarrollo	Menor a un día
2	Sí	Pruebas	Menor a un día
3	No	-	-
4	Sí	Desarrollo	3
		Pruebas (2 veces)	9
			7
5	Sí	Análisis y Diseño	1
		Desarrollo (2 veces)	Menor a un día
			Menor a un día
6	No	-	-
7	Sí	Análisis y Diseño (2 veces)	7
			Menor a un día
		Desarrollo	7
Pruebas	5		

D1.6. Datos Eficiencia

Los gráficos presentados a continuación tienen por objetivo otorgar a los usuarios de JIRA una herramienta para determinar si los datos del proceso actual pueden usarse para prever el rendimiento futuro del proyecto. Para lograr esto, los gráficos presentan una serie de elementos que permiten determinar la visibilidad, la eficiencia y la predecibilidad.



Figura D.30 Información Proporcionada por JIRA de Cómo Leer el Gráfico.

En el caso particular de este estudio, el enfoque con el cual se examinarán estos gráficos radica principalmente en el análisis de cómo los grupos han evolucionado durante el desarrollo de sus proyectos. Observando el promedio móvil y la desviación estándar se podrá determinar si la eficiencia de los equipos de trabajo ha aumentado o disminuido con el paso del tiempo y con los valores atípicos se podrá determinar aquellas incidencias que presentan algún comportamiento extraño, lo cual podrá ser posteriormente contrastado con los valores obtenidos en los gráficos anteriores y con esto, corroborar si alguno de esos valores se vio influenciado por estas anomalías.

Para que el lector sea capaz de entender adecuadamente la información proporcionada por los gráficos, deberá tener en cuenta lo siguiente:

- Los valores atípicos son contabilizados desde el momento en que las incidencias son movidas desde el lane “Por hacer” hacia cualquiera de los otros lanes de la pizarra.
- Para la creación de estos gráficos sólo fueron considerados los días hábiles, por lo tanto, fines de semana y feriados quedaron excluidos.
- **Promedio móvil:** para cada incidencia que se muestra en el gráfico, el promedio móvil (en ese momento) se calcula promediando los tiempos de ciclo de la incidencia en sí, las X incidencias previas y las X incidencias posteriores, siendo X el número de incidencias correspondiente a cada período (o ventana).
- La desviación estándar se calcula en cada punto utilizando el promedio móvil de dicho punto.
- Los números con los que se identifican las incidencias en estos gráficos son asignados automáticamente por JIRA, por lo que no guardan relación con la numeración utilizada en las tablas del cálculo de Lead Time / Touch Time.

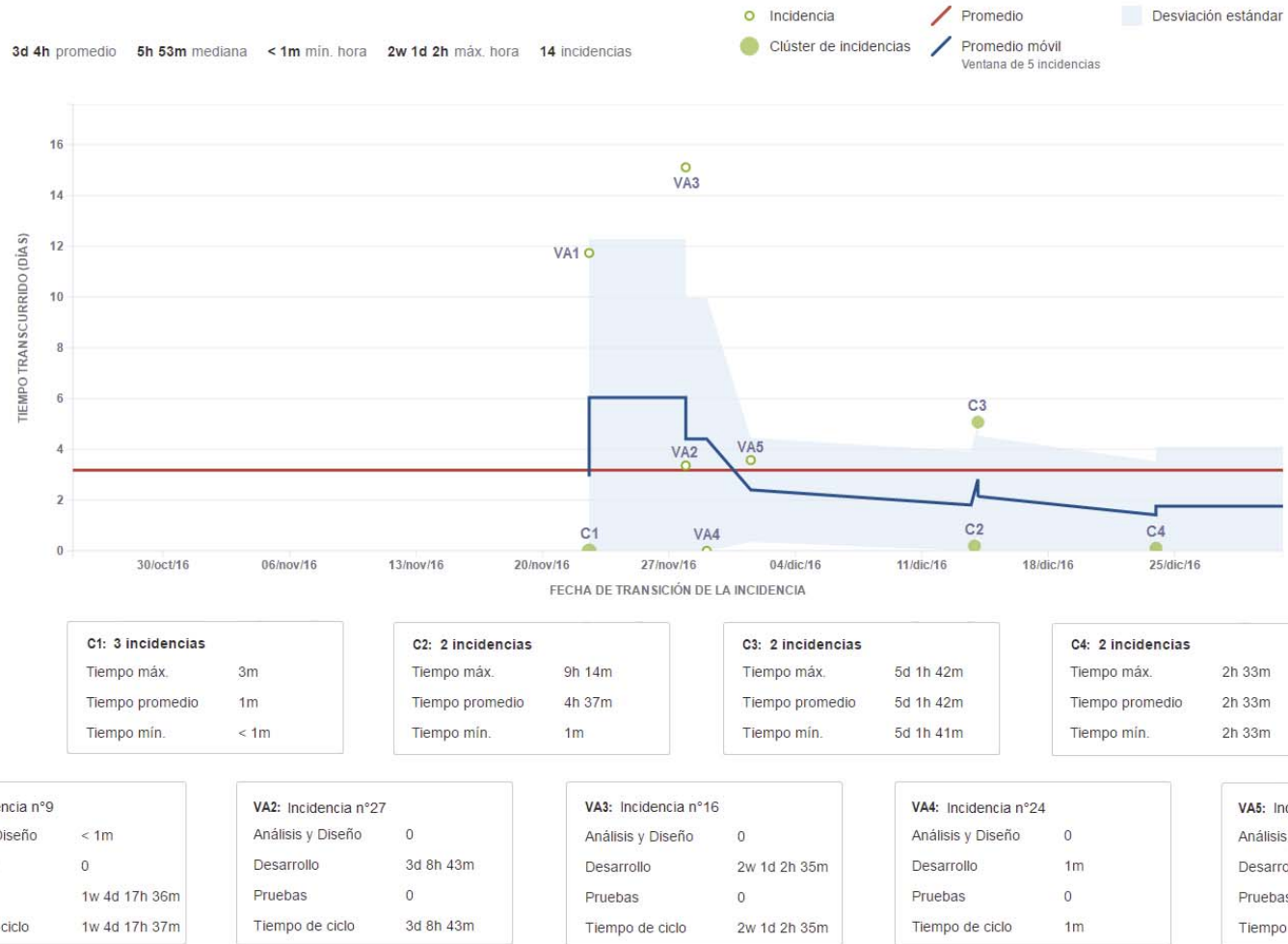


Figura D.31 Gráfico Eficiencia Grupo 1.



C1: 4 incidencias	
Tiempo máx.	< 1m
Tiempo promedio	< 1m
Tiempo mín.	0

C2: 2 incidencias	
Tiempo máx.	1d 7h 10m
Tiempo promedio	1d 7h 10m
Tiempo mín.	1d 7h 10m

C3: 2 incidencias	
Tiempo máx.	2d 20m
Tiempo promedio	2d 20m
Tiempo mín.	2d 20m

VA1: Incidencia n°3	
Análisis y Diseño	1w 1d 8h 57m
Desarrollo	0
Pruebas	0
Tiempo de ciclo	1w 1d 8h 57m

VA2: Incidencia n°11	
Análisis y Diseño	0
Desarrollo	0
Pruebas	< 1m
Tiempo de ciclo	< 1m

Figura D.32 Gráfico Eficiencia Grupo 2.

