



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## FACULTAD DE INFORMÁTICA

# TESINA DE LICENCIATURA

**TÍTULO:** El desafío de Scrum distribuido en diferentes locaciones

**AUTORES:** Grimaldi, Pablo David; Salazar, Joaquin Ismael

**DIRECTOR:** Dra. Pons, Claudia

**CODIRECTOR:**

**ASESOR PROFESIONAL:**

**CARRERA:** Grimaldi, Pablo - Lic. en Sistemas; Salazar, Joaquin - Lic. en Informática

### Resumen

*En las últimas décadas la tecnología ha avanzado rápidamente y con ella la forma de trabajo de todas las personas relacionadas con IT, hoy en día es totalmente normal que un equipo esté integrado por personas que están en diferentes ciudades del mundo, trabajando de manera remota o con diferentes husos horarios e idiomas. Al mismo tiempo, el uso de las metodologías ágiles; principalmente Scrum, han tenido un gran crecimiento en su implementación. Por esta razón es oportuno poder realizar un análisis de todos los desafíos que implica usar Scrum de manera distribuida, brindando además un aporte de posibles soluciones y consejos para afrontarlos.*

### Palabras Clave

*Manifiesto ágil, Scrum, Scrum Distribuido, Desafíos, Comunicación, Equipos Distribuidos, Eventos Distribuidos, Home Office.*

### Trabajos Realizados

*Se llevó a cabo una investigación sobre diferentes equipos que utilizan Scrum de manera distribuida.*

*Se realizó una encuesta entre cincuenta personas, para analizar cuáles son los desafíos y beneficios más comunes de utilizar Scrum de manera distribuida.*

*Se realizó un acompañamiento a un equipo real, identificando desafíos e implementando posibles soluciones a lo largo de los Sprints.*

*Además, la tesis fue realizada utilizando Scrum de manera distribuida.*

### Conclusiones

*Luego de recopilar toda la información necesaria acerca de la metodología ágil, de Scrum, del trabajo distribuido y de evaluar y analizar las experiencias de múltiples personas, los autores dejan evidencia de sus experiencias, aportando y detallando diferentes soluciones a los problemas más comunes de utilizar Scrum Distribuido. Con la finalidad de ver el real alcance de las soluciones planteadas, los autores identificaron un equipo que se enfrentara a alguno de los desafíos descriptos con el objetivo de acompañarlos y trabajar con ellos en la implementación de alguna de las soluciones planteadas en la tesis; realizando además, un análisis detallado del caso y los resultados obtenidos.*

### Trabajos Futuros

*Ampliar la investigación a más equipos y realizar una encuesta a un número más grande de personas.*

*Seguir acompañando al equipo con el que se trabajó, para ver el avance y estado de las soluciones planteadas.*

*Continuar buscando soluciones para los desafíos relacionados con la diferencia horaria y el idioma.*

*Hacer el experimento con otro framework y realizar comparaciones.*

Fecha de la presentación: Febrero 2022

# UNIVERSIDAD NACIONAL DE LA PLATA

## FACULTAD DE INFORMÁTICA

### TESINA DE GRADO



---

## El desafío de Scrum distribuido en diferentes locaciones

---

Autores:

Grimaldi, Pablo David

Salazar, Joaquin Ismael

Directora:

Dra. Pons, Claudia

**Febrero 2022**

## Agradecimientos

### **Pablo:**

*“A mi madre, que me apoyó desde el minuto cero para hacer este recorrido y nunca dejó de tener fe en mí a pesar de los altibajos, también a mis familiares y en especial a mi novia que me motivó constantemente para no bajar los brazos en este tramo final.”*

### **Joaquin:**

*“A mi familia que me apoyó a lo largo de todos estos años y me dieron la oportunidad de poder estudiar la carrera de Licenciatura en Informática, realizando un esfuerzo muy grande para que me mudara de mi Gualeguaychú natal a la ciudad de La Plata en plena crisis económica luego del 2001.*

*A mis abuelas, que ya no están en este plano, pero siempre prendían una velita antes de rendir un examen y depositaron toda su fe en mí.”*

### **En conjunto:**

*A Claudia nuestra directora de tesis que nos ayudó a concretar y no abandonar nuestro objetivo en el camino, aun cuando estuvimos muy cerca de hacerlo.*

