

2019-02

# PROPUESTA BASADA EN MÉTODOS ÁGILES PARA DESARROLLO DE PROYECTOS CAPSTONE DE EMPRENDIMIENTO E INNOVACIÓN

CALVO ERICES, RAFAEL ANGEL

---

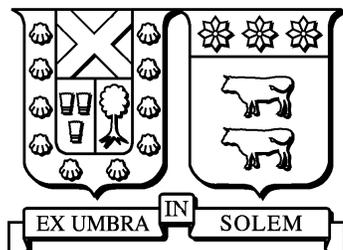
<https://hdl.handle.net/11673/48248>

*Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA*

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

SANTIAGO – CHILE



“PROPUESTA BASADA EN MÉTODOS ÁGILES  
PARA DESARROLLO DE PROYECTOS  
CAPSTONE DE EMPRENDIMIENTO E  
INNOVACIÓN”

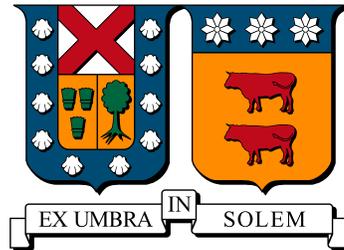
RAFAEL ANGEL CALVO ERICES

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: LIOUBOV DOMBROVSKAIA

FEBRERO 2019

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA**  
**DEPARTAMENTO DE INFORMÁTICA**  
**SANTIAGO – CHILE**



**“PROPUESTA BASADA EN MÉTODOS ÁGILES  
PARA DESARROLLO DE PROYECTOS  
CAPSTONE DE EMPRENDIMIENTO E  
INNOVACIÓN”**

**RAFAEL ANGEL CALVO ERICES**

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL INFORMÁTICO**

**PROFESOR GUÍA: LIOBOV DOMBROVSKAIA**

**PROFESOR CORREFERENTE: MARCELLO VISCONTI**

**FEBRERO 2019**

**MATERIAL DE REFERENCIA, SU USO NO INVOLUCRA RESPONSABILIDAD DEL AUTOR O DE LA INSTITUCIÓN**

# Agradecimientos

A mi familia, por apoyarme y respaldarme a lo largo de toda mi formación y aprendizaje. Tal vez no haya salido como el médico de la casa, pero al menos podré arreglar siempre el WiFi.

A mis amigos en la universidad, quienes directa o indirectamente hayan logrado apoyarme o ayudarme para salir adelante semestre a semestre. En un ambiente de estrés el tener quién te escuche o comparta contigo es algo indispensable, y puedo decir que nunca me sentí solo gracias a ellos.

A los profesores de informática en campus Santiago, quienes, a pesar de la responsabilidad que involucra su rol, siempre tuvieron una actitud cordial y cercana conmigo. En particular a mi jefe de carrera, puesto que más de una vez me sacó de aprietos, y a mis profesores guía y correferente, puesto que gracias a ellos, y a pesar de la adversidad, logré sacar este trabajo adelante.

# Resumen

El presente documento expone los resultados de un diagnóstico a 2 asignaturas consecutivas de ingeniería de software en la Universidad técnica Federico Santa María y otorga una propuesta para mejorar su afinidad ante las metodologías ágiles. El diagnóstico se realizó mediante el análisis a legados de alumnos que han rendido estas asignaturas en los años 2014 a 2018, contrastando estas observaciones con la ejecución de dichas asignaturas durante el año 2018, encontrándose que los conflictos encontrados son consecuencias de un problema raíz: la cultura organizacional. Para contrarrestar esto, se otorga una propuesta que especifica roles, herramientas y pautas para las pruebas que se utilizan en las asignaturas, en pos de eliminar las inconsistencias con los valores y principios del manifiesto ágil.

Debido a que la propuesta se redactó durante la ejecución de estas asignaturas, no fue posible evaluar su implementación, siendo esta validada mediante opinión experta a través de ex-alumnos con experiencia profesional en métodos ágiles y a través de profesores de dicha universidad. Las conclusiones obtenidas luego de dicho proceso fueron que lo fundamental al querer incorporar metodologías ágiles es no enfocarse en las prácticas y ceremonias, sino incorporar los valores y principios de estas a todos los involucrados. No basta con pedirle a los equipos que sean ágiles, ya que si no hay colaboración o consistencia con el resto de la organización, en vez de un aporte esto resulta en un impedimento.

# Abstract

This document presents the results of a diagnosis to 2 consecutive software engineering courses at Federico Santa María Technical University and offers a proposal to improve its affinity with agile methodologies. The diagnosis was made by analyzing legacies of students who have completed these subjects in the years 2014 to 2018, contrasting these observations with the execution of said subjects during the year 2018, finding that the conflicts found are consequences of a root problem: the organizational culture. To counteract this, a proposal is given that specifies roles, tools and guidelines for the tests used in the subjects, in order to eliminate inconsistencies with the values and principles of the agile manifesto.

Due to the fact that the proposal was drafted during the execution of these subjects, it was not possible to evaluate its implementation, being validated by expert opinion through alumni with professional experience in agile methods and through professors of said university. The conclusions obtained after this process were that the fundamental thing when wanting to incorporate agile methodologies is not to focus on the practices and ceremonies, but to incorporate the values and principles of these to all those involved. It is not enough to ask the teams to be agile, because if there is no collaboration or consistency with the rest of the organization, instead of a contribution this results in an impediment.

# Índice de Contenidos

<b>Agradecimientos</b>	<b>III</b>
<b>Resumen</b>	<b>IV</b>
<b>Abstract</b>	<b>V</b>
<b>Índice de Contenidos</b>	<b>VI</b>
<b>Lista de Tablas</b>	<b>X</b>
<b>Lista de Figuras</b>	<b>XI</b>
<b>Glosario</b>	<b>XIII</b>
<b>INTRODUCCIÓN</b>	<b>1</b>
<b>1. DEFINICIÓN DEL PROBLEMA</b>	<b>3</b>
<b>2. MARCO CONCEPTUAL</b>	<b>6</b>
2.1. Restricciones de un proyecto <i>Capstone</i> . . . . .	6
2.1.1. Tiempo y dedicación . . . . .	7
2.1.2. Nivel de experiencia . . . . .	7
2.1.3. Alcance y complejidad . . . . .	7

2.1.4.	Tecnología necesaria . . . . .	8
2.1.5.	Criterios de evaluación . . . . .	8
2.2.	Métodos y Metodologías de Desarrollo . . . . .	9
2.2.1.	Software Development Life Cycle (SDLC) . . . . .	9
2.2.2.	Desarrollo con orientación a objetos . . . . .	10
2.2.3.	Gestión de proyectos tradicional . . . . .	10
2.2.4.	Team Software Process (TSP) . . . . .	12
2.2.5.	Metodologías Ágiles . . . . .	13
2.2.6.	Desarrollo a Medida (Ad-Hoc) . . . . .	23
2.2.7.	Comentarios Adicionales . . . . .	23
<b>3.</b>	<b>DIAGNÓSTICO</b>	<b>25</b>
3.1.	Feria de Desarrollo de Software . . . . .	26
3.2.	Metodología de desarrollo . . . . .	26
3.2.1.	Participantes y Roles . . . . .	27
3.3.	Primer Semestre: Gestión de Proyectos de Informática . . . . .	30
3.3.1.	Entrega 1: Licitación de Ideas de Proyecto . . . . .	31
3.3.2.	Entrega 2: Documento de Identificación de Historias de Usuarios . . . . .	32
3.3.3.	Entrega 3: Defensa del Producto Mínimo Viable . . . . .	33
3.3.4.	Entrega 4: Evaluación Económica del Proyecto . . . . .	34
3.3.5.	Entrega 5: Informe de Plan de Proyecto para Segundo Semestre . . . . .	35
3.3.6.	Actividades Individuales y Grupales en Clases . . . . .	36
3.4.	Segundo Semestre: Taller de Desarrollo de Proyectos Informáticos . . . . .	37
3.4.1.	Informe de Marketing . . . . .	39
3.4.2.	Entrega 1: Primer Sprint . . . . .	39
3.4.3.	Presentación de posters . . . . .	41

3.4.4.	Entrega 2: Segundo Sprint . . . . .	42
3.4.5.	Reporte de marketing . . . . .	42
3.4.6.	Entrega 3: Último Sprint . . . . .	42
3.4.7.	Evento de Feria de Software . . . . .	43
3.4.8.	Entrega del Documento de Legado . . . . .	44
3.5.	Análisis a legados de generaciones anteriores (2014-2018) . . . . .	45
3.5.1.	Resumen de Resultados . . . . .	46
3.6.	Conclusiones del Diagnóstico . . . . .	48
<b>4.</b>	<b>PROPUESTA DE SOLUCIÓN</b>	<b>51</b>
4.1.	Tradicional vs. Agilidad . . . . .	52
4.2.	Estructura de la Propuesta . . . . .	55
4.2.1.	El <i>Framework</i> Scrum . . . . .	55
4.2.2.	Contexto de la asignatura . . . . .	55
4.2.3.	Participantes y Roles . . . . .	56
4.3.	Evaluaciones y Criterios . . . . .	58
4.3.1.	Etapas Principales . . . . .	59
4.4.	Estructura Detallada: Primer Semestre . . . . .	61
4.5.	Estructura Detallada: Segundo Semestre . . . . .	69
<b>5.</b>	<b>VALIDACIÓN DE LA SOLUCIÓN</b>	<b>75</b>
5.1.	Resultados de la evaluación . . . . .	76
5.2.	Recepción de la propuesta . . . . .	78
	<b>Conclusiones</b>	<b>79</b>
5.3.	TRABAJO FUTURO . . . . .	81
5.3.1.	Respecto de la Metodología . . . . .	81

5.3.2.	Respecto del Desarrollo y los Roles de Equipo . . . . .	81
5.3.3.	Respecto de Retrospectivas y Feedback . . . . .	82
5.4.	Uso de Lean Inception . . . . .	83
<b>Anexo</b>		<b>84</b>
5.5.	Detalle de legados de años 2014-2018 . . . . .	84
5.5.1.	Año 2014 . . . . .	84
5.5.2.	Año 2015 . . . . .	88
5.5.3.	Año 2016 . . . . .	91
5.5.4.	Año 2017 . . . . .	94
5.5.5.	Año 2018 . . . . .	98
5.6.	Aclaraciones y Sugerencias . . . . .	107
5.6.1.	Mapa Simplificado del Producto . . . . .	107
5.6.2.	Diagrama de Flujo Acumulado . . . . .	108
5.7.	Comparación de Evaluaciones de Primer Semestre . . . . .	109
5.7.1.	Licitación de Proyectos . . . . .	109
5.7.2.	Sprint de PMV . . . . .	114
5.7.3.	Planificación del proyecto . . . . .	117
5.8.	Comparación de Evaluaciones de Segundo Semestre . . . . .	119
5.8.1.	Marketing . . . . .	119
5.8.2.	Evaluaciones de Final de Sprint . . . . .	124
5.8.3.	Evento Feria de Software . . . . .	129
5.8.4.	Documento Legado . . . . .	132
<b>Bibliografía</b>		<b>134</b>

# Índice de tablas

2.1. Propósito y actividades definidas para Extreme Programming, basado en [1].	16
2.2. Propósito y actividades definidas para Scrum, basado en [1]. . . . .	20
3.1. Ponderación de entregables de gestión de proyectos de informática. . . . .	31
3.2. Ponderación de evaluaciones del segundo semestre. . . . .	38
4.1. Comparación entre las mentalidades fija y de crecimiento. . . . .	53
5.1. Grupo de expertos que evaluaron la propuesta. . . . .	75

# Índice de figuras

3.1. Descripción gráfica de los roles de la asignatura y su interacción. . . . .	29
3.2. Representación visual del primer semestre. . . . .	30
3.3. Representación visual del segundo semestre. . . . .	37
3.4. Porcentajes de equipos, de los 5 años de legados analizados, que realizaron críticas a la asignatura. . . . .	46
3.5. Diagrama resumen de la feria de software 2018, sus evaluaciones y roles. . . . .	50
4.1. Comparación de enfoques entre proceso predictivo y adaptativo. . . . .	54
4.2. Descripción gráfica de los roles de la propuesta y su interacción. . . . .	57
4.3. Representación visual de las 10 etapas de la presenta propuesta. . . . .	60
4.4. Representación visual de la propuesta de primer semestre. . . . .	61
4.5. Product Vision Board a utilizar en la licitación de ideas. . . . .	63
4.6. Descomposición de elementos, desde temas hasta tareas, en donde éstas últimas se definen según las historias a realizar en el actual Sprint. . . . .	65
4.7. Representación visual de la propuesta de segundo semestre. . . . .	69
4.8. Diagrama resumen de la propuesta, con sus evaluaciones y roles. . . . .	74

5.1. Apreciación de la propuesta desde el punto de vista de los expertos. . . . .	76
5.2. Porcentaje de equipos que presentaron críticas, en base a los 16 presentes en el año 2014. . . . .	85
5.3. Porcentaje de equipos que presentaron críticas, en base a los 13 presentes en el año 2015. . . . .	88
5.4. Porcentaje de equipos que presentaron críticas, en base a los 9 presentes en el año 2016. . . . .	91
5.5. Porcentaje de equipos que presentaron críticas, en base a los 7 presentes en el año 2017. . . . .	94
5.6. Porcentaje de equipos que presentaron críticas, en base a los 15 presentes en el año 2018. . . . .	98
5.7. Ejemplo de Mapa Simplificado del Producto a usar en documento de planificación tentativa. . . . .	107
5.8. Ejemplo de Diagrama de Flujo Acumulado. . . . .	108

# Glosario

- UTFSM: Abreviación para “Universidad Técnica Federico Santa María”, universidad en donde se aplica la presente propuesta. También conocida como USM.
- Ayudante: Alumno que ya ha cursado la asignatura y es contratado por los profesores como asistente mediante una beca.
- Framework: Un conjunto estandarizado de conceptos, prácticas y criterios que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar, pudiendo enfocarse en un tipo de problemática particular.
- Stakeholders: Todas las personas u organizaciones que se relacionan con las actividades y decisiones de una empresa, tales como: empleados, proveedores, clientes, gobierno, entre otros.
- Frontend: En diseño de software, es la parte del software que interactúa con los usuarios.
- Backend: En diseño de software, es la parte que procesa los datos que se ingresan desde el front-end.
- QA: Sigla para aseguramiento de calidad (Quality Assurance). Es un rol que se enfoca en el compromiso de calidad, procurando evitar errores, bugs o cualquier condición que haga fallar al software.
- Coaching: Proveniente del inglés (to coach) es un método que consiste en acompañar, instruir o entrenar a una persona o a un grupo de ellas, con el objetivo de conseguir cumplir metas o desarrollar habilidades específicas.
- Demo: Programa informático de demostración que es una versión reducida en prestaciones de un programa para poder utilizarlo y evaluarlo antes de su compra.

# INTRODUCCIÓN

El proceso de desarrollo de software es complejo, abarcando su diseño, especificación, programación, documentación, testeo y mantenimiento, considerándose estos pasos esenciales para crear o mejorar un software o sus componentes. Es un proceso extenso y que, en caso de empresas o grupos de trabajo, requiere de un buen control y seguimiento para lograr asegurar un desarrollo eficaz, en donde además, se espera que un trabajador posea las cualidades aptas para cumplir con dicha tarea.

Desde el contexto de la formación de futuros ingenieros, instancias como los proyectos *Capstone* buscan que los estudiantes adquieran una experiencia previa a ser introducidos a un entorno laboral. Dichos proyectos tienen por finalidad que los estudiantes, trabajando en equipos, integren todos sus conocimientos y habilidades para crear un producto, proceso o servicio que constituya una solución práctica a un problema del mundo real.

El problema es que, como consecuencia, surge una problemática a nivel de instituciones educacionales, y es que para organizar proyectos *Capstone* hay muchas aristas que se deben considerar: factores como la experiencia y el tiempo disponible de los alumnos, el alcance y complejidad de los proyectos, la tecnología necesaria y los criterios de evaluación. También se debe adoptar una metodología de trabajo: ¿enseñar desarrollo tradicional o ágil? ¿enfatar el conocimiento técnico o aplicación práctica?. El simple hecho de planificar los eventos a lo largo de este proyecto resulta un desafío, ya que cruza elementos tanto del área de gestión de proyectos como de desarrollo de software, considerando las restricciones que involucra un entorno académico.

El presente trabajo diagnostica el proceso de gestión y desarrollo de software de las asignaturas de “Gestión de proyectos de informática” y de “Taller de desarrollo de proyectos informáticos” de la Universidad Técnica Federico Santa María (en adelante USM), estudiando sus metodologías de gestión de proyectos y desarrollo de software. Luego, se entrega una propuesta nueva basada en metodologías de desarrollo ágil, con el fin de promover buenas prácticas en el desarrollo de software y una metodología acorde a su contexto educacional, aportando a la formación de futuros ingenieros civiles en informática, y siendo útil como referencia a futuras investigaciones que busquen implementar agilidad en ambientes similares.

A continuación, en el capítulo 1 se definen los conceptos clave de la problemática mencionada a nivel global y en el capítulo 2 se presenta el marco conceptual requerido para el contexto de la propuesta. En el capítulo 3, se presenta el cómo se estructuran las asignaturas y qué falencias se identifican, las cuales se buscan mejorar a través de la propuesta en el capítulo 4. Finalmente, en el capítulo 5 se valida la propuesta mediante un grupo de expertos y ante los profesores de las asignaturas para, finalmente, entregarse conclusiones en el capítulo final.

# Capítulo 1

## DEFINICIÓN DEL PROBLEMA

El proyecto tipo *Capstone* fue introducido en la década de los 90' en Estados Unidos, Canadá y Europa, en respuesta a la insatisfacción declarada de parte de la industria en cuanto a las falencias detectadas en los egresados universitarios [2]. Si bien los estudiantes mostraban dominios técnicos y habilidades de comunicación, sus competencias de trabajo en equipo y liderazgo quedaban muy por debajo en relación a las expectativas de la propia industria. De esta manera, principalmente en las ingenierías, se comienza a expandir esta idea para poner a prueba los conocimientos adquiridos de los alumnos de manera práctica, preparándolos para su futuro labor profesional.

En pocas palabras, el proyecto *Capstone* es una modalidad de enseñanza orientada a estudiantes de cursos superiores, quienes deben resolver problemas complejos reales de la industria de manera grupal, integrando saberes y movilizandoo habilidades de manera colaborativa. Se describe como una experiencia genuina de aprendizaje que amplía las potencialidades profesionales del estudiante al integrar conocimientos y habilidades adquiridas previamente en la carrera.

De manera específica su creador, Gerard Lachiver, lo define como: “(...) *una demostración de los conocimientos y habilidades que el estudiante ha aprendido durante su carrera. Normalmente los estudiantes son convocados para emprender un proyecto de ingeniería substancial, preparando un reporte que describe el trabajo realizado, los resultados, las conclusiones*

*y recomendaciones. Se requiere que los estudiantes ejecuten el proyecto de acuerdo a los estándares de ingeniería demostrando habilidades profesionales como experticia técnica, administración de proyectos, ética y comunicaciones” [2].*

Al igual que en en Canadá y Estados Unidos, la modalidad en la USM se extiende por dos semestres pero, contrario a estos, el proyecto se realiza previo al trabajo de titulación, teniendo cada alumno que dedicarse tanto a esta como a otras asignaturas en paralelo. A diferencia de la planificación por defecto de un proyecto *Capstone*, en la carrera de ingeniería civil informática se da énfasis a realizar un proyecto de innovación, dándole a los alumnos una oportunidad de emprendimiento. Para esto, los alumnos son guiados tanto por los profesores de la asignatura, como por un ayudante, el cual es un alumno que ya ha cursado la asignatura, siendo uno asignado a cada equipo de desarrolladores.

La dificultad actual de los proyectos *Capstone* es que sus resultados dependen de su planificación y desempeño, no existiendo una pauta universal acerca de cómo llevarlo a cabo. La modalidad de este tipo de proyectos posee una definición base, la cual es adaptada por cada institución en base a sus objetivos. En consecuencia, existe la posibilidad de que instituciones incorporen esta estructura sin obtener resultados favorables o que, a pesar de efectuarse el curso, no se logre en este el aprendizaje esperado [3].

El presente trabajo surge desde la base de que en la actualidad la planificación de las asignaturas de “Gestión de proyectos de informática” y “Taller de desarrollo de proyectos informáticos” (la “feria de software”) no se ajusta a los valores y principios de las metodologías ágiles que señala impartir. Las asignaturas presentan una buena estructura como proyecto *Capstone*, pero al momento de estudiarlas en detalle se observan falencias y contradicciones ante la metodología de desarrollo en la que se basan. El presente documento diagnostica dicha estructura, presentando una propuesta que otorgue mayor claridad a sus procesos y que oriente a ambas asignaturas a los pilares de las metodologías ágiles, dada su cada vez mayor presencia en la industria.

Bajo dichos términos es que se define al objetivo principal de esta memoria como:

Entregar una propuesta que mejore el desempeño, control y aprendizaje de los equipos de las asignaturas de “Gestión de proyectos de informática” y “Taller de

desarrollo de proyectos informáticos”.

Y, en paralelo, se definen los siguientes objetivos secundarios:

- Diagnosticar la situación actual de las asignaturas.
- Detectar y analizar los factores que puedan entorpecer el desarrollo y evaluación de cada equipo.
- Identificar oportunidades de mejoras de herramientas utilizadas, tanto en la gestión y evaluación, como en el mismo desarrollo de software y su coordinación.
- Establecer una guía adecuada, que defina roles y responsabilidades para que cualquiera pueda comprender el cómo aplicar la presente propuesta.
- Definir pautas y criterios claros y concisos para cada hito de las asignaturas, eliminando ambigüedades al evaluar.
- Establecer métricas que den paso a posteriores análisis de productividad.

# Capítulo 2

## MARCO CONCEPTUAL

### 2.1. Restricciones de un proyecto *Capstone*

La finalidad de los proyectos *Capstone* es darle una experiencia previa a los alumnos acerca de lo que es el desarrollo y entrega de un producto, como un símil a lo que podría ser su futuro trabajo como ingeniero de software. El entorno educacional supone 5 restricciones a considerar al momento de plantear estos proyectos [3]:

1. El tiempo y dedicación disponible
2. El nivel de experiencia previo
3. El alcance y complejidad de los proyectos
4. La tecnología requerida
5. Los criterios de evaluación involucrados

Estas restricciones son los principales elementos que marcan la diferencia entre un proyecto real en un entorno de trabajo y un proyecto *Capstone* en un ambiente educacional.

### **2.1.1. Tiempo y dedicación**

Típicamente, los proyectos de desarrollo de software en la industria poseen cierta flexibilidad, permitiendo en algunos casos semanas extra para asegurar la completitud del trabajo, mientras que un proyecto *Capstone* debe ser terminado entre uno o dos semestres. Además, se debe tomar en cuenta la disponibilidad de los estudiantes a lo largo del año, puesto que tienen varias asignaturas en paralelo. Esto es comparable a un grupo de profesionales que participan en más de un proyecto a la vez, debido a que a cada asignatura requiere de su propia dedicación. Los alumnos deben administrar sus tiempos y reservar horarios en los que puedan coincidir con el resto de su equipo.

### **2.1.2. Nivel de experiencia**

Debido a que los estudiantes se encuentran en un proceso de aprendizaje, se asume que la experiencia que tienen en proyectos de desarrollo es baja, considerando que un desarrollo abarca múltiples áreas (diseño, backend, frontend, QA, entre otras), variados lenguajes de programación y herramientas, las que pueden variar según cada proyecto. Además, se requiere de experiencia en gestión de proyectos, de software, mantener el trabajo en equipo y el contacto con los *stakeholders* del proyecto. En algunos casos los mismos instructores pueden no tener experiencia suficiente si, por ejemplo, el proyecto requiriese de tecnologías fuera de su dominio.

### **2.1.3. Alcance y complejidad**

Al momento de analizar cada proyecto *Capstone*, este debe ser lo suficientemente complejo para promover la participación de un pequeño equipo de desarrollo (2 a 5 miembros), pero no tan complejo como para sobrepasar la disponibilidad de los estudiantes, pudiendo ser terminado a tiempo.

Determinar la complejidad y el alcance de un proyecto *Capstone* es de gran dificultad, ya que ambas características están relacionadas y, lo ideal, es que luego de determinadas no sean

alteradas [4]. Esto debido a que podrían afectar al proyecto como herramienta de evaluación y, además, porque es importante mantener un balance entre la dificultad y alcance de todos los proyectos.

#### **2.1.4. Tecnología necesaria**

Debido a la variedad de los proyectos algunos de ellos pueden requerir de alguna inversión que los haga inviables para los estudiantes, tomando en cuenta que gran parte de ellos no cuenta con ingresos estables. Es importante establecer un filtro basado en un límite presupuestario, tomando en cuenta tanto hardware como software al momento de evaluar un proyecto. Usualmente se opta por las alternativas tradicionales: lenguajes y conceptos orientados a objetos, desarrollo de aplicaciones o contenido Web y uso de bases de datos relacionales.

#### **2.1.5. Criterios de evaluación**

Las evaluaciones imponen restricciones particulares en los proyectos *Capstone*. No solo se debe asegurar la satisfacción del cliente, también se debe asegurar que las metodologías se hayan puesto en práctica, logrando cierto nivel de aprendizaje. Esto usualmente incorpora fechas límite y tareas extra a lo largo de los proyectos, además de elementos típicos como entregas de documentos, presentaciones, registros de avance e informes de retrospectiva [5, 6].

Junto con las restricciones de los proyectos *Capstone*, hay que tomar en cuenta las alternativas respecto de cómo se realiza y gestiona un desarrollo de software, debido a la gran cantidad de variantes existentes hasta la fecha.

## 2.2. Métodos y Metodologías de Desarrollo

Tal como lo menciona Jayaratna [7], ya en mediados de los años 90 se registraban más de 1000 metodologías de desarrollo y, desde entonces, muchísimas más han visto la luz, considerando temas como la orientación a objetos, desarrollo Web y métodos ágiles. Se debe tomar en cuenta cuáles de estas son acordes para los objetivos del ciclo de enseñanza a impartir, considerando su periodo de duración fija y las mencionadas restricciones educacionales. Además, se debe considerar que algunas metodologías se enfocan en el desarrollo de software, mientras que otras enfatizan la gestión de proyectos, y otras balancean ambos conceptos, lo que se debe tener en cuenta al momento de implementar alguna de ellas.

Según [3], ninguna alternativa es perfecta para los proyectos *Capstone*, pero estas aportan otorgando una estructura base para el desarrollo de un software.

### 2.2.1. Software Development Life Cycle (SDLC)

Uno de los primeros documentos que daba una estructura al desarrollo de software fue el que se le atribuye a Royce en 1987 [8], el cual entrega información acerca de prototipado, iteraciones y testeado de programas. A pesar de su utilidad, puesto que se diseñó pensando en programas de los años 90, presenta problemas debido a que los requerimientos del software se asumían observables. En dicha época se buscaba, en general, automatizar procesos de maquinaria, idea que vuelve al proceso incompatible con tareas que no tienen un predecesor manual como el desarrollar una aplicación Web, además de que en dicha época no existía orientación a objetos.

Existe una diferencia de opiniones acerca de la utilidad de SDLC en proyectos *Capstone*. Este considera el abandono de tareas en las fases finales si llegasen a ocurrir atrasos en la programación, lo que ocurre con frecuencia en equipos sin experiencia. Esta idea no es compatible con un contexto educacional, puesto que implicaría no abarcar todo el ciclo de enseñanza. Además, al ser una metodología tradicional, entrega un prototipo sólo en las etapas más avanzadas del proyecto, dificultando revisar y controlar el avance del resultado final. Por último, no otorga instancias para revisión y mejoras al código ya desarrollado en

etapas anteriores.

### **2.2.2. Desarrollo con orientación a objetos**

Estas metodologías suelen tener una estrecha relación con los lenguajes de desarrollo *C++* y *Java*, los cuales eran de uso común en 1990. Destacan por proveer de técnicas de análisis y diseño para el desarrollo de sistemas con orientación a objetos y generalmente respaldan el uso de procesos iterativos e incrementales [3].

De esta categoría, la metodología más conocida y con mayor herramientas de apoyo es la de *Rational Unified Process* (RUP) [9], siendo su uso en proyectos *Capstone* avalado en [10] en conjunto con una herramienta para el lenguaje UML. RUP consiste en un proceso de desarrollo incremental que involucra todo el ciclo de vida del producto, desde su comienzo hasta el lanzamiento del mismo. Sus iteraciones se basan en la entrega de requerimientos, dándole prioridad a aquellos de mayor riesgo. La metodología especifica roles, actividades y fases, pudiendo ser adaptado tanto para grandes como pequeños proyectos. Dicha adaptación, al momento de aplicarlo a proyectos *Capstone*, involucra optar por simplificar RUP de forma genérica para todos los proyectos, perjudicando a algunos, o simplificar RUP para cada uno, lo que requeriría de personal con experiencia en esta metodología.

Existe una alternativa pre-simplificada de RUP conocida como UPEDU (Unified Process for EDUCation), enfocada en proyectos estudiantiles [11], pero no existe documentación acerca de implementaciones exitosas de dicho método.

### **2.2.3. Gestión de proyectos tradicional**

Siendo una metodología enseñada frecuentemente en cursos computacionales, destaca por considerar cuatro principales fases en el desarrollo: concepto, desarrollo, implementación y cierre del proyecto. Está principalmente enfocado en la administración, calendarización, costos, calidad, riesgos, manejo y comunicación de personal en el proyecto (como recurso).

Una de las principales bases de esta metodología es el *project management body of knowledge* (PMBOK) [12], el cual encapsula buenas prácticas para variados tipos de productos. Sin embargo, [13] señala la importancia de diferenciar entre el proyecto en sí y el ciclo de vida del producto, sugiriendo que en el caso de desarrollos de software, el ciclo tradicional de gestión de proyectos sea adaptado al ciclo de vida del producto, pudiendo ser este último de cualquier tipo: SDLC tradicional (o cascada), espiral, incremental, prototipado, *Rapid Application Development* (RAD), o por metodologías ágiles.

Respecto a proyectos *Capstone*, no hay gran cantidad de documentación en donde se aplique esta metodología, no presentándose detalles acerca del proceso de desarrollo en los existentes [14]. A pesar de ello, se destaca en la literatura que, pese a sus puntos a favor, esta metodología no detalla temas como la documentación, a excepción de los documentos clave de cada fase como son el caso comercial, la especificación de objetivos y el detalle de la planificación. En pocas palabras, no entrega guía o técnicas para desarrollo de software.