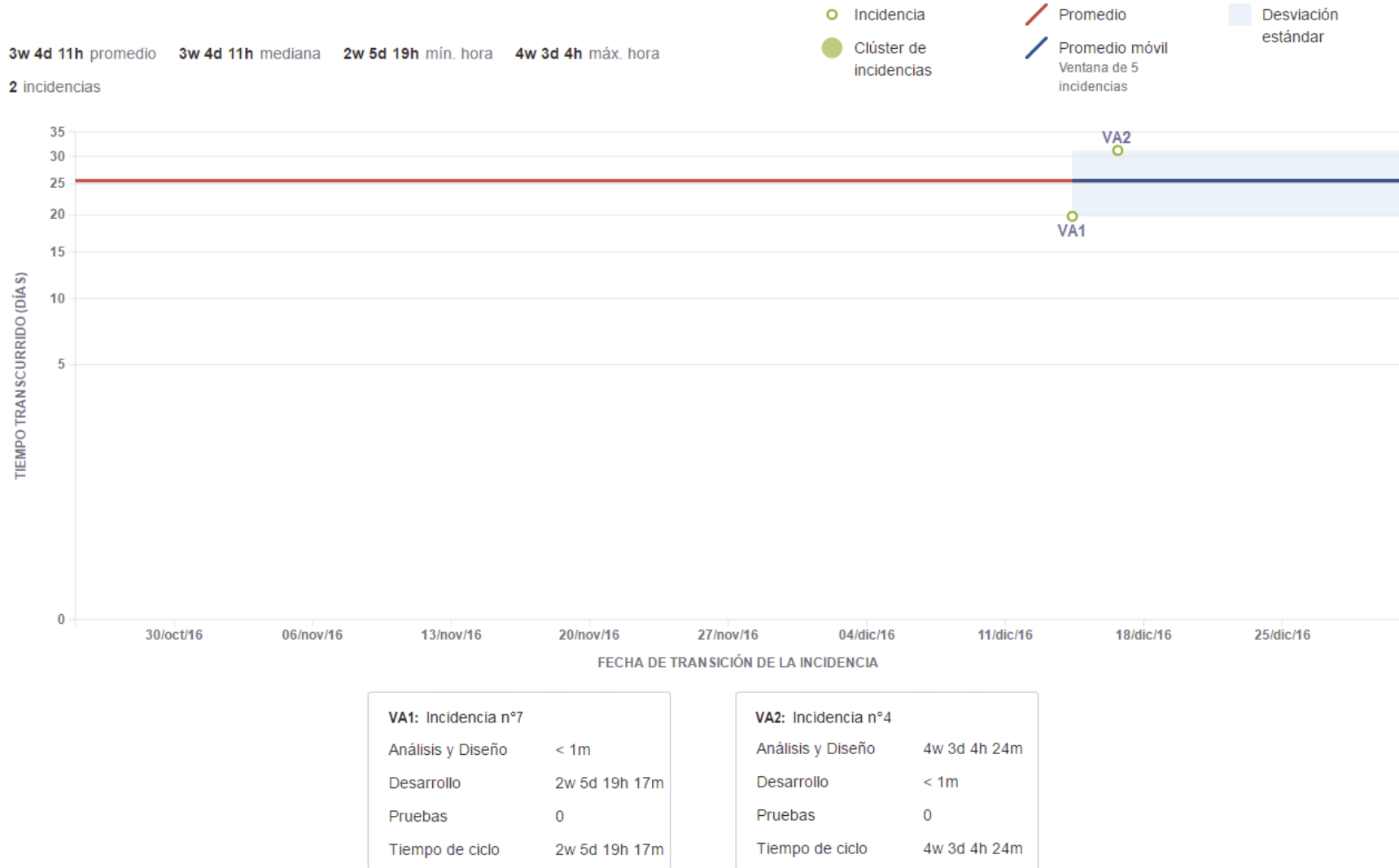


Figura D.33 Gráfico Eficiencia Grupo 3.

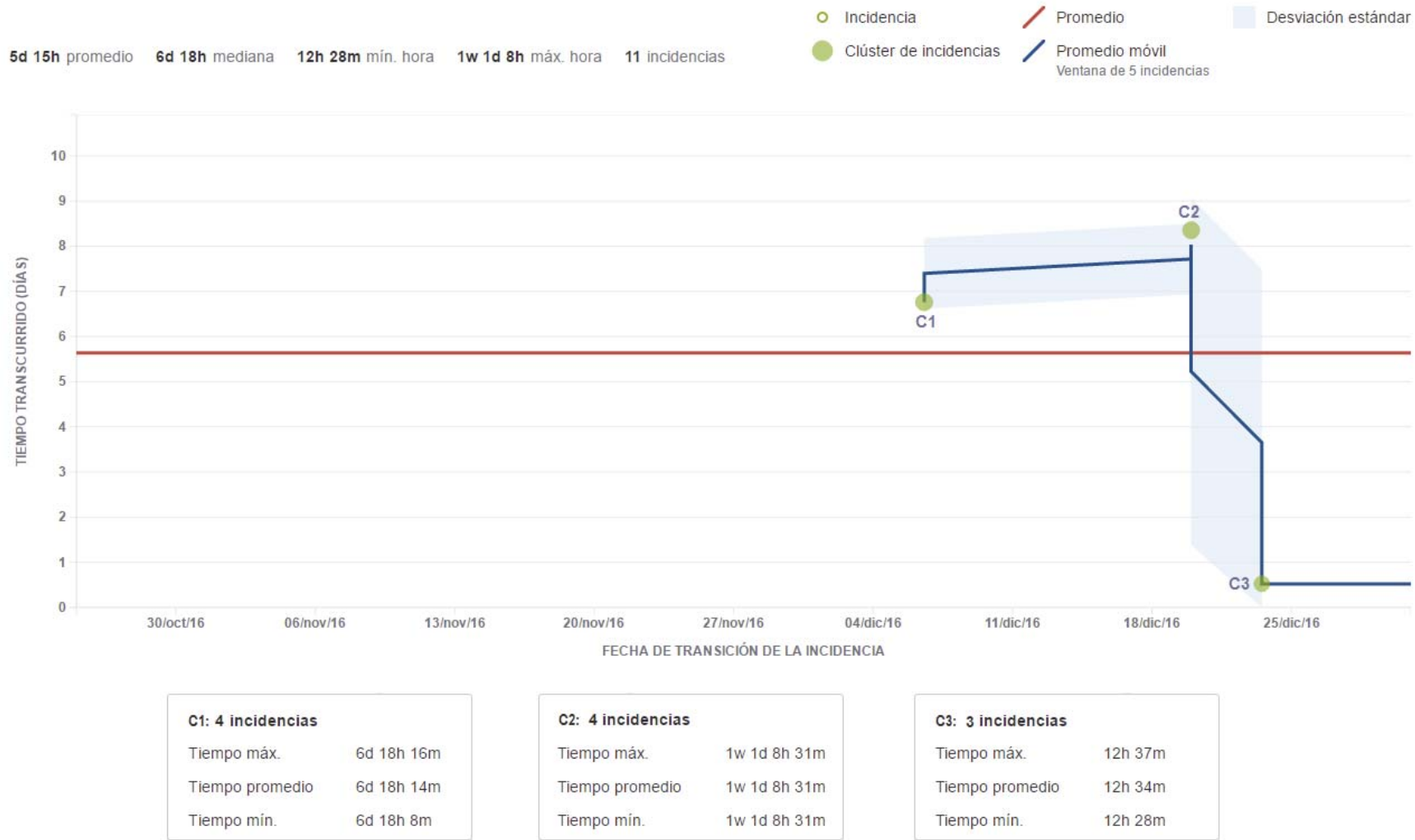


Figura D.34 Gráfico Eficiencia Grupo 4.