*A nuestros compañeros y amigos que nos acompañaron en este recorrido.*

# ÍNDICE DE CONTENIDO

CAPÍTULO 1: Introducción.....	10
1.1 Motivación.....	10
1.2 Objetivo.....	11
1.3 Organización/Estructura.....	11
CAPÍTULO 2: Estado del arte.....	13
CAPÍTULO 3: Metodologías ágiles.....	16
3.1 Introducción.....	16
3.2 Origen de las metodologías ágiles.....	16
3.3 Historia del Manifiesto ágil.....	22
3.4 Valores del Manifiesto para el desarrollo de software ágil.....	24
3.4.1 Individuos e interacciones sobre procesos y herramientas.....	24
3.4.2 Software funcionando sobre documentación extensiva.....	25
3.4.3 Colaboración con el cliente sobre negociación contractual.....	25
3.4.4 Respuesta ante el cambio sobre seguir un plan.....	25
3.5 Conclusión.....	26
CAPÍTULO 4: Scrum.....	27
4.1 Introducción.....	27

4.2 Historia de Scrum .....	27
4.3 Definición de Scrum .....	31
4.4 Valores de Scrum .....	31
4.5 Scrum Team (El equipo Scrum).....	32
4.5.1 Product Owner (Propietario del producto) .....	33
4.5.2 Scrum Master .....	35
4.5.3 Developers (Desarrolladores).....	36
4.6 Rol por fuera del Scrum: Stakeholder (No definido en “La guía de Scrum”).....	37
4.7 Eventos de Scrum .....	37
4.7.1 El Sprint.....	38
4.7.2 Sprint Planning (Planificación de Sprint).....	39
4.7.3 Daily Scrum.....	40
4.7.4 Sprint Review (Revision del Sprint).....	41
4.7.5 Sprint Retrospective (Retrospectiva del Sprint).....	41
4.7.6 Refinement.....	42
4.8 Artefactos de Scrum .....	42
4.8.1 Product Backlog.....	43
4.8.2 Sprint Backlog.....	44

4.8.3 Increment (Incremento) .....	44
4.9 Conclusión .....	44
CAPÍTULO 5: Scrum Distribuido.....	45
5.1 Introducción .....	45
5.2 Niveles de equipos distribuidos.....	46
5.2.1 Equipo co-locado .....	47
5.2.2 Equipo parcialmente co-locado .....	48
5.2.3 Distribuido con horas superpuestas .....	49
5.2.4 Distribuido sin horas superpuestas .....	51
5.3 Modelos de equipos distribuidos.....	52
5.3.1 Scrum aislados.....	53
5.3.2 Scrum de Scrum distribuidos.....	53
5.3.3 Scrum totalmente integrados.....	54
5.4 Conclusión .....	54
CAPÍTULO 6: Desafíos de utilizar Scrum Distribuido .....	55
6.1 Introducción .....	55
6.2 Desafíos más frecuentes.....	56
6.2.1 Diferencias culturales .....	56

6.2.2 Diferencias de idioma .....	57
6.2.3 Diferencia horaria.....	57
6.2.4 Comunicación .....	58
6.2.5 Coordinación de calendario .....	58
6.2.6 Desarrollo sobre una misma funcionalidad.....	59
6.2.7 Tamaño del equipo .....	59
6.2.8 Inserción de nuevos miembros en un equipo distribuido .....	60
6.3 Eventos de Scrum distribuido y sus desafíos .....	61
6.3.1 Sprint Planning Distribuida .....	61
6.3.2 Daily Scrum Distribuida .....	61
6.3.3 Review del Sprint distribuida .....	62
6.3.4 Sprint Retrospective distribuida.....	62
6.4. Home office en Scrum Distribuido.....	63
6.4.1 Por qué home office?.....	63
6.4.2 Retos implementar Scrum con home office .....	64
6.5 Conclusión .....	66
CAPÍTULO 7: Encuesta sobre Scrum Distribuido .....	67
7.1 Introducción .....	67