Los principios de la gestión tradicional de proyectos, a pesar de ser útiles para un proyecto genérico, no son directamente aplicables en pequeños equipos *Capstone* de desarrollo, debido a que le da gran importancia al manejo de costos y planificación basándose en empresas tradicionales. Uno de sus puntos a favor es el énfasis que le da a la comunicación y feedback desde los equipos de trabajo, pero solicita una estructura detallada de plan de trabajo y una calendarización junto con una apreciación de costos lo que, debido a la inexperiencia de los desarrolladores, resulta en apreciaciones poco realistas. Además, esta metodología tiende a dejar el alcance del proyecto fijo desde el inicio, lo que ha quedado atrás desde 1990 con las ideas de desarrollo iterativo e incremental.

Según [3], por sí sola la gestión tradicional de proyectos no es suficiente para llevar a cabo desarrollo de software. Debe ser usada en conjunto con alguna metodología de desarrollo para que sea efectiva, lo que se traduce en que cursos que empleen esta alternativa deberán incorporar elementos extra para lograr un avance en el desarrollo de un proyecto de software.

#### 2.2.4. Team Software Process (TSP)

Es planteado a finales de los años 90 para proyectos de ingeniería de software. Se enfoca en el progreso y calidad del producto, en donde cada miembro del equipo de desarrollo aprende acerca de la aplicación de *Personal Software Process* para posteriormente trabajar en conjunto como equipos [15, 16].

En esta modalidad, cada miembro cumple un rol determinado y se compromete a trabajar para otorgar un seguimiento de su trabajo y del proyecto, comprometiéndose a cumplir con los estándares de calidad necesarios. Cada 2 a 3 meses se produce un proceso de re-lanzamiento, en donde se replantean los planes y se expone el conocimiento adquirido desde el último ciclo respecto del producto, del progreso y del proceso en sí. La metodología presenta estándares para guiar a los equipos y es recomendado el empleo de *coaching* para equipos sin experiencia.

Para medir el avance del proyecto se define un valor numérico para cada tarea en base a su porcentaje estimado del total. Si la tarea se cumple, entonces se adquiere dicho valor, lo que permite medir el avance periódicamente. Para esto TSP especifica tareas de trabajo, seguimiento de calidad y reportes de progreso semanales.

En contexto de proyectos *Capstone*, TSP ha sido probado en [17], señalándose que puede ser usado para todo tipo de proyecto, independiente de su tecnología e incluyendo proyectos que no sean de software. En dicho documento se adaptó el proceso TSP para que fuese de 2 iteraciones, en donde los estudiantes deben emitir una gran cantidad de documentos aparte del producto mismo. Se destaca que gracias a la forma de trabajar, los alumnos lograron comprender la diferencia entre un trabajo individual y un desarrollo en equipos.

En relación a las críticas de la metodología, se señala a TSP como un proceso que involucra gran cantidad de documentación y que no incluye mecanismos de análisis, diseño, desarrollo de código o desarrollo iterativo, por lo que los instructores son los que deben compensar con dicho material, regulando cuánto es suficiente documentar o definiendo qué funcionalidades priorizar para cada iteración. Sumado a esto, TSP considera en sus publicaciones proyectos

con una duración suficiente para realizar re-lanzamientos, en donde cada uno ocupa usualmente la duración de un proyecto *Capstone*, por lo que sus etapas deberían ser descartadas o acortadas considerablemente para poder encajarlas en un proyecto de 1 o 2 semestres.

### 2.2.5. Metodologías Ágiles

A pesar de ser una tendencia reciente respecto del resto de las metodologías, existen en la actualidad una gran cantidad de alternativas a considerar. En [18] se realiza un análisis a estas, dentro de las cuales se señalan:

- Extreme Programming (XP) [19]
- Scrum [20]
- Lean Development [21]
- Adaptative Software Development (ASD) [22]
- Crystal Methods [23]
- Feature Driven Development (FDD) [24]
- Dynamic Systems Development Method (DSDM) [25]

Cada una de estas metodologías es independiente de la otra, y cada una cumple con un propósito diferente, aunque se basen completa o parcialmente en los principios y prácticas del manifiesto ágil [18]. Estas metodologías tienen en común que todas son de desarrollo iterativo incremental (IID) [1], asumiendo que el concepto de orientación a objetos es algo común en el equipo desarrollo [26]. El manifiesto ágil [27] define 4 valores principales:

- Colaboración con el cliente por sobre negociación de contratos.
- Individuos e interacciones por sobre procesos y herramientas.
- Software funcionando por sobre documentación extensiva.
- Respuesta al cambio por sobre apearse a un plan preestablecido.

También, [27] define 12 principios, complementando la postura que una organización debe adoptar para considerarse ágil:

- La mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
- El software funcionando es la medida principal de progreso.
- Aceptar que los requisitos cambien, incluso en las etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventajas competitivas al cliente.
- Los procesos ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios deben ser capaces de mantener un ritmo constante de forma indefinida.
- Entregar software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
- La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
- Los responsables del negocio y los desarrolladores trabajan juntos de forma cotidiana durante todo el proyecto.
- La simplicidad es esencial, desarrollar lo justo y necesario para cumplir con el trabajo necesario.
- Los proyectos progresan gracias a la motivación de sus individuos. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
- Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
- El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
- A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

La agilidad basa su mentalidad en confiar en las personas, aceptar los cambios y fomentar la sencillez en sus procesos, buscando darle un acercamiento más “humano” a lo que tradicionalmente se emplea en los procesos industriales. Esto es conocido como el *mindset* ágil.

Debido a ser una tendencia reciente respecto de las otras, la presencia de documentos acerca de experiencias en la aplicación de cada metodología no es muy amplia respecto de proyectos *Capstone*. El caso de Extreme Programming (XP) es la excepción, en donde su uso en la industria es común [28], teniendo su propio campo de investigación enfocado a la aplicación

de sus técnicas en la sala de clases [4, 29]. En el caso de otros como Scrum, cuyo uso comienza a difundirse, y de DSDM, que es un producto comercial, no existe documentación suficiente respecto de su uso en proyectos *Capstone*.

Debido a la amplia variedad de metodologías, y para cuidar la extensión del presente documento, se hará hincapié en dos metodologías en particular: XP, debido a su amplio estudio y documentación en proyectos *Capstone*, y Scrum, debido a que en Chile es la metodología más difundida y la cual posee expertos a nivel local.

### **Extreme Programming (XP)**

Es un método ágil ampliamente conocido. Se describe como una metodología que enfatiza la colaboración, promueve el desarrollo rápido y anticipado de software, y entrega prácticas hábiles de desarrollo [1]. Se basa en cuatro valores: comunicación, sencillez, *feedback* y coraje. Además de emplear un desarrollo iterativo-incremental (IID), recomienda el uso de 12 prácticas esenciales:

1. Planning Game.
2. Entregas pequeñas y frecuentes.
3. Metáforas en el sistema.
4. Diseño simple.
5. Testing.
6. Refactorización frecuente.
7. Programación en pares.
8. Responsabilidad compartida de código.
9. Integración continua.
10. Ritmo sostenible.
11. Todo el equipo unido.
12. Estándares de código.

XP busca obtener la satisfacción del cliente a través de la creación de software de valor de manera rápida, con técnicas de desarrollo hábiles y sostenibles, y flexibilidad ante cambios. Está definido para equipos relativamente pequeños y con fechas de entrega de menos de un año, en donde sus iteraciones van usualmente entre una y tres semanas. Notar que, tal como señala su nombre, es una metodología enfocada a equipos de desarrolladores, entregando técnicas y prácticas específicas para mejorar el desempeño de estos.

A nivel general se compone de 5 etapas: Exploración, Planificación, Iteraciones, Producción

y Mantenimiento detallándose sus propósitos y actividades en la tabla 2.1. Junto con esto, en XP se detallan una serie de herramientas de trabajo y roles clave que complementan el desarrollo y elaboración del producto, junto con promover la participación de los involucrados, los cuales se detallan a continuación:

Fase	Propósito	Actividades
Exploración	<ul style="list-style-type: none"> <li>- Definir suficientes y bien estimadas story cards para el primer release.</li> <li>- Asegurar factibilidad.</li> </ul>	<ul style="list-style-type: none"> <li>- Prototipos.</li> <li>- Investigación de la tecnología requerida.</li> <li>- Definición y estimación de story cards.</li> </ul>
Planificación	<ul style="list-style-type: none"> <li>- Definir fecha e historias a entregar en el primer release.</li> </ul>	<ul style="list-style-type: none"> <li>- “Release Planning Game”</li> <li>- Definición y estimación de story cards.</li> </ul>
Iteraciones	<ul style="list-style-type: none"> <li>- Implementar un sistema testeado y listo para lanzarse al mercado.</li> </ul>	<ul style="list-style-type: none"> <li>- Programación y testeo.</li> <li>- Iteration Planning Game.</li> <li>- Definición y estimación de tareas.</li> </ul>
Producción	<ul style="list-style-type: none"> <li>- Despliegue operacional.</li> </ul>	<ul style="list-style-type: none"> <li>- Documentación.</li> <li>- Capacitación.</li> <li>- Marketing.</li> <li>- ...</li> </ul>
Mantenimiento	<ul style="list-style-type: none"> <li>- Mejoras y arreglos.</li> <li>- Construcción de mayores releases.</li> </ul>	<ul style="list-style-type: none"> <li>- Puede involucrar la repetición de otras fases para release incrementales.</li> </ul>

Tabla 2.1: Propósito y actividades definidas para Extreme Programming, basado en [1].

■ Herramientas:

- Story Cards: Tarjetas en donde son escritas breves características requeridas. Son referencias que el cliente después debe detallar. Poseen una granularidad de 2 a 10 días.

- CRC Cards: Tarjetas donde se definen breves ideas del tipo *class-responsibility-collaborator*. Se complementan con sketches para diseñar el producto.
  - Lista de tareas: Lista en donde son escritas las tareas para cada historia en una iteración, considerando tareas de 1 a 2 días.
  - Gráficos: Herramienta a la vista que expone el estado del desarrollo. El contenido que muestra dependerá del equipo. Por ejemplo, total de pruebas versus pruebas aprobadas.
- Roles:
    - Cliente: Se encarga de definir historias y pruebas de aceptación y escoge las historias para cada iteración y entrega.
    - Programador: Desarrolla pruebas, diseños y código, refactoriza cuando es necesario e identifica tareas y estimaciones.
    - Tester: Ayuda al cliente a plantear y desarrollar las pruebas.
    - Coach: Encargado de la instrucción y orientación de los miembros del equipo. Mantiene el enfoque del equipo en los procesos y regula la adaptabilidad de estos.
    - Tracker: Realiza seguimiento de las métricas y el progreso, además de entregar retroalimentación en las malas estimaciones.
    - Consultant: Realiza coaching y consultoría técnica complementaria.
- Las consideraciones que se deben tener en cuenta para aplicar XP son:
    - Requiere de la presencia constante del cliente, lo cual usualmente no es posible e imposibilita la realización de ciertas prácticas.
    - Emplea explicaciones habladas para conocer detalles acerca del diseño y requerimientos, lo que complica el recibir nuevos miembros y escalar el proyecto. No hay una forma estándar de documentación para facilitar el aprendizaje.
    - XP requiere que se realicen todas sus practicas para que las falencias de una se compensen con la otra, lo que no permite seleccionar sólo las esenciales.
    - Existen desarrolladores que no se sienten cómodos realizando programación en pares o se resisten a hacerlo.
    - No hay una estandarización de documentos que puedan compartirse de proyecto a proyecto, lo que impide la reutilización de herramientas y un vocabulario común

- en grandes organizaciones.
- No posee elementos que fomenten diseño de la arquitectura del producto, surgiendo esta a medida que se va desarrollando y corrigiendo este.

Respecto de proyectos *Capstone*, lo que se destaca de XP en la literatura es:

- En [3] se considera a XP junto con DSDM como las metodologías más apropiadas para proyectos *Capstone*.
- En [30] luego de un estudio empírico, se obtuvieron resultados variados para cada técnica, señala que no basta con llegar y aplicar la metodología.
- En [31] se comparan ciertas metodologías y se señala que, a pesar de que algunas de las técnicas de XP son “engañosamente difíciles”, se considera apropiado debido a que es lo suficientemente adaptable como para ajustarse a la diversidad de cada proyecto y la habilidad de cada estudiante.
- En [32] se señala que es importante enseñar las técnicas y herramientas necesarias de XP antes de que el proyecto empiece y enfatiza el realizar *coaching* a lo largo de este.
- [33] evidenció un mayor conocimiento de las necesidades del cliente y de problemas de *testing* al usar este método.

XP aporta tanto ventajas como desventajas a los proyectos *Capstone* [3]. Posee técnicas útiles, está diseñado para equipos pequeños y aumenta la comunicación entre sus miembros. Además, define técnicas de generación de código, desarrollo enfocado a *testing* e integración continua entre otras, que favorecen la producción de software de calidad. Junto con esto, define roles y responsabilidades para cada miembro del equipo, pudiendo adaptarse incluso a condiciones y situaciones particulares [19]. A pesar de esto, posee desventajas para los proyectos *Capstone*: la documentación no se considera como parte del proceso de desarrollo sino que se incluye como parte del código, lo que dificulta las evaluaciones académicas. Recomendaciones de cómo manejar documentación en metodologías ágiles se encuentran presentes en [34].

Sumado a lo anterior, en el caso de proyectos *Capstone* no es factible tener al cliente constantemente supervisando el proyecto (customer-on-site) y tampoco las 40 horas semanales

consideradas por XP son factibles en un ambiente educativo con otras clases en paralelo. Se recomienda en la guía de Beck la técnica de Sistem Metaphor pero está vagamente explicada, por lo que se requiere de una alternativa [19]. Los instructores deberían tener experiencia en XP y tener suficiente disponibilidad de tiempo para hacer *coaching* a los equipos a menos que puedan usar una entidad extra y otorgarle a los alumnos un curso previo de uso de XP. El concenso presente en general es que la metodología es demasiado compleja y extensa como para poder incorporarla de manera inmediata [35, 36, 37] e incluso algunos rechazan la idea de emplear metodologías ágiles debido a la falta de documentación y a la impresión de que estas realmente corresponden a un nuevo tipo de metodologías ad-hoc [38, 39, 40].

La compatibilidad de XP en un contexto educacional depende del objetivo del curso. Si se busca enseñar a los alumnos a desarrollar proyectos siguiendo una metodología tal cual es, no es viable, debido a que no cumple con muchas de las restricciones de un proyecto *Capstone* [40]. Si, por otro lado, se quiere enseñar el cómo trabajar bien en pequeños equipos de desarrollo, desarrollar habilidades de *testing* y recibir software funcional en un periodo corto de tiempo, entonces esta metodología puede ser útil para el curso en cuestión.

## Scrum

Scrum se considera un *framework* para administración de proyectos que involucra una serie de prácticas simples pero que, en consecuencia, logran marcar una gran diferencia en la experiencia de trabajo [1]. Sus características clave son:

1. Equipos auto-gestionados.
2. No intervención de un ciclo una vez se comienza.
3. Reuniones diarias estandarizadas.
4. Iteraciones de 30 días.
5. Demostrar ante stakeholders al final de cada iteración.
6. Planear y priorizar iteraciones según lo que necesita el cliente.

Lo que enfatiza Scrum son una serie de valores y prácticas a ejercer en la administración de un proyecto, no abarcando temas como la toma de requerimientos, implementación, u otros. En consecuencia, esta estructura es fácil de combinar o complementar con otros métodos,

dando origen a metodologías mixtas como la mezcla entre Kanban y Scrum (Scrumban) [41, 42].

A nivel general se compone de 4 etapas: Planificación, Puesta en marcha, Desarrollo y Lanzamiento, siendo estas dos primeras también llamadas usualmente *pre-game* debido a que son la etapa de preparación previa al desarrollo. El propósito y actividades de cada una de ellas se detallan en la tabla 2.2.

Fase	Propósito	Actividades
Planificación	<ul style="list-style-type: none"> <li>- Establecer la visión.</li> <li>- Definir expectativas.</li> <li>- Asegurar financiamiento.</li> </ul>	<ul style="list-style-type: none"> <li>- Definir visión y presupuesto.</li> <li>- Product backlog inicial y estimación de items.</li> </ul>
Puesta en marcha	<ul style="list-style-type: none"> <li>- Identificar requerimientos y priorizarlos lo suficiente como para una primera iteración.</li> </ul>	<ul style="list-style-type: none"> <li>- Planificación inicial.</li> <li>- Diseño exploratorio y prototipos.</li> </ul>
Desarrollo	<ul style="list-style-type: none"> <li>- Implementar un sistema listo para lanzarse al mercado en 30 días.</li> </ul>	<ul style="list-style-type: none"> <li>- Sprint planning.</li> <li>- Definición de sprint backlog y sus estimaciones.</li> <li>- Reuniones semanales.</li> <li>- Sprint review.</li> </ul>
Lanzamiento	<ul style="list-style-type: none"> <li>- Despliegue operativo del producto desarrollado.</li> </ul>	<ul style="list-style-type: none"> <li>- Documentación.</li> <li>- Capacitación.</li> <li>- Marketing y ventas.</li> <li>- ...</li> </ul>

Tabla 2.2: Propósito y actividades definidas para Scrum, basado en [1].

Junto con lo anterior, en Scrum se detallan una serie de herramientas de trabajo y roles clave que complementan el desarrollo y elaboración del producto, además de promover la comunicación y participación de los involucrados. Dichos elementos se detallan a continuación:

- Herramientas:

- Product Backlog: Herramienta que encapsula los requerimientos del producto a

desarrollar, considerando características, casos de uso, mejoras, defectos y tecnologías necesarias, incorporando aquellos factores que surgen a lo largo del desarrollo.

- Sprint Backlog: Herramienta que contiene las tareas escogidas del Product Backlog para preparar en el actual sprint, con una granularidad usual de 4 a 16 horas cada una.
- Gráfico del Backlog: Herramienta gráfica que muestra el avance a lo largo de los sprints, muestra el trabajo restante en función del tiempo invertido en el proyecto.

■ Roles:

- Product Owner:
  - En representación del cliente, prioriza los elementos del Product Backlog y los escoge para el actual sprint.
  - Revisa, junto con otros stakeholders, el sistema resultante al final de cada Sprint.
- Scrum Team: Grupo de desarrolladores que se encarga de trabajar en los elementos del Sprint Backlog. Su deber es hacer todo lo que sea necesario para llevar a la vida la visión que se tiene del producto.
- Scrum Master:
  - Conoce y refuerza la visión y los objetivos de cada iteración y del proyecto.
  - Promueve que se sigan y respeten los valores y ceremonias de Scrum.
  - Actúa como intermediario entre la directiva y los desarrolladores.
  - Realiza un seguimiento del progreso y soluciona los impedimentos.
  - Dirige las reuniones diarias (Daily Scrum) y presenta los Sprint Review.
  - Puede ser un desarrollador que dedica la mitad de su esfuerzo a la gestión del desarrollo, lo que usualmente no se recomienda ya que, al formar parte del equipo, este suele relajar la exigencia de las normas.

■ Las consideraciones que se deben tener en cuenta para aplicar Scrum son:

- Es una metodología para administración de proyectos, por lo que no entrega guías para el desarrollo de software o toma de requerimientos.

- No posee estándares para la documentación, por lo que diferentes proyectos entregarán diferentes guías y reportes a menos que se defina un estándar a nivel interno.

Respecto de proyectos *Capstone*, lo que se destaca del uso de Scrum en [43] es:

- El enseñar métodos ágiles se vuelve mucho más sencillo a través de proyectos y trabajo práctico.
- El rol de Product Owner es crucial para el éxito de un proyecto Scrum.
- El Scrum Master debe asumir el rol de guardián, apoyando las prácticas de Scrum en el proyecto y eliminando impedimentos que dificulten su avance.
- Desde el comienzo y lo antes posible se debe enseñar e impartir el cuándo se da por finalizada una tarea.
- Para que el trabajo práctico logre generar aprendizaje, por ejemplo, en historias de usuario, es esencial una buena comunicación entre el Product Owner y el equipo de desarrollo.
- Las reuniones “Daily Scrum” deben ser para reportar el estado del proyecto a todo el equipo, no solo al Scrum Master.
- Estimar y planear es una tarea difícil, pero va mejorando con la práctica.
- De incorporarse un enfoque tradicional, es importante hacerlo en una manera apropiada que preserve la agilidad.

Scrum presenta la misma desventaja que PMBOK: es una guía metodológica que no especifica guías o pautas para el desarrollo de software. A pesar de esto, es útil para equipos de cualquier tamaño, ya que ayuda a aumentar la comunicación entre sus miembros [1], siendo apropiado para contextos de desarrollo de software debido a su modalidad iterativo-incremental.

Scrum presenta pocos roles, algo favorable para un entorno educacional, ya que usualmente se cuenta con poco personal y una gran cantidad de equipos. También presenta pautas para sus documentos y ceremonias, siendo su aplicación un proceso sencillo y replicable.

Al igual que en XP, en el caso de proyectos *Capstone* no es factible tener al cliente constantemente supervisando el proyecto, lo que se compensa con el rol de Product Owner. Debido a que sus ciclos son de 30 días y requiere de reuniones diarias, la decisión usual que se toma es la de adaptar estas normas para hacerlas compatibles con un ambiente académico.

La compatibilidad de Scrum en un contexto educacional depende de qué tan al pie de la letra se pretenda aplicar, compensando de alguna forma la poca orientación en términos de desarrollo de software. Si se busca enseñar Scrum completamente, no es compatible. Por otro lado, si se quiere enseñar a los estudiantes cómo organizarse en un proyecto y gestionar su avance, entonces puede ser útil.

### **2.2.6. Desarrollo a Medida (Ad-Hoc)**

Consiste en la ausencia de cualquier tipo de metodología conocida, adaptando el desarrollo a una estructura hecha a la medida del producto que se requiere. Aunque esto no es considerado una alternativa apropiada para un contexto educacional [6], es muy común en procesos industriales. De hecho, según [19] en un equipo pequeño de dos personas no es necesario el empleo de una metodología. Ejemplos de esta idea se presentan en Personal Software Process [15], una adaptación de RUP [9], y en el empleo de buenas prácticas en la programación descritas en Programación Pragmática [44]. Dado que el objetivo de este trabajo es proponer una metodología, no se indagará en esta alternativa.

### **2.2.7. Comentarios Adicionales**

Como se señala en [3], no hay una metodología que se ajuste perfectamente a los proyectos *Capstone*. Lo que se sugiere es adaptar una o varias metodologías, siendo más recomendadas las iterativo-incrementales por sobre desarrollo en cascada, ya que las iteraciones proveen de puntos naturales para evaluar avances, permitiendo cambios o ajustes controlados en los requerimientos y ajustándose mejor al desarrollo ideal de un software, siendo validado constantemente el producto con lo que el cliente necesita.

Se resalta la importancia de emplear tiempo en la sala de clases enseñando la filosofía y el trasfondo de las tecnologías empleadas, realizando una preparación previa antes de iniciar los proyectos. **Se debe convencer tanto a los profesores como a los estudiantes de por qué es importante el uso de dicha tecnología.** Finalmente, es importante que la metodología empleada sea evaluable a lo largo del curso, para invitar a los participantes a reflexionar acerca de las ventajas y desventajas de ponerse esta en práctica.

La metodología a utilizar es un aspecto importante en proyectos de software. Existen numerosas alternativas y variantes, tanto a nivel personal como para grupos grandes de desarrollo, teniendo todas ellas puntos a favor y en contra al momento de aplicarse a proyectos *Capstone*. Tal como se mencionó anteriormente, existen metodologías de gestión de proyectos y de desarrollo de software, por lo que antes de optar por una alternativa, es importante que los instructores se informen bien de lo que su curso necesita. Es una decisión importante, puesto que la experiencia y aprendizaje de los estudiantes de hoy influyen las prácticas en las industrias del futuro.

## Capítulo 3

# DIAGNÓSTICO

Para el presente trabajo se analizaron las asignaturas de “Gestión de proyectos de informática” y “Taller de desarrollo de proyectos informáticos”, también conocidas como “feria de desarrollo de software”. Para esto, se consideraron 3 puntos de vista:

- Según lo observado durante la ejecución de la asignatura en el año 2018, considerando problemas en su ejecución.
- Desde las observaciones señaladas en documentos legados, redactados por alumnos de 2014 a 2018, las cuales se analizan en la sección [5.5](#).
- Desde las inconsistencias detectadas en la planificación respecto de la información expuesta en la sección de marco conceptual.

### 3.1. Feria de Desarrollo de Software

El nombre de “feria” o “feria de software” es debido al evento de cierre de dicho ciclo de desarrollo, en donde los proyectos que terminan el curso son expuestos en una feria de entrada liberada, siendo presentado cada uno en un puesto designado para su equipo durante un día completo, siendo dichos proyectos calificados por un jurado y por un comité evaluador.

Las asignaturas de “Gestión de proyectos de informática” y “Taller de desarrollo de proyectos informáticos” son dos asignaturas secuenciales que abarcan un año académico completo y se realizan en el quinto año del programa de estudios de ingeniería civil informática de la USM. En estas, los alumnos en equipos de 5 integrantes deben proponer una idea de proyecto de software, ya sea para solucionar una problemática real, o para aprovechar una oportunidad existente, exigiéndose que sea una idea innovadora (no encontrándose en internet ideas similares).

La finalidad de las asignaturas es que los alumnos simulen ser una pre-empresa en busca de lograr el emprendimiento, por lo que se considera a la feria de software como la formulación de un proyecto que abarca 3 tipos de contextos posibles:

- **Desarrollo de software:** Poniendo a prueba los conocimientos vistos por los alumnos a lo largo de toda su formación académica hasta la fecha.
- **Emprendimiento:** Trabajando en equipo para sacar adelante un proyecto de software.
- **Innovación:** Identificando oportunidades de productos que actualmente no se encuentren en el mercado.

### 3.2. Metodología de desarrollo

Inicialmente las asignaturas fueron basadas en lo expuesto en la metodología de Gestión de proyectos tradicional, expuesta anteriormente en la sección 2.2.3, siendo aplicadas las prácticas expuestas en el PMBOK [12]. Luego, considerando la tendencia emergente de metodologías ágiles, se consideró incorporar estas a la feria de software. Debido a la resistencia

ante la idea de cambiar la estructura actual, se decidió por parte de la coordinación de dicha asignatura considerar tanto aspectos tradicionales como ágiles, resultando una “metodología híbrida”. El resultado de esto fue un cruce, en el cual el control y documentación de los proyectos se realizó de forma tradicional, mientras que el desarrollo empleó las prácticas propuestas por Scrum [20]. Según la misma coordinación, los alumnos deben poseer conocimientos tanto de desarrollo tradicional como de agilidad y no pudiendo descartarse lo expuesto en el texto guía, el cual es el PMBOK.

PMBOK no otorga guías ni técnicas para el desarrollo de software. Sus principios, a pesar de ser útiles para un proyecto industrial cualquiera, no son directamente aplicables a un proyecto *Capstone* en donde no se requiere de manejar costos a un nivel empresarial [3]. Además de esto, se enfoca en solicitar un detalle de planificación que un equipo sin experiencia no es capaz de generar con precisión, lo que resulta en estimaciones poco realistas. Esto de paso se contradice con los principios del manifiesto ágil, por lo que la “metodología híbrida” que se señala es infactible, ya que pretende fijar alcance, costos y tiempos del proyecto al mismo tiempo, cosa que no es acorde a los principios tradicionales o ágiles [45]. Las consecuencias de esta decisión se evidencian más adelante en el detalle de cada semestre en las secciones 3.3 y 3.4.

### 3.2.1. Participantes y Roles

La estructura de roles de las asignaturas consta de 3 entidades principales: profesor, ayudante y alumno, pudiendo variar la cantidad de ayudantes si la cantidad de alumnos lo requiere. Además, en particular se incorporan un cliente y un Jefe de Proyecto. Las labores de cada uno de estos miembros son:

- **Profesor:** Responsable de realizar clases y evaluaciones cuando sea necesario. Controla el estado actual de los equipos de la asignatura y delega responsabilidades a los ayudantes. Uno de ellos, el profesor coordinador, se encarga de controlar que el material académico, los hitos y las evaluaciones se entreguen a tiempo, siendo quien toma

las decisiones de la asignatura. En las asignaturas, los profesores se consideran **la gerencia o directiva** de los equipos y, además, los **Product Owner**.

- **Ayudante:** Alumno que, luego de cursada la asignatura, decide contribuir como asistente en futuras versiones de esta, siendo el nexo entre profesores y alumnos, controlando el avance y desempeño de estos y pudiendo calificar evaluaciones si se es requerido. Uno de ellos, el ayudante coordinador, además de asistir a profesores actúa como canal de comunicación entre estos y otros ayudantes, coordinando el quehacer de estos últimos. En las asignaturas, estos se consideran los **Scrum Master y Product Owners** de los equipos.
- **Equipo:** Alumnos cursando la asignatura, quienes realiza el desarrollo de software y son sujetos a evaluaciones. En las asignaturas, 5 alumnos conforman una **pre-empresa**.
- **Jefe/líder de Proyecto:** Miembro de un equipo que, además de ejercer como desarrollador, se ofrece como encargado y representante ante las otras entidades de la asignatura.
- **Cliente:** Supervisor voluntario y externo al proyecto, el cual cumple el rol de un inversionista. Debe declararse conforme con el producto final para que los alumnos puedan aprobar la asignatura.

Debido a que el ciclo de la feria de software se compone de dos asignaturas, cada una cuenta con sus propios hitos, documentación y eventos, detallados en la siguiente sección.

#### Inconsistencia

El rol de *Product Owner* definido en la estructura oficial es inconsistente con Scrum. Este es quien se contacta con el cliente del proyecto para definir lo que necesita, lo que es responsabilidad de los alumnos, no de los profesores o del ayudante.



Figura 3.1: Descripción gráfica de los roles de la asignatura y su interacción.

### Responsabilidades

A pesar de que cada miembro busca cumplir sus responsabilidades, al no estar estas claras, estas son aprendidas en la marcha pasando algunas desapercibidas. En legados de años anteriores (anexo 5.5) se señalan 3 situaciones de este tipo:

- Hubo casos en donde los ayudantes dejaron de controlar a sus equipos, sus avances o la calidad de su producto.
- Ciertos clientes buscaron apropiarse del software una vez terminado, considerando a los alumnos como una consultoría de bajo costo o, alternativamente, cobrándoles a estos un porcentaje de los ingresos al momento de salir al mercado, lo que contradice el emprendimiento que se busca impulsar en la asignatura.
- Al definirse los profesores como una gerencia o directiva, se produce, en consecuencia, que estos actúen como jefes por sobre su rol de educadores, esperando que los alumnos resuelvan sus problemas solos y causando que estos se sientan poco apoyados.