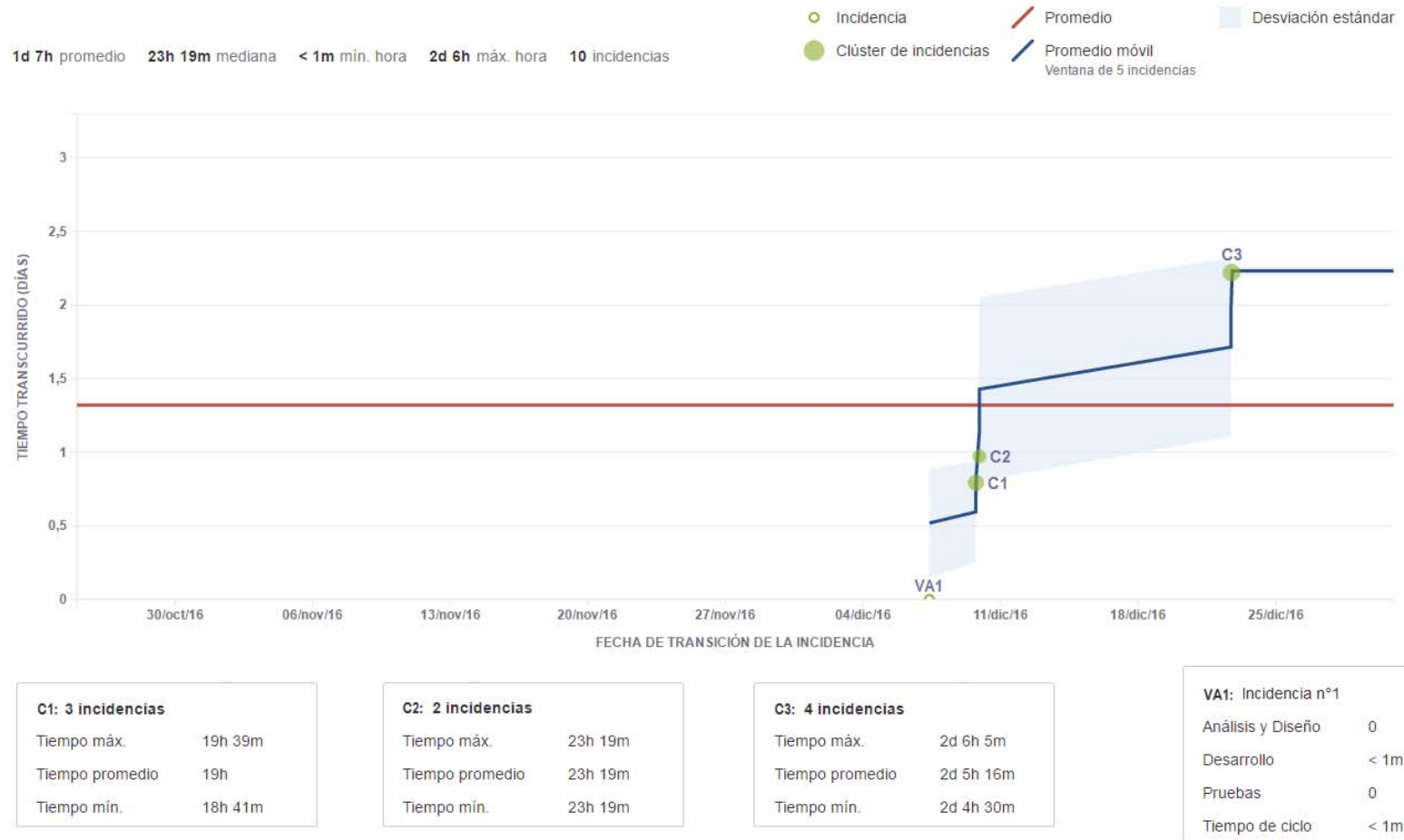


Figura D.35 Gráfico Eficiencia Grupo 5.

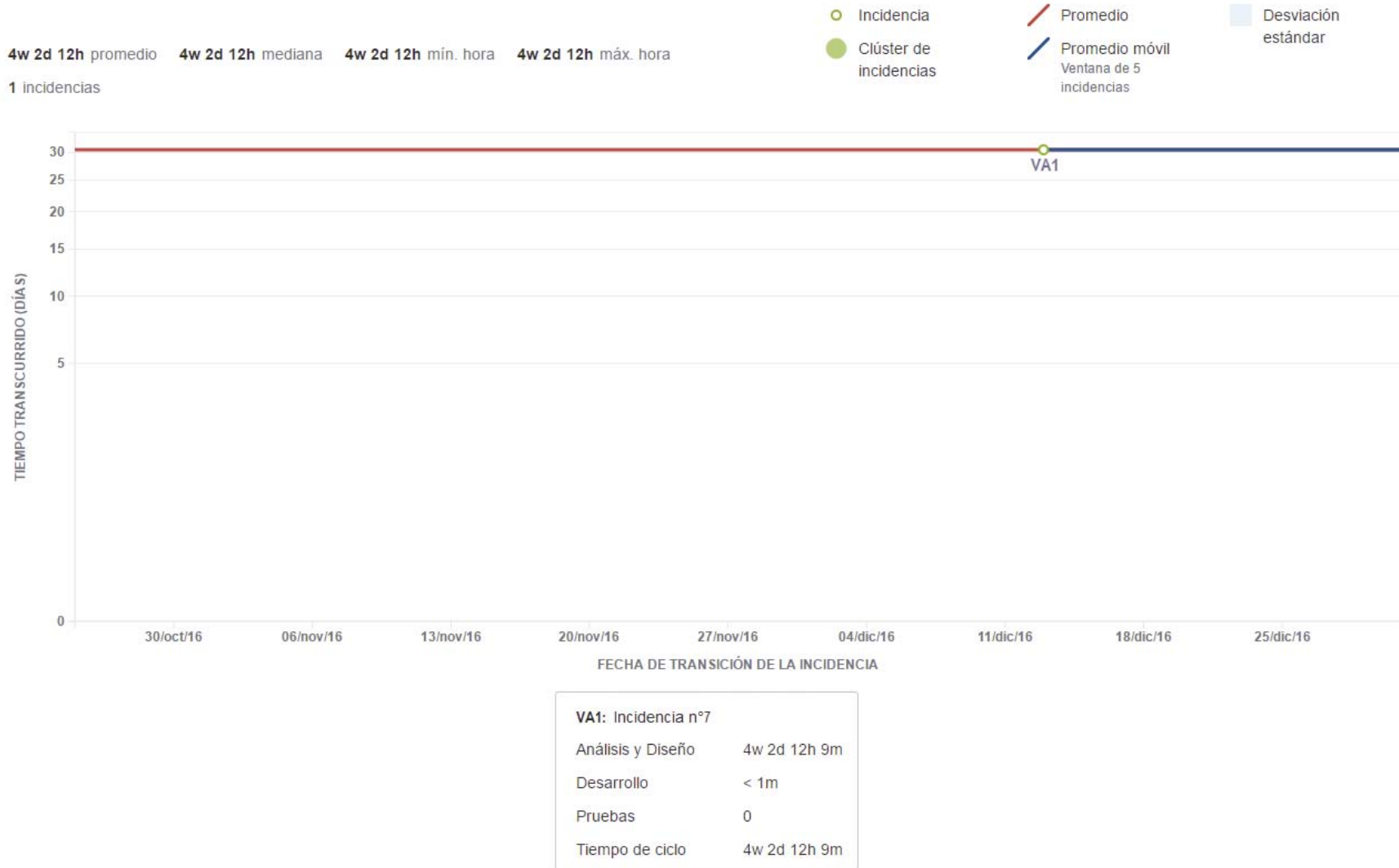


Figura D.36 Gráfico Eficiencia Grupo 6.