7.2 Descripción de la población. ....	67
7.3 Desafíos de Scrum Distribuido.....	68
7.4 Beneficios de Scrum Distribuido .....	70
7.5 Conclusión .....	70
CAPÍTULO 8: Experiencias personales de los autores .....	72
8.1 Introducción .....	72
8.2 Proyectos distribuidos en los que trabajó Pablo:.....	72
8.2.1 Primer proyecto.....	72
8.2.2 Segundo proyecto.....	73
8.2.3 Tercer proyecto .....	74
8.2.4 Cuarto proyecto .....	75
8.3 Proyectos distribuidos en los que trabajó Joaquin: .....	76
8.3.1 Primer proyecto.....	77
8.3.2 Segundo proyecto.....	78
8.3.3 Tercer proyecto.....	79
8.3.4 Cuarto proyecto .....	80
8.4 Conclusión .....	81
CAPÍTULO 9: Planteo de soluciones.....	82



9.1	Introducción .....	82
9.2	Posibles soluciones para los desafíos más frecuentes.....	82
9.2.1	Diferencias culturales.....	82
9.2.2	Diferencias de idioma .....	83
9.2.3	Diferencias horaria.....	84
9.2.4	Comunicación .....	86
9.2.5	Coordinación de calendario.....	87
9.2.6	Desarrollo de una misma funcionalidad .....	88
9.2.7	Tamaño del equipo .....	89
9.2.8	Inserción de nuevos miembros en un equipo distribuido .....	90
9.3	Consejos para llevar adelante eventos distribuidos .....	90
9.3.1	Sprint Planning.....	91
9.3.2	Daily Scrum.....	93
9.3.3	Sprint Review .....	94
9.3.4	Sprint Retrospective.....	96
9.3	Recomendaciones en general.....	98
9.5	conclusión .....	100
CAPÍTULO 10: Implementación de la propuesta en un equipo.....		101

10.1 Introducción .....	101
10.2 El equipo y sus desafíos .....	101
10.3 Soluciones planteadas .....	103
10.4 Resultados obtenidos.....	104
CAPÍTULO 11: Conclusiones.....	106
Referencias.....	110
Anexo.....	112
Tesis realizada usando Scrum Distribuido.....	112

# CAPÍTULO 1: Introducción

## 1.1 Motivación

En las últimas décadas la tecnología avanzó rápidamente y con ella la formas de trabajo de todas las personas relacionadas con IT, ya casi no existe la idea de tener a todo un equipo de desarrollo ubicado en un mismo edificio y mucho menos que los miembros de éste concurren a la oficina todos los días. Todo ha ido virando hacia el trabajo de forma remota, desde el hogar y situados en cualquier parte del mundo.

Hoy en día es totalmente normal, que un equipo esté integrado por personas situadas en diferentes ciudades del mundo, trabajando de manera remota, con diferentes husos horarios e idiomas.

Al mismo tiempo, el uso de las metodologías ágiles en el desarrollo de software es cada vez más normal, gran cantidad de equipos alrededor del mundo utilizan algún framework agile, particularmente Scrum que es el que se va a analizar en la presente tesis.

Estos puntos mencionados (trabajo distribuido y uso de Scrum) son aspectos que los autores de la presente tesis han tenido la oportunidad de experimentar hace ya más de cinco años. Por esta razón se consideró oportuno poder realizar un análisis de todos los desafíos que implica usar Scrum de manera distribuida, brindando además un aporte de posibles soluciones y consejos para afrontarlos.

Además de esto, se plantea la idea de buscar un equipo de Scrum que esté teniendo desafíos al trabajar de manera distribuida, identificar los mismos y ayudarlos a que los puedan afrontar de manera más ágil para realizar un aporte real en base a todo lo aprendido al momento de realizar la tesis.

## **1.2 Objetivo**

El primer objetivo es realizar un análisis detallado de los desafíos que implican utilizar Scrum en equipos que se encuentran distribuidos en diferentes países y husos horarios, haciendo foco también en los que están distribuidos en la misma región pero que trabajan mediante la modalidad home office.

Por otra parte, se plantearán soluciones a los desafíos encontrados y se implementarán las mismas en un equipo de trabajo real, detallando los resultados obtenidos.

También se plantea como objetivo realizar una introducción contando la historia y el surgimiento de la metodología , para luego focalizarse en la historia de Scrum.

## **1.3 Organización/Estructura**

En el capítulo 2 se realiza un análisis del estado del arte.

En el capítulo 3 se menciona la historia de las metodologías ágiles y del Manifiesto ágil [6]; además, los valores de éste.

En el capítulo 4 se menciona la historia del surgimiento de Scrum, la definición del mismo, como así de los roles y eventos que posee el framework.