### 3.3. Primer Semestre: Gestión de Proyectos de Informática

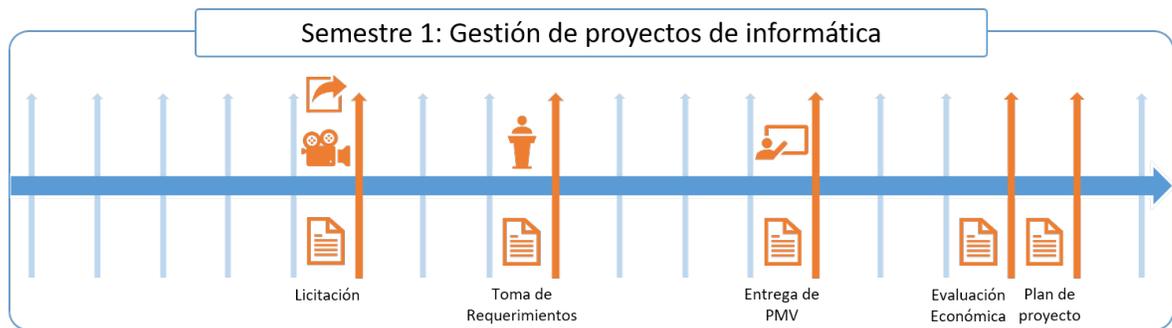


Figura 3.2: Representación visual del primer semestre.

La primera mitad del ciclo anual consiste en el planteamiento de un proyecto de software, su definición, el desarrollo de un PMV [46], su evaluación económica y la planificación del desarrollo para el siguiente semestre, siendo esta asignatura descrita en su documento introductorio [47] como:

*”Los estudiantes, trabajando en equipo analizan procesos, métodos, técnicas y herramientas sugeridas para formular y gestionar proyectos informáticos en un contexto de licitaciones orientadas al emprendimiento innovador con base en Tecnología de la Información (TI). Formulan un proyecto que integre las etapas de inicio y planificación, basados en una problemática real para dar continuidad a este proyecto en el semestre siguiente.”*

Tal como se señala, esta parte de la planificación consiste en la formulación del proyecto como tal y una implementación inicial, por lo que considerando que esta es una asignatura, fueron definidos hitos (de ahora *entregas* o *entregables*) que, en conjunto con otros factores, controlan el avance académico de los proyectos.

En la figura 3.2 se muestran 9 evaluaciones para el primer semestre, representados como iconos de color naranja y detallándose estos en la sección 5.7. Dichos entregables son agrupados en 5 entregables, detallándose su correspondiente porcentaje de la nota final en la tabla 3.1.

Actividades individuales y grupales.	10 %
Entrega 1: Licitación de ideas de proyecto.	20 %
Entrega 2: Informe y defensa de requerimientos.	20 %
Entrega 3: Defensa del producto mínimo viable.	15 %
Entrega 4: Evaluación económica del proyecto.	15 %
Entrega 5: Informe y defensa de Plan de Proyecto.	20 %

Tabla 3.1: Ponderación de entregables de gestión de proyectos de informática.

### 3.3.1. Entrega 1: Licitación de Ideas de Proyecto

Este entregable consiste en la definición como tal de la idea del proyecto y sus características fundamentales, además del establecimiento del equipo como una pre-empresa. Dicha revisión se compone de 3 evaluaciones:

- **Un informe de licitación** detallando al equipo y su potencial idea de proyecto en 14 items, incluyendo desde descripciones hasta diagramas de la idea.
- **Un video** de 2 a 3 minutos, evaluado posteriormente por un jurado externo a las asignaturas para obtener opiniones desde diversos entornos respecto de cada proyecto.
- La creación de **cuentas en LinkedIn** a modo de currículos vitae, con el fin de que los alumnos asimilen el concepto de pre-empresa en un ambiente laboral, los cuales son revisados por los profesores para corroborar que su contenido sea acorde.

Estos tres elementos son entregados en *YouNoodle* [48], una plataforma online. De aprobar tanto el jurado como los profesores la idea se procede a la siguiente etapa, de lo contrario, los profesores pueden solicitar una reformulación o determinar el rechazo de la idea de proyecto.

#### Licitación

En las primeras semanas no se realiza seguimiento ni se entrega orientación a los equipos mientras estos buscan una idea de proyecto. En consecuencia, la mayoría de las ideas propuestas son rechazadas, generando los alumnos ideas de proyectos improvisadas o de alta complejidad por temor a reprobar, evidenciado en las críticas de legados de años anteriores.

#### Informe de Licitación

Hay elementos solicitados en el documento de licitación que cumplen la misma función, resultando en un texto con información redundante. Esto causó descontento en los alumnos y trabajo adicional para quienes revisaron dichos informes, respaldando la “sobrecarga” mencionada en los legados.

#### Video Online

Las pautas para la entrega del video fueron poco claras y, al no entenderlas, los estudiantes realizaron un video comercial, pero debido a que aún no había software desarrollado esto requirió de mas trabajo del necesario.

Comentarios de legados anteriores señalan que el jurado le da énfasis al modelo de negocios de cada proyecto, esperando que sean sostenibles económicamente. A pesar de que este elemento no forma parte de la pauta del video, causó descuentos en los equipos y la desaprobación de algunas ideas por parte de dichos evaluadores.

### **3.3.2. Entrega 2: Documento de Identificación de Historias de Usuarios**

Esta entrega consiste en la presentación de un documento que detalla al producto propuesto por el equipo, añadiendo las características técnicas que este requiere. Los temas de dicho documento son posteriormente expuestos ante profesores y ayudantes en una presentación evaluada en donde se revisa la idea, características del proyecto y las historias fundamentales de este, además de proponerse un Producto Mínimo Viable a desarrollar (prototipo inicial).

#### Contenidos del documento

Se solicita definir el alcance del proyecto en 20 historias para que se distribuyan 5 entre las 4 entregas de software del año, pero esto va en contra de lo que busca el enfoque ágil, cerrando la estructura de las asignaturas a un enfoque tradicional. Esto, en consecuencia, dio origen a muchas “falsas historias”, las cuales eran usadas para poder cumplir con el número solicitado y siendo estas criticadas, ya sea por no cumplir con el criterio INVEST o por consistir en tareas y no historias (crear un login, hacer click derecho, etc.). En dicho informe, además, se piden elementos innecesarios, como una tabla de tiempos invertidos por cada alumno en la elaboración del documento (un dato innecesario) y conclusiones por parte de los alumnos respecto de lo expuesto en el documento, lo que no es acorde a un informe técnico.

### 3.3.3. Entrega 3: Defensa del Producto Mínimo Viable

En la siguiente etapa se busca estimar las historias de usuario planteadas en el documento anterior y la formulación de sus criterios de aceptación, entregado esto en un informe al ayudante. Además, inicia un Sprint de desarrollo, enfocado en realizar un prototipo siguiendo la lógica de un Producto Mínimo Viable (PMV) [46].

En esta etapa el ayudante debe reunirse semanalmente con sus equipos para controlar el avance y desarrollo de cada prototipo el cual, luego de un determinado plazo, es presentado ante los profesores para su evaluación. Finalmente, los alumnos deben entregar un informe de retrospectiva, analizando los hechos ocurridos durante el Sprint.

#### Producto Mínimo Viable

Respecto del PMV, dicho concepto se origina desde *The Lean Startup* [49], donde es definido como una versión simplificada del producto final que puede ser utilizada por sus usuarios, pudiéndose observar el comportamiento entre estos y dicho PMV. Esto no es lo que se realizó en la asignatura, puesto que el software nunca fue usado por sus posibles usuarios.

#### Estimación de historias

Al estimar historias de usuario no se entregó un método estándar, por lo que cada equipo lo realizó de manera diferente. Es importante utilizar un método que considere el juicio inexperto de los alumnos, para evitar que sub-estimen o sobre-estimen sus historias. Por otro lado, como cada equipo redactó su informe de forma diferente, la coordinación señaló a aquellos con mejor apariencia y le señaló al resto que siguieran su ejemplo, contradiciendo la buena convivencia y el ambiente de motivación que busca fomentar agilidad.

#### Presentación de PMV

Para las presentaciones de software no hubo claridad respecto de los tiempos disponibles para cada equipo al exponer y, debido a que el periodo de defensa del prototipo no era controlado, las presentaciones tendieron a alargarse y salirse del margen.

#### Informe de Retrospectiva

Se solicitaron 6 elementos en el informe de retrospectiva, de los cuales solo uno correspondió al estándar de Scrum. El resto fueron tablas de tiempos, diagramas y otros elementos no acordes con dicha ceremonia, solicitándose al jefe de equipo que evaluara el desempeño de sus compañeros, contradiciendo nuevamente el valor de buena convivencia y motivación del agilismo al hacer que el líder juzgue a su equipo.

### 3.3.4. Entrega 4: Evaluación Económica del Proyecto

La siguiente entrega es un documento en donde se estima el total de inversión requerido por el proyecto de cada equipo, simulando ser una empresa tradicional. Se le entrega a cada equipo una planilla excel como referencia, debiendo los alumnos calcular costos de sueldos, instalaciones, herramientas, licencias y cualquier otro elemento que fuese requerido, asumiendo que cada miembro trabaja por horas y que no cuentan con materiales inicialmente. Además, se debe entregar una reflexión tanto del aprendizaje de este entregable, como del monto del proyecto resultante.

#### Contenidos y evaluación

Debido a la ocurrencia de un paro de estudiantes en la universidad se realizó una única clase para complementar esta entrega, realizada el mismo día de la evaluación y realizada por un profesor del área de gestión. Los alumnos señalaron que no estaban preparados para la evaluación, sintiéndose pasados a llevar al no darse dicha información de manera anticipada.

El estudio económico de un proyecto y su viabilidad es algo que debiese enseñarse en un ramo enfocado a temas económicos, el cual, según la malla curricular de la carrera, ya existe: “Introducción a la matemática financiera”. Como alternativa, se sugiere orientar respecto de modelos de negocio para mantener un software, puesto que ha sido solicitado en legados anteriores.

### **3.3.5. Entrega 5: Informe de Plan de Proyecto para Segundo Semestre**

Es el documento considerado como el más importante del semestre, dando cierre al primer ciclo de la feria de software. Consiste en un documento extenso que recopila los detalles de las entregas anteriores y fija una planificación de lo que se desarrollará en cada Sprint del siguiente semestre.

#### Contenidos

Si el lector se dirige a la sección 5.7 de anexo, se dará cuenta de que incluso la descripción del informe resulta extensa. La confección de este documento y su posterior revisión resulta una tarea extenuante, resultando en informes con una extensión de 30 páginas en promedio.

En dichos informes, se observa que muchos elementos solicitados buscan analizar la misma información o cumplen el mismo objetivo. El informe resultante presenta, en su mayoría, información redundante.

#### Desarrollo Tradicional

En este informe se fija el alcance del software mediante una planificación de las siguientes entregas, transformando la actual metodología en un desarrollo en cascada que usa herramientas ágiles. El enfoque ágil busca fijar los tiempos y costos, negociando el alcance del producto, lo cual no se cumple si se fija todo desde un comienzo, contradiciendo la metodología “hibrida” descrita en el curso.

### **3.3.6. Actividades Individuales y Grupales en Clases**

Consiste en la evaluación de la asistencia de los alumnos y su participación en las distintas actividades en clases, las cuales buscan fomentar la participación de los alumnos en cada clase y complementar el aprendizaje práctico que se ve en el desarrollo del proyecto.

### 3.4. Segundo Semestre: Taller de Desarrollo de Proyectos Informáticos



Figura 3.3: Representación visual del segundo semestre.

En esta parte del ciclo se da inicio al desarrollo del proyecto como tal y en paralelo comienza su “marketing” en redes sociales. Esto con el fin de difundir tanto al software que se está produciendo como al evento mismo de feria de software que se efectuará a fin de año. La descripción presente en el documento introductorio de esta asignatura la define de la siguiente forma:

*”Descripción: El estudiante integra la metodología de proyectos a través de los conocimientos adquiridos en su plan de estudio, aplicándolos para obtener una solución informática a un problema real y multidisciplinario. El resultado del proyecto es expuesto al término de la asignatura, en una Feria abierta a la comunidad interna y externa que cumple el rol de un examen final. Esta asignatura potencia en los estudiantes competencias diferenciadoras, asociadas al emprendimiento, creatividad, innovación, auto-aprendizaje, trabajo en equipo y gestión de proyectos.”*

Para este periodo, al igual que el anterior se piden diversos trabajos y presentaciones y, además de esto, debido a que se le da relevancia al desarrollo y a su presentación final se exige la aprobación por separado del software en su última entrega y de su presentación ante el público durante el día del evento de feria de software, requiriéndose la aprobación como tal de dicho producto para poder ser presentado en tal día.

Cada ícono en la figura 3.3 corresponde a una evaluación, siendo de color gris aquellas relacionadas con marketing. Sus ponderaciones se muestran en la tabla 3.2 y la descripción de cada uno de estos elementos se presenta a continuación, adjuntándose un detalle de ellos en la sección 5.8.

Elementos de Marketing (Imagen, sitio Web, video y póster)	10 %
Entrega 1: Primer Sprint (en laboratorio)	13 %
Entrega 2: Segundo Sprint (con cliente presente)	13 %
Entrega 3: Tercer Sprint (revisión completa del software)	20 %
Evento Feria de software (evalúa coordinación y jurado)	30 %
Evaluación de compromiso (Asistencia, retrospectivas, reuniones)	10 %
Documento Legado	4 %

Tabla 3.2: Ponderación de evaluaciones del segundo semestre.

#### Observaciones

El evento de feria de software posee una ponderación del 30 %, siendo dicha cantidad para asegurar la participación de los alumnos en el evento. Se sugiere balancear más equitativamente los porcentajes, puesto que la participación se puede lograr estableciendo una nota mínima para el evento, dándole más importancia a las entregas de software y a la calidad de su desarrollo.

### 3.4.1. Informe de Marketing

Consiste en el desarrollo de un sitio Web que describa al producto y al equipo que lo desarrolla, además de un video que promocioe dicho software, los logos de este y la creación de páginas en redes sociales que hablen del mismo.

En el caso del video, éste se revisa desde el mismo sitio Web hecho por los estudiantes, el cual es cargado en la página de la feria de software para que quede expuesto públicamente [50], mientras que los logos, descripción del producto y enlaces a redes sociales deben ser presentados en un informe.

#### Contenidos

Hubo confusión debido a que se pidió valor agregado en el sitio Web. Esto usualmente va en las presentaciones e incluye factores que no involucran desarrollo, lo cual en este caso no se pudo mostrar, causando descuentos en los alumnos que no supieron qué presentar.

#### Malas Prácticas

Durante la revisión de los sitios Web, se observó que el profesor coordinador editó las calificaciones de otros evaluadores, lo que no pasó desapercibido. Dicho profesor señaló que se confundió, pero al revisar el historial de cambios se observó que esto fue algo intencional. Se requiere de mayor control en cómo califica cada evaluador.

### 3.4.2. Entrega 1: Primer Sprint

Al igual que la presentación del PMV, este consiste en la presentación de los equipos ante profesores para mostrar el avance que tiene su desarrollo, ahora siendo la exposición en un laboratorio más amplio.

Para este segundo semestre se trató de cambiar las rúbricas por parte de quienes estaban a favor del cambio, pero luego de un conflicto interno la coordinación decidió mantener sus propias pautas, aceptando solamente una nueva pauta para los informes de retrospectiva, la

cual se detalla en el anexo 5.8. Además, se solicitó un nuevo informe redactado por el ayudante de cada grupo evaluando el avance de cada uno de sus equipos, denominado “Rúbrica del Product Owner”.

#### Documentación innecesaria

A pesar de modificarse la pauta de la retrospectiva, esta incluye un Sprint Review, el cual corresponde a otra ceremonia de Scrum, la cual es una entrega presencial. Comparado a los hitos de la asignatura, el Sprint Review corresponde a las “entregas” de software, por lo que este documento es redundante e innecesario. La misma información también se solicita en la rúbrica del Product Owner, por lo que se presenta 3 veces la misma información de distinta manera, resultando en 2 documentos innecesarios.

#### Sobrecarga a evaluadores

La rúbrica del Product Owner consistía en un reporte semanal redactado por el ayudante, siendo todos estos revisados a fin de año por el ayudante coordinador. Tomando en cuenta los 15 equipos y las 10 semanas aproximadas de desarrollo, esto implicaba revisar 150 informes. Al señalarse esto, se decidió solicitar 1 reporte por Sprint, resultando un total de 45. Tomando en cuenta que una persona los revisa, resultó en un trabajo exhaustivo para revisar información que ya se encontraba en otras entregas.

#### Errores Recurrentes

Luego de presentar, cada equipo se retira del salón sin ver las presentaciones de otros. Al no tenerse interacción con otros equipos, ninguno aprende de los errores de sus compañeros y se fomenta la competitividad. Se sugiere incentivar a los alumnos a ver las presentaciones de sus compañeros luego de exponer, puesto que ayuda a que los alumnos tomen dicha experiencia como retroalimentación.

#### Falta de información

Dos grupos presentaron problemas debido a que las plataformas utilizadas inicialmente no fueron compatibles con su proyecto, debiendo volver a desarrollar todo con una nueva tecnología. Al no conversarlo con la coordinación, no consideraron la posibilidad de intercambiar o postergar algunas de sus historias, resultando con notas deficientes por no estar al tanto de dicha posibilidad.

### 3.4.3. Presentación de posters

Posterior a la entrega 1 también se solicita un “póster”, el cual consiste en un cartel promocional. Este impreso debe ser entregado durante clases a los profesores, quienes los colocan en la universidad para promocionar el evento de feria de software.

#### Problemas de Rúbrica

La estructura solicitada en los posters fue la de un documento de investigación, resultando tan densos en texto que no invitaban a la lectura en absoluto. Los estudiantes optaron por dos alternativas: seguir al pie de la letra la pauta, generando un póster denso y poco llamativo, o privilegiar la apariencia y calidad de este, sacrificando sus calificaciones. Además, no se ofreció orientación para la elaboración de estos, resultando estos con apariencia deficiente, contenido poco claro o textos muy pequeños, y las dimensiones solicitadas no fueron acordes al lugar donde estos fueron colocados, dificultando su lectura.

#### Problemas de Fondos

El financiamiento de cada póster fue con dinero de los alumnos, teniendo todos los equipos un capital base, pero debiendo pagar por su cuenta el excedente. El tema de costos es recurrente en los legados, señalados en el anexo 5.5. En la propuesta del capítulo 4 se presenta una alternativa a esta evaluación.

### 3.4.4. Entrega 2: Segundo Sprint

La diferencia entre esta instancia y la entrega anterior es que el cliente debe estar presente para que se compruebe si este valida o no la propuesta de producto que hasta ahora el equipo ha estado desarrollando.

#### Valoración del Cliente

A pesar de que agilidad busca fomentar la interacción con el cliente, este sólo se involucró en dos instancias: la segunda entrega de software y el evento final. Ya que el entregable 2 es una entrega parcial no le aportó valor al cliente, puesto que sólo se le muestra una versión incompleta del software final y, debido a lo caótico del evento y la alta afluencia de invitados, el cliente pasa desapercibido durante este, lo que fue criticado en legados del año 2017.

### 3.4.5. Reporte de marketing

Consiste en un informe que expone los resultados logrados por los medios de marketing del equipo (sitio Web y redes sociales).

#### Penalizaciones Extra

A pesar de que el reporte de *marketing* sólo solicitó en su pauta un análisis, se penalizó por elementos como el no tener el nombre del producto en el archivo, no incluir un contador de visitas en la página Web, no viralizar lo suficiente el video y no presentar suficiente *marketing* en medios tradicionales, lo que no se mencionó en la rúbrica.

### 3.4.6. Entrega 3: Último Sprint

Debido a ser la entrega final, ésta pone a prueba todos los aspectos que debiese realizar el software como si se tratase de una prueba antes de comercializarse. De aprobarse éste, se autoriza al equipo a presentarse en el evento final, de otro modo se otorga un periodo de

recorrección o se reprueba a dicho equipo según sea la gravedad de la situación.

Para esta entrega se consideran dos partes:

- Exposición y defensa de historias de usuario, tal como en entregas anteriores.
- La exposición de uno o varios casos de prueba, que demuestren un flujo a lo largo de toda la aplicación para demostrar su funcionamiento completo.

#### Modificaciones

La estructura inicial de la última entrega de software pretendía ser igual a la de aquellas anteriores, pero la semana anterior al día de la revisión incorporó también casos de prueba para el software. Debido a que la exposición de historias de usuario y casos de prueba al mismo tiempo resultaría engorrosa, el ayudante coordinador sugirió separar la entrega en dos partes, lo que en consecuencia duplicó la duración de las presentaciones y agotó a los evaluadores.

### 3.4.7. Evento de Feria de Software

Consiste en un día completo dedicado a la exposición de los productos que cada equipo desarrolló, el cual involucra dos evaluaciones, una desde el grupo coordinador (profesores y ayudantes), enfocada en la apariencia y calidad de su exposición y el software, y otra desde un grupo de jurados invitados, enfocada en el producto y su potencial de innovación.

#### Criterios ocultos

Durante el evento se le notificó a los alumnos que la carta de aceptación del cliente requería de un formulario de evaluación y de una foto que respalde la presencia de este, debiendo además, presentarse físicamente dentro de un sobre a la coordinación. Los estudiantes debieron conseguir dichos materiales sin dejar de lado al público asistente, puesto que hacerlo implicaba una penalización.

#### Pauta de revisión

El 54 % de los elementos de la evaluación que realizó la coordinación consistieron en elementos relacionados a gastos monetarios como la presencia de regalos, decoraciones o la presencia de pendones y tarjetas de presentación. Esto causa que, en dicha evaluación, los equipos con mayores fondos tuvieran más oportunidades de sacar mejores calificaciones.

#### Valor Agregado

Actualmente se descuenta parte de todas las calificaciones si no se entrega un elemento adicional valorable por los profesores. Esto contradice el concepto de “agregado”, además de que resulta redundante en evaluaciones como la del equipo durante el evento, en donde se le exige a estos un valor agregado y además la “presencia de otros elementos de ambientación interesantes”.

### 3.4.8. Entrega del Documento de Legado

Luego del evento se les pide a los equipos un “documento de legado”, el cual resume su apreciación personal respecto de las lecciones aprendidas como equipo, las fortalezas y debilidades encontradas en las asignaturas, además de una evaluación respecto del desempeño de sus compañeros y una autoevaluación. Dicho documento se recibe a través de un formulario en *Google Forms*, redactándose uno por persona.

#### Sobrecarga a evaluadores

En versiones anteriores de la asignatura se entregaba un documento de legado por cada equipo. El 2018 se realizó uno por persona, resultando 75 informes a revisar, dificultando la revisión de feedback y resultando en opiniones repetidas entre compañeros de equipo.

La entrega de este documento y una última clase finalizan la estructura de la feria de software, la cual se resume en la figura 3.5.

### **3.5. Análisis a legados de generaciones anteriores (2014-2018)**

Dada la existencia de los legados, documentos que recopilan comentarios positivos y negativos hacia la asignatura, se logró recopilar aquellos de las últimas 5 generaciones (2014 hasta 2018), de los que se tomó en cuenta toda observación, crítica o sugerencia que se encontró en estos. Con este material se buscaron los temas más recurrentes por año, los cuales fueron encapsulados en las siguientes seis categorías, respecto de:

- Contenidos y estructura de la asignatura.
- Profesores y coordinación.
- Evaluaciones y pautas.
- Evento y premios de éste.
- Financiamiento de la asignatura.
- Sobrecarga académica.

Con esto, para cada año se procedió a resumir las observaciones entregadas a cada una de estas áreas y, además, a cuantificar cuántos equipos respecto del total en la asignatura presentaron observaciones en cada categoría. Con tales datos, se buscó respaldar el motivo del actual trabajo de titulación, el cual es la necesidad de una propuesta más acorde para el contexto de la presente estructura anual de la feria de software. Dicha información es en especificada en la sección 5.5 del anexo, analizándose los contenidos año a año. El resumen de sus contenidos se presenta a continuación.

### 3.5.1. Resumen de Resultados

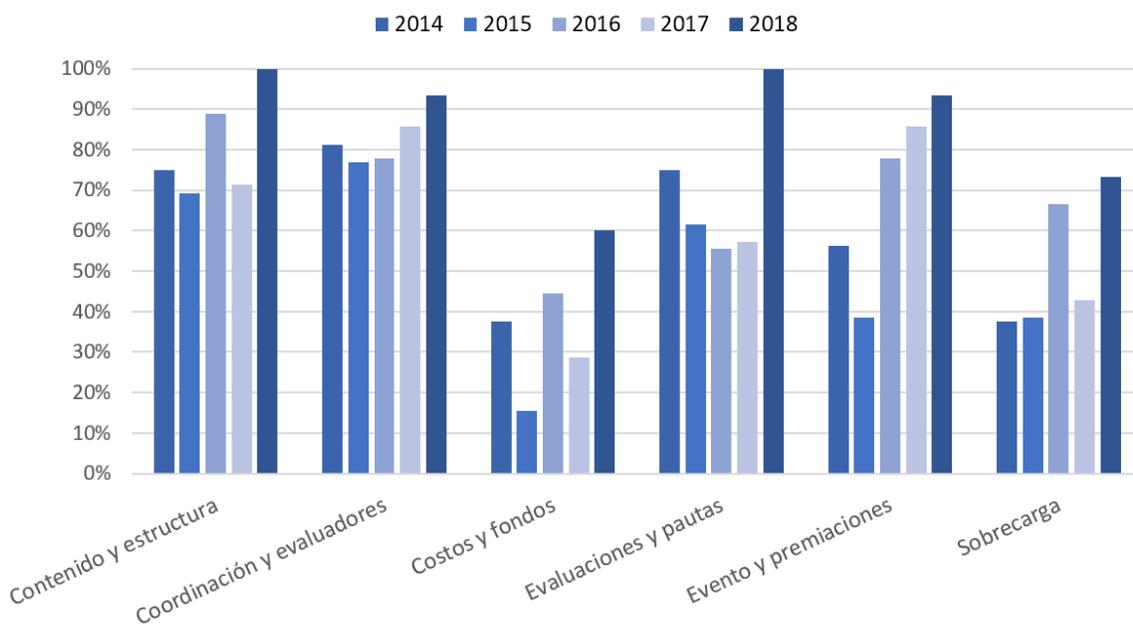


Figura 3.4: Porcentajes de equipos, de los 5 años de legados analizados, que realizaron críticas a la asignatura.

Se observó que las dos temáticas más criticadas son la mala coordinación y control que ejercen los profesores durante la asignatura y lo obsoleto o poco práctico de los contenidos que se ven a lo largo de esta misma, tal como se observa en la figura 3.4. A excepción de un caso, la mitad o más de los equipos estuvieron descontentos con la coordinación, los contenidos, el evento y las evaluaciones, solicitando algunos equipos en sus legados temas que consideraran necesarios y señalando estos que hay malas prácticas que no debiesen impartirse, lo que sucedió en más de un año. Es importante recalcar que, según los alumnos, la postura de los profesores se considera muy distante, lo que se considera un factor importante para lograr el buen desenvolvimiento en la asignatura y la buena formación académica de estos. Notar que, justamente, el año más reciente es donde el descontento es más alto.

Comparando dichas observaciones con lo analizado en el presente diagnóstico de cada entrega, se observa que muchas de estas problemáticas son debido a la presencia de redundancias en las evaluaciones, información poco clara, pautas redundantes o engorrosas, contenidos obsoletos y una coordinación poco estable, lo que se puede mejorar con una estandarización

de roles y procesos, además de hitos claros que permitan un control constante de avance y desempeño.

Al examinar la ejecución de las asignaturas asociada a la feria de software y al complementarla con la información otorgada en los legados se evidencia la necesidad de un replanteamiento en cómo se lleva a cabo la feria de software. Al querer incorporarse metodologías ágiles se obtuvo como resultado una estructura contradictoria, debido a que se pasan a llevar los valores y principios de este enfoque y se sigue trabajando y evaluando con un enfoque tradicional. Debido a esto, la planificación se vuelve rígida y, en consecuencia, las pautas y mecanismos de control en vez de aportar al desarrollo de los proyectos, dificultan su desempeño, no solo por parte de los alumnos, sino de la asignatura en general y de los resultados que esta busca lograr.

### 3.6. Conclusiones del Diagnóstico

Luego de investigar y observar el comportamiento dentro de las asignaturas, se concluye que el origen de los problemas observados se atribuyen a los siguientes factores, que a su vez están relacionados entre sí:

La coordinación decidió optar por incorporar agilidad, pero parte de ellos no estuvo de acuerdo con cambiar su forma de trabajar. En consecuencia se aplicó agilidad sólo en las actividades que realizan los alumnos, pero no en las evaluaciones y criterios solicitados ni en el cómo interactúan los involucrados. Dicha situación se conoce como “resistencia al cambio” [51], y coincide con el problema identificado por VersionOne en su encuesta [52], la cual identifica a la principal razón por la que falla el implementar agilidad como el que la cultura organizacional no sea capaz de adaptarse a los valores y principios ágiles. En consecuencia, en vez de asimilarse la lógica del agilismo como tal, sólo se agregaron nuevos elementos a las evaluaciones, conservando estas una estructura en cascada. Esto justifica la confusión y contradicciones señaladas por los alumnos. Además, las nuevas herramientas ágiles también se controlaron de forma tradicional: a través de documentación. Esto justifica la sobrecarga mencionada por los estudiantes y la existencia de puntos redundantes.

En los legados se señala que la coordinación de la asignatura presenta un modo de enseñanza “anticuado” y distante, en donde el profesor se enfoca en las evaluaciones y se desentiende del alumno. No se le da importancia suficiente a la interacción positiva y a la retroalimentación, evidenciado al observar la falta de orientación y apoyo señalado por los alumnos durante el periodo de licitación, contradiciendo la postura de agilidad de privilegiar personas y relaciones por sobre procesos y herramientas. Junto con esto, se nota una estructura jerárquica en los profesores, lo que no permite una colaboración entre ellos debido a la actitud impositiva de algunos.