C1: 6 incidencias	
Tiempo máx.	2d 7h 15m
Tiempo promedio	2d 6h 23m
Tiempo mín.	2d 4h 16m

C2: 2 incidencias	
Tiempo máx.	23h 52m
Tiempo promedio	23h 3m
Tiempo mín.	22h 15m

C3: 4 incidencias	
Tiempo máx.	3d 4h 10m
Tiempo promedio	3d 3h 49m
Tiempo mín.	3d 2h 46m

VA1: Incidencia n°37	
Análisis y Diseño	13m
Desarrollo	42m
Pruebas	21m
Tiempo de ciclo	1h 17m

C4: 8 incidencias	
Tiempo máx.	1h 25m
Tiempo promedio	43m
Tiempo mín.	4m

C5: 2 incidencias	
Tiempo máx.	3d 5h 57m
Tiempo promedio	3d 5h 25m
Tiempo mín.	3d 4h 52m

C6: 2 incidencias	
Tiempo máx.	4d 6h 6m
Tiempo promedio	4d 5h 59m
Tiempo mín.	4d 5h 52m

VA2: Incidencia n°36	
Análisis y Diseño	16m
Desarrollo	39m
Pruebas	1d 2h 51m
Tiempo de ciclo	1d 3h 47m

Figura D.37 Gráfico Eficiencia Grupo 7.

Tabla D.12 Análisis General Gráficos Eficiencia.

Grupo	Análisis promedio móvil	Análisis desviación estándar	Análisis valores atípicos
1	<p>Entre 22/nov – 28/nov: ascendente.</p> <p>Entre 28/nov – final: descendente.</p> <p>Inclinación: 1° tramo: pronunciada. 2° tramo: leve.</p> <p>Conclusión: se observa que el grupo tuvo algunas dificultades durante su primera semana, pero supieron sobreponerse y mantuvieron un trabajo eficiente durante el resto del proyecto.</p>	<p>Entre 22/nov – 1/dic: gruesa.</p> <p>Entre 1/dic – final: media.</p> <p>Conclusión: similar al promedio móvil, salvo por la primera semana de trabajo, la desviación estándar presenta un grosor medio en la mayor parte del tiempo de desarrollo, lo que indica un alto índice de predecibilidad de los datos. Dicho de otro modo, hay una alta probabilidad de que, si se decidiera continuar el proyecto en el futuro, este mantendría el mismo comportamiento mostrado durante el último período de desarrollo.</p>	<p>Se han encontrado un total de 14 incidencias con valores atípicos, las cuales se relacionan con las siguientes causas:</p> <ul style="list-style-type: none"> • Gran discrepancia en los tiempos de permanencia entre un lane y otro, siendo el caso más grave VA3, que permaneció 2 semanas en Desarrollo y 0 días en el reto de los lanes. • Las incidencias fueron resueltas en un período muy corto, por ejemplo, VA4, que se resolvió en tan sólo 1 minuto. • Para el caso de los clústeres, estos son creados cuando varias incidencias son resueltas durante el mismo día o con muy pocas horas de diferencia. En este caso, se observan un total de 4 clústeres. <p>Todas las causas expuestas anteriormente se pueden asociar a una incorrecta utilización de la aplicación JIRA, aunque pueden existir más causas involucradas. En el caso particular de los clústeres, el hecho de que dos o más incidencias se resuelvan el mismo día, perfectamente puede deberse a una coincidencia, al fin y al cabo, no es raro que dos tareas se realicen de forma simultánea, pero cuando la cantidad de clústeres es muy alta, la probabilidad de que sea una coincidencia se reduce bastante.</p>

2	<p>Entre 27/nov – 12/dic: ascendente.</p> <p>Entre 12/dic – 13/dic: descendente.</p> <p>Entre 13/dic – final: ascendente.</p> <p>Inclinación: 1° tramo: leve. 2° tramo: pronunciada. 3° tramo: leve.</p> <p>Conclusión: se mantuvo un promedio móvil con tendencia ascendente, lo que señala un bajo nivel de eficiencia por parte del equipo, sin embargo, dada la leve inclinación de la recta, se deduce que su ineficiencia también fue leve.</p>	<p>Entre 27/nov – 13/dic: gruesa.</p> <p>Entre 13/dic – final: estrecha.</p> <p>Conclusión: contrario a lo observado en el promedio móvil, la desviación estándar comienza siendo bastante gruesa pero mejora notablemente con el paso del tiempo, lo que indica alta predecibilidad.</p>	<p>Se ha determinado que, en todos los gráficos de eficiencia, las incidencias con valores atípicos comparten las mismas causas explicadas en el análisis del grupo 1, por lo tanto, a partir de ahora sólo se mencionará la cantidad total de incidencias detectadas y la cantidad de clústeres generados, ya que repetir las causas resultaría redundante. En este caso, se detectaron un total de 10 incidencias con valores atípicos, de las cuales hay 8 que forman parte de algún clúster. Se formaron 3 clústeres en total.</p>
3	<p>Debido al bajo número de incidencias (con valor atípico) detectadas, el promedio móvil en este caso es una simple recta horizontal que coincide con el promedio general, por lo tanto, no es posible efectuar una conclusión al respecto.</p>	<p>Análogo al promedio móvil, el número de incidencias impide el análisis de la desviación estándar.</p>	<p>Sólo se detectaron 2 incidencias con valores atípicos.</p>
4	<p>Entre 6/dic – 20/dic: ascendente.</p> <p>Entre 20/dic – final: descendente.</p> <p>Inclinación: 1° tramo: leve. 2° tramo: pronunciada.</p> <p>Conclusión: durante la primera mitad del proyecto, se aprecia una leve ineficiencia en el grupo, la cual es compensada por una notoria mejoría en la segunda mitad.</p>	<p>Entre 6/dic – final: estrecha.</p> <p>Conclusión: la desviación estándar se mantiene estrecha durante todo el tiempo de desarrollo, señalándolo una alta predecibilidad.</p>	<p>Se detectaron un total de 11 incidencias con valores atípicos, todas ellas forman parte de algún clúster. Se formaron 3 clústeres en total.</p>