En el capítulo 5 se realiza un análisis de por qué se utiliza Scrum distribuido, así como también de los diferentes niveles y modelos de equipos distribuidos posibles.

En el capítulo 6 se plantean los desafíos de utilizar Scrum distribuido, mencionando también los desafíos de la modalidad del trabajo home office.

En el capítulo 7 se analizan los datos obtenidos de una encuesta realizada por los autores a cincuenta personas que trabajan utilizando Scrum de manera distribuida.

En el capítulo 8 los autores describen sus experiencias personales trabajando en equipos Scrum de manera distribuida.

En el capítulo 9 se plantean posibles soluciones a los desafíos mencionados en el capítulo 6, relacionados al uso de Scrum distribuido.

En el capítulo 10 se menciona y analiza la experiencia de implementar algunas de las soluciones planteadas en el capítulo 9 en un equipo de Scrum distribuido.

Finalmente, en el capítulo 11 se plantea una conclusión general del trabajo de investigación realizado y las prácticas implementadas.

Adicionalmente se agrega un anexo en el cual se detalla la forma en la cual se ha realizado la tesis, utilizando Scrum de manera distribuida.

## CAPÍTULO 2: Estado del arte

El análisis del estado del arte que aquí se desarrolla se basa en investigaciones que se han realizado acerca de Scrum y de su utilización en entornos distribuidos. La mayoría de los textos con los que se ha trabajado hacen referencia a Scrum Distribuido; pero en diferentes locaciones. Es claro que hoy en día (en especial a partir de la pandemia por el covid-19) el Scrum Distribuido mediante home office ha crecido en forma exponencial.

Esquivel en su trabajo integrador “Scrum en entornos geográficamente dispersos” [1], tuvo como objetivo la recolección de información relacionada al trabajo Scrum con equipos distribuidos geográficamente.

Para esto hizo un análisis de los diferentes casos de estudio relacionados al uso de Scrum en entornos distribuidos mencionando diferentes desafíos o problemas, todo esto limitado a equipos con miembros en diferentes ciudades. El trabajo comenta brevemente los roles, eventos y artefactos de Scrum.

La investigación se basa en diferentes artículos que documentan casos de estudio con diferentes resultados en la utilización de Scrum dentro del contexto de entornos geográficamente dispersos.

A partir de la documentación relevada, el autor formula conclusiones acerca de cómo Scrum deja al descubierto en entornos distribuidos ciertos problemas, pero también que las soluciones dependen del contexto. El autor considera importante el abordar a futuro la adopción de Scrum en una modalidad de “home office” total, siendo ese marco muy cuestionado por la industria (Lo cual hoy en día es más bien aprovechado por éstas).

Herranz, Mamoghli, Yazzi, Vera, González, Matulis, Ratón, Salas, Arauzo, Muñoz y Farias en su trabajo de investigación llamado “Scrum distribuido” [2], buscaron comprender los diferentes aspectos de implementar Scrum en un contexto distribuido, teniendo en consideración el progreso de las herramientas tecnológicas de esos años, las cuales beneficiaban al trabajo colaborativo. Partiendo de experiencias obtenidas de libros, artículos y blogs, el trabajo posee definiciones base, se menciona el beneficio, las problemáticas relacionadas y brinda unas pocas recomendaciones.

Woodward, Surdeky y Gianis en su libro “A Practical Guide to Distributed Scrum”[3], sentaron precedente para la época en cuanto a la descripción de cómo implementar Scrum distribuido en grandes empresas.

El libro se centra en las experiencias obtenidas al trabajar con Scrum en IBM, las mismas se remontan a personas que usaban teleconferencia, teléfono, etc. No existían herramientas tecnológicas como las de hoy en día.

Paasivaara, Durasiewicz y Lassenius escribieron un artículo llamado “Using Scrum in a Globally Distributed Project: A Case Study”[4], en el cual se comenta que existen múltiples proyectos chicos que aplican agile distribuido, pero no hay mucho estudio acerca de agile distribuido en proyectos grandes. Se describe cómo se aplicaron satisfactoriamente las prácticas de Scrum en una organización distribuida, y también menciona el uso de XP.

En una sección del texto se mencionan las visitas entre equipos, una técnica muy poco utilizada hoy en día, dado que muchas empresas no lo hacen por los costos que esto conlleva y por la contaminación al utilizar aviones.