Junto a lo anterior, no todos los docentes en la asignatura contaron con los mismos conocimientos de agilidad, lo que causó diferencias en el apoyo que recibió cada equipo según quién les ayudó. Una evidencia de esto se dio al presentar las historias de usuario, en donde los equipos fueron criticados por la mala calidad de estas. Esta situación justifica la “falta de apoyo” y las diferencias en los rendimientos de los equipos a lo largo del año.

En las asignaturas, a pesar de utilizarse herramientas y técnicas de agilidad, hasta la fecha no se ha podido llegar a una planificación estable. Esto, debido a que se buscó aplicar agilidad, pero no fue asimilado lo que esto implica, pasando a llevar los pilares y principios en los que se basa. Se asumió como suficiente llegar e implementar una serie de prácticas, cuando agilidad implica una transformación a nivel organizacional y cultural del que todos deben formar parte, cambiando el enfoque de negociar tiempos y costos al de negociar el alcance de los proyectos, y enfatizando a las personas y al software funcionando por sobre reportes y documentación innecesaria.

Los problemas identificados son consecuencia de un problema raíz en común: la cultura organizacional. Para llegar a una buena planificación se requiere de no solo incorporar técnicas ágiles, sino de formar parte del cambio y comprender lo que se busca lograr con el concepto de agilidad, considerando el enfoque que se requiere para que su ejecución sea exitosa. Es importante que todos los involucrados conozcan los conceptos expuestos en el manifiesto ágil y asimilen la lógica detrás de esta nueva forma de realizar desarrollo de software, fomentando la comunicación e interacción entre alumnos y profesores, factores que son la actual falencia de las asignaturas, y lo que causa problemas en la toma de decisiones en estas.

En el siguiente capítulo se toman en cuenta dichos problemas identificados para plantear una propuesta de trabajo que facilite la ejecución de la presente asignatura en un año académico general, generando una estructura de trabajo basada en las metodologías ágiles que se utilizan frecuentemente en la industria.

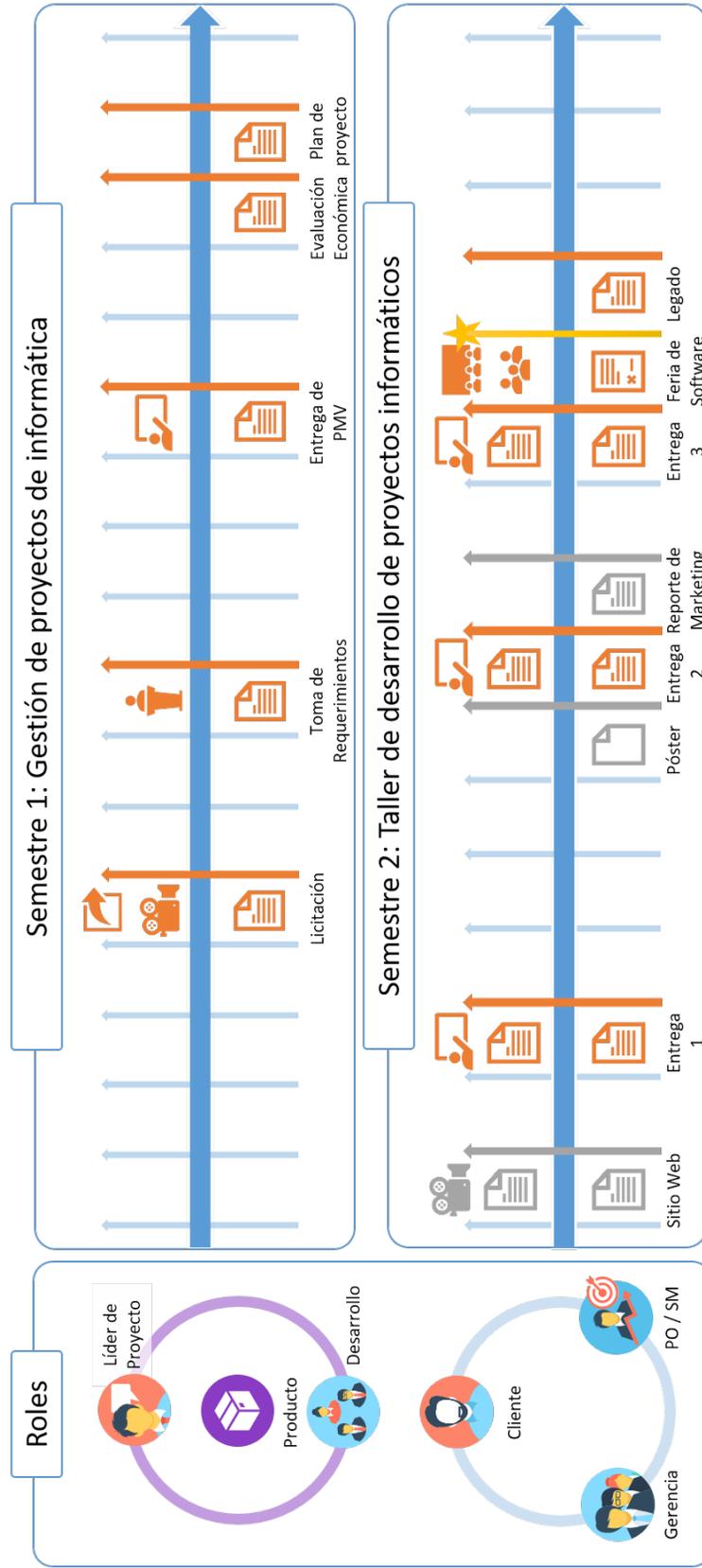


Figura 3.5: Diagrama resumen de la feria de software 2018, sus evaluaciones y roles.

## Capítulo 4

# PROPUESTA DE SOLUCIÓN

Luego de analizarse los elementos que se presentan en la actual planificación de la feria de software, se observa que a nivel general su estructura es válida, puesto que contiene un periodo de definición previa de la idea, desarrollo de un prototipo, elaboración del Producto y exposición del resultado final que facilitan el controlar el avance de los proyectos de cada equipo. El defecto encontrado yace en que, a pesar de tener la intención de aplicar agilidad, se ha caído en el error de querer incorporar un nuevo enfoque sin querer asimilar la mentalidad que este involucra, causando que en la práctica no se obtengan los resultados deseados.

Se señala, entonces, que no basta con aplicar una serie de prácticas a una asignatura para que esta pase de desarrollo tradicional a agilidad, puesto que esto involucra incorporar una filosofía diferente de trabajo que debe ser asimilada tanto por los desarrolladores como por su entorno, siendo este último aspecto el más difícil de lograr. En pocas palabras, y aplicándolo a nuestro contexto, no es lo mismo decirle a los alumnos que sean ágiles, al tener que cambiar la manera en que se evalúa y coordina una asignatura en pos de lograr la agilidad.

Para aclarar dicha situación, a continuación se presenta una descripción de las diferencias entre el enfoque tradicional y ágil, con el objetivo de que cualquier persona que quisiese incorporar agilidad a un contexto de proyectos entienda lo que esto involucra. Entendiéndose dicha diferencia, se dará paso a la estructura general que la presente propuesta ofrece para dar un acercamiento a una asignatura basada en agilidad, seguido de un detalle de las actividades

involucradas en la ejecución de esta y adjuntándose un detalle de las pautas de evaluación sugeridas para estas mismas en en anexo, secciones 5.7 y 5.8.

## 4.1. Tradicional vs. Agilidad

Para comprender rápidamente la diferencia entre la forma tradicional de trabajar y la que se propone en agilidad, se deben de comprender las diferencias entre las mentalidades que involucran ambas posturas. Dichas posturas presentan las mismas características de las mentalidades fija y de crecimiento señaladas en [53], comparándose en la tabla 4.1.

Estas son las características principales que se mencionan al comparar ambas formas de trabajo (desde el punto de vista de la agilidad), en donde el lado ágil se complementa con los 4 valores y 12 principios del manifiesto ágil señalados en la sección 2.2.5.

Mentalidad Tradicional (fija)	Mentalidad Ágil (de crecimiento)
Aprender pensando, evitar el error	Aprender haciendo, aprender del error
Trabajar bajo roles, donde cada uno tiene una labor determinada	Pensar como equipo, con labores y responsabilidades compartidas
Evitar el fracaso, no cambiar lo que ya funciona	Aceptar los cambios, abrazar los desafíos
Fijar la planificación desde el comienzo	Planificar dinámicamente, adaptándose mientras el trabajo progresa
Entregar todo al primer intento	Realizar entregas rápidas, pequeñas y lo más temprano posible
Quedarse con lo que uno ya domina	Nunca dejar de aprender y progresar
Controlar mediante métricas e indicadores el progreso del personal	Confiar en que la gente sabe cómo hacer su trabajo
Apuntar a maximizar las ganancias	Apuntar al valor del usuario
Respetar el conducto regular, comunicarse mediante correos u otros medios de manera formal	Escuchar al otro. Fomentar la interacción cara a cara. Incorporar a los demás, dejar que participen

La persona a cargo toma las decisiones, puesto que es la persona más capacitada para hacerlo	No hay alguien a cargo, sólo roles que colaboran entre sí respaldándose mutuamente
Mantener todo documentado y detallado, registrando cada suceso y planificación	Mantener las cosas lo más simple posible y registrar sólo lo importante y esencial

Tabla 4.1: Comparación entre las mentalidades fija y de crecimiento.

Para aplicar esta filosofía se deben permitir los errores y aceptar los cambios, lo que en un ambiente académico es difícil de controlar. El fomentar la sencillez en agilismo va en contra de la documentación que usualmente se pide, ya que esta se usa como registro del avance y participación de los alumnos. Al momento de considerar agilidad, una importante cantidad de barreras se deben sobrepasar para planear una asignatura. A pesar de esto, dichas consideraciones no son un gran impedimento, puesto que es posible generar instancias de control que se rijan por dichas condiciones.

La diferencia fundamental entre lo tradicional y el agilismo, en el contexto de gestión, radica en el llamado “triángulo de hierro de la administración de proyectos” (*The iron triangle of project management [45]*), el cual define tres conceptos principales que se deben considerar al momento de gestionar un proyecto:

- **Los costos** que requerirá el proyecto.
- **El tiempo** que necesitará para completarse.
- **El alcance** que tendrá el producto resultante.

El desarrollo tradicional se considera un proceso predictivo puesto que, inicialmente, el cliente fija el alcance que requiere que cumpla el proyecto para, a continuación, que el equipo estime cuánto será el costo asociado a este y el tiempo que requerirá. La agilidad, por otro lado, se rige bajo un proceso adaptativo, en donde el cliente define inicialmente lo que está dispuesto a pagar y el tiempo que dispone para el proyecto, para que el equipo estime el alcance que lograría cumplir del proyecto, tal como lo muestra la figura 4.1.

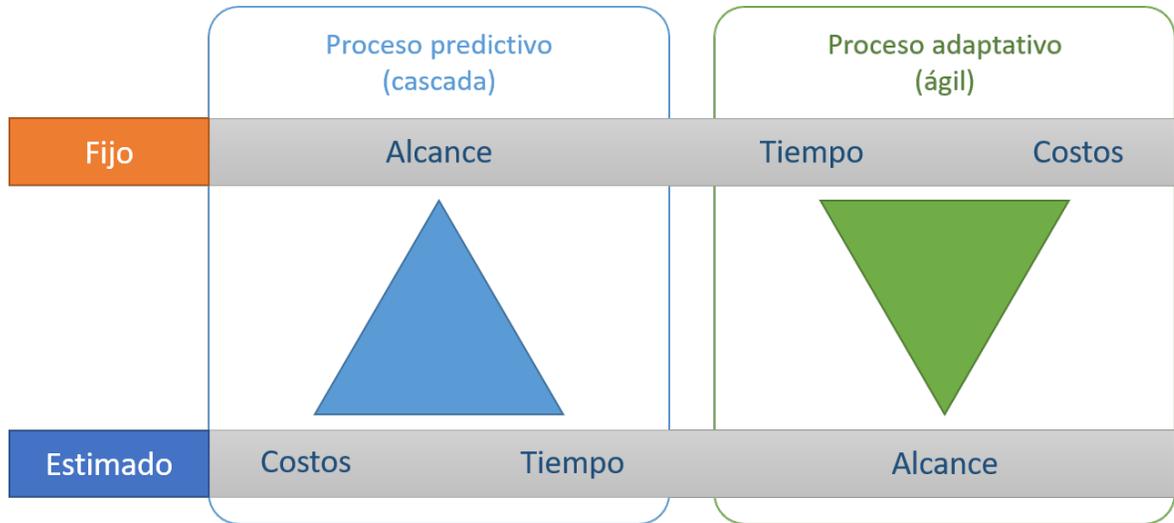


Figura 4.1: Comparación de enfoques entre proceso predictivo y adaptativo.

No considerar esta diferencia al implementar agilidad en una organización causa que la mentalidad siga siendo tradicional, no lográndose buenos resultados a pesar de respetar las estructuras y prácticas que definen los métodos ágiles. En consecuencia, para aplicar la presente propuesta a un entorno académico es necesario considerar que se requiere cambiar el enfoque de la asignatura, fijando desde el inicio los fondos disponibles y el tiempo para cada proyecto y, lo más importante, dando instancias para que los alumnos puedan negociar el alcance de su proyecto constantemente con la coordinación. Notar que la duración de cada semestre es fijada desde el comienzo junto con los fondos disponibles para cada equipo pero, debido a la baja experiencia de los equipos, no se conoce el alcance de cada proyecto, volviéndose este difícil de fijar o planificar.

La presente memoria, entonces, se plantea bajo el supuesto de que, tanto los alumnos como la coordinación, están dispuestos a cambiar el enfoque de la asignatura para respetar el enfoque ágil, puesto que no basta con seguir al pie de la letra las prácticas que se definen en esta propuesta y hacer que los estudiantes las apliquen, requiriéndose cooperación y un apoyo mutuo.

## 4.2. Estructura de la Propuesta

### 4.2.1. El *Framework* Scrum

Según contenidos de asignaturas previas a la feria de software, los alumnos cuentan con conocimientos básicos de Scrum. Siendo este el modelo más popular en Chile, se decide usar dicho *framework* como base para la gestión de proyectos. A pesar de esto, debido al contexto educacional no es posible su implementación al pie de la letra, pero se complementa su ejecución con otras herramientas y técnicas que facilitan la fluidez de las asignaturas, buscando un acercamiento inicial a la lógica de los pilares y principios del manifiesto ágil [27]. Como referencia se deja al libro Scrum Body of Knowledge [54] en las asignaturas, debido a su similitud con el texto anterior.

#### Aclaración

Scrum es una estructura de trabajo, la cual exige que se cumplan sus hitos al pie de la letra. En consecuencia, si una metodología no respeta dichos hitos o normas en su totalidad, no cumple con este *framework*. Esta es la razón de que la presente propuesta se defina como “**basada** en Scrum”.

### 4.2.2. Contexto de la asignatura

Los estudiantes señalan en sus legados que no encuentran de utilidad las labores de gestión y la redacción de documentos en la asignatura, puesto que según ellos, su labor en el campo laboral sólo involucra el desarrollo de un software a nivel técnico. Esto en la vida real no es así, puesto que la carrera en la que se ven involucrados dichos estudiantes es la de Ingeniero civil informático, en donde el área de “informático” considera la parte técnica del desarrollo, y el área de “civil” la de gestión de proyectos, considerando todo lo que esto abarca.

Se sugiere en la presente propuesta que tanto alumnos como ayudantes y profesores entiendan la asignatura bajo el siguiente contexto:

*La asignatura representa a una gran empresa llamada FESW Ltda, en donde los alumnos actúan como micro-empresas (los equipos) que postulan para financiar el desarrollo de sus proyectos de innovación. Luego de aceptados, estos pasan a un periodo de desarrollo iterativo incremental en donde, en paralelo, van publicitando el producto. Finalmente, el ciclo culmina en una exposición abierta de los resultados durante un evento que busca encontrar posibles interesados, en pos de lograr un posible emprendimiento.*

### **4.2.3. Participantes y Roles**

Considerando el anterior contexto, los roles que cada miembro de la asignatura deberá cumplir en la organización serán:

- **Stakeholder:** Rol de los profesores y ayudantes durante las entregas, en donde actúan como **la directiva** de la empresa FESW. Evalúan el avance del proyecto y los resultados de cada Sprint en los *Sprint Review*.
- **Coach:** Rol de los profesores durante clases, en donde orientan y apoyan a sus alumnos para que estos puedan sacar adelante sus proyectos.
- **Team Lead:** Rol de los ayudantes asignados a cada equipo durante los Sprint, puesto que interactúan con estos apoyándolos, controlando su avance y eliminando sus impedimentos.
- **Equipo:** Representan a una **empresa postulante** que busca que su proyecto sea aceptado por los stakeholders. Luego de verse aceptados, son contratados como un **equipo Scrum**, el cual debe cumplir todo lo que sea necesario para llevar a la vida la visión del Producto.
- **Scrum Master:** Antiguo “líder de equipo”. Desarrollador que dedica parte de su esfuerzo a la gestión del equipo, siguiendo el progreso de sus compañeros y actuando como intermediario entre la directiva y los desarrolladores. Representa a un **líder coach**.
- **Product Owner:** Segundo líder en el equipo de alumnos, el cual es responsable de contactarse con el cliente piloto o usuario representativo y de cuidar la buena definición

del producto, orientando este a otorgar valor a dicho cliente y cuidando que las épicas e historias de usuario puedan definirse y cumplirse apropiadamente. Representa a un **líder técnico**.

- Dependiendo de si la idea surge desde los alumnos o si es entregada por alguien externo, el antiguo “cliente” puede asumir dos posibles cargos:
  - **Cliente Piloto:** Aporta su idea de proyecto a la asignatura, la cual es aceptada como aporte voluntario por un equipo. Puede optar a beneficios o descuentos una vez desarrollado el producto, pero no tiene permitido adueñarse de la idea.
  - **Usuario Representativo:** Representante externo del segmento objetivo de usuarios o clientes de la idea de proyecto, dispuesto a orientar al equipo en base a su experiencia en el contexto del proyecto.

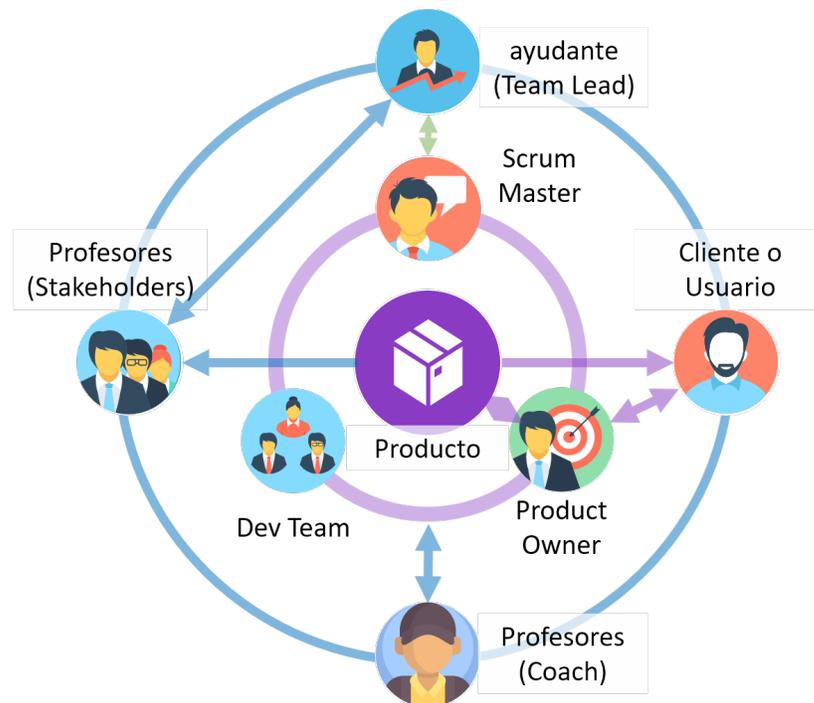


Figura 4.2: Descripción gráfica de los roles de la propuesta y su interacción.

#### Team Lead vs. Scrum Master

El cargo de Team Lead en pequeños equipos ágiles genéricos es el mismo que el de Scrum Master en Scrum [55], la diferencia de nombres es sólo para diferenciar al ayudante del estudiante.

#### Emprendimiento

No se le permite al “cliente” cobrar un porcentaje de ingresos del proyecto o adueñarse del software resultante una vez terminado este, puesto que el objetivo de la asignatura es dar paso a un posible emprendimiento en los alumnos. He aquí el motivo del término “aporte voluntario”. Se debe evitar que los equipos sean considerados una consultoría de bajo costo o que se vean obligados a entregar su software una vez terminado.

### 4.3. Evaluaciones y Criterios

Considerando 18 semanas por semestre, la actual propuesta considera 5 instancias de evaluación para la primera asignatura y 8 instancias de evaluación para la segunda. A pesar de cumplir una finalidad similar a las que se imparten actualmente, estas se diferencian en los criterios que utilizan para cumplir sus objetivos, cambiando considerablemente sus pautas en pos de facilitar la fluidez de la asignatura, las que se presentan en las secciones 5.7 y 5.8 del anexo.

Debido a que los reportes o informes que se entregan simulan ser una empresa, se considerará para cada documento el descontar 5 puntos en caso de no borrar los textos guía incluidos en cada uno de ellos debido a que no corresponde que estos aparezcan en un informe de este contexto (entiéndase con esto aquellos textos como “colocar aquí un diagrama de (...)” incluidos en las pautas).

#### Modo de evaluación

Las evaluaciones consistirán en una serie de criterios a cumplir, evaluándose en base a una escala Likert [56] con valores de 0 a 4 de la siguiente forma:

- 0: No se entrega lo solicitado.
- 1: Se demuestra que hubo trabajo pero no funciona.
- 2: Funciona, pero no cumple con todas las expectativas.
- 3: Se entrega todo, pero tiene leves observaciones.
- 4: Lo entregado cumple y funciona como se esperaba.

Elementos fundamentales poseerán doble o triple puntaje (representado con  $x2$  o  $x3$ ), siendo la suma total de los valores de cada pregunta el puntaje máximo, ajustándose este posteriormente a la escala de evaluación necesaria.

#### Valor agregado

En el caso de valores agregados, estos se considerarán como una mención adicional en las presentaciones presenciales, pudiendo cada equipo obtener puntos adicionales al puntaje máximo.

### 4.3.1. Etapas Principales

La propuesta considera diez etapas, tal como se muestra en la figura 4.3. Estas se agrupan en las categorías de *Postulación de Proyectos*, *Realización del Prototipo*, *Desarrollo del Producto y Exposición* y *Cierre de Proyectos*, cada una de ellas posee sus propios hitos y evaluaciones.

- **Postulación de Proyectos:**

- 1) Llamado a postulación.
- 2) Definición inicial del proyecto.
- 3) Defensa de la idea de proyecto.

- **Realización del Prototipo:**

- 4) Desarrollo de un prototipo inicial.
- 5) Redacción de Planificación Tentativa.

■ **Desarrollo del Producto:**

- 6) Sprint 1: Desarrollo inicial.
- 7) Sprint 2: Desarrollo considerando feedback anterior.
- 8) Sprint 3: Incluye revisión completa junto al cliente.

■ **Exposición y Cierre de Proyectos:**

- 9) Realización del evento Feria de Software.
- 10) Periodo de cierre de proyectos.



Figura 4.3: Representación visual de las 10 etapas de la presenta propuesta.

La diferencia de la propuesta a lo que se utiliza en la actualidad, a nivel general, es que se cambia el detalle de las evaluaciones y se incorpora un periodo de cierre de proyectos, el cual consiste en una ceremonia que servirá como instancia para que el feedback de los alumnos se le haga llegar a los profesores y ayudantes de forma más efectiva, siendo el mayor cambio el cómo se realiza cada uno de estos hitos y las pautas que estos siguen, tema que se detalla a continuación.

## 4.4. Estructura Detallada: Primer Semestre

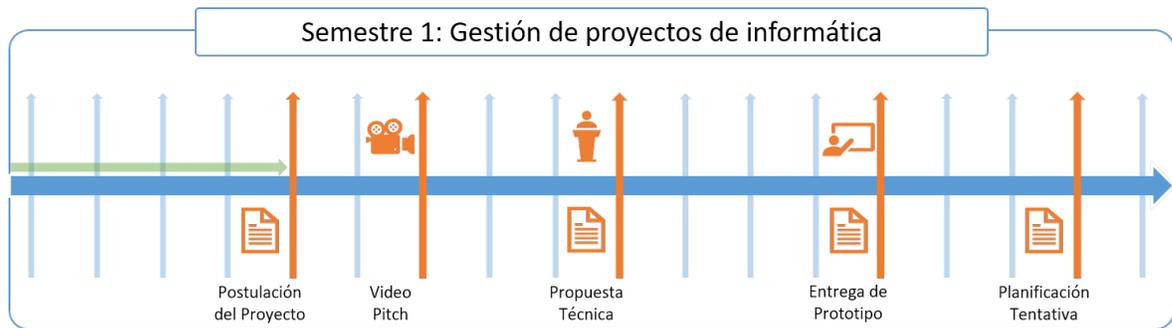


Figura 4.4: Representación visual de la propuesta de primer semestre.

Para el primer semestre se consideran dos periodos principales: Postulación de Proyectos y Realización de Prototipo. Debido a que se introduce a todos los miembros a un contexto específico y que todos estos poseen roles y responsabilidades diferentes, **se les hará entrega de un documento inicial tanto a profesores, alumnos y ayudantes**. Esta medida busca que todos los miembros tengan a su disposición un documento breve dedicado a ellos, en donde se definan sus roles, responsabilidades y exigencias, de forma de que estos tengan claridad respecto de en qué consiste la asignatura y, a su vez, fomentando el principio de transparencia que fomenta la agilidad, permitiendo, además, que estos puedan prepararse y organizarse anticipadamente.

### Postulación de Proyectos

Consiste en el periodo compuesto desde el inicio de la asignatura hasta la definición inicial de la idea de cada proyecto. Debido a esto es que este periodo se separa en tres etapas: Postulación, definición y defensa.

#### ■ Llamado a postulación:

- **Objetivo:** Que cada equipo logre formular una idea de proyecto de emprendimiento innovador aceptable por los profesores.
- **Herramientas:** Product Vision Board, Video Pitch del Proyecto.

- **Evaluaciones:** Documento de postulación de proyectos, Evaluación externa del video.

Los alumnos forman equipos y, semanalmente, deberán presentar ante los profesores un *Product Vision Board* (PVB) [57], el cual definirá la idea de un posible proyecto para que esta reciba la aprobación, observaciones o rechazo del profesor o profesores encargados. Esta herramienta busca fomentar la interacción entre profesores y alumnos, llegando los equipos a la siguiente etapa con una idea potencialmente aceptable. La estructura de dicho PVB se muestra en la figura 4.5, siendo la utilidad de esta herramienta que su información es útil para posteriores entregas.

El PVB debiese ser realizado de manera física por medio de notas adhesivas, de forma de que sea fácil incorporar o eliminar factores del Producto ante alguna observación del profesor, pero se aceptan alternativas virtuales.

Nota

Si el equipo tiene más de una idea de proyecto puede ir planteándoselas al profesor en más de un PVB. Un ejemplo de esta herramienta se encuentra en [58].

El presente hito culmina con la presentación de un **documento de postulación de proyectos**, el cual define inicialmente la idea que pretende desarrollar cada empresa. Dicho documento es entregado a los profesores para evaluar si pasan a la siguiente etapa. Caso contrario, se les dará un periodo corto de re-corrección en donde puedan buscar o redefinir su idea, teniendo como nota máxima el valor de aquel informe dentro de los aprobados cuya calificación sea la más baja.

Seguido de esto se debe confeccionar un **video en formato pitch** de la idea del proyecto, el cual es evaluado por un jurado externo a la asignatura. Esto, con el fin de recibir segundas opiniones respecto de las ideas que estarán elaborando los alumnos en la asignatura desde un punto de vista diferente al académico.

Terminados ambos hitos anteriores, se da por aprobada la idea del proyecto, dando paso al periodo de definición técnica, en donde se requiere que el equipo especifique más el proyecto

 <b>Nombre del Producto</b>	 <b>Frase Representativa (Visión del Producto)</b>		
Nombre del Software que buscan desarrollar.	Frase corta y precisa que resuma qué es lo que aspira a ser el producto.		
 <b>Segmento Objetivo</b>	 <b>Necesidades</b>	 <b>Funcionalidades</b>	 <b>Beneficios</b>
¿Quiénes utilizarán el producto? (usuarios). ¿A quiénes va dirigido el producto? (clientes).	¿Qué necesidades del grupo de usuarios satisface el producto?	¿Qué características del producto satisfacen las necesidades de los usuarios?	¿Qué valor agrega el producto a lo que ya existe en la industria? (Resaltar factores de innovación)

Figura 4.5: Product Vision Board a utilizar en la licitación de ideas.

y su ejecución.

■ **Definición inicial del Proyecto:**

- **Objetivo:** Definir el qué busca solucionar el Producto y cómo lo hará, especificando sus historias de usuario y un prototipo.
- **Herramientas:** Product Vision Board, Historias de Usuario, Planning Póker.
- **Evaluaciones:** Documento de Propuesta Técnica, Presentación de Propuesta.

En la presente etapa se especifica la definición técnica del proyecto, dando un primer acercamiento a las historias de usuario que este abarcará y cuáles de estas son aptas para un prototipo inicial. Todas estas historias deben ser priorizadas según el valor que entregan para su cliente o usuario, siendo estimadas aquellas que correspondan al prototipo.

### Definición de Prototipo

Se considera como historias aptas para un prototipo a aquellas que le aporten alto valor al cliente o usuario, pero que sean de menor dificultad respecto del resto de igual importancia. Cualquier historia que aporte suficiente valor para el usuario es apta para un prototipo inicial y, de no ser suficientemente, pueden seleccionarse más de una, complementándose estas entre sí o aportando valor de manera independiente.

La propuesta técnica busca con sus contenidos que los alumnos asimilen el feedback recibido en instancias anteriores para mejorar la idea inicial, realizando una exposición presencial de esta y la propuesta de un **prototipo** como acercamiento inicial al software resultante.

### ¿Por qué no usar un Producto Mínimo Viable?

Un PMV requiere que el desarrollo expuesto esté en producción o listo para estarlo. Al presentarse, ya debería poder ser usado a través de un sitio Web o de algún servicio [59]. Esto en la actualidad no se realiza en la asignatura, presentándose el prototipo a nivel interno. El planteamiento de un PMV como tal se deja como trabajo a futuro en la sección 5.3.