5	<p>Entre 7/dic – final: ascendente. Inclinación: pronunciada – leve. Conclusión: el promedio móvil se mantiene con una tendencia ascendente durante todo el período de trabajo y, exceptuando el lapsus comprendido entre el 9/dic – 22/dic, la inclinación en general es bastante pronunciada, por lo que se puede concluir que el trabajo de este equipo fue bastante ineficiente.</p>	<p>Entre 7/dic – 9/dic: estrecha. Entre 9/dic – final: gruesa. Conclusión: la desviación estándar muestra un grosor estrecho los primeros días, pero esto cambia rápidamente y se torna gruesa durante el resto del tiempo, demostrando baja predecibilidad.</p>	<p>Se detectaron un total de 10 incidencias con valores atípicos, salvo por una, todas forman parte de algún clúster. Se formaron 3 clústeres en total.</p>
6	<p>Análogo al grupo 3, incidencias insuficientes como para ser evaluadas.</p>	<p>No se puede evaluar.</p>	<p>Sólo se detectó 1 incidencia con valor atípico.</p>
7	<p>Entre 7/dic – 13/dic: ascendente. Entre 13/dic – 19/dic: descendente. Entre 19/dic – final: ascendente. Inclinación: 1° tramo: leve – pronunciada. 2° tramo: pronunciada. 3° tramo: pronunciada. Conclusión: en general se aprecia un promedio móvil ascendente, sólo hubo una semana durante todo el desarrollo que mostró un cambio en esta tendencia. Por otro lado, la recta presenta una inclinación pronunciada a lo largo de todo el proyecto, por lo que se puede concluir que, a nivel general, el equipo de trabajo fue bastante ineficiente.</p>	<p>Entre 7/dic – 12/dic: media. Entre 12/dic – 13/dic: estrecha. Entre 13/dic – 19/dic: gruesa. Entre 19/dic – final: estrecha. Conclusión: la desviación estándar en general se mantuvo estrecha, sólo hubo una semana en la cual se engrosó, pero luego se volvió a estrechar, por lo que se asume un alto índice de predecibilidad.</p>	<p>Se encontraron un total de 26 incidencias con valores atípicos, de las cuales sólo hay dos que no pertenecen a algún clúster. Se formaron 6 clústeres en total.</p>

D2. Métricas Cualitativas

Para la obtención de los datos cualitativos se han efectuado un total de 2 encuestas, una al inicio del semestre orientada a la medición de los conocimientos de los estudiantes respecto al concepto de metodologías ágiles, y otra al término del semestre orientada a comprobar las apreciaciones y satisfacción de los estudiantes en relación a las metodologías utilizadas. A continuación, se presentarán las preguntas efectuadas en cada una de las encuestas junto con los respectivos resultados obtenidos de estas. Cabe recalcar que en las preguntas de tipo abierta se han omitido las respuestas particulares de cada encuestado y sólo se expondrá un pequeño resumen con los datos más relevantes extraídos de ellas.

D2.1. Encuesta 1: Conocimientos previos

Describe brevemente qué entiende Ud. por metodología ágil. (16 respuestas)

Tu respuesta

Figura D.38 Pregunta 1: Abierta.

De los 16 encuestados, sólo 3 mostraron un cierto nivel de conocimiento sobre el concepto de metodología ágil, y de estos, sólo uno demostró poseer un conocimiento ligeramente más avanzado que los demás, el resto sólo dominaba el concepto de forma muy general. Con esto, se puede concluir que los alumnos sometidos a este estudio no contarán con los conocimientos suficientes a la hora de desarrollar sus respectivos proyectos, lo que posiblemente repercutirá en su eficiencia al momento de trabajar, ya que deberán ir adquiriendo dicho conocimiento durante el transcurso del semestre.

¿Ha tenido alguna experiencia trabajando con metodologías ágiles?
(16 respuestas)

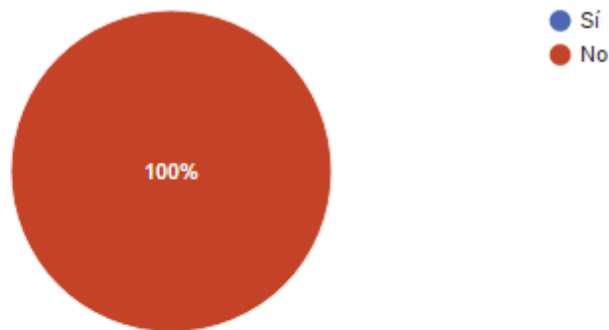


Figura D.39 Pregunta 2: Elección Única (Dicotómica).

En esta pregunta se ve claramente que ninguno de los estudiantes posee algún tipo de experiencia relacionada con el método de trabajo que deberán utilizar, este dato se deberá tomar en consideración a la hora de evaluar su desempeño en los proyectos.

De las siguientes metodologías ágiles, seleccione todas aquellas que Ud. conozca o con las cuales Ud. haya trabajado.
(16 respuestas)

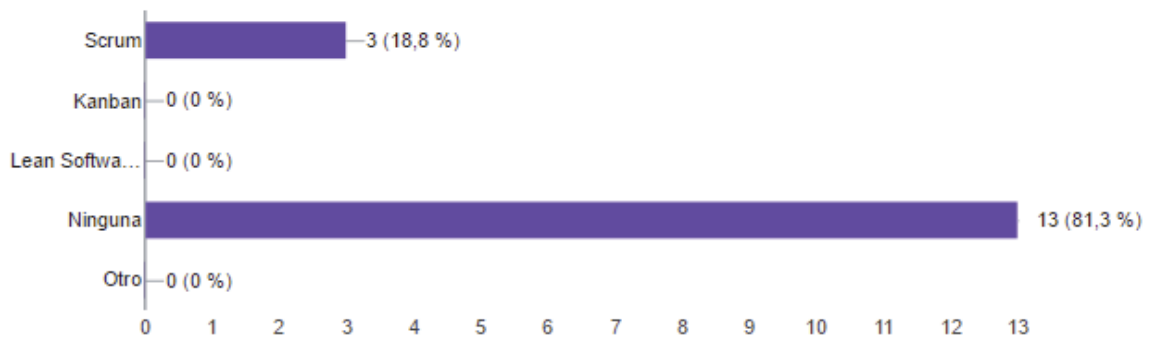


Figura D.40 Pregunta 3: Elección Múltiple.

Si bien la encuesta realizada fue de carácter privado, es decir, no es posible saber quién respondió qué, en esta pregunta se aprecia una clara coincidencia con los datos recabados en la pregunta 1, en la cual sólo tres estudiantes mostraron un cierto nivel de conocimiento sobre las metodologías ágiles, por lo cual se puede concluir que dicho conocimiento viene dado por sus nociones de Scrum.

Respecto a la metodología Scrum: Seleccione su nivel de conocimiento. (16 respuestas)

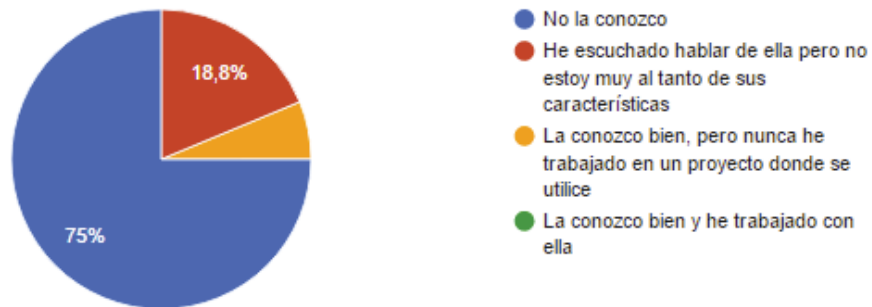


Figura D.41 Pregunta 4: Elección Única (Politómica).

En este caso, 3 de los 16 encuestados han indicado tener una cierta idea sobre lo que es Scrum y sólo uno ha indicado conocerlo bien. Es muy posible que este último coincida con aquel encuestado que dio la respuesta más certera y completa en la pregunta 1.

Respecto a la metodología Kanban: Seleccione su nivel de conocimiento. (16 respuestas)



Figura D.42 Pregunta 5: Elección Única (Politómica).