Se obtuvieron buenos resultados aplicando Scrum en un equipo pequeño, pero eso hace que la investigación sea limitada y plantean en un futuro entrevistar a gente de otros equipos, diferentes compañías o con otros roles.

Por último, es muy importante destacar que Sutherland, Viktorov, Blount y Puntikov(2007) publicaron un caso de estudio llamado “Agile Project Management with Outsourced Development Teams”[5], que fue base para gran cantidad de documentos que los autores de la presente tesis fueron leyendo a lo largo del desarrollo de la misma. En este texto se mencionan 3 tipos de equipos distribuidos, de esto hace mención el libro de IBM que se ha mencionado anteriormente. El objetivo fue demostrar que; aplicando eficientemente Scrum, los equipos distribuidos pueden ser tan productivos como un pequeño equipo ubicado en el mismo lugar.

Este documento analiza y recomienda las mejores prácticas para los equipos ágiles distribuidos globalmente y menciona el por qué de que el mercado tienda a la lógica de expandirse a otros países y usar equipos distribuidos.

Por todo esto, la tesis no solo va a estar basada en visibilizar los desafíos de Scrum aplicados a equipos distribuidos, sino que se buscará complementar mostrando, escenarios reales vividos por los autores, planteo de soluciones en base a la experiencia de los mismos y todo esto en un contexto más actual a la bibliografía existente.



## **CAPÍTULO 3: Metodologías ágiles**

### **3.1 Introducción**

En el presente capítulo se describirá la historia del origen de las metodologías ágiles y como están impactaron favorablemente en el desarrollo de software a lo largo de los últimos años, así mismo se menciona la reunión en Snowbird en el año 2001 en donde el Manifiesto ágil [6] surgió.

### **3.2 Origen de las metodologías ágiles**

Los orígenes de la metodología se remontan a muchos años atrás, específicamente a la década del 30, cuando aún no se denominaba ágil o agile.

Los primeros registros que se pueden encontrar, datan de la década del 30 y hacen mención al ciclo PDCA (Plan, Do, Check, Act), propuesto por Walter Shewhart, en donde se plantea la idea de: planificar lo que se va a realizar, hacerlo, verificar y luego actuar en consecuencia.

El siguiente evento importante para destacar, sucede en los años 40, con la inclusión de los métodos Kanban, el sistema de producción de Toyota y Lean manufacturing. Éstos sentaron un precedente y son de las principales fuentes de inspiración para el movimiento ágil.

Unas décadas después, más precisamente en el año 1974, el Dr. Ernest Edmonds publica un artículo en donde se presenta el concepto de “Proceso de desarrollo de Software Adaptativo”. Asimismo, también durante los 70, Tom Gild publica conceptos sobre la Gestión de Proyectos Evolutiva (EVO).

En los años 90 se comienza a manifestar una insatisfacción muy marcada en los trabajadores de la industria del software con los resultados de los modelos tradicionales de gestión (los procesos llegaban a durar 3 años). Es así como, en el año 1992, Alistair Cockbur presenta los Métodos Crystal[9] y marca el inicio de la evolución de las metodologías de desarrollo de software.

Continuando con esta idea de la búsqueda de mejoras, en el año 1993 Bill Opdyke presenta el concepto de "Refactorización" en un texto académico denominado "Creando Superclases Abstractas por medio de la Refactorización". En el mismo se plantea la idea de reestructurar partes del código existente, alterando su estructura interna sin afectar el comportamiento del mismo.

El año 1995 se puede considerar como uno de los más importantes en base a los estudios que se publicaron y de como éstos marcaron la forma de trabajo, que es incluso utilizada hoy en día.

A principios de ese año se presenta el concepto de Programación en Pares (Pair Programming), el mismo fue simultáneamente ideado, pero de forma independiente por varios autores. Por un lado, Jim Coplien publicó un texto, en el cual definió la "Programación en Pares" como un patrón de desarrollo de software y por otro lado Larry Constantine definió los "dúos dinámicos" en su libro "Constantine on Peopleware".

Ese mismo año; en Octubre, se realiza la décima conferencia anual OOPSLA 95 (Object-Oriented Programming, Systems, Languages & Applications) en Austin (Texas), en donde Ken Schwaber y Jeff Sutherlan presentan al mundo el método Scrum.