Tomando en cuenta que los alumnos deben definir historias de usuario en esta etapa de la asignatura, se sugiere que estos sigan las siguientes indicaciones:

1. En base a la columna de funcionalidades del PVB, formular una lista de posibles historias de usuario siguiendo la idea descrita en la figura 4.6.
2. Verificar que cumplan con el criterio INVEST [60].
3. Para cada historia resultante, estudiar si es posible realizarla en menos de dos semanas (tomando en cuenta el tiempo disponible del equipo). Si la respuesta es si, aceptar la historia, caso contrario, se considera una épica y se descompone en historias más específicas.
4. Tomar las historias de usuario y validarlas con un profesor o un ayudante.
5. Priorizar las historias, asignándoles las letras I, E, D, O, según sean las más Importantes, Esenciales, Deseables u Opcionales, incluyendo en I las de innovación.
6. Entre las Importantes y Esenciales, seleccionar aquellas que se consideran aptas para

un prototipo y:

- Estimarlas, a través de planning poker [61] junto a su ayudante usando una escala lineal de puntos de historia (considerar 1 punto como 5 horas de trabajo).
- Definir los criterios de aceptación de estas.

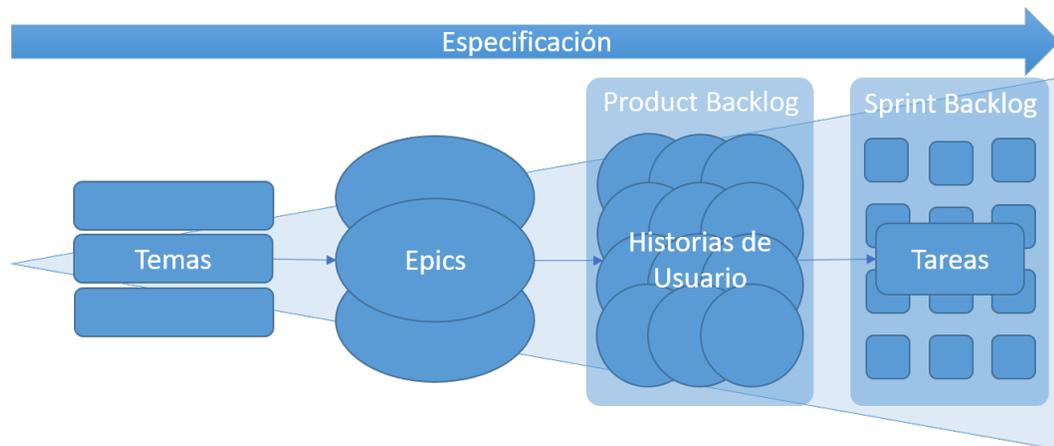


Figura 4.6: Descomposición de elementos, desde temas hasta tareas, en donde éstas últimas se definen según las historias a realizar en el actual Sprint.

#### Consideraciones y sugerencias al estimar

Al momento de estimar, es fundamental que los alumnos tengan claro que cada historia abarca tanto diseño como desarrollo e investigación previa. El estimar considera todo lo necesario para poder entregar la historia como terminada. Además, los equipos deben utilizar las mismas escalas de estimación (lineal), puesto que esto permite posteriores estudios de productividad.

La información de las historias de usuario es recopilada junto con información del proyecto en un **Documento de Propuesta Técnica**, siendo entregado como registro de lo que se planea realizar en el proyecto. En paralelo a esto, se realiza una presentación frente a los evaluadores de la asignatura, con el fin de simular la propuesta de dicho proyecto ante los representantes de la empresa FESW.

De aceptarse dicha propuesta este pasa a la siguiente etapa. Caso contrario, se da un periodo de reformulación que sigue los mismos términos del documento de postulación de proyectos.

Dicha aceptación simboliza la autorización de los proyectos, dando luz verde al inicio de su desarrollo para el siguiente periodo.

#### ¿Cómo regular el alcance de cada proyecto?

Una modalidad ágil no fija el alcance. Lo que se sugiere en esta propuesta es que los alumnos desarrollen en orden descendente según la priorización de sus historias, cumpliendo de inmediato con aquellas necesarias para cumplir con la idea objetivo a través de las historias importantes y esenciales. Si faltan historias a lo largo del desarrollo, deberán proponerse adicionales, en caso contrario, ya que las relevantes se desarrollaron al inicio, se pueden negociar las restantes, ya sea para modificarlas o dejarlas de lado, puesto que estas en su mayoría serían opcionales.

### Realización del Prototipo

#### ■ Desarrollo de un prototipo inicial:

- **Objetivo:** Elaborar un prototipo del Producto a desarrollar, además de dar experiencia a los alumnos acerca de las herramientas técnicas a utilizar.
- **Herramientas:** Product Backlog, Sprint Backlog, Diagrama de Flujo Acumulado.
- **Evaluaciones:** Sprint Review, Sprint Retrospective.

Luego de aprobada la idea de proyecto y su prototipo cada empresa pasa a ser un *Scrum Team*, comenzando el desarrollo de software en Sprints controlados por el Team Lead en reuniones semanales de avance. En estas, se debe presentar un diagrama de flujo acumulado, controlándose el progreso del Sprint Backlog [62]. Las etapas de los Sprint se detallan en la sección 4.5.

Este periodo finaliza con la presentación y defensa presencial del prototipo ante los *stakeholders* en un **Sprint Review**, el cual es complementado con un posterior **Sprint Retrospective** de manera interna entre los miembros del equipo, en donde se sugiere que el ayudante participe para verificar que este se efectúe de manera correcta.

#### Orientación complementaria

Se sugiere incorporar clases acerca de herramientas y técnicas para fomentar buenas prácticas de desarrollo, como el uso de reuniones virtuales, comunicaciones a través de [Slack](#) u otro producto similar, además de estándares de control de versiones y el uso de Product Backlog y Sprint Backlog.

#### Retrospective de Profesores

El equipo de profesores también debiese realizar una retrospectiva por Sprint. Esto ayuda a identificar posibles problemas recurrentes y a mantener a los equipos coordinados, especialmente considerando una asignatura impartida en más de un campus a la vez.

#### ■ Redacción de Planificación Tentativa:

- **Objetivo:** Definir completamente al proyecto y bosquejar su planificación para el siguiente semestre.
- **Herramientas:** Lean Canvas, Historias de usuario, Planning Póker, Mapa simplificado de Producto.
- **Evaluaciones:** Informe de Planificación Tentativa.

Luego de aprobado el prototipo, los equipos reciben la retroalimentación de las pasadas entregas para realizar un último **Informe de Planificación Tentativa**, el cual establece definitivamente la idea del proyecto y bosqueja una distribución del trabajo a lo largo de los siguientes 3 Sprints, distribuyendo en estos las historias en orden descendente desde las de innovación hasta las opcionales en un mapa simplificado del producto (descrito en el anexo [5.6.1](#)). En el caso de las opcionales, estas pueden asignarse parcialmente o no asignarse, según la cantidad de historias que se alcancen a abarcar, para posteriormente negociarse en caso de requerir un ajuste.

El documento de planificación tentativa solicita estimar todas las historias de usuario, basándose en la experiencia adquirida durante el desarrollo del prototipo. Esto, posteriormente, da paso a la formulación del Product Backlog del segundo semestre.

### ¿Por qué planificación “tentativa”?

El nombre se debe a que el alcance en agilidad es adaptable y negociable, por lo que no se puede interpretar dicho documento como un compromiso. Es importante dar instancias de negociación en donde los alumnos puedan re-estructurar el alcance y desarrollo de sus proyectos, lo que se busca transmitir en el nombre del documento.

## 4.5. Estructura Detallada: Segundo Semestre



Figura 4.7: Representación visual de la propuesta de segundo semestre.

El segundo semestre considera dos periodos: desarrollo del Producto, que consta de 3 Sprints, y cierre de la asignatura, que considera el evento y las ceremonias de cierre. Cada Sprint de desarrollo considera, además, una actividad adicional de *marketing* que se desarrolla en paralelo.

### Desarrollo del Producto

Para evitar malos entendidos respecto del ciclo de desarrollo de software en Scrum, se mantendrán los nombres de las ceremonias de este modelo en inglés, las cuales involucran cuatro etapas:

- ***Sprint Planning***: Reunión inicial donde, en conjunto con el ayudante, se descomponen las historias de usuario del Sprint en tareas, registrándose en el Sprint Backlog y siendo posteriormente asignadas a cada integrante a medida que se van realizando. En el caso de quedar una historia pendiente de un Sprint anterior o de surgir una no considerada, esta también es incorporada al Sprint Backlog, siendo opcional el realizar un *Product Backlog Refinement* [63].
- **Desarrollo**: Periodo en que cada integrante realiza sus tareas asignadas. En reuniones semanales se le reporta al Scrum Master y al ayudante de cada equipo como en un “Daily Scrum” de mayor duración, siendo el ayudante responsable de controlar el

avance. Además, periódicamente se debe registrar el avance del Sprint Backlog mediante un diagrama de flujo acumulado [62].

- ***Sprint Review***: Revisión presencial de lo desarrollado en el Sprint. Cada equipo presenta sus avances ante los *stakeholders*<sup>1</sup> dentro de un tiempo determinado, mostrándose las historias comprometidas y los resultados obtenidos. Luego, se da un periodo de defensa en donde se recibe la retroalimentación de los evaluadores, a las que el equipo puede apelar. En Scrum el expositor es el Scrum Master, pero para fomentar la participación se sugiere ir rotando expositores. Además, considerar la participación del cliente o usuario para fomentar su interacción con el proyecto.
- ***Sprint Retrospective***: Reunión de cierre que analiza los factores tanto positivos como negativos del equipo a lo largo del Sprint. Dicha reunión también debe realizarse por parte de los profesores para evaluar el desempeño actual en la asignatura y prevenir posibles conflictos. Con este hito se da por finalizado el Sprint y se pasa al siguiente, comenzando otro ciclo.

### Actividades complementarias

Dentro de cada Sprint se consideran actividades adicionales, las cuales se deben realizar en paralelo al ciclo de desarrollo:

- **Sprint 1**: Considera la entrega de un **Sitio Web** promocional que describa al Producto y al equipo, además de crear dos redes sociales a modo de contacto con el equipo de desarrolladores, las cuales deben mencionarse en dicho sitio y deben tener publicaciones periódicamente.
- **Sprint 2**: Requiere de un **Video Comercial**, el cual debe mostrar al producto siendo usado y al equipo, siendo este video incorporado a la página oficial de la Feria de software.
- **Sprint 3**: Considera un **reporte de marketing** con un análisis a los avances obtenidos en las redes sociales del proyecto. Dicho documento consulta además si cada equipo tiene planeado realizar algún tipo de concurso durante el día del evento, para así

---

<sup>1</sup>Debido a la diferencia de experiencia, se sugiere que las calificaciones de los ayudantes sean de menor ponderación.

controlar la viabilidad de estos.

Semanas antes del evento final se realiza un **periodo de puesta en marcha**, en donde cada equipo expone su producto ante un público externo a la asignatura. Dicha iniciativa no es evaluada, pues busca que los alumnos practiquen su discurso para el evento y, a su vez, que se le extienda una invitación a personas que pudieran verse interesadas en dicho evento.

### **Consideraciones para el desarrollo**

- **Reuniones semanales:** Estas poseen como finalidad reportar el estado de las tareas que realiza cada integrante, indicando las complicaciones que estos pudieran tener. Es esencial que los alumnos entiendan la finalidad de dicha reunión, ya que no es una instancia para juntarse a desarrollar. Se señalan reuniones semanales y no diarias con el ayudante debido a que el contexto académico no lo facilita. A pesar de esto, se sugiere fomentar que los alumnos tengan reuniones virtuales, ya sea diarias o cada *X* días.
- **Proactividad en los equipos:** Durante el desarrollo es importante la presencia constante del ayudante, ya sea a modo presencial o virtual. Esto, debido a que con su experiencia puede ser capaz de reconocer problemas de forma anticipada. También se sugiere mostrar avances constantemente a los profesores, de forma de recibir sus observaciones de manera oportuna y no en forma de descuentos durante las evaluaciones.
- **Errores Recurrentes en Sprint Review:** Durante cada Sprint Review el resto de los equipos debiese estar presente, puesto que los errores que cometen algunos equipos los cometen otros en el siguiente Sprint. Con esta medida se pueden evitar problemáticas recurrentes, aprendiendo los estudiantes de las presentaciones de sus compañeros.

## Cierre de la Asignatura

Abarca desde la exposición final de los proyectos en el evento de la Feria de Software, hasta el cierre de estos y del ciclo académico. Durante las exposiciones de dicho evento se realizan 3 actividades en paralelo:

- **Evaluación de coordinadores:** Realizada por profesores y ayudantes, enfocada en el producto desarrollado y su presentación durante el evento. El total de calificaciones de la asignatura hasta este punto dan origen al *Premio por desempeño académico*.
- **Evaluación del Jurado y del cliente:** Realizada por un grupo de jurados invitados, quienes califican al producto y su potencial en la industria considerando su propuesta de innovación. En el caso del cliente, este debe presentarse ante los profesores para verificar su asistencia al evento. El proyecto mejor evaluado por el jurado recibe el *Premio al emprendimiento innovador*, el cual considera **primer, segundo y tercer lugar**.
- **Evaluación del Público:** Encuesta realizada a los asistentes del evento en donde estos escogen los 3 productos que consideran más útiles mediante un sistema de puntajes. La idea más votada recibe el *Premio al proyecto por votación del Público*, el cual considera **primer, segundo y tercer lugar**.

Finalizado el evento se da paso al periodo de finalización de la asignatura, en donde se emiten dos documentos por parte de los equipos y se realiza una última clase a modo de cierre.

El primer documento consiste en un **documento legado**, donde los equipos señalan la experiencia adquirida a lo largo de la asignatura y detallan consejos a futuros alumnos de la asignatura. El segundo documento consiste en un **documento de feedback académico**, donde cada equipo entrega su retroalimentación respecto de ambas asignaturas, otorgando ideas de posibles mejoras o aspectos deficientes de su ejecución.

El **documento de feedback académico** solicita información basada en los 6 temas recurrentes de los documentos legado de años anteriores:

- Los contenidos y la estructura de la asignatura.

- Los profesores y la coordinación.
- Las evaluaciones y las pautas.
- El evento y los premios de éste.
- El financiamiento en la asignatura.
- La sobrecarga académica.

Dicho documento **no es evaluado**, puesto que esto influencia la opinión de los alumnos, evitando la transparencia que fomenta el agilismo. Al segmentarse por temas, se puede apreciar rápidamente los elementos más criticados, facilitando un rápido análisis del feedback de cada año.

La última clase sigue la estructura de una retrospectiva, involucrando tanto a profesores como ayudantes y alumnos. Esta instancia busca fomentar el diálogo entre todos los miembros de la asignatura, considerando los resultados obtenidos en los últimos dos documentos en un ambiente de respeto y transparencia. Se sugiere estructurar dicha clase en base a los 6 temas mencionados anteriormente.

Esta clase finaliza el ciclo académico de la asignatura, pudiendo incorporarse el feedback otorgado por los alumnos a futuras versiones de la asignatura. La figura 4.8 resume la estructura general de la propuesta.

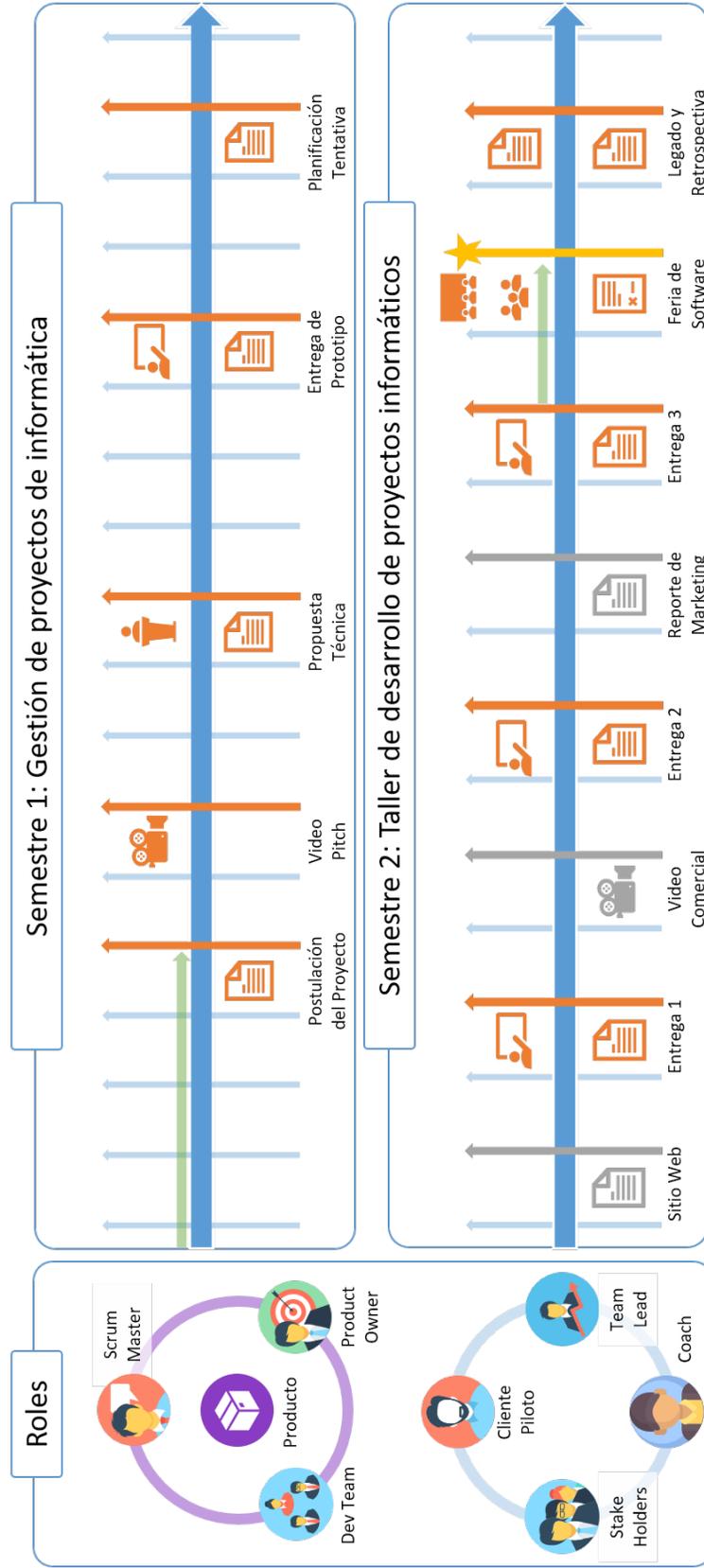


Figura 4.8: Diagrama resumen de la propuesta, con sus evaluaciones y roles.

## Capítulo 5

# VALIDACIÓN DE LA SOLUCIÓN

Debido a que no fue permitida la implementación de la propuesta durante el año 2018, esta fue validada a través de un grupo de expertos. Dicho grupo consiste en 6 Ingenieros Civiles en Informática de la UTFSM con experiencia de al menos 1 año trabajando con métodos ágiles, detallados en la tabla 5.1.

Nombre y Apellido	Cargo u ocupación	Experiencia
Daniel Basterrica	Agile Product Owner - Equifax	12 años
Alejandro Bedini	Agile Enterprise Coach - Resource IT Solutions	8 años
Patricia Trejo	Project Manager - ThoughtWorks	3 años
Álvaro Hernández	Consultor/Desarrollador- ThoughtWorks	3 años
Axel V. Simonsen	Big Data Software Engineer - Equifax	2 años
Ignacio Sfeir	Developer - Equifax	1 año

Tabla 5.1: Grupo de expertos que evaluaron la propuesta.

Estos, considerando su experiencia previa en la asignatura y sus conocimientos de métodos ágiles, evaluaron la presente propuesta bajo 4 criterios:

- Su coherencia con las metodologías ágiles.
- Su aporte al desarrollo de software.

- Su aporte al realizar proyectos de innovación.
- Su aporte a la formación de futuros desarrolladores profesionales.

La evaluación consistió en una escala Likert [56], evaluando qué tan de acuerdo cada evaluador estaba respecto del aporte de la propuesta a cada criterio, pudiendo adjuntar observaciones o sugerencias. Los resultados son resumidos en la figura 5.1.

## 5.1. Resultados de la evaluación

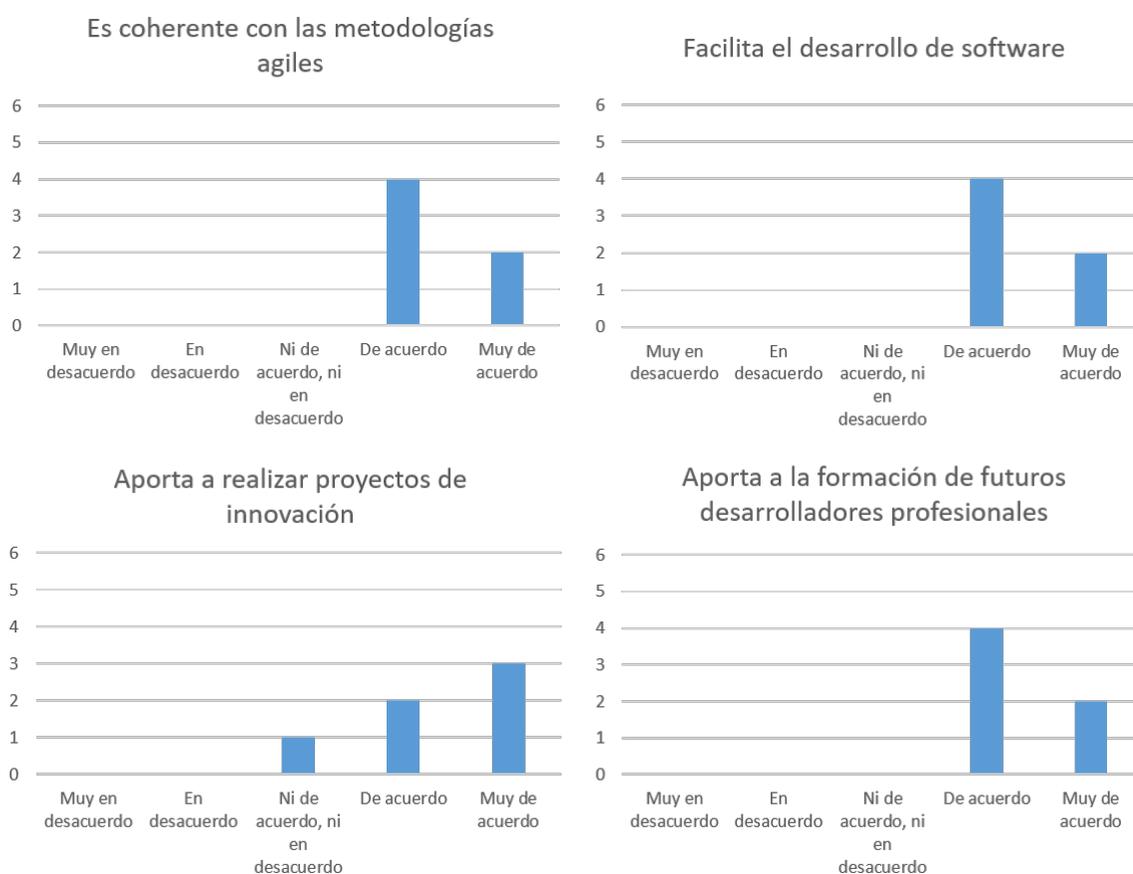


Figura 5.1: Apreciación de la propuesta desde el punto de vista de los expertos.

La recepción de la propuesta fue positiva en los cuatro aspectos. Hubo solo una respuesta neutral respecto a su aporte a proyectos de innovación lo que, según el evaluador, no es debido a que la propuesta se considere deficiente, sino porque se es necesario complementarla

con actividades y mayor orientación a lo largo de los contenidos vistos en clase para que esta sea efectiva. Debido al alcance del presente trabajo y a que dicho factor es definido por los profesores, esto no estaba contemplado, pero es considerado como trabajo futuro. También se sugirió el uso de “Lean Inception” [64] en clases, lo cual es detallado en las conclusiones.

## 5.2. Recepción de la propuesta

Luego de exponerse la propuesta a la coordinación, esta fue recibida y aceptada, realizándose dos reuniones en pos de planificar su ejecución para la feria de software del año 2019. A pesar de esto, la propuesta se vio modificada, cambiando ciertos contenidos y términos usados en las evaluaciones e incorporando nuevos elementos:

- Se cambia el término “prototipo” usado en el primer semestre por “prototipo de mitigación de riesgos”, debido a que en asignaturas anteriores se utiliza dicho término para hacer referencia a un producto desarrollado de forma rápida e improvisada, pudiendo ser desechado.
- Después del Sprint Review 3 se incorpora la entrega de un video que muestre a un usuario utilizando la aplicación, probando las historias más importantes del proyecto con el fin de validar que el software completo funcione antes de su demostración en el evento.
- Se reincorpora una entrega de estimación del costo del proyecto a la actual planificación como elemento dentro del documento de planificación tentativa, señalando que dicho elemento está incluido dentro de los aprendizajes esperados de la asignatura y, por ende, no puede ser descartado.
- Se reduce el plazo para la entrega del documento de Sprint Retrospective de 1 semana a 4 días, justificando que demasiado tiempo baja la productividad de los equipos.
- El coordinador de la asignatura modificó los porcentajes propuestos para las evaluaciones (entregas, marketing, participación).
- Se decide que la reunión de retrospectiva de profesores se hará una vez cada dos semanas desde el comienzo de la asignatura. Esto, al notar ellos la falta de comunicación presente a lo largo del año, lo que causaba imprevistos al ejecutar la planificación.

# CONCLUSIONES

El objetivo principal de la presente propuesta, mencionado en el capítulo 1, se definió como “Entregar una propuesta que mejore el desempeño, control y aprendizaje de los equipos de las asignaturas de Gestión de proyectos de informática y Taller de desarrollo de proyectos informáticos”. En vista del feedback positivo, tanto del grupo de profesionales como de los profesores de las asignaturas, se señala dicho objetivo como logrado. En relación a los objetivos secundarios, se señala que:

- Un diagnóstico a las asignaturas se cumple con lo expuesto el capítulo 3, analizándose los factores que entorpecen el desarrollo y evaluación de cada equipo en la sección 3.6.
- Fueron mejoradas las herramientas utilizadas, tanto en la gestión, evaluación, y en el mismo desarrollo de software y su coordinación a través de la implementación de un *product vision board* y el diagrama de flujo acumulado, siendo descartadas herramientas redundantes y elementos innecesarios de las evaluaciones.
- Se logra establecer una guía adecuada, definiendo roles y responsabilidades para cada integrante de la asignatura, las que se entregarán al inicio de la asignatura.
- Se elaboran nuevas pautas, claras y concisas, para cada hito de la asignatura, reduciendo las ambigüedades al evaluar y adjuntándose estos en los Anexos 5.7 y 5.8.
- Se establece un diagrama de flujo acumulado y estandariza la forma de estimar de los alumnos, dando paso a posibles análisis de productividad.

En base a lo observado durante el planteamiento de la presente propuesta, se señala que los factores que esta buscaba solucionar no resultaron ser el origen del problema de la asignatura, sino consecuencias de un problema raíz, el cual es señalado en [52]: la mayor dificultad

al implementar agilidad es el lidiar con el cambio en la cultura organizacional. Esto fue evidenciado durante el diagnóstico, en donde cada profesor realizó su trabajo como acostumbra hacerlo en sus asignaturas y donde, incluso en la presentación de la propuesta ante dichos profesores, se observó una jerarquía en la toma de decisiones, lo que fue criticado en los legados del año 2018: se le exige a los alumnos trabajar como equipo en agilidad, pero los profesores no aplican dicha estructura ni trabajan como tal.

El presente documento deja como legado a futuras iniciativas en este ámbito que el primer paso hacia el trabajo con metodologías ágiles es el contar con un ecosistema académico que facilite el cambio y **sea parte de él**, no solo coordinándolo sino que abrazando y comprendiendo los valores y principios de esta cultura de trabajo. No basta con implementar las prácticas y ceremonias, ya que lo más importante es que se incorpore su cultura a todos los involucrados de manera horizontal. No es suficiente pedirle a los equipos que sean ágiles, puesto que **si el resto de la organización no está dispuesta a cambiar, se dará paso sólo a confusión e impedimentos.**

## 5.3. TRABAJO FUTURO

En vista de que esta propuesta es el primer acercamiento en pos de lograr una asignatura en base a agilidad, se observan muchos factores a incorporar, puesto que un proyecto de este tipo incorpora elementos tanto académicos como técnicos y teóricos. Por razones de alcance y tiempo hubo elementos que no fueron incorporados, pero se señalan a continuación.

### 5.3.1. Respeto de la Metodología

Uno de los focos al usar agilidad es reconocer que al comienzo del proyecto no se sabe exactamente cómo va a funcionar el sistema ni qué es lo que va a tener más impacto en el negocio, por lo que se debe permitir “pivotar” (cambiar el enfoque de la idea). Esto es fundamental considerando un contexto de innovación en donde es posible un futuro emprendimiento, puesto que esto puede ayudar a manejar situaciones en donde el profesor rechaza las ideas de proyectos en etapas preliminares.

Debido a que Scrum se enfoca en la gestión de proyectos, sería ideal complementarlo con otras metodologías que potencien el desarrollo, el escalamiento o el emprendimiento, tales como SAFe [65], Lean Startup [49], Running Lean [66] o Lean Inception [64].

### 5.3.2. Respeto del Desarrollo y los Roles de Equipo

En la actualidad sólo se realiza un acercamiento al software resultante mediante un “prototipo”. Sería ideal incorporar un producto mínimo viable como tal, siendo puesto en producción o, al menos, siendo probado por sus usuarios desde el inicio.

En el documento de planificación tentativa se utiliza un *user story map* simplificado. Es necesario reemplazar dicha medida por algo estándar como un *product roadmap* o un *User Story Map* [67], lo que es más acorde al objetivo que se quiere lograr en dicha etapa.

Respecto de los roles, sería de gran utilidad incorporar un cargo de UX a los equipos (*User Experience Design*), encargado del *benchmarking* de producto, de la interacción con usuarios

y el feedback de estos, entregando información constante sobre la apreciación del producto y de posibles mejoras para este.

Los expertos evaluadores de la propuesta sugirieron que los roles de cada alumno vayan rotando, buscando que más de un alumno aprenda el cómo realizar las responsabilidades de cada cargo. Esto sería de gran utilidad para fomentar el aprendizaje de los estudiantes, pero dificultará el progreso del software, por lo que no se espera incorporarlo en el corto plazo.