Como era de esperarse considerando los datos anteriores, la mayor parte de los estudiantes no conocen esta metodología.

Respecto a la metodología Lean Software Development: Seleccione su nivel de conocimiento.

(16 respuestas)



Figura D.43 Pregunta 6: Elección Única (Politómica).

Mismo caso que la pregunta 5.

Creo que el uso de metodologías ágiles será provechoso para mi proyecto de Ingeniería Web

(16 respuestas)

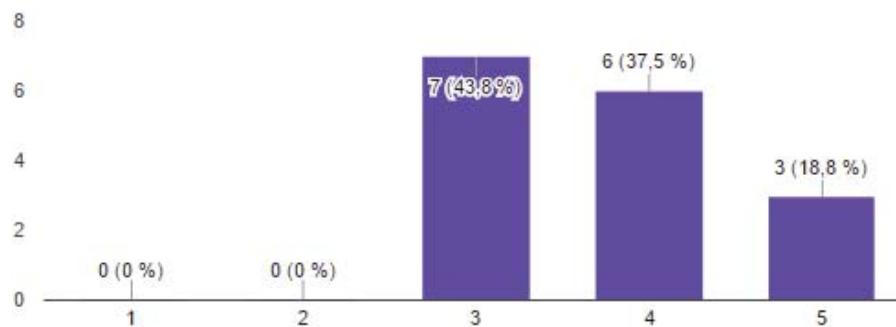


Figura D.44 Pregunta 7: Escala Numérica (donde 1 es muy en desacuerdo y 5 es totalmente de acuerdo).

En este caso, se les realizó una pregunta del tipo escala numérica para corroborar qué tanto creen los alumnos que, con los conocimientos que estos poseían en el instante en que se les aplicó esta encuesta, la implementación de una metodología ágil les sería beneficioso en la realización de sus proyectos. Esta información sería luego contrastada con los resultados obtenidos en la encuesta 2.

D2.2. Encuesta 2: Apreciaciones generales de la metodología utilizada

¿Qué tan difícil le resultó comprender y aplicar la metodología? (11 respuestas)

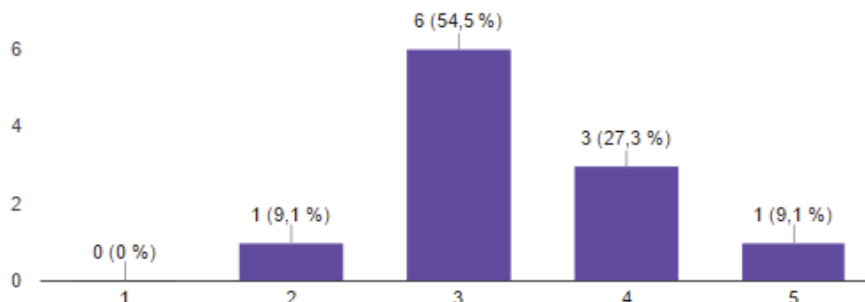


Figura D.45 Pregunta 1: Escala Numérica (donde 1 es muy difícil y 5 es muy fácil).

La presente pregunta está formulada con el fin de entender las dificultades a nivel general que pudiesen haber tenido los estudiantes a la hora de aplicar la metodología, se ve que, si bien no les resultó del todo fácil, tampoco les supuso una complicación excesiva.

¿Cree usted que la metodología le ofrece las herramientas necesarias para una fácil adopción?

(11 respuestas)

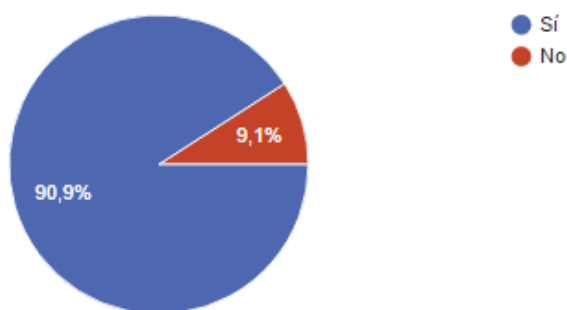


Figura D.46 Pregunta 2: Elección Única (Dicotómica).

Si bien a los estudiantes se les aplicó una capacitación previa a la elaboración de sus proyectos, explicándoles los conceptos más relevantes de la metodología a aplicar y ofreciéndoles documentación apropiada, la propia metodología ofrece también una serie de pasos para ser aplicada, pasos que ya fueron explicados a lo largo del presente documento. El objetivo de esta pregunta era justamente comprobar eso, las herramientas propias de la metodología, y los resultados reflejan que a la mayoría de los estudiantes dichas herramientas les fueron de utilidad.

¿Siente que la metodología utilizada ha propiciado el trabajo en equipo?
(11 respuestas)

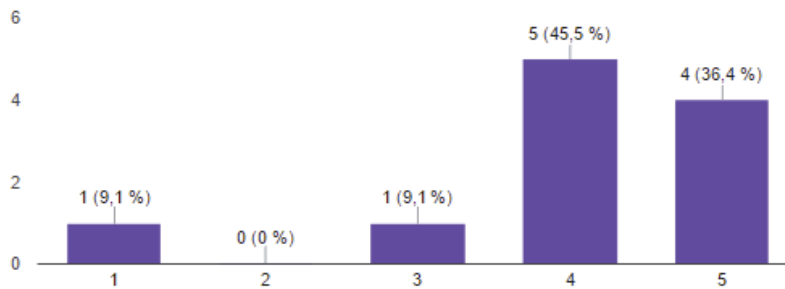


Figura D.47 Pregunta 3: Escala Numérica (donde 1 es totalmente en desacuerdo y 5 totalmente de acuerdo).

En vista de que el trabajo en equipo es un factor sumamente importante en cualquier metodología ágil, conocer las apreciaciones de los estudiantes respecto a este concepto permitirá tener una idea más concreta respecto al funcionamiento de la metodología aplicada. Los resultados demuestran que, salvo casos concretos en los cuales los estudiantes tuvieron problemas internos de grupo, a la mayoría les resultó bastante cómodo trabajar en equipo utilizando esta metodología de trabajo.

¿Considera que ha formado parte de un equipo sólido y bien cohesionado?
(11 respuestas)

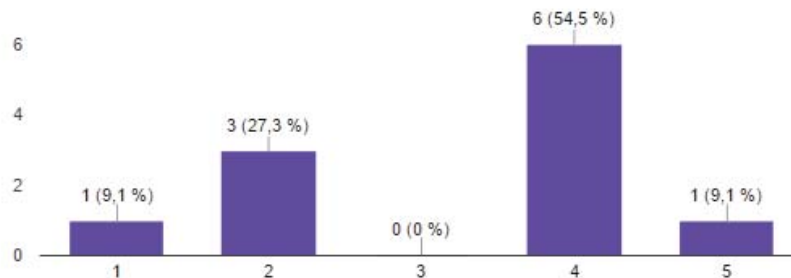


Figura D.48 Pregunta 4: Escala Numérica (donde 1 es totalmente en desacuerdo y 5 totalmente de acuerdo).

Esta pregunta viene a reforzar los resultados obtenidos en la pregunta anterior y también sirve para distinguir de alguna manera aquellos casos en que efectivamente hubo problemas internos de grupo. En este caso, se aprecia que cuatro estudiantes dicen haber pertenecido a grupos con poca cohesión, mientras que en la pregunta anterior sólo dos estudiantes respondieron con valores iguales o inferiores a 3. Esta discrepancia numérica podría indicar que, si bien hubo un porcentaje importante de grupos cuyos miembros no se sintieron del todo unidos a sus compañeros a la hora de trabajar, en algunos casos la metodología sí les resultó útil y les permitió mejorar su organización y trabajo en equipo.