Ya avanzado el año 97 Jeff De Luca y Peter Coad idean un enfoque de desarrollo ágil de software, denominado FDD (Feature Driven Development). En él se definen mejores prácticas tales como: Modelado de objetos de dominio, Desarrollo por funcionalidades, Propiedad individual de las clases (Código), Equipos de trabajo por funcionalidad, Inspecciones, Gestión de Configuración, Compilaciones regulares (periódicas) y visibilidad del avance y resultados.

A finales del siglo 20; en el año 1999, Jim Highsmith formaliza el concepto de “Desarrollo de Software Adaptativo” y publica un libro del mismo nombre. La idea creció y evolucionó hacia la metodología de Desarrollo Rápido de Aplicaciones (RAD), la cual propone un ciclo de vida de tres fases: Especulación, Colaboración y Aprendizaje.

Ese mismo año Kent Beck publica; con la ayuda de Ward Cunningham y Ron Jeffries, un libro titulado "Extreme Programming Explained"[11] (Programación Extrema Explicada), concepto que desarrolló mientras trabajaba en Chrysler.

Otro fuerte suceso acontecido ese mismo año, fue la definición de “Integración continua” de la mano de Kent Beck, pero fue un artículo de Martin Fowler [10] el que realmente lo popularizó.

En la búsqueda de algo más oportuno y receptivo, varios líderes de pensamiento de software comienzan a reunirse informalmente. Es así como Bob Martin, reúne a otros dieciséis líderes del movimiento ágil, para escribir; en el año 2001, el Manifiesto ágil [6], que engloba las metodologías que hasta ese momento se les conocía como "Metodologías de Desarrollo de Software de peso liviano"(historia que se desarrolla en el siguiente capítulo).

A partir de la publicación del manifiesto, se siguieron desarrollando nuevos conceptos, como por ejemplo el desarrollo guiado por pruebas (TDD) y la técnica de estimación Planning Poker, ideada por James Greening.

En el año 2003, Mary y Tom Poppendieck presentan lo que para muchos es considerado como el próximo eslabón en la evolución del desarrollo ágil, el "Lean Software Development". El mismo hace referencia a siete principios claves: eliminar desperdicio, amplificar el aprendizaje, decidir tan tarde como sea posible, entregar lo más rápido posible, dar poder al equipo, construir integridad y ver la totalidad.

Tres años más tarde; en el 2006, Dan North presenta el concepto de "Desarrollo guiado por comportamiento" (Behavior Driven Development" o BDD), un método que combina las principales ideas y técnicas del TDD con las ideas del Diseño guiado por dominio y el Análisis y Diseño orientado a objetivos.

En el 2007 se presentan dos obras importantes, la primera fue "Agile Retrospective" escrita por Esther Derby y Diana Larsen, estableciendo las reuniones retrospectivas como práctica estándar dentro del mundo ágil. La siguiente obra fue presentada por David Anderson, ésta se llamó "Kanban", método que se enfoca en la entrega "justo a tiempo" y en no sobrecargar a los desarrolladores de software (tal como su precursor el Kanban para manufactura, perfeccionado por Toyota).

Comenzando la segunda década del siglo 21; en el año 2010, se escribe la primera versión de "La guía de Scrum", que contiene la definición de Scrum, de la cual se escribirá en otro capítulo de esta tesis.

Ya en el año 2011 se introduce el concepto de SAFe, el cual en sus orígenes fue llamado “Agile Enterprise Big Picture” por el veterano de la industria del software Dean Leffingwell.

En base a los sucesos antes mencionados, se puede concluir que el desarrollo ágil es un concepto que evolucionó a partir de varios métodos que se fueron publicando a lo largo de los últimos noventa años.

En la siguiente línea de tiempo (Figura 1), se presentan los principales eventos en la historia del movimiento ágil, que fueron descritos en este capítulo.