Respecto de los proyectos, sería importante considerar una diferenciación entre proyectos de tipo social, de investigación o emprendimiento, puesto que actualmente las asignaturas de la feria de software evalúan a todos sus proyectos bajo los mismos criterios y exigencias, siendo que algunas de estas no aplican a casos puntuales.

### **5.3.3. Respeto de Retrospectivas y Feedback**

El feedback entre las personas del equipo, o incluso entre el equipo y stakeholders, es algo que debe ser fomentado constantemente y es necesario saber cómo realizarlo correctamente para que sea positivo y constructivo. Las retrospectivas son herramientas para una mejora continua, siendo uno de los grandes objetivos del desarrollo ágil en comparación al desarrollo tradicional. En las asignaturas, es necesario incorporar contenidos acerca de la cultura del feedback, enseñando cómo guiar una retrospectiva saludable donde nadie se sienta atacado y en donde se mantenga el enfoque en lo que salió bien y lo que se podría mejorar.

Debe aumentarse la interacción entre alumnos, profesores, ayudantes y clientes, con el objetivo de recibir el mayor feedback posible a tiempo. Las retrospectivas no debiesen ser sólo en un punto sino continuas, por ejemplo cada 2 o 3 semanas de trabajo, identificándose los avances y mejoras del equipo constantemente. Un feedback constante es necesario al desarrollar productos de manera ágil y es una diferencia del método tradicional general.

## 5.4. Uso de Lean Inception

Dos de los expertos que evaluaron esta propuesta señalaron que, para una siguiente iteración de estas mejoras, es importante coordinar tanto los contenidos de las asignaturas, como las evaluaciones y el enfoque de esta nueva propuesta ágil. Dichos evaluadores sugirieron incorporar las actividades de Lean Inception [64] al inicio de la asignatura, puesto que son actividades probadas en la industria y que demuestran resultados concretos.

Evaluando su compatibilidad con las asignaturas, para las semanas de elaboración del *product vision board* resultan útiles las primeras actividades de Lean Inception, en donde se realiza un *Elevator Pitch* y se definen objetivos para el producto. La metodología se puede incorporar completamente, resultando, a finales de esta, una definición de lo que es dicho producto, siendo esta asimilada por todos los miembros del equipo, algo útil para el periodo de licitación inicial. Por otra parte, también ayuda a determinar un *backlog* de historias de usuario, aportando a la presentación de la propuesta técnica.

Luego del primer hito de Lean Inception pueden incorporarse presentaciones a través de *demos*, mostrando los avances del proyecto a nivel de definición de proyecto, definición técnica o de evolución del software. Es posible incluso usar un *MVP Canvas* (o *Release Canvas*) en la Planificación Tentativa, el que también es una herramienta de Lean Inception.

# Anexo

## 5.5. Detalle de legados de años 2014-2018

A continuación se presenta un detalle con los contenidos de cada año por separado, destacándose aquellos comentarios que reflejaron mayor disconformidad.

### 5.5.1. Año 2014

Para el 2014 lo que más se destaca son las críticas a la coordinación, la falta de pautas y a los contenidos y estructura de la asignatura.

#### **Contenidos y estructura de la asignatura:**

- Se critican los contenidos del ramo, siendo estos contradictorios con asignaturas anteriores donde se señala que carta Gantt y cascada no aplican a proyectos informáticos hoy en día, en donde la estructura del ramo fue desorganizada y mal planificada y muchas sesiones se realizaron de forma apurada, generando dudas y dejando cosas poco claras.
- Se pide que se enseñe acerca de modelo de negocios y potencial de emprendimiento, lo que más revisa el jurado en la licitación y el evento, puesto que no se toca ese tema en todo el año, dar más clases de difusión y promoción del producto y darle más importancia al prototipo.
- Se critica la poca definición de roles, donde el jefe de proyecto no tiene claro cuáles

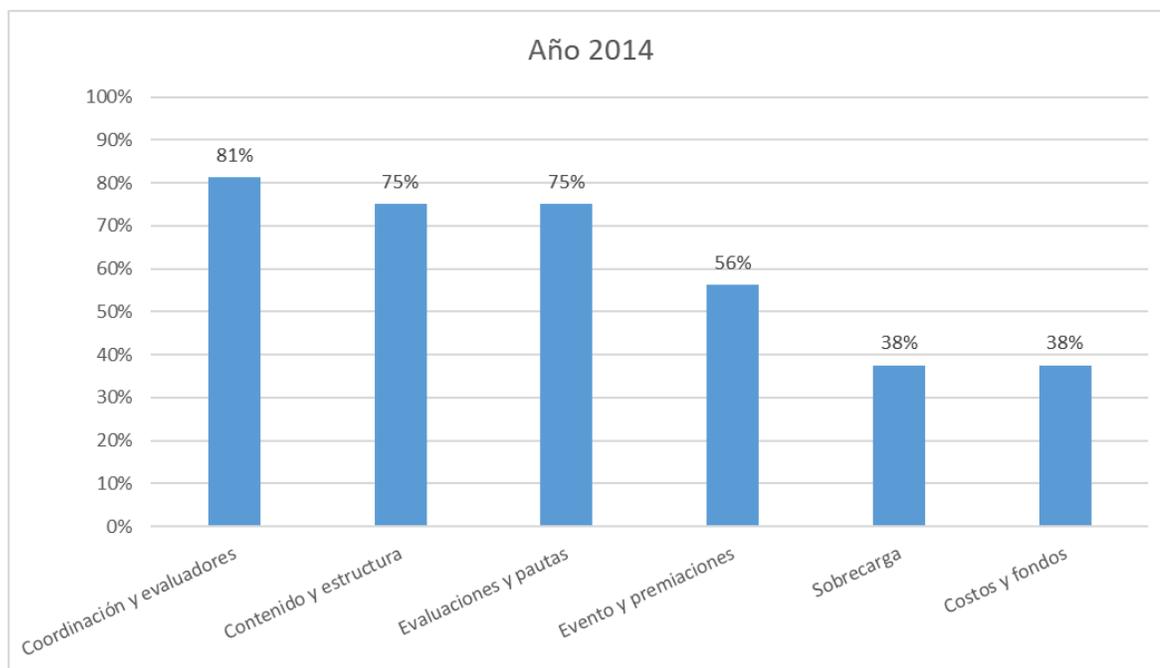


Figura 5.2: Porcentaje de equipos que presentaron críticas, en base a los 16 presentes en el año 2014.

son sus deberes, y donde los clientes externos no entienden de desarrollo, tratando a los equipos como contratistas de bajo costo.

### **Profesores y coordinación:**

- Se critica el desempeño de los profesores, en donde hubo atraso en la entrega de notas, al punto de no estar para el segundo semestre, y que trabajos aceptados al inicio se rechazaron al segundo semestre, destacándose el poco compromiso de uno de ellos que llegó a realizar revisiones por Skype. El mismo profesor se desligó después durante el segundo semestre y no se vio hasta el cierre de la asignatura.
- Por otro lado, se critica el poco control del desempeño de los ayudantes. Algunos fueron más atentos que otros, lo que causó ventaja en los equipos con mayor respaldo.
- Se observa que falta feedback para saber qué aspectos debe mejorar cada equipo además de orientación, como lo es en el caso de la licitación. Se señala que en esta última se le da muy poco tiempo y apoyo a la búsqueda de una idea de proyecto. Al

rechazar se daba un “NO” y no se daba justificaciones o una ayuda para concretar un proyecto aceptable. Se señala “falta de tacto” al rechazar ideas.

### **Evaluaciones, criterios y pautas:**

- Se critica la falta de pautas de evaluación, puesto que hubo contradicción entre lo que dicen profesores y ayudantes. A pesar de tenerse pautas los profesores penalizaron por cosas que no se solicitaban.
- Se señala además que hay muy poca flexibilidad, penalizándose cambios por parte de los equipos e imponiéndose prácticas que a veces eran contraproducentes para estos, como el exigir marketing a pesar de que el cliente haya especificado discreción y/o confidencialidad del proyecto. Además de esto se pide que se acepten proyectos de carácter científico, social o de inteligencia de negocios, para generar experiencia en las áreas que a cada uno más le interese.
- Se señala falta de transparencia. Debiesen mostrarse todas las pautas y fechas desde el inicio, puesto que surgieron nuevas evaluaciones en la marcha y esto afectó en la planificación de los estudiantes respecto de otras asignaturas.

### **Evento y premiaciones:**

- Se critica al evento, en donde el jurado solo se enfoca en apariencias y no en funcionalidades, no hay información de lo que se requerirá para tal fecha por parte de los estudiantes y no hay transparencia acerca de quién asigna los premios, bajo qué criterios, o el desempeño de cada equipo en todas las premiaciones.
- Se señala que se presentaron problemas graves de transporte y nula atención a asistentes del evento aparte de lo que ofrecían los estudiantes.

### **Financiamiento de la asignatura:**

- Se critica el poco financiamiento. El costo de algunos proyectos ya era elevado y salió peor con los gastos de presentación. Los fondos concursables terminaron siendo una carga extra que generó conflicto en los miembros del equipo y los fondos dados por

el departamento no fueron suficientes, puesto que a medida que avanzaba el taller los costos se hicieron mayores.

**Sobrecarga académica:**

- Se señala que el ramo es exigente y quita mucho tiempo respecto de sus créditos. Se cuenta con algunos conocimientos, pero falta orientación al momento de implementarlos.

**Comentarios destacados:**

- *“En feria la gran mayoría de los productos son olvidados e incluso nunca publicados luego de ésta”.*
- *“El ramo no es como lo venden”.*
- *“Para tener un buen desempeño y optar por buenas calificaciones se necesitó descartar otros ramos, trasnochar y dejar de lado ciertas responsabilidades familiares y laborales”.*
- *“Se piden cosas que un informático no debiese realizar en su campo laboral. No es útil el rol de pre-empresa si uno no trabaja así”.*

### 5.5.2. Año 2015

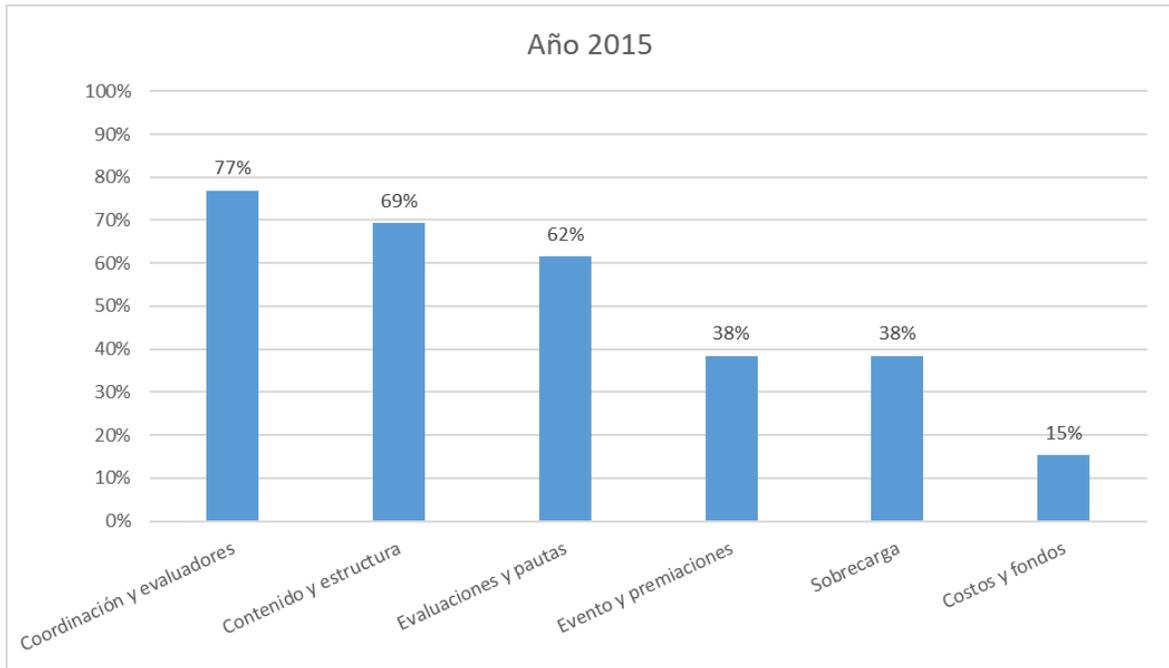


Figura 5.3: Porcentaje de equipos que presentaron críticas, en base a los 13 presentes en el año 2015.

A pesar de ser una cantidad menor de equipos, siguen predominando los mismos temas que el año pasado.

#### **Contenidos y estructura de la asignatura:**

- Se señala que falta actualizar los tópicos del ramo, ya que muchos ya están obsoletos y muy poco de lo que se enseña es realmente lo que se ocupa. Además, se señala que muchas de las técnicas enseñadas ya se ven en ramos futuros o paralelos a feria. Evidencia de lo poco actualizado de la asignatura fue que un cliente pidió trabajar con historias de usuario en vez de casos de uso.
- Se pide, como el año pasado, orientación para realizar modelo de negocio. Un cliente al final tuvo que ayudar a un equipo a formularlo mejor. Además de esto, se solicita orientación en herramientas básicas de desarrollo y en diseño de interfaces, además de cómo exponer, es decir, cómo llegar al posible cliente.

- Se pide que se reorganice el orden de las charlas, puesto que se hacen cuando ya no es útil, y que en la estructura del ramo se le dé más prioridad al aspecto de proyectos de innovación por sobre proyectos de emprendimiento.
- Se critica nuevamente la falta de orientación en la elección de una idea para la feria, y se señala que los conocimientos otorgados por la asignatura de *Evaluación de Proyectos* harían mucho más fácil la elección de un buen y adecuado proyecto para la Feria.
- Se señala que la herramienta base para los proyectos no cumplía con las expectativas, complicándose el desempeño del grupo en todo el año y que se debiese de orientar a los alumnos acerca de herramientas útiles para la gestión de equipos, tales como trello.

#### **Profesores y coordinación:**

- Se aprecia poca coordinación por parte de los profesores respecto de pautas y trabajos. Las pautas y evaluaciones aparecían con muy poca anticipación y la información no se entregaba correctamente. Puntualmente se modificaron pautas días antes de una entrega y la información en la plataforma Moodle era presentada de forma dispersa y difícil de encontrar.

#### **Evaluaciones, criterios y pautas:**

- Los ayudantes, debido a la falta de rúbricas, corregían según lo que ellos recordaban de otros ramos, lo cual es subjetivo. Debido a esto se pide mayor atención a la creación de pautas o rúbricas en la asignatura. La falta de estas en algunas entregas hace que sea difícil entender qué es lo que se pedirá o revisará. Se señala además que las pautas ya existentes deben ser más claras, y que no se deben modificar sin previo aviso.
- Por otro lado, se señala que la rigidez de las evaluaciones perjudicó algunos proyectos debido a que marketing y publicidad en redes sociales afectaban algunas ideas de proyecto que requerían discreción.
- Se pide que, ya que los jurados de la asignatura son de diferentes áreas, que cada uno analice el área que es su fuerte y no todo.

### **Evento y premiaciones:**

- Se pide nuevamente mejorar el traslado para el día del evento y solicitar autenticación para los premios, ya que algunos equipos hicieron trampa al votar más de una vez dando los rut de varias personas. Sumado a esto uno de los stands se desarmó en pleno evento y no se tenían las herramientas para arreglarlo. Se señala que estos son muy pequeños y que también lo es el espacio del mismo evento.

### **Financiamiento de la asignatura:**

- Se critica el poco financiamiento, teniendo muchos equipos que pagar stand, merchandising y material técnico de sus fondos personales. Más que depender de fondos concursables, el departamento debería apoyar más a sus estudiantes.

### **Sobrecarga académica:**

- Se critica que los plazos son muy cortos y hace falta tiempo. Debiesen ser tiempos más realistas y una exigencia acorde a los créditos asignados al ramo. Se destaca por varios equipos que hay una cantidad agobiante de informes y requerimientos en el segundo semestre.

### **Comentarios destacados:**

- *“No cumple con las expectativas de un ramo de tal envergadura”.*
- *“Asignatura no cumple expectativas. A pesar de que la feria de software es un evento que lleva realizando muchos años, aún existen muchas cosas por mejorar”.*
- *“Aumentar los requisitos o desplazar el ramo hacia el final, ya que no todos tenían suficiente experiencia técnica”.*
- *“Un ramo con entregables, controles y además un proyecto startup en paralelo no corresponde a 4 créditos”.*

### 5.5.3. Año 2016

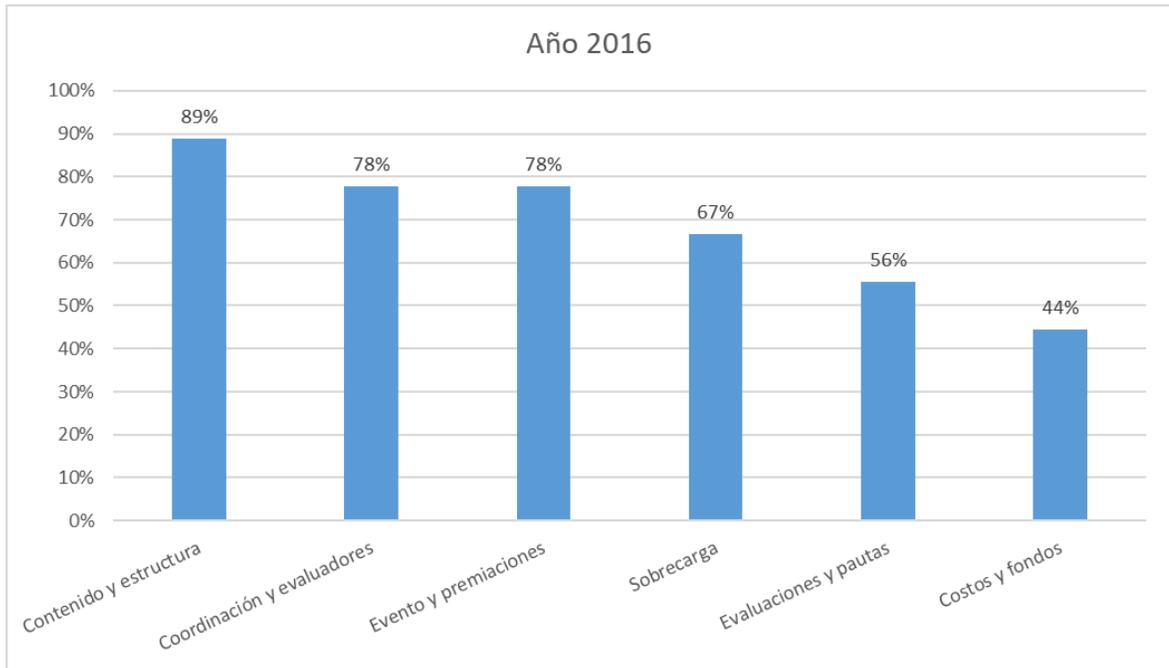


Figura 5.4: Porcentaje de equipos que presentaron críticas, en base a los 9 presentes en el año 2016.

Durante este año, se critica principalmente a la coordinación y los contenidos, siendo el evento criticado debido al atraso en el armado previo de este, lo que perjudicó a los alumnos y causó penalizaciones y problemas de transporte.

#### **Contenidos y estructura de la asignatura:**

- Nuevamente se pide mayor orientación técnica.
- Se solicita respaldo del plan de negocios, más charlas para emprendimiento y el cómo realizar una buena exposición (repetido en años anteriores).
- Se pide repartir más apropiadamente las charlas, ya que se hicieron cuando sus temas ya no aportaban mucho.
- Se señala que, a pesar de mencionarse, nunca se publicaron los legados de años anteriores.

### **Profesores y coordinación:**

- Se pide más apoyo para los proyectos y mejorar los feedback. En revisiones se recibían respuestas como “está bien”.
- Nuevamente se solicita mayor respaldo a la búsqueda de ideas. Además, mencionar con qué cosas cuenta la universidad (herramientas, convenios).
- Se critica la falta de organización y coordinación en entregas, además de una mala coordinación entre campus.
- Se pide usar Moodle para la información esencial, puesto que se envió mucho mail y se perdía el dónde se mencionaba cada cosa.

### **Evaluaciones, criterios y pautas:**

- Se pide nuevamente definir las fechas de entregas con tiempo suficiente, estas eran comunicadas con poca anticipación. Además de esto, muchas veces las pautas no estaban disponibles con el tiempo de antelación necesario.
- Se sugiere mostrar ejemplos de años pasados para los entregables como una guía.
- Se solicita estandarizar criterios de evaluación entre campus para evitar ambigüedades y contradicciones entre evaluadores.
- Se sugiere extender las defensas de avances a 2 bloques, ya que el tiempo es muy corto.
- Se señala que hay entregables que se consideraron una pérdida de tiempo y ánimos, sin entregar ningún beneficio al proyecto.

### **Evento y premiaciones:**

- Se critica el que los puntos que impliquen gasto de dinero en el evento formen parte de la nota en vez de considerarse un valor adicional, puesto que se le da más puntaje a quienes contratan una promotora e invierten más dinero que a aquellos que cumplen con un buen proyecto.
- Se señala una mala organización del evento. El armado del evento se atrasó, lo que hizo que grupos se perdieran el bus al hotel. Además, hubo problemas técnicos de internet el día del evento lo que le jugó en contra a varios proyectos.

- Se pide mejorar la apariencia del folleto (tríptico) ya que es la imagen para familiares, amigos e invitados al asistir.
- Se señala nuevamente que los stands fueron muy pequeños.
- Se agradece la iniciativa de hacer un streaming online con el evento, pero el entrevistador fue muy aburrido y repetitivo.

#### **Financiamiento de la asignatura:**

- Nuevamente se señala la falta de apoyo económico. No hay suficiente dinero si no se postula a fondos concursables, pero estos consumen mucho tiempo y requieren dedicación. Además, al solicitarse materiales a alguno de ellos, estos llegaron tarde.

#### **Sobrecarga académica:**

- Se resalta una carga académica muy alta, en donde personas con trabajos part-time o a cargo de una familia se vieron colapsadas.

#### **Comentarios destacados:**

- *“Que los puntos que impliquen gasto de dinero en el evento sean extras y no parte de la nota. Se le dio más puntaje a quienes contrataron una promotora e invirtieron más dinero”.*
- *“Tristemente, el equipo con más fondos es el que más posibilidades tiene de destacar y obtener mejores notas”.*
- *“Sorprendidos por problemas de gestión del ramo a pesar de impartirse hace tantos años”.*
- *“Esperábamos aprender más sobre gestión e ingeniería de software. Todo se había visto en fundamentos de ingeniería de software”.*

#### 5.5.4. Año 2017

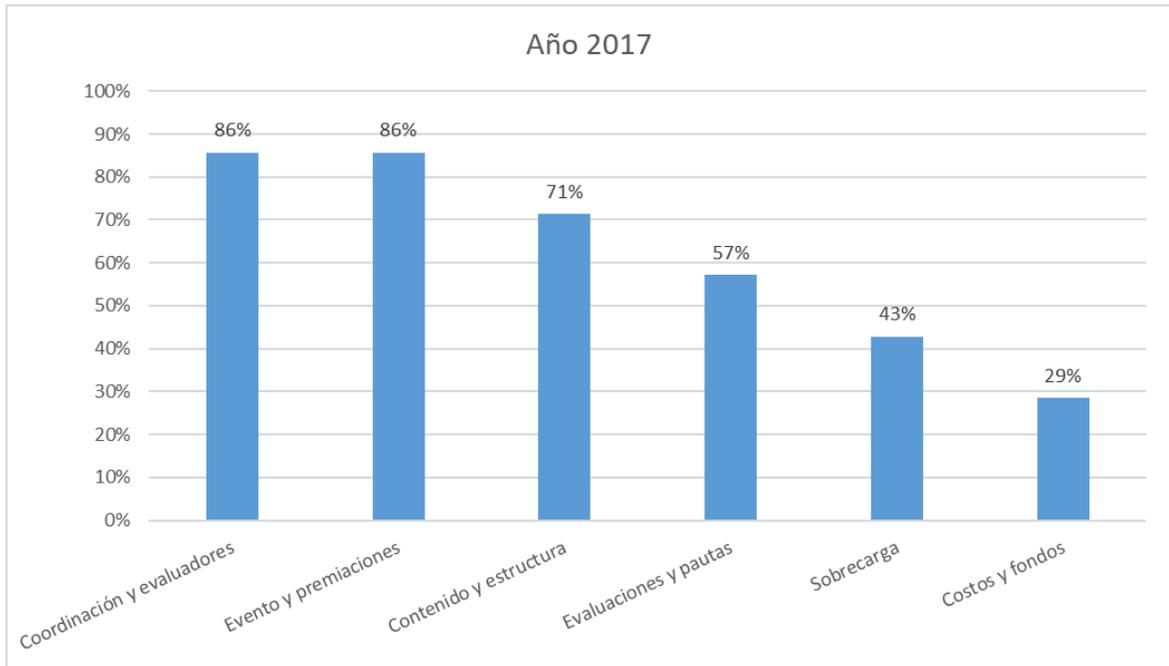


Figura 5.5: Porcentaje de equipos que presentaron críticas, en base a los 7 presentes en el año 2017.

Nuevamente se repiten los temas anteriores, pero predominando las críticas al evento debido a las prácticas ilícitas de un equipo para ganar la premiación.

#### Contenidos y estructura de la asignatura:

- Se señala que se tratan de abordar demasiados contenidos que no se ven en profundidad.
- Se pide desaparecer la metodología de cascada y potenciar Scrum, puesto que la metodología impuesta fue agotadora y hasta considerada “*vergonzosa*”.
- Nuevamente se solicita mejorar o cambiar la herramienta usada en la asignatura, ya que fallaba y era demasiado detallada. Además, se pide fomentar uso de herramientas de comunicación como Discord<sup>1</sup>.
- Se critica que, a pesar de su gran cantidad, los informes del primer semestre no fueron de utilidad para el segundo y, en particular, se critica la exigencia de un informe

<sup>1</sup><https://discordapp.com/>

económico, ya que no aplica al contexto de la asignatura y es tan detallado que hace perder el foco de su motivo o utilidad. Debido a que la mayoría de los proyectos eran digitales la coordinación obligó a rellenar con supuestos poco realistas. Además de esto, a pesar de tratar de defender su importancia, sólo se entregó una planilla de calidad cuestionable para la realización de este informe.

- Se pide, al igual que en años anteriores, que se le dé mayor importancia al modelo de negocios, ya que es de lo más preguntado por las entidades de importancia que visitan el evento.
- Se sugiere como idea que a futuro se permita la colaboración entre proyectos, puesto que hay ideas que pueden usarse en conjunto.

### **Profesores y coordinación:**

- Nuevamente se critica el poco control al aporte de ayudantes puesto que la dedicación de estos fue dispareja, lo que se notó en la licitación de ideas.
- Se critica muy frecuentemente al comité organizador, debido a la mala organización y planificación a lo largo del año. Se modificaron pautas en la marcha y no se respetaron las fechas de evaluación. Esto último causó problema a uno de los clientes que debió pedir permiso dos veces en su trabajo para asistir a la defensa del software.
- Se señala que difundir el evento de la feria debería ser responsabilidad del comité organizador y no de los alumnos.

### **Evaluaciones, criterios y pautas:**

- Se pide seriedad y consistencia al momento de evaluar. Durante la licitación los jueces solo podían hacer 3 acotaciones y 1 miembro del equipo responder. Luego la situación se relajó, pudiendo responder cualquiera y recibéndose feedback de todos los jurados, lo que marcó una enorme diferencia en las calificaciones.
- Se señala que hubo descuentos por elementos que, si se encontraban en los informes, pero que para notarlo se debía leer este por completo, lo que no se hizo debido a que se repartieron los informes por secciones. No hubo compensación para aquellos que apelaron.

- Se critica el que hubiese informes que se avisaron 2 días antes de su entrega. Además, se agregaron entregables, pautas y ponderaciones a última hora, como la evaluación económica.

### **Evento y premiaciones:**

- Se critica que el colocar una nota de por medio en las apreciaciones personales hace que sólo se responda con cosas que dejen contento al evaluador. Por ejemplo, el presente documento exige que se digan aspectos buenos y malos de la asignatura.
- Se pide que se controle previamente lo que cada equipo planea realizar en el evento de feria, ya que surgen estrategias poco éticas.
- Se pide que se muestren los resultados de las premiaciones. El premio de innovación es poco claro y abstracto. No se sabe cómo se evalúa o qué criterios sigue. Se critica la poca claridad y orden en las premiaciones.
- Nuevamente se pide ampliar los stands, orientar y recibir mejor a los visitantes del evento y difundir este de mejor manera, sin depender del trabajo de los alumnos. Se señala que el espacio del evento es muy reducido.

### **Financiamiento de la asignatura:**

- Se señala que cada año se pide mayor decoración en los stands, pero se da poco apoyo para esta, faltando fondos para todo lo que se pide. Además, a pesar de tenerse 3 fondos concursables estos no aportan una ayuda real al momento de necesitarlos.
- Por ejemplo, se realizaron solicitudes a PIE>A a finales de agosto y los materiales llegaron después del día del evento. Es importante conocer los requisitos de cada uno.
- Se critica que no se mencionan desde un principio todos los gastos que requerirá la asignatura durante el año, lo que impide considerar dichos gastos durante el proceso de postulación a fondos.

### **Sobrecarga académica:**

- Se pide considerar el tiempo que requiere realizar tanto informe, estimación y medición, ya que la carga académica es muy alta.
- Se critica el exceso de indicadores solicitados, ya que se agobia al equipo solicitando tanta información. Se considera que los registros de avances eran demasiado tediosos e innecesarios.
- Se hace énfasis en la gran cantidad de informes, en especial durante el segundo semestre. Se señala que no aportan y solo retrasan el avance del proyecto.