¿Qué tan útiles han resultado para usted las reuniones con su equipo de trabajo?

(11 respuestas)

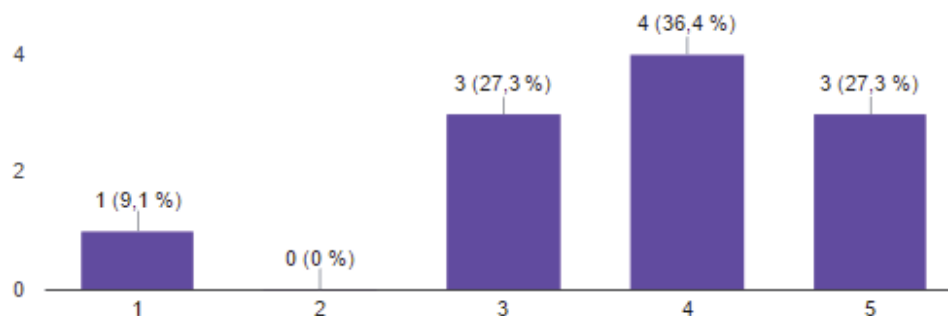


Figura D.49 Pregunta 5: Escala Numérica (donde 1 es ninguna utilidad y 5 muy útil).

Esta pregunta nace de la idea de comprobar la utilidad de las Daily Scrum para los estudiantes, ya que, tal y como se ha explicado en este documento, una de las debilidades que pretende mejorar la propuesta planteada es la rigidez que puede llegar a presentar Scrum a causa de las reglas que conforman la metodología, y un ejemplo claro de esa rigidez son las Daily Scrum, ya que dependiendo del caso y proyecto en el que se esté trabajando, pueden no ser estrictamente necesarias o realizarse con menor frecuencia. En base a los resultados se puede observar que en general a la mayoría les resultó de mucha ayuda, con la única excepción de un estudiante que, posiblemente, coincida con aquel que respondió 1 en las dos preguntas anteriores, lo que refleja un claro descontento por su parte con la relación que mantuvo con su grupo de trabajo, lo cual demuestra claramente qué pasa cuando el grupo de trabajo no funciona correctamente dentro de una metodología ágil.

¿Cuán eficiente considera que fue su equipo a la hora de resolver problemas?
(11 respuestas)

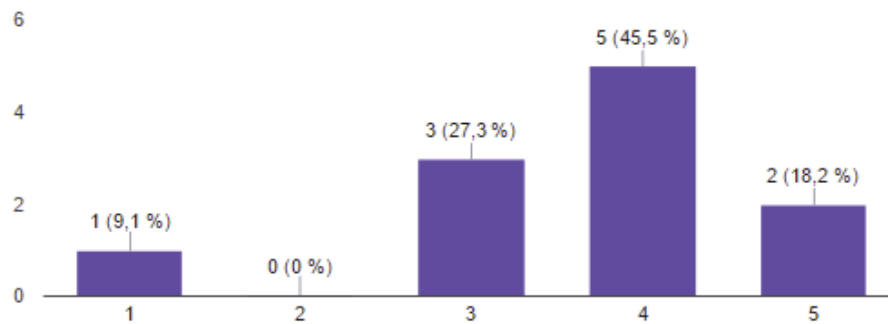


Figura D.50 Pregunta 6: Escala Numérica (donde 1 es muy ineficiente y 5 muy eficiente).

En cualquier proyecto es sumamente importante poder resolver los problemas con rapidez, más aún si se está aplicando una metodología ágil, ya que justamente ese es uno de sus objetivos. En esta pregunta se puede observar que salvo 1 estudiante que, nuevamente, debe ser el mismo que ha respondido con un 1 en las preguntas anteriores, la mayoría de los estudiantes pudieron resolver los problemas que se les fueron presentando con relativa rapidez.

¿Con qué nivel de éxito los integrantes del equipo han compartido la responsabilidad por las tareas?
(11 respuestas)

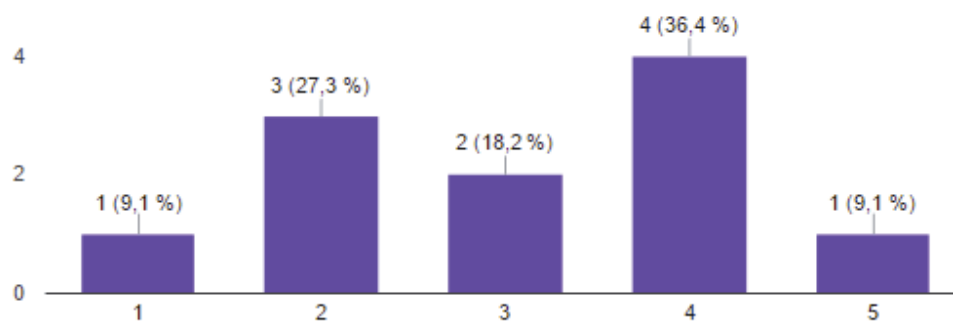


Figura D.51 Pregunta 7: Escala Numérica (donde 1 es muy bajo y 5 muy alto).

Esta es sin duda una de las preguntas con mayor volatilidad en sus respuestas, queda bastante claro que el trabajo en equipo durante el desarrollo de los proyectos no estuvo exento de problemas, lo que sin duda debe haber perjudicado el rendimiento de los distintos grupos.

¿Cuán efectiva considera que fue la labor que realizaron el o los jefes del equipo?

(11 respuestas)

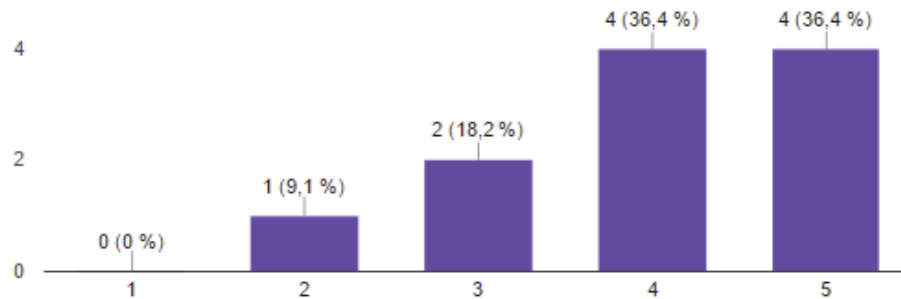


Figura D.52 Pregunta 8: Escala Numérica (donde 1 es muy deficiente y 5 muy efectiva).

Un factor de gran relevancia dentro de una metodología ágil son sus líderes, quienes además de dirigir el proyecto debe velar por la correcta adopción de la metodología. En esta pregunta se observa una apreciación bastante positiva de los encuestados hacia quienes adoptaron el liderazgo de cada grupo, por lo que se puede deducir que los problemas internos presentados por los grupos a lo largo del período de trabajo no fueron causados por irresponsabilidad de sus líderes.

¿Cree usted que la metodología ha favorecido la visibilidad del flujo de trabajo?