# HISTORIA

## AGILE

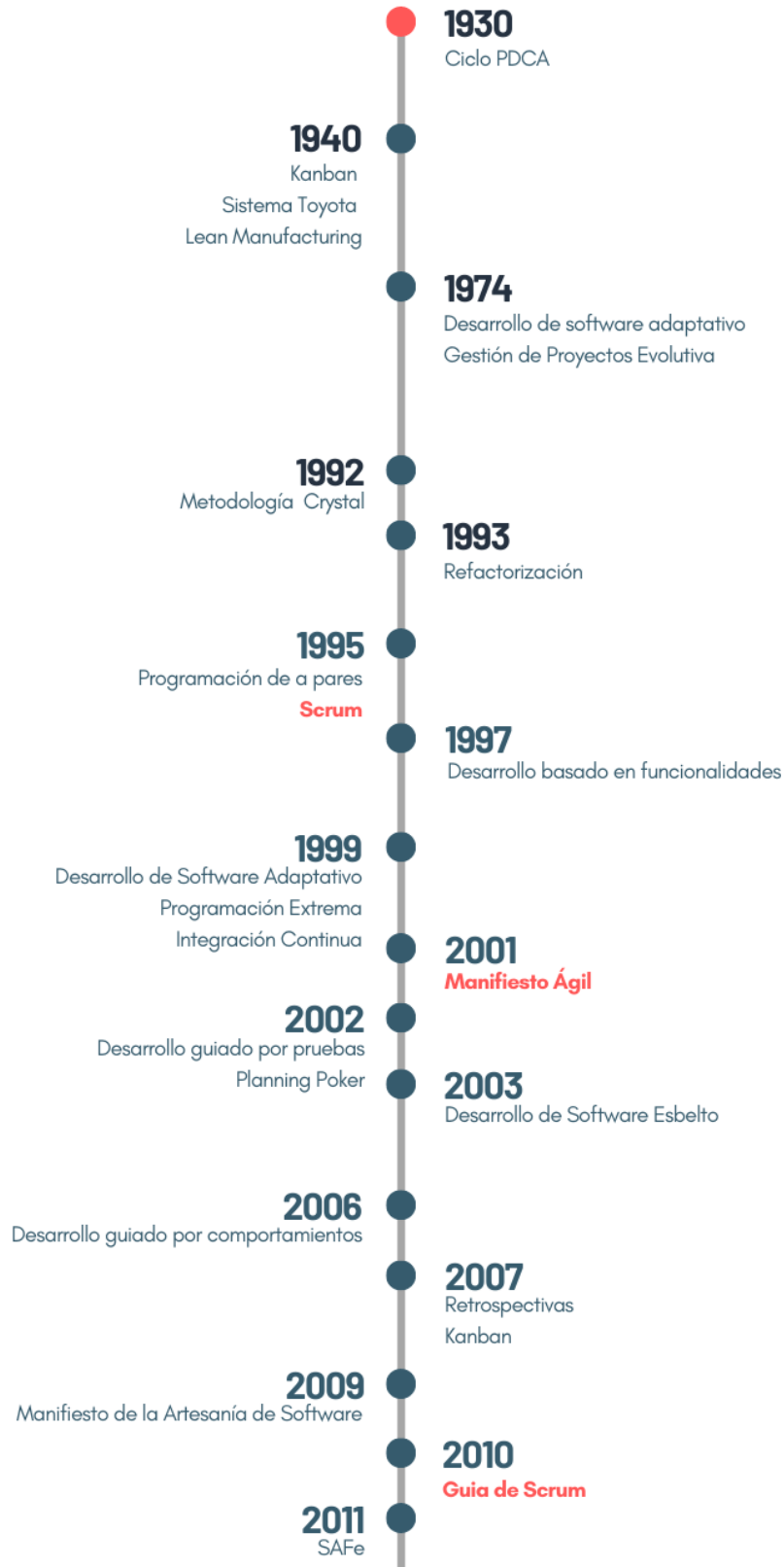


Figura 1 - Cronología de Agile

### 3.3 Historia del Manifiesto ágil

Según describe Jim Highsmith, uno de los firmantes del Manifiesto ágil [6]:

*“Del 11 al 13 de febrero de 2001, en la estación de esquí The Lodge at Snowbird en las montañas Wasatch de Utah, diecisiete personas se reunieron para hablar, esquiar, relajarse y tratar de encontrar puntos en común y, por supuesto, para comer”*

De esta reunión en las montañas de Utah surgió lo que hoy se conoce como el Manifiesto ágil [6], estas diecisiete personas a las cuales hace referencia Highsmith eran especialistas de diversas ramas del desarrollo de software que buscaban una alternativa a los procesos de desarrollo de software de gran peso impulsados por la documentación.

Este grupo de pensadores independientes sobre el desarrollo de software y; a veces, competidores entre sí, tomó el nombre de “La Alianza Ágil” y estuvo de acuerdo con el “Manifiesto para el desarrollo de software ágil”.