### **Comentarios destacados:**

- *“Ni siquiera se daban el tiempo de leer (todo) lo que habíamos escrito en los entregables. ¿Para que se realizan entonces? Esto perjudica mucho a los alumnos, en especial a aquellos como nosotros, que realmente se preocupan de desarrollar un producto funcional y de calidad”.*
- *“Proyección económica (fue) tan detallada que hace perder el foco de para qué se hace. Obliga a rellenar con supuestos que hacen perder validación a los números que entrega este informe”.*
- *“Consideramos inaceptable que a pesar de que se nos dijo que las pautas estaban subidas desde principio de año, esto no haya sido así. Existía un 30 % final que no se especificaba cómo se evaluaba”.*

### 5.5.5. Año 2018

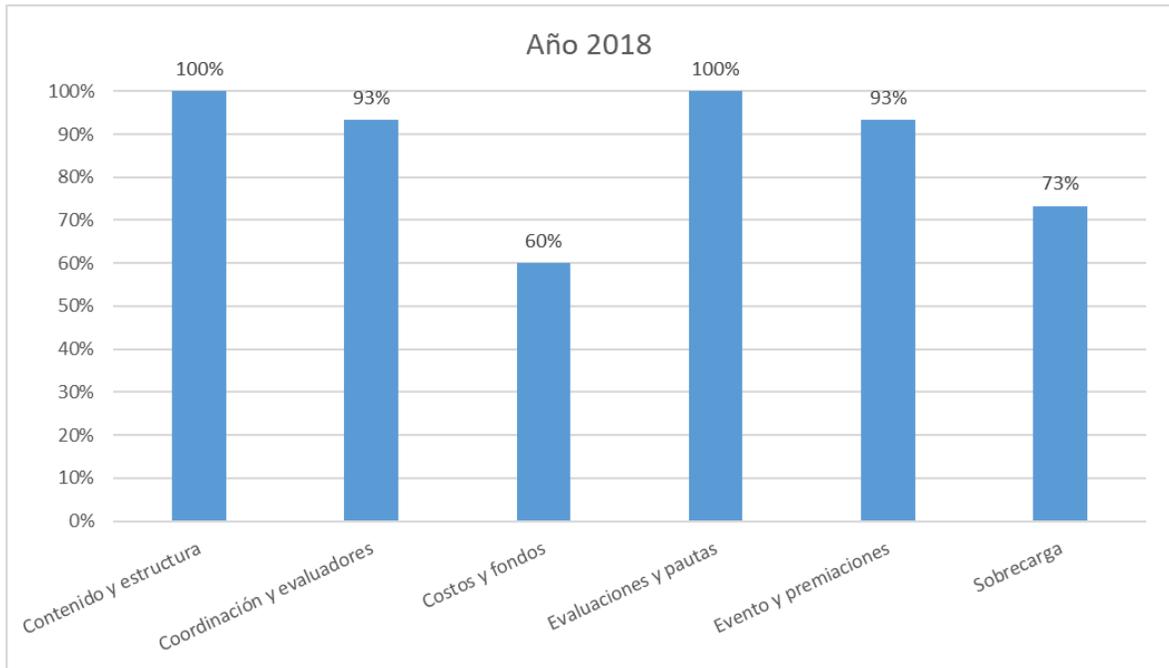


Figura 5.6: Porcentaje de equipos que presentaron críticas, en base a los 15 presentes en el año 2018.

Para este año, en particular, se decidió que la entrega del documento legado se realizaría de forma individual, lo que causó un número desbordante de entregas a revisar. Este año es el que presenta la mayor cantidad de descontento hasta la fecha, criticando duramente tanto a los profesores como sus contenidos y las pautas que se utilizan en la asignatura, lo que justamente es una de las áreas que aborda la presente propuesta.

#### **Contenidos y estructura de la asignatura:**

- De los contenidos exigidos en la asignatura, se señala una deficiente o nula orientación respecto de lo que se exige. Sólo se entrega material como diapositivas o textos estándar que no aportan al aprendizaje o desempeño de los equipos
- Se critica lo anticuado de la asignatura, señalando contradictorio el uso de elementos como una carta Gantt.

- Se señala que, de lo enseñado, es poco o nulo el aporte al aprendizaje y desarrollo, siendo las clases realizadas cuando ya no eran de utilidad.
- Se critica la relevancia que se le da a la Innovación, pidiendo “bajarle la importancia”. Se pide considerar aspectos como la resolución de problemas emergentes o antiguos que afectan a personas o empresas. Se señala el tomar en cuenta el impacto que puede causar un proyecto, no solo “que tan innovador es”.
- Se critica la mala comunicación y el poco Feedback entre profesores y alumnos, señalando que este no era un aporte debido al tono de crítica que se recibía, y debido a que usualmente el feedback llegaba cuando ya no era de utilidad.
- Se señala la poca consistencia en los temas enseñados en la asignatura, en donde no se condice lo que se enseñó con lo que se practicó. Se destaca el tema del trabajo en equipo, lo que no se vio por parte de los profesores, y la aplicación de la metodología de agilidad que nunca se vió, sino que sólo se usó como otra evaluación.
- Se sugiere, ya que se requiere de márketing y creación de videos, el interactuar con miembros de otros departamentos, y se señala que “El proyecto de feria es un desafío multidisciplinario, debiésemos trabajar con diseñadores, ingenieros comerciales, publicistas, ya que nos vimos obligados a adquirir conocimientos no residentes en nuestra área y poco útiles, lo que le quito tiempo y finalmente calidad al resultado final”.

### **Profesores y coordinación:**

- Se señala como hipócrita el que los profesores enseñen buenas prácticas, las exijan y no las apliquen entre ellos ni con sus alumnos.
- Se critican las pautas, ya que a veces no se entendía lo solicitado y se penalizaba por factores poco claros.
- Se señala desmotivación, debido a que una idea que fue previamente seleccionada por los profesores y expuesta a los alumnos, al momento que estos la emplearon, se les dijo, no era suficiente, por lo que debían plantear otro proyecto.
- Se critica la falla en algunas planificaciones, ya sea por agregar cosas para los entregables en último momento o por agregar nuevos entregables a principio del segundo semestre que no estaban planeados a principio del año.
- Se pide apoyar más a los equipos con problemas de convivencia.

- Se pide mayor transparencia con las pautas y exigencias, puesto que se dieron varias situaciones en donde cambiaron estas o se añadían elementos a última hora.
- Se señala una evidente desorganización en la asignatura y en los profesores, y un ambiente de “improvisación”, llegando a contradecirse lo que decían ayudantes y profesores.
- Se critica la falta de anonimato en el presente documento legado.
- Se señala la falta de interés de los profesores con ciertos proyectos, en donde el único apoyo que recibían era el del ayudante.
- Se observa favoritismo, en especial en eventos y reportajes, donde siempre se mostró y mencionó al mismo proyecto.
- Se critica que, a pesar de pedirse el correcto uso de GitHub, no se dio ningún feedback “real” y sólo se tuvo una clase para introducir el correcto funcionamiento de este. Se señaló una penalización en caso de no cumplir con esta estructura, pero esto nunca se volvió a mencionar.
- No se especificaron los gastos monetarios que implicarían los proyectos, por lo que al postularse a fondos concursables, los alumnos iban a ciegas sin saber cuánto necesitarían. Junto con esto, los alumnos sin fondos debieron aportar con ahorros propios debido a las pautas que exigían elementos pagados.
- Se señala la actitud de los profesores como “destruktiva y en ocasiones poco educada”, llegando a faltarle el respeto a los alumnos al momento de exponer. También se define como “desconsiderada” al no tomar en cuenta los tiempos de los alumnos, por ejemplo, al dar 30 minutos para realizar la entrega de estimación económica.
- Se expresa repudio ante un profesor que, luego de un conflicto a nivel de profesores, expuso a su colega delante del grupo de alumnos, culpándolo del dilema y presentándose él como quién arregló la situación.

### **Evaluaciones, criterios y pautas:**

- Muchas de las peticiones pedidas en los ramos, como la asistencia o los informes, no generaron mucho aporte para el desarrollo del proyecto, incluso fueron las razones que generaron complicaciones durante el desarrollo.
- Fijar el tamaño del proyecto a un mínimo de “20 historias de usuario” fue arbitrario

y dio a los equipos mucho pie para crear historias de usuario “de relleno” que no aportaban al valor real del producto.

- El primer semestre se centra demasiado en informes, lo cual no permite avanzar en el proyecto en sí de forma adecuada. Si hubiésemos tenido más tiempo para desarrollar (código y no informes) el proyecto habría quedado mucho mejor.
- Algunas de estas entregas (sobre todo en Gestión de Proyectos) tienen extensiones completamente ridículas, pidiendo cosas como Cartas Gantt y autoevaluaciones por parte del jefe de proyecto que no se condicen con una metodología ágil de desarrollo, y que no hacen más que causar demoras (y posibles tensiones, en el caso de la evaluación) dentro del equipo de trabajo.
- Se debe enfocar los entregables de manera que entreguen información útil para el proyecto, puesto que la mayoría se sintió como una imposición en vez de algo que aportara valor al equipo o al proyecto.
- La regla del 10 % por cosas extras es muy subjetiva y no debería influir en la nota del 100, quizás un extra del 100-110 sería lo mejor, esto refleja que si yo hago lo que debo hacer y lo hago bien, solo puedo obtener un 90. Falta especificar qué se considera como valor agregado.
- El feedback entregado fue de muy baja calidad. Se recibió comentarios como “malo”, “ya existe”, etc. sin ahondar en detalles.
- Detallar mejor las rúbricas, puesto que son poco específicas. Debido a esto se recibieron penalizaciones que no estaban en las pautas.
- El entregable de costos es inútil, el profesor alarga la clase innecesariamente, quitando tiempo para la realización de este entregable y no explicando claramente.
- Los profesores no evalúan a los equipos de ambos campus, esto solo se da en una instancia lo cual considero insuficiente, ya que deja ver en el evento de la Feria de Software las deficiencias de cada campus.
- Se considera entendible pedir elementos de marketing, pero no que la exigencia sea al nivel de un experto, puesto que en la carrera no se enseña cómo realizarlo. Además, no se considera correcto que dicho marketing esté sujeto a invertir dinero por parte de los miembros de cada equipo.

## **Evento y premiaciones:**

- Falta preparación para las entrevistas, cómo lidiar con cierto tipo de personas, etc.
- El día mismo puede llegar a ser muy caótico, además en esta feria en específico hizo mucho calor y el aire acondicionado no daba abasto, esto junto a lo cerrado del espacio que tienen los auditorios provocó que nos deshidratáramos mucho en el transcurso del día.
- Fue un problema la organización a nivel académico, diciéndonos el mismo día de feria que teníamos que entregar documentos en un formato que no estaba establecido y cuando nos tenían restringida la movilidad del stand, agregando pautas que en un comienzo nunca se mencionaron. No se respetó el límite de los expositores en varios casos, donde el(la) promotor(a) se encontraba junto a las puertas captando gente. Falta establecer una cantidad límite de promotores.
- Los premios generan gran controversia, donde algunos pueden ser muy “manipulables”, como lo es el premio por público. Si bien exigen ciertos criterios a respetar, por ejemplo, la cantidad de promotores o el espacio del stand, al momento de la feria no se revisa o penaliza a quienes incumplen tales reglas.
- El evento final me pareció que no cumplió ningún fin como aprendizaje, fue exigente, cansador, caro y no aprendí nada, aunque me parece muy conveniente para los fines publicitarios de la universidad post PSU.
- Para la cantidad de gente que asiste a la feria, más los expositores, el aire se sentía muy pesado, por lo que podría haber sido ideal el tener alguna fuente de ventilación.
- Limitar la cantidad de premios que se le entrega a cada grupo, si la mayoría se los lleva uno, entonces los otros no tienen capital.
- Eliminar el premio al rendimiento académico ya que la nota del informe es bastante subjetiva y muchas veces el equipo que se gana el premio da pena (HolSense año 2017).
- “No le encontré sentido a tener jueces separados por categoría, si al final sólo habrá un ganador. También podría haber 2do y 3er lugar para este premio, estaría muy bien”.
- Mayor transparencia con respecto a los votos de los jurados, no con intenciones de saber si está bien el conteo, sino para saber que opinan las personas que pasaron por el stand desde un punto de vista industrial.

- Se deberían premiar a los 3 primeros lugares de cada apartado repartiendo así el monto actual de manera proporcional, evitando así que exista un gusto amargo en la mayoría de los equipos y ayudando, aunque sea, a subsanar la inversión que sale de los bolsillos de los participantes. Convertir a la feria en una competencia por montos de dinero ostentosos para nuestros bolsillos estudiantiles solo creará un ambiente más hostil de lo que necesita ser.
- Podrían hacer la feria el sábado, ya que más gente podría asistir, en especial los familiares, puesto que en algunos casos la familia no pudo asistir.
- Distribuir mejor los stands ya que la mayoría de la gente está en el centro
- No se considera la inclusión para la preparación. En un futuro se debería pensar cómo disponer la exposición para que pueda disfrutarla toda persona, más aún cuando hay proyectos relacionados a la inclusión. Si no se enseña muy pocos estudiantes lo tendrán en cuenta.
- Se dio plazo para salir a almorzar en la peor hora, cuando el casino estaba lleno con todos los otros alumnos de la universidad.
- El premio de innovación no tiene respaldo claro, considero que los últimos años el premio ha sido mal otorgado (este año incluido).
- El cocktail debiera estar servido en mesas posicionadas previamente en forma de cuadrado o círculo e ir reemplazando las bandejas vacías, ya que con garzones se genera que los alumnos se amontonen en la salida de estos y no se distribuya de manera homogénea. Además, hay que elegir mejor el lugar, ya que cualquier persona que iba pasando podía sacar. Deberían dejar de fijarse en que sea un cocktail caro y que sea llenador. Los informáticos de por sí comen mucho, más si han estado parados todo el día.
- Definan de mejor forma los horarios de cierre de votaciones y hora de la chala de cierre, esto para que los alumnos no tengan que estar hasta tan tarde desarmando su stand.
- Se señala la mala atención a los invitados en el evento, puesto que las charlas quedaron a cargo de los ayudantes, sin ningún profesor que supervisara el asunto.

### **Financiamiento de la asignatura:**

- Una debilidad de la asignatura es el costo que involucra, no siempre hay retorno de lo invertido.
- Sería ideal orientar mejor a los alumnos sobre los financiamientos, ya que incluso dentro de la misma universidad, las otras instituciones que aportan fondos no respetan los contratos como tales y hacen casos omisos, entregando una falsa expectativa. En el caso de los fondos de la universidad, había limitaciones al intentar invertir el dinero, limitando su utilidad.
- Los requisitos del stand implican una fuerte inversión por parte de los estudiantes, que poseen pocos o nulos ingresos. Los fondos entregados por la asignatura fueron \$55.000, lo que no es suficiente considerando tarjetas de presentación, poleras, afiche, pendón, un logo del equipo en el stand, promotora, póster, etc.
- No se considera apropiado el descontar nota por factores que involucran inversión de dinero.
- Se pide adelantar desde el inicio de la asignatura cuánto se requerirá de inversión monetaria, para tener dicha cifra en cuenta al postularse a fondos.
- A pesar de existir, los fondos concursables son difíciles de cobrar, restringen y quitan tiempo.

### **Sobrecarga académica:**

- El proyecto no se desarrolló de la mejor forma posible debido a que es casi incompatible con la carga académica de esta universidad, considerando que en el mismo semestre se tienen entre 19 y 21 créditos, lo cual significa trabajar bajo presión y buscar sobrevivir al estrés.
- Fue un desafío mantener la fuerza laboral en horarios extras y la disponibilidad hacia mis compañeros.
- Su fortaleza es que ayuda para poder crear un speech y desarrollar habilidades blandas, hablar con una cantidad de gente con distintos trasfondos y distintos conocimientos ayuda a ganar confianza con el proyecto que se tiene. Sus debilidades son que la carga que se siente de la misma no es equiparable a nada visto anteriormente.

- Desde el ámbito de gestión, hubo momentos en donde quedamos con tiempo muerto y no desarrollamos, lo que generó una cola de trabajo muy cerca de las fechas de entrega, lo que sobre exigía al equipo cercano a fechas límites.
- La presentación de informes extensos de manera recurrente aleja demasiado el enfoque del desarrollo a uno ágil.
- El tiempo fue un gran problema ya que por un lado teníamos el desarrollo del proyecto y por otro, los fondos a los que teníamos que postular y las reuniones con distintas personas para buscar apoyo.
- Una debilidad son los tiempos, tanto para uno personalmente como para el equipo. Había que hacerse los tiempos para poder cumplir con las tareas, dando mucha prioridad a los desarrollos, pero donde debido también a descoordinaciones, hacen que se atrasen trabajos relacionados.
- La cantidad de entregables que se realizan a lo largo de ambos cursos se vea reducido, ya que creo que la cantidad de entregas que se deben realizar en la actualidad es completamente desproporcionada.

#### **Comentarios destacados:**

- *“Mi experiencia en Feria fue positiva desde la perspectiva de trabajo en equipo, y fue pésima como asignatura. El hecho de que intentaron enseñarnos una metodología ágil sin en realidad aplicarla me pareció algo súper negativo. Ya mencioné que mi sugerencia es implementar de lleno SCRUM, sin obligar a realizar documentación, fomentando la interacción de los equipos con sus clientes por sobre todo. Por favor, no hagan a los equipos redactar documentos gigantes si al final éstos no servirán de nada para el resto del proyecto.”*
- *“Se le da demasiado enfoque a ”desarrollar proyectos innovadores”, hasta el punto en que uno siente que el verdadero objetivo es generar publicidad para la carrera y la universidad. Uno puede tener una buena idea o proyecto que le gustaría sacar adelante, pero si no cumple con lo que el jurado opina, la idea se descarta.”*
- *“En gestión de proyectos, al momento de comentar las ideas a los profesores, estos las destrozan inmediatamente, sin una segunda opinión, ni mediante un dialogo constructivo.”*

- *“Falta que los profesores escuchen lo que los alumnos tienen que decir, y para eso debiese haber instancias que hoy no existen, aparte de esta encuesta que asumo nadie lee (o al menos nadie lee en serio).”*
- *“Convertir a la feria en una competencia por montos de dinero ostentosos para nuestros bolsillos estudiantiles solo creará un ambiente más hostil de lo que necesita ser.”*

## 5.6. Aclaraciones y Sugerencias

### 5.6.1. Mapa Simplificado del Producto

En el actual *Informe de Planificación Tentativa* lo ideal sería incorporar un *User Story Map* como el señalado por Visual Paradigm [67], en donde se distribuyen las actividades de un producto, sus tareas y las épicas asociadas a estas a lo largo de *releases* (en nuestro caso Sprints). Debido a que el uso de estas herramientas requeriría que tanto profesores, como ayudantes y alumnos aprendan a elaborar estas herramientas y tuviesen un criterio para evaluarlas, se deja este factor como tema a incorporar en un posible trabajo a futuro. Para compensar, se utilizará una versión simplificada de un *product roadmap*, que señala las historias a cumplir en cada etapa de desarrollo como se muestra en la figura 5.7. Se señalan también aquellas historias opcionales al final, a modo de tener claro las posibles historias a negociar en caso de ser necesario.

<b>Sprint 1</b>	Historia 1	Historia 3
<b>SP: 15</b>	Historia 5	
<b>Sprint 2</b>	Historia 2	Historia 4
<b>SP: 17</b>	Historia 6	
<b>Sprint 3</b>	Historia 7	Historia 8
<b>SP: 12</b>	Historia 10	
<b>Extras</b>	Historia 9	Historia 11
	Historia 12	Historia 13
	...	

Figura 5.7: Ejemplo de Mapa Simplificado del Producto a usar en documento de planificación tentativa.

## 5.6.2. Diagrama de Flujo Acumulado

Conocido en inglés como *Cumulative Flow Diagram*, consiste en un diagrama de Kanban, la cual permite controlar el avance logrado cada ciertas unidades de tiempo en un tablero determinado [62]. Debido a esto, es una herramienta ideal para controlar el avance de cada equipo de estudiantes puesto que, aplicado al Sprint Backlog, puede otorgar un reporte rápido del progreso del equipo en el actual Sprint, pudiendo revisarse semanalmente durante las reuniones con el ayudante o, idealmente, dos o tres veces por semana para asegurar un avance continuo. Un ejemplo de esta herramienta se puede apreciar en la figura 5.8, en donde se consideran 4 etapas: en backlog, en desarrollo, en verificación y terminado.

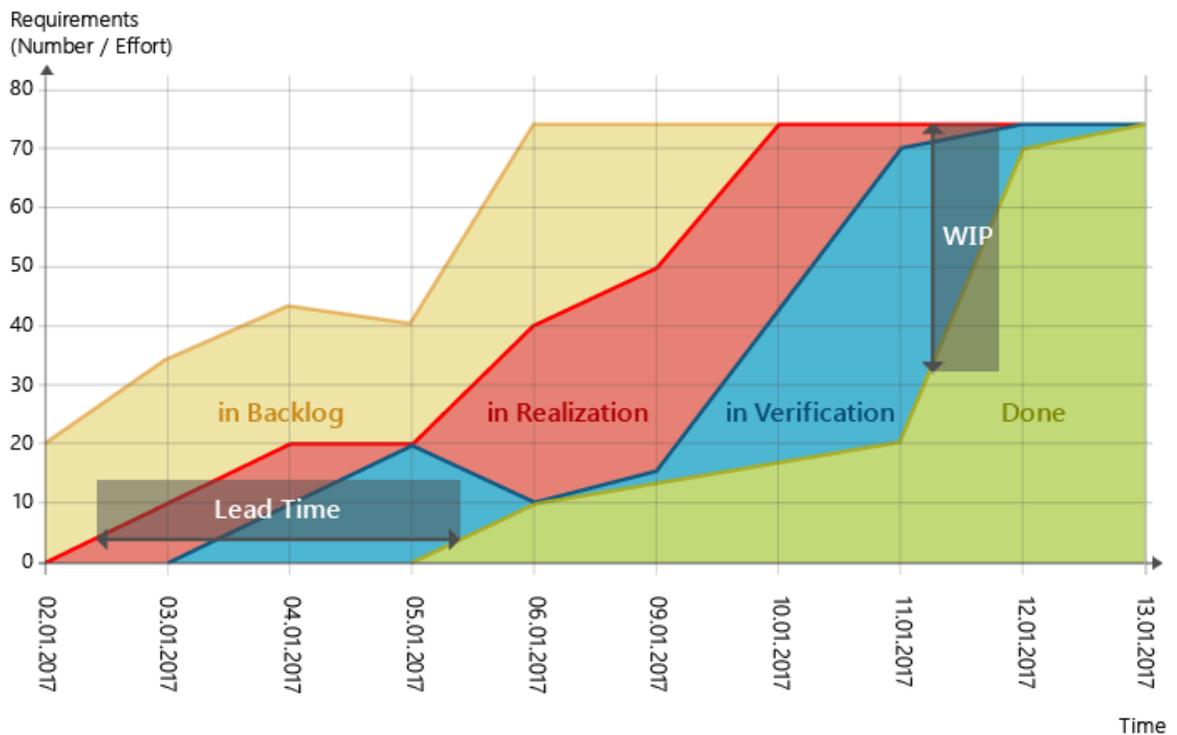


Figura 5.8: Ejemplo de Diagrama de Flujo Acumulado.

Fuente: MicroTool.de [62]

## 5.7. Comparación de Evaluaciones de Primer Semestre

### 5.7.1. Licitación de Proyectos

#### Informe de licitación (antes):

- Una portada con el nombre del producto a desarrollar, descripción de este en 7 palabras y detalle de sus integrantes (simulando una pre-empresa).
- 5 párrafos acerca del producto con:
  - El *Problema u oportunidad* a resolver y que le dan origen.
  - La *Solución propuesta* en el proyecto.
  - Un análisis a su *segmento de clientes* objetivo.
  - La *propuesta de valor* que propone el proyecto (en qué aporta el proyecto a sus clientes y/o usuarios).
  - El *grado de innovación* que este aporta (lo que lo diferencia de lo que ya hay en el mercado).
- Las palabras clave que identifican al proyecto (keywords).
- Un diagrama de Ishikawa (diagrama de causa y efecto) para identificar el problema o necesidad involucrados.
- Una descripción de la solución propuesta en el proyecto y las dependencias que posee para realizar su correcta ejecución.
- Una Descripción y cuantificación del segmento de clientes potenciales para el proyecto.
- Una Descripción y fundamentación del valor que aporta el producto a sus clientes y/o usuarios.
- El grado de innovación del proyecto respecto de su competencia global y local.
- Un mapa de empatía del contexto del cliente o usuario (para definición del segmento de clientes “Early Adopters”).
- Un lean canvas que resuma el contexto del proyecto (para análisis visual del modelo de negocio).

## **Cuentas LinkedIn (antes)**

- Foto donde se vea al alumno.
- Encabezado con datos de identificación. Experiencia (siempre con lugar y fechas) ordenada desde la última o la más importante:
  - Proyectos, breve descripción y logros alcanzados (ojalá cuantificables).
  - Asistencia a Congresos, Seminarios, etc.
  - Académica como prácticas y ayudantías.
  - Otros estudios cursados de perfeccionamiento en los últimos años.
- Descripción Personal (su definición más personal, su saber ser):
  - Competencias Específicas.
  - Competencias Transversales.
  - Participación en eventos con un rol importante.
  - Logros, como becas y otros aspectos meritorios.
  - Otras habilidades (artísticas, deportivas, de responsabilidad social, hobbies, etc.).
  - Idiomas con su nivel de manejo (ojalá acreditado).
- Recomendaciones (altamente deseable, a través de la misma red).
- Estructura similar al enlace que se adjunta en el documento a modo de ejemplo.

## **Formulario de postulación de proyectos (después):**

- Portada con nombre de equipo y empresa.
- Introducción con una descripción breve de ¿de qué se trata el proyecto?.
- Detalle de los integrantes de la empresa: nombre, campus, correo y celular de contacto.
- Enlace a currículum en LinkedIn con:
  - Foto tipo carné (opcional).
  - Descripción personal.
  - Mención de educación previa.
  - Mención de sus experiencias previas (ej: prácticas).
- Problema identificado y la solución propuesta en el proyecto.
- A quiénes estará dedicado el producto (clientes y usuarios), con una cuantificación estimada de posibles usuarios.
- Descripción de por qué es importante para sus clientes/usuarios (valor que aporta).
- La innovación que el proyecto aporta, describiendo qué lo diferencia de lo que actualmente hay en el mercado (mencionando productos que busquen solucionar lo mismo o un problema similar, total o parcialmente).
- Product Vision Board para visión general del producto (adjunto).

### **Video para jurado externo (antes):**

- El problema o necesidad identificados.
- La solución propuesta en el proyecto (el producto).
- Características relevantes de sus posibles interesados.
- La propuesta de valor que entrega dicho producto.
- Su grado de innovación y diferenciación de su posible competencia.

### **Video Pitch Online (después):**

#### **■ Contenidos del Video:**

- Descripción del problema o necesidad (responde el por qué).
- Solución propuesta (responde el qué).
- Descripción de a quienes estará destinado el proyecto y el por qué es importante para ellos (responde el para quién).
- Comparación respecto de lo que actualmente se encuentra en el mercado, mencionando su innovación.
- Descripción de cómo se sustentará el proyecto económicamente.
- Se menciona al equipo y a la feria de software.

#### **■ Estructura del Video:**

- Duración de 3 minutos  $\pm$  30 segundos.
- La apertura del video es llamativa.
- El audio es claro, sin ruido de fondo, ni música molesta.

### **Documento de identificación de historias de usuario (antes):**

- Portada con los detalles del equipo y el producto.
- Introducción al problema a resolver, solución propuesta y contenidos del resto del informe.
- Descripción de la situación actual.
- Descripción de la solución propuesta.
- Objetivo general del producto.
- Objetivos específicos del producto que respeten el estándar SMART (Specific, Measurable, Attainable, Relevant, Timely).
- Detalle de principales actores que interactuarán con el producto a desarrollar.
- Historias de usuario propuestas para el desarrollo del producto (se exigen 20) bajo el criterio de INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable).
- Requerimientos no funcionales del software.
- Conclusiones del aporte, aprendizaje y reflexión de cada miembro del equipo.
- Tabla de tiempos empleados por cada integrante en la planificación, búsqueda, análisis, desarrollo, edición reuniones y revisiones del presente documento basada en el estándar SCT de Chile [68].

### **Propuesta Técnica del Proyecto (después):**

- Portada con detalles de equipo y producto.
- Introducción describiendo los contenidos del documento.
- Descripción breve del problema que da origen al producto.
- Explicación de la Visión del producto (lo que busca llegar a ser o solucionar).
- Detalle de los tipos de usuarios que interactúan con el producto y que se mencionarán en las historias de usuario.
- Lista de historias de usuario, ordenadas desde importantes, esenciales, deseables y opcionales, descomponiendo como épicas aquellas que sean demasiado complejas o extensas, donde cada una está estimada.
- Historias de usuario seleccionadas para PMV, especificando los requisitos que necesitan cumplir para que estas se consideren terminadas.

## **5.7.2. Sprint de PMV**

### **Defensa de PMV (antes):**

- Que se presente una contextualización breve del proyecto al inicio de la presentación y las historias de usuario comprometidas en el informe de la entrega anterior, en conjunto con sus respectivos criterios de aceptación definidos en el informe entregado al ayudante.
- Que se entregue algún hecho o aporte como valor agregado en dicha presentación.
- Que el ayudante haya revisado previamente el informe con estimaciones de historias de usuario.
- Que se cumpla la entrega del reporte de retrospectiva solicitado por la coordinación, con un plazo de 2 días posterior a la presentación del PMV.

### **Sprint Review: PMV (después):**

- Se describe brevemente el proyecto.
- Se mencionan las historias de usuario que serán expuestas, mostrando en qué parte del software se presentarán.
- Demostración de lo desarrollado en el software y defensa de lo expuesto ante observaciones de los profesores.
- Historias presentadas en documento complementario a defensa son revisadas y aprobadas, siendo re-correctas y validadas aquellas con observaciones.
- Presentación de elementos adicionales, en caso de existir.

### **Informe de retrospectiva de primer semestre (antes):**

- Tabla de tiempos SCT empleados en el sprint.
- Detalle de qué se acordó a entregar.
- Línea de tiempo de los eventos relevantes a lo largo del sprint.
- 3 párrafos con qué funcionó bien, qué no funcionó bien y qué podría mejorarse respectivamente.
- Conclusiones individuales con el aporte y aprendizaje de cada alumno.
- Una evaluación por parte del jefe de proyecto en donde se le asigna nota a cada integrante según su trabajo y avance.

### **Sprint Retrospective (después):**

- 3 párrafos: lo que funcionó bien, mal y qué podría mejorarse, donde los criterios en cada uno son:
  - Todo el puntaje si se dan respuestas junto a su reflexión o justificación.
  - Medio puntaje si solo mencionan temas a modo de lista.
  - Sin puntaje si se encuentra vacío o posee respuestas básicas como “no hubo problemas”.