(11 respuestas)

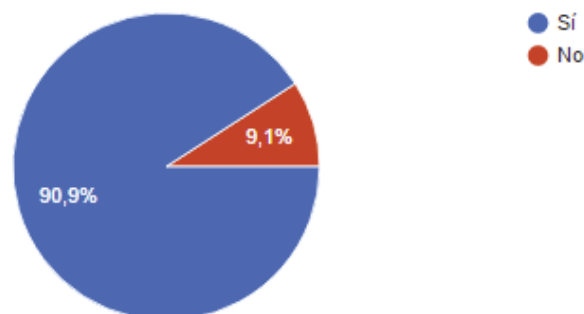


Figura D.53 Pregunta 9: Elección Única (Dicotómica).

Esta pregunta pretende determinar qué tan útil les ha resultado la utilización de Kanban a la hora de organizar y distribuir sus tareas. Se puede ver claramente que a la gran mayoría de los estudiantes les resultó bastante útil.

¿Qué tan fácil le ha resultado organizar el proyecto en iteraciones?

(11 respuestas)

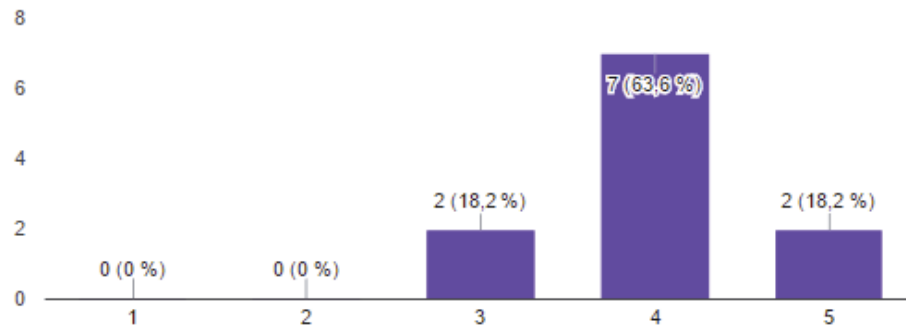


Figura D.54 Pregunta 10: Escala Numérica (donde 1 es muy difícil y 5 muy fácil).

El objetivo de esta pregunta es principalmente corroborar qué tan complejo les ha resultado a los estudiantes realizar la separación del proyecto en iteraciones, es decir, en proyectos más pequeños, especialmente considerando que casi ninguno de ellos poseía conocimientos previos de estas metodologías. Pese a lo anterior, el gráfico revela que a los estudiantes no les ha resultado demasiado complejo llevar a cabo esta tarea.

¿Cuán difícil le resultó planificar la cantidad de trabajo a realizar en cada iteración?

(11 respuestas)

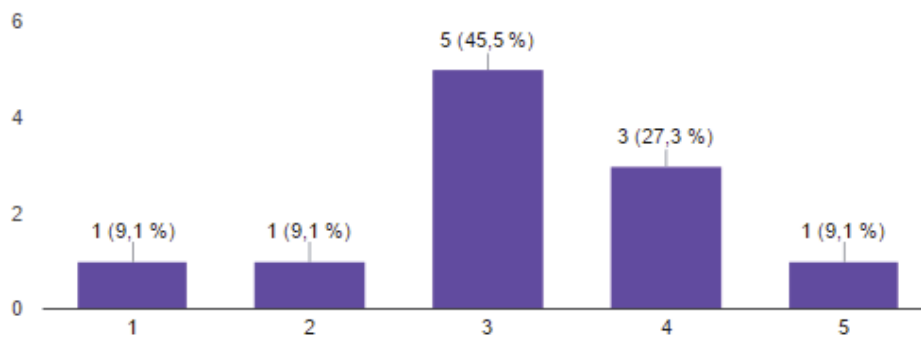


Figura D.55 Pregunta 11: Escala Numérica (donde 1 es muy difícil y 5 muy fácil).

Una vez creadas las diferentes iteraciones es importante determinar las tareas y la carga que cada una de estas tendrá. En este aspecto, se observa una discrepancia mucho mayor en las respuestas en comparación a la creación de la propia iteración, habiendo estudiantes a los que les pareció fácil, otros difícil, pero la mayoría se mantuvo en el punto medio de la escala.

¿Qué tan satisfecho se siente con la metodología utilizada? (11 respuestas)

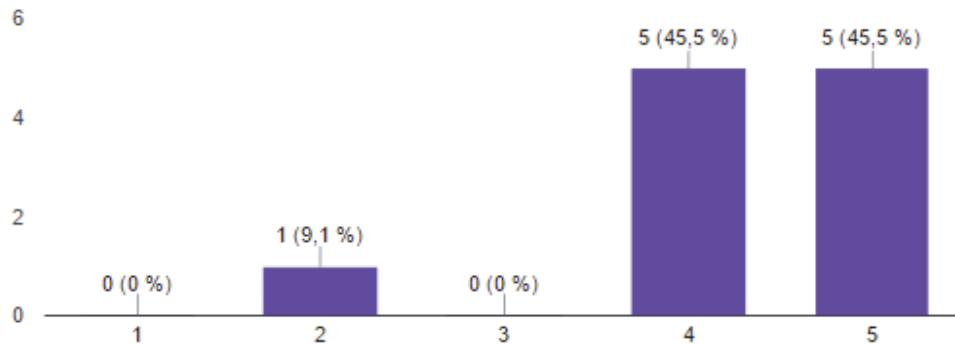


Figura D.56 Pregunta 12: Escala Numérica (donde 1 es muy insatisfecho y 5 muy satisfecho).

Luego de todo un semestre trabajando regidos por una metodología de trabajo desconocida para muchos, es de suma importancia saber qué les ha parecido la experiencia, y en este aspecto, la mayoría ha quedado satisfecho con la metodología utilizada.

Explique brevemente qué fue lo más difícil para usted a la hora de aplicar la metodología.

(11 respuestas)

Tu respuesta

Figura D.57 Pregunta 13: Abierta.

En esta pregunta hubo bastante variedad de respuestas, pero lo más destacable entre todas ellas fue la dificultad en la organización del trabajo en equipo y la consiguiente asignación de las tareas, el aprendizaje de nuevas herramientas desconocidas y la distribución de los tiempos.

¿Volvería a utilizar esta metodología en el futuro? Explique. (11 respuestas)

Tu respuesta

Figura D.58 Pregunta 14: Abierta.

Todos los estudiantes que respondieron esta encuesta coincidieron en que sí, la volverían a usar, pero a la vez varios hicieron hincapié en que, para utilizarla, se necesita un buen equipo de trabajo.

¿Cree usted que la metodología ha incentivado la capacidad de planificación del equipo de trabajo? Explique.

(11 respuestas)

Tu respuesta

Figura D.59 Pregunta 15: Abierta.

Nuevamente todos los encuestados han coincidido en que sí, y su principal justificación radicaba en la visibilidad que les aportaba la metodología respecto a las tareas faltantes y la organización y división de estas, además, destacaban el beneficio que obtuvieron de las herramientas propias de la metodología.

¿Qué aspectos de la metodología considera que deberían mejorarse? Explique.

(11 respuestas)

Tu respuesta

Figura D.60 Pregunta 16: Abierta.

Lamentablemente en esta pregunta muchos estudiantes dieron respuestas poco precisas o directamente respondían una cosa que no tenía relación con lo preguntado sino más bien con el curso de Ingeniería Web. Por lo anterior, las únicas conclusiones que se pudieron extraer de ellas, es que, hablando concretamente de la metodología, los estudiantes no señalaron ningún aspecto a mejorar.