### **Evaluación económica del proyecto (antes):**

- Estimar el total de inversión del proyecto, señalando cantidad de ítem utilizados, el costo unitario y total de estos en UF y el costo unitario y total en pesos CLP considerando:
  - Inversiones fijas tangibles.
  - Inversiones fijas intangibles.
  - Sueldos personales.
  - Sueldos por gestiones.
  - Gastos de puesta en marcha.
- Señalar el aprendizaje en esta entrega.
- Entregar una reflexión acerca de lo visto en la presente sesión.

### **(Descartado)**

Dichos contenidos ya son evaluados en otra asignatura, por lo que es redundante incorporarlos en la presente planificación.

Sus contenidos se reemplazan por un planteamiento del modelo de negocios a lo largo de las entregas del primer semestre.

**Nota:** La propuesta a implementar por los profesores de la universidad incorpora esta entrega como un elemento adicional en el posterior documento de planificación tentativa. Debido a que no fue parte de la propuesta inicial, no se incorpora en el presente documento.

### 5.7.3. Planificación del proyecto

#### Informe de Plan de Proyecto (antes):

- Información del equipo, una frase descriptiva del producto y los logos de ambos.
- 5 párrafos acerca del producto con:
  - El *Problema u oportunidad* a resolver y que le dan origen.
  - La *Solución propuesta* en el proyecto.
  - Un análisis a su *segmento de clientes* objetivo.
  - La *propuesta de valor* que propone el proyecto.
  - El *grado de innovación* que este aporta.
- Un detalle del cliente que representa al proyecto.
- Un lean canvas y un mapa de empatía.
- Una descripción de la solución propuesta por el producto y un diagrama que lo explique.
- El objetivo general del producto y sus objetivos específicos (mostrando por qué son SMART).
- Los factores críticos o requisitos que son necesarios para que el proyecto tenga éxito.
- Una lista de todas las historias de usuario identificadas para el software, junto con su estimación y sus respectivos criterios de aceptación.
- Una estructura detallada que muestre, de forma gráfica, cómo se ha distribuido y distribuirá la realización de estas historias de usuario a lo largo del año.
- Una carta Gantt que que muestre cómo será la distribución de tiempos que se invertirán en cada historia a entregar en el siguiente semestre.
- Un detalle de los costos asociados al proyecto, suponiendo 9 meses con 9 subproductos y que cada integrante trabaja 450 horas aproximadamente.
- Una lista de los stakeholders del proyecto (Interesados claves, usualmente involucra a posibles clientes y usuarios), detallando nombre, cargo, organización, su posible rol en el proyecto, nivel de conocimiento de los temas asociados al mismo, su apoyo actual a este y su influencia en su éxito.
- Una lista con los actores o roles que posee el proyecto (tipos de usuarios), con una descripción breve y sus principales tareas.

- Un detalle de cada miembro del equipo y las cualidades que este posee, junto con un diagrama que muestre cómo se relacionan entre sí considerando sus roles y responsabilidades.
- Una lista de acciones de marketing a realizar en redes sociales o en otros medios y cómo planean influir para subir sus respectivos indicadores además de los elementos que planean utilizar el día del evento final (stand, trípticos, tarjetas, etc.).
- 12 riesgos importantes que puedan afectar el éxito del proyecto, siendo categorizados y fundamentados, detallando un plan de contingencia para los 3 más importantes.
- Conclusiones individuales por cada integrante del equipo de mínimo media página cada uno y tiempos SCT empleados en el desarrollo del informe.
- Una calificación por parte del líder de proyecto a sí mismo y a sus compañeros que represente su aporte total al desarrollo y al equipo.

#### **Informe de Planificación Tentativa (después):**

- Portada con logo tentativo del producto y nombre de documento seguida de descripción breve en 1 párrafo del producto, nombre y logo de la empresa e integrantes.
- 5 párrafos: problema, solución, segmento objetivo, propuesta de valor e innovación.
- Descripción del cliente del proyecto.
- Lista de riesgos técnicos importantes y requisitos críticos que puedan afectar el éxito del proyecto, especificando un plan de contingencia para los más probables y relevantes.
- Descripción de los tipos de usuario que interactúan con el producto
- Lista de historias de usuario ordenadas por valor y complejidad de forma descendente (primero las historias críticas, las fundamentales, las deseables, opcionales, etc).
- Tabla resumen que indique una distribución tentativa de las historias a lo largo de 3 Sprints.
- Detalle de cómo se planea sustentar el proyecto financieramente (modelo de negocios).
- Lean canvas para contextualización del proyecto.

## 5.8. Comparación de Evaluaciones de Segundo Semestre

### 5.8.1. Marketing

Entrega de sitio Web y Video (antes):

■ **Sección I. Accesibilidad:**

- Diseño Responsivo del sitio Web.
- Uso adecuado de contraste colores, texto, fuentes y fondos.
- Tiempo de carga del sitio es razonable.

■ **Sección II. Identidad:**

- Logos del producto (y pre-empresa) fácilmente identificables en todo el sitio Web.
- Página principal asimilable asertivamente.
- Información clara del equipo.
- Información del equipo y pre-empresa con forma de contacto.

■ **Sección III. Navegación:**

- Menú principal de navegación fácilmente identificable.
- Etiquetas de navegación claras y concisas.
- Cantidad razonable de botones y enlaces.
- Enlaces consistentes y fáciles de identificar.
- Video del producto incrustado.
- Contenido crítico por encima del doblez.

■ **Sección IV. Contenido:**

- Títulos principales claros y descriptivos.
- Estilos y colores consistentes con su imagen corporativa.
- Descripción clara del producto y keywords (doble puntaje).
- Texto conciso y explicativo.
- Habilitan métricas de Google (Google Analytics).

■ **Sección V. Valor Agregado:**

- Agregan valor por Creatividad, Innovación o Aporte Extraordinario (doble puntaje).

■ **Sección VI. Video:**

- El Discurso o Pitch del video es claro y entendible (narración orientada hacia satisfacer a los futuros clientes y/o usuarios).
- Contextualiza el problema a resolver y como se soluciona a través del producto.
- La narración es asertiva, acorde al segmento de clientes, y volumen adecuado.
- Tiempo de Duración del Video (1:30 ± 15 segundos).
- Apelan a las Emociones de quién observa el video.
- Usan Música y Elementos Multimedia, haciéndolo entretenido y asertivo.

■ **Sección VI. Descuentos:**

- Se aplicarán descuentos por: ausencia de contenido, animaciones continuas, flash para la construcción de todo el sitio, links rotos, errores ortográficos, y funcionalidades inapropiadas o invasivas.

## **Sitio Web Publicitario (después):**

### **■ Accesibilidad:**

- Tiempo de carga del sitio es razonable.
- Adecuado contraste entre texto y el fondo.
- Tamaño de fuente / espaciado fácil de leer.
- Uso moderado de animaciones y otros complementos.

### **■ Identidad:**

- Logos fácilmente identificables.
- Descripción clara del producto y su propósito.
- Información de la empresa claramente indicada.
- Menciona a los miembros de la empresa incluyendo foto tipo carné de sus integrantes.
- Información de contacto claramente indicada.

### **■ Navegación:**

- Menú principal de navegación fácilmente identificable.
- Etiquetas de navegación claras y concisas.
- Logo del producto vincula a la página principal.
- Enlaces son consistentes y fáciles de identificar.
- Página principal asimilable en 5 segundos.

### **■ Contenido:**

- Títulos claros y descriptivos.
- Contenido crítico visible sin scroll vertical.
- Estilos y colores consistentes.
- Texto conciso y explicativo.
- Video pitch incorporado en la página.
- Incluye enlaces a redes sociales principal y secundaria.

### **Informe de marketing (antes):**

- Tabla con Datos del proyecto y equipo:
  - Nombre y logo del proyecto.
  - Nombre y logo de la pre-empresa.
  - Descripción breve del proyecto.
  - Jefe de proyecto, encargado de Marketing y resto del equipo.
  - Campus en donde está el equipo.
- Enlaces a redes sociales de:
  - Pagina Web en feriadesoftware.cl.
  - Página de facebook.
  - Página en otra red social.
  - URL del video.

### **(Redundante)**

Dichos contenidos ya fueron incorporados en la evaluación de Sitio Web.

### **Video en Sitio Web (antes):**

El video se evalúa en el sitio Web:

- El Discurso o Pitch del video es claro y entendible (narración orientada hacia satisfacer a los futuros clientes y/o usuarios).
- Contextualiza el problema a resolver y como se soluciona a través del producto.
- La narración es asertiva, acorde al segmento de clientes, y volumen adecuado.
- Tiempo de Duración del Video (1:30 ± 15 segundos).
- Apelan a las Emociones de quién observa el video.
- Usan Música y Elementos Multimedia, haciéndolo entretenido y asertivo.

### **Video Comercial del Producto (después):**

#### ■ **Estructura del video:**

- 3 minutos ± 30 segundos
- La apertura es llamativa
- Audio es claro, sin ruido de fondo, ni música molesta
- Se muestra el software en el video

#### ■ **Contenidos del video:**

- Descripción del problema o necesidad (responde el por qué)
- Solución propuesta (responde el qué)
- Comparación respecto de lo que actualmente se encuentra en el mercado, mencionando su innovación
- Se menciona al equipo y a la feria de software

## **5.8.2. Evaluaciones de Final de Sprint**

### **Entregable de Desarrollo (antes):**

- Contextualización breve del proyecto al inicio de la presentación y exposición de las historias de usuario comprometidas.
- (Sólo Sprint 3) Se realiza un recorrido por toda la aplicación, a modo de casos de prueba.
- Algún hecho o aporte como valor agregado.
- Descuentos por mala interfaz, poca claridad u otros factores.

### **Sprint Review (después):**

- Se describe brevemente el proyecto.
- Se mencionan las historias de usuario que serán expuestas, mostrando en qué parte del software se presentarán.
- Demostración de lo desarrollado en el software y defensa de lo expuesto ante observaciones de los profesores.
- Presentación de elementos adicionales, en caso de existir.

### **Reporte de Sprint Review y Sprint Restrospective (antes):**

- Tabla con datos del equipo, del producto y de las historias de usuarios expuestas en la revisión contando cuales fueron completadas, cuales tuvieron observaciones y cuales fueron incompletas.
- Sprint Review. Respecto de lo desarrollado en el proyecto:
  - ¿Qué acordaron cumplir en el actual sprint?
  - ¿Qué se logró desarrollar?
  - ¿Qué presentó problemas?
- Sprint Retrospective. Respecto del desempeño y labor de los miembros del proyecto:
  - ¿Qué debería empezar a hacer el equipo?
  - ¿Qué se debería dejar de hacer en el equipo?
  - ¿Qué se debería mantener en el equipo?

### **Documento de Sprint Review (después):**

**Redundante:** Ya se realiza un Sprint Review presencial, no es necesario redactarlo.

### **Documento de Sprint Retrospective (después):**

- 3 párrafos: lo que funcionó bien, mal y qué podría mejorarse, donde los criterios en cada uno son:
  - Todo el puntaje si se dan respuestas junto a su reflexión o justificación.
  - Medio puntaje si solo mencionan temas a modo de lista.
  - Sin puntaje si se encuentra vacío o posee respuestas básicas como “no hubo problemas”.

### **Rúbrica de Product Owner (antes):**

- Tabla con información general del equipo detallando nombre del ayudante, del Producto y la pre-empresa, fecha de reunión y lugar, hora de inicio y término de cada reunión.
- Tabla de asistencia de cada miembro, justificando inasistencias.
- Avance logrado según lo comprometido en el sprint.
- Problemas con el avance y propuesta factible de solucionarlos.
- Trabajo en equipo: asistencia, participación, integración, relaciones, etc.
- Observación a relación entre equipo y cliente o usuario premium.
- Evaluación general del product owner.

### **(Redundante)**

Dicha evaluación no aporta valor o información que otras entregas ya presenten, por lo que se descarta.

### **Presentación de posters (antes):**

- El diseño invita a la lectura, la cual es fácil de realizar.
- Contextualiza el problema a resolver.
- Muestra aspectos claves de la solución o producto de software que lo resuelve.
- Explica el segmento de clientes, usuarios o beneficiarios.
- Presenta los aspectos innovadores de la solución lograda.
- Describen brevemente metodología y herramientas utilizadas.
- Usan código QR e invita a visitar páginas en Redes Sociales.
- Encabezado del póster incluye logos USM, DI, Feria de Software y producto.
- Al pie del póster, se identifican con sus nombres y agradecen apoyos recibidos.
- Agregan Valor por Creatividad, Innovación o Aporte Extraordinario.

### **(Reemplazado)**

La finalidad del póster es la de publicitar al proyecto y al evento feria de software, pero esto resulta costoso y muy limitado puesto que involucra implementar un diseño llamativo, pagar por su impresión e instalarlo en algún lugar de la universidad, en donde puede ser dañado o donde pudiese pasar desapercibido por quienes pasen por el sector. En consecuencia, la presente propuesta reemplaza el póster por el periodo de puesta en marcha, el cual es una actividad en donde se logra publicitar cada producto, publicitar la feria de software y, de paso, ayudar a los alumnos a tener un acercamiento a la experiencia de describir su aplicación a una audiencia, tal como se realizará posteriormente en el evento.

### **Reporte de Marketing Digital (antes):**

- Análisis del comportamiento de los usuarios en el sitio Web de feria de software, considerando las estadísticas que google analytics ofrece.
- Análisis al video del producto presente en youtube, considerando tiempo de visualización, duración media de reproducciones, visualizaciones, cantidad de Me Gusta y comentarios.
- Analizar la interacción de usuarios con la página de facebook creada para el producto, en donde se analizan las estadísticas que este mismo entrega.
- Presentar un análisis a cualquier otra red social utilizada, presentando el mismo análisis mencionado anteriormente.
- Presentar cualquier otra mención en medios tradicionales, considerando imágenes o enlaces a evidencias que lo respalden.

### **Reporte de Marketing Digital (después):**

- **Tabla con Datos del proyecto y equipo:**
  - Nombre y logo del proyecto.
  - Nombre y logo de la pre-empresa.
  - Descripción breve del proyecto.
- **Video comercial:**
  - Presentar estadísticas del video promocional
  - Analizar los resultados relevantes o la falta de ellos
- **Red Social Principal:**
  - Presentar estadísticas de la página del producto
  - Analizar los resultados relevantes o la falta de ellos
- **Red Social Secundaria:**
  - Presentar estadísticas de la página del producto
  - Analizar los resultados relevantes o la falta de ellos
- **Menciones adicionales (opcional):**
  - Presentar cualquier mención del producto en algún otro medio no mencionado, con alguna evidencia que respalde a esta (+5 ptos. a esta entrega).
  - Mencionar si el equipo considera realizar algún tipo de concurso para el día del evento.

### 5.8.3. Evento Feria de Software

#### Evaluación del Jurado y del cliente (antes):

- **Accesibilidad y navegación:** La apariencia de la aplicación es apropiada y esta es fácil de usar.
- **Calidad de la aplicación:** Producto no presenta imprevistos y posee tiempos de carga apropiados.
- **Enfoque de la solución:** Se define apropiadamente lo que pretende solucionar el producto, considerando sus clientes y/o usuarios.
- **Ambientación:** Se promueve el producto de forma buena y creativa.
- **Expositores:** Disponibilidad y buena comunicación de estos. Buena argumentación.
- **Propuesta de Valor:** El valor del producto para sus clientes objetivo, destacando su diferenciación de la competencia.
- **Proyección:** Claridad en futuras funcionalidades del producto que le permitan penetrar en el mercado y/o potenciar su uso.
- **Innovación:** Introduce alguna novedad en un ámbito, contexto o producto.

#### Evaluación del Jurado y del cliente (después):

- **Calidad de la aplicación:** Producto no presenta imprevistos y posee tiempos de carga apropiados.
- **Enfoque de la solución:** Se define apropiadamente lo que pretende solucionar el producto, considerando sus clientes y/o usuarios.
- **Discurso:** El discurso es simple y fácil de entender. Se demuestra dominio del tema..
- **Expositores:** La actitud de los expositores es de cordialidad y demuestra interés por llamar la atención del oyente.
- **Propuesta de valor:** Queda claro qué es lo que lo diferencia de lo que actualmente hay en el mercado.
- **Proyección:** Hay claridad acerca de futuras funcionalidades que pudiese tener el producto para otorgar mayor valor a sus usuarios.
- **Innovación:** Introduce alguna novedad en el contexto actual del producto.

## **Evaluación del Grupo Coordinador (antes):**

- **Calidad asociada al stand:**
  - Imagen corporativa reflejada en mantel, pendón y vestimenta del equipo.
  - Equipamiento adecuado para hacer una buena demostración del producto.
  - Otros elementos de ambientación interesantes.
- **Equipo y personas:**
  - Portan credenciales y tarjetas de presentación que los individualizan.
  - Aporte y/o presencia del cliente-usuario en el evento.
  - Se observa trabajo en equipo y colaborativo respetando buenas prácticas.
- **Elementos de difusión en feria:**
  - Ofrecer flyers, regalos y/o dulces por la visita al stand.
  - Promueven su stand y producto a través de elementos llamativos.
  - Códigos QR entregan buena información complementaria.
- **Atención a los expositores:**
  - Cordialidad: atentos para tratar y hacer grata la experiencia al visitante.
  - Compromiso: puntualidad, tener todo listo en revisiones previas, etc.
  - Disponibilidad de los expositores en el transcurso de la feria.
- **Funcionamiento del producto en la feria**
- **Agregan valor por creatividad, innovación o aporte extraordinario**

## **Evaluación del Grupo Coordinador (después):**

### **■ Calidad y elementos del stand:**

- Imagen del producto y la empresa están presentes y son fáciles de identificar.
- Hay cómo demostrar el funcionamiento del producto.
- Se muestra en alguna parte las redes sociales del equipo.
- El stand demuestra dedicación en su decoración.

### **■ Calidad del equipo y la presentación:**

- Se observa buena coordinación y colaboración entre los miembros del equipo.
- La actitud de los expositores es de cordialidad y demuestra interés por llamar la atención del oyente.
- El discurso es simple y fácil de entender. Se demuestra dominio del tema.
- El stand presenta al menos 3 miembros del equipo en todo momento.
- Presentan medios para que el visitante pueda seguir en contacto con el equipo.

### **■ Producto:**

- El producto funciona correctamente en la demostración (x3).

## 5.8.4. Documento Legado

### Documento Legado - individual (antes):

- Aprendizajes: Responder lo aprendido bajo los siguientes contextos.
  - Lecciones técnicas.
  - Al trabajo en Equipo.
  - Principal Problema o Dificultad enfrentado.
  - Respecto a la Gestión del Proyecto.
- ¿Es usted el o la Líder del Proyecto? (Si/No)
- Fortalezas y Debilidades: Señalar claramente cuáles son Fortalezas y cuáles son Debilidades respecto...
  - De la formación recibida a lo largo de la carrera para enfrentar este desafío.
  - De las asignaturas Gestión de Proyectos y Taller de Desarrollo.
  - De la Feria de Software.
- Evaluación. Considerando los siguientes ítems (...). Califique a cada integrante con PUNTAJE entre uno (1) y cinco (5), siendo 1 el peor y 5 el mejor.
  - Líder de Proyecto y Puntaje
  - Nombre de cada Integrante y Puntaje
  - Que nota se pone Ud. por su propio aprendizaje general logrado (0 a 10)
- Comentarios Generales y Sugerencias.

## **Documento Legado - grupal (después):**

### **■ Acerca del proyecto:**

- Breve descripción del proyecto.
- Descripción de la experiencia de llevarlo a cabo (expectativas, desarrollo, etc.).
- Resultado y futuro del proyecto.

### **■ Introspección:**

- ¿qué se hizo bien en el equipo durante el año?
- ¿qué salió mal en el equipo durante el año?
- Expectativas iniciales de la asignatura y sus resultados.

### **■ Sugerencias a futuros equipos:**

- Sugerencias profesionales: Respecto de la interacción con miembros externos al equipo.
- Sugerencias administrativas: Respecto de cómo llevar a cabo la organización interna del equipo.
- Sugerencias técnicas: Respecto del uso de tecnologías y herramientas.
- Referencias recomendadas: Enlaces y datos que puedan ser de utilidad para futuros proyectos.

## **Feedback Académico (nuevo)(No involucra nota)**

- Respecto de los contenidos y estructura de la asignatura.
- Respecto de los profesores y la coordinación.
- Respecto de las evaluaciones y pautas.
- Respecto del evento y las premiaciones.
- Comentarios adicionales.

# Bibliografía

- [1] Craig Larman. *Agile and iterative development: a manager's guide*. Addison-Wesley Professional, 2004.
- [2] Marcela Ziede Bize. *Capstone project: Evaluación de productividad y efectividad de equipos. El caso de ingeniería civil industrial, universidad católica del norte*.
- [3] Diane E Strode and Jill Clark. Methodology in software development capstone projects. *Proceedings of the 20th Annual Nafional Advisory Committee on Compufing Qualificafions*, pages 243–251, 2007.
- [4] Laurie Williams, Sarah E Smith, and Michael Rappa. Resources for agile software development in the software engineering course. In *Software Engineering Education & Training, 18th Conference on*, pages 236–238. IEEE, 2005.
- [5] Sue Chard, Brenda Lloyd, DE Strode, and Nick Wempe. Student industry projects: Streamlining the process for a win-win. *Proc. Eighth Annual New Zealand Association for Co-operative Education NZACE*, 2004.
- [6] Tony Clear, Michael Goldweber, Frank H Young, Paul M Leidig, and Kirk Scott. Resources for instructors of capstone courses in computing. *ACM SIGCSE Bulletin*, 33(4):93–113, 2001.
- [7] Nimal Jayaratna. *Understanding and evaluating methodologies : NIMSAD, a systemic framework / Nimal Jayaratna*. McGraw-Hill London ; New York, 1994.
- [8] Winston W Royce. Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering*, pages 328–338. IEEE Computer Society Press, 1987.
- [9] Philippe Kruchten. *The rational unified process: an introduction*. Addison-Wesley Professional, 2004.
- [10] Robert F Roggio. A model for the software engineering capstone sequence using the rational unified process®. In *Proceedings of the 44th annual Southeast regional conference*, pages 306–311. ACM, 2006.

- [11] P. N. Robillard, P. Kruchten, and P. d’Astous. Yoopeedoo (upedu): a process for teaching software process. In *Software Engineering Education and Training, 2001. Proceedings. 14th Conference on*, pages 18–26, 2001.
- [12] PMBOK Guide. A guide to the project management body of knowledge. In *Project Management Institute*, volume 3, 2004.
- [13] Kathy Schwalbe. *Information technology project management*. Cengage Learning, 2015.
- [14] Liz Adams, Mats Daniels, Annegret Goold, Orit Hazzan, Kathy Lynch, and Ian Newman. Challenges in teaching capstone courses. In *ACM SIGCSE Bulletin*, pages 219–220. ACM, 2003.
- [15] Watts S Humphrey. *Introduction to the personal software process (sm)*. Addison-Wesley Professional, 1996.
- [16] Watts S Humphrey. *The Team Software Process (sm)(TSP (sm))*. Carnegie Mellon University, Software Engineering Institute, 2000.
- [17] Richard Conn. A reusable, academic-strength, metrics-based software engineering process for capstone courses and projects. In *ACM SIGCSE Bulletin*, pages 492–496. ACM, 2004.
- [18] DE Strode. Agile methods: a comparative analysis. In *Proceedings of the 19th annual conference of the national advisory committee on computing qualifications, NACCCQ*, volume 6, 2006.
- [19] Kent Beck and Erich Gamma. *Extreme programming explained: embrace change*. addison-wesley professional, 2000.
- [20] Ken Schwaber and Mike Beedle. *Agile software development with Scrum*, volume 1. Prentice Hall Upper Saddle River, 2002.
- [21] Mary Poppendieck and Tom Poppendieck. *Lean Software Development: An Agile Toolkit: An Agile Toolkit*. Addison-Wesley, 2003.
- [22] Jim Highsmith. Adaptive software development. *Dorset House*, 2000.
- [23] Alistair Cockburn. *Agile software development*, volume 177. Addison-Wesley Boston, 2002.
- [24] Peter Coad and S Palmer. Feature-driven development. *Java Modeling in Color with UML*, pages 182–203, 1999.
- [25] Jennifer Stapleton. *DSDM, dynamic systems development method: the method in practice*. Cambridge University Press, 1997.

- [26] Diane Elizabeth Strode. The agile methods: An analytical comparison of five agile methods and an investigation of their target environment. *Unpublished Master of Information Sciences (Information Systems), Massey University, Palmerston North*, 2005.
- [27] Manifesto for Agile Software Development. <https://agilemanifesto.org/>, 2016.
- [28] Michele Marchesi, Giancarlo Succi, Don Wells, Laurie Williams, and James Donovan Wells. *Extreme programming perspectives*, volume 176. Addison-Wesley, 2003.
- [29] Charlie McDowell, Linda Werner, Heather E Bullock, and Julian Fernald. Pair programming improves student retention, confidence, and program quality. *Communications of the ACM*, 49(8):90–95, 2006.
- [30] Noel F LeJeune. Teaching software engineering practices with extreme programming. *Journal of Computing Sciences in Colleges*, 21(3):107–117, 2006.
- [31] David A Umphress, T Dean Hendrix, and James H Cross. Software process in the classroom: The capstone project experience. *IEEE software*, 19(5):78–81, 2002.
- [32] Karen Keefe and Martin Dick. Using extreme programming in a capstone project. In *Proceedings of the Sixth Australasian Conference on Computing Education-Volume 30*, pages 151–160. Australian Computer Society, Inc., 2004.
- [33] Yael Dubinsky and Orit Hazzan. The role of a project-based capstone course. In *Proceedings of the 27th international conference on Software engineering*, pages 645–646. ACM, 2005.
- [34] Scott Ambler. *Agile modeling: effective practices for extreme programming and the unified process*. John Wiley & Sons, 2002.
- [35] Owen Astrachan, Robert C Duvall, and Eugene Wallingford. Bringing extreme programming to the classroom. In *XP Universe*, volume 2001. Citeseer, 2001.
- [36] David H Johnson and James Caristi. Extreme programming and the software design course. In *Proceedings of XP Universe*. Citeseer, 2001.
- [37] Rick Mugridge, Bruce MacDonald, Partha Roop, and Ewan Tempero. Five challenges in teaching xp. In *International Conference on Extreme Programming and Agile Processes in Software Engineering*, pages 406–409. Springer, 2003.
- [38] Christian Bunse, Raimund L Feldmann, and Jörg Dörr. Agile methods in software engineering education. In *International Conference on Extreme Programming and Agile Processes in Software Engineering*, pages 284–293. Springer, 2004.
- [39] Dean Sanders. Student perceptions of the suitability of extreme and pair programming. *Extreme programming perspectives*, pages 168–174, 2002.

- [40] J-G Schneider and Lorraine Johnston. extreme programming at universities-an educational perspective. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 594–599. IEEE, 2003.
- [41] Corey Ladas. *Scrumban-essays on kanban systems for lean software development*. Lulu. com, 2009.
- [42] Ajay Reddy. *The Scrumban [r] evolution: getting the most out of Agile, Scrum, and lean Kanban*. Addison-Wesley Professional, 2015.
- [43] Viljan Mahnic. A capstone course on agile software development using scrum. *IEEE Transactions on Education*, 55(1):99–106, 2012.
- [44] Andrew Hunt and David Thomas. *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [45] Dennis Lock. Project management (9e ed.) gower publishing, 2007.
- [46] Lessons Learned: Minimum Viable Product: a guide. <http://www.startuplessonslearned.com/2009/08/minimum-viable-product-guide.html>, 2019.
- [47] *Programa de la Asignatura Gestión de Proyectos de Informática (INF-360)*, 2018.
- [48] YouNoodle — The #1 Startup Evaluation Platform. <https://younoodle.com/>, 2019.
- [49] Francisco Javier Llamas Fernández and Juan Carlos Fernández Rodríguez. La metodología lean startup: desarrollo y aplicación para el emprendimiento. *Revista EAN*, pages 79–95, 2018.
- [50] XXVI Feria de Software — Departamento de Informática UTFSM. <http://www.feriadesoftware.cl/>, 2018.
- [51] Ayesha Amjad and Muqqadas Rehman. Resistance to change in public organization: Reasons and how to overcome it.
- [52] VersionOne 12th Annual State of Agile Report. <https://explore.versionone.com/state-of-agile/versionone-12th-annual-state-of-agile-report>, 2019.
- [53] Carol S Dweck. *Mindset: The new psychology of success*. Random House Digital, Inc., 2008.
- [54] A Guide to the Scrum Body Of Knowledge. <https://www.scrumstudy.com/SBOK/SCRUMstudy-SBOK-Guide-2016.pdf>, 2016.

- [55] Roles on Agile Teams: From Small to Large Teams. <http://www.ambyssoft.com/essays/agileRoles.html>, 2019.
- [56] I Elaine Allen and Christopher A Seaman. Likert scales and data analyses. *Quality progress*, 40(7):64–65, 2007.
- [57] Product Vision Tools Part 1: The Product Vision Board - LeadingAgile. <https://www.leadingagile.com/2014/03/Product-vision-tools/>, 2018.
- [58] 2. Product Vision Board - scrumgym. <https://sites.google.com/site/scrumgym/product-vision-board>, 2019.
- [59] What is a Minimum Viable Product (MVP)? — Agile Alliance. <https://www.agilealliance.org/glossary/mvp/>, 2019.
- [60] What does INVEST Stand For? — Agile Alliance. <https://www.agilealliance.org/glossary/invest/>, 2019.
- [61] What is Planning Poker? — Agile Alliance. <https://www.agilealliance.org/glossary/poker/>, 2019.
- [62] What is a Cumulative Flow Diagram? – Knowledge Base. <https://www.microtool.de/en/knowledge-base/what-is-a-cumulative-flow-diagram/>, 2019.
- [63] Scrum Alliance Member-Submitted Informational Articles: Product Backlog Refinement. <https://www.scrumalliance.org/community/articles/2014/october/product-backlog-refinement>, 2014.
- [64] Lean Inception. <https://martinfowler.com/articles/lean-inception/>, 2018.
- [65] Scaled Agile Framework – SAFe for Lean Enterprises. <https://www.scaledagileframework.com/>, 2019.
- [66] Ash Maurya. *Running lean: iterate from plan A to a plan that works*. .°Reilly Media, Inc.”, 2012.
- [67] What is User Story Mapping? <https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story-mapping/>, 2014.
- [68] SCT. <http://sct-chile.consejoderectores.cl/>, 2019.