Según comenta el relator de la historia; en esa reunión de tres días, todos se sintieron privilegiados de trabajar con un grupo de personas que tenían un conjunto de valores compatibles, centrados en la confianza, el respeto mutuo y la promoción de modelos organizativos basados en personas, colaboración y construcción de los tipos de comunidades organizativas en las que les gustaría trabajar.

Esta reunión en Snowbird a la cual se hace referencia, se pensó un año antes en otro encuentro organizado por Kent Beck en Rogue River Lodge, Oregón. Allí según cuentan, los asistentes expresaron su apoyo a los procesos “ligeros”; dado que, como se pudo ver en la historia de las metodologías ágiles, en años anteriores se publicaron diversos

artículos en donde se hacía referencia a las metodologías “ligeras”, programación extrema, desarrollo de software adaptativo y Scrum.

Desde las primeras charlas, siempre fue un punto de debate el término “ligero”, uno de los firmantes, Alistair Cockburn expresó lo siguiente:

*"No me importa que se llame a la metodología ligera, pero no estoy seguro de querer que me llamen un liviano asistiendo a una reunión de metodólogos livianos. De alguna manera suena como un grupo de personas livianas delgadas, débiles mentales que intentan recordar qué día es" [6].*

Se terminó optando por nombrar a la metodología como “Agile” en base a un libro que Alistair Cockburn había leído: Agile Enterprise. Así mismo, otras versiones indican que originalmente el nombre pensado para la metodología era Adaptive, pero desistieron de usarlo porque uno de los futuros firmantes del manifiesto tenía registrado dicho término a su nombre.

Finalmente, luego de pasar tres días en los cuales surgieron muchas ideas y propuestas, pensaron en plasmarlas en lo que hoy conocemos como el “Manifiesto para el desarrollo de software ágil”.



### 3.4 Valores del Manifiesto para el desarrollo de software ágil

Según expresan los autores del manifiesto para el desarrollo de software ágil:

*“Estamos descubriendo mejores formas de desarrollar software haciéndolo y ayudando a otros a hacerlo. A través de este trabajo hemos llegado a valorar:”*

- Individuos e interacciones sobre procesos y herramientas.
- Software funcionando sobre documentación extensiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

En este punto se hace una aclaración importante, en base a como leer e interpretar estos valores: *“Si bien hay valor en los elementos de la derecha, valoramos más los elementos de la izquierda”*.

El segundo y cuarto valor fueron introducidos por la gente de XP (Programación Extrema); que estaba más relacionada al software, y el primero y el tercer valor fueron acordados por el resto.

A continuación, se analiza cada uno en detalle:

#### 3.4.1 Individuos e interacciones sobre procesos y herramientas

Este valor hace referencia a que se valora mucho más la interacción entre los individuos de un equipo ágil, que los procesos o herramientas que se utilicen. La idea principal es fomentar las conversaciones cara a cara, por sobre el correo electrónico, herramientas de comunicación y similares, se busca que los integrantes busquen conectarse entre ellos de una manera más natural.

### **3.4.2 Software funcionando sobre documentación extensiva**

El segundo valor busca diferenciarse de las formas de desarrollo más utilizadas hasta ese entonces, como por ejemplo el modelo en cascada. En este modelo se utiliza mucha documentación, incluso se plantea una etapa de diseño y documentación antes de comenzar a desarrollar el software.

Se considera que es más valioso un prototipo de la aplicación funcionando, que un manual con páginas y páginas de documentación.

De todos modos, esto no quiere decir que la metodología no utilice documentación, solamente se usa en menor medida, lo justo y necesario.

### **3.4.3 Colaboración con el cliente sobre negociación contractual**

El tercer valor establece la importancia de que el cliente esté más involucrado en todo el proceso de desarrollo, y no solamente al inicio.

La idea es que el cliente esté trabajando y disponible para el equipo la mayor parte del tiempo.

### **3.4.4 Respuesta ante el cambio sobre seguir un plan.**

Este valor representa uno de los pilares del agilísimo, que es la posibilidad de responder al cambio, nada está escrito en piedra y se busca que un equipo de desarrollo tenga la posibilidad de adaptarse a los cambios para responder de una manera más eficiente a los requerimientos que vayan surgiendo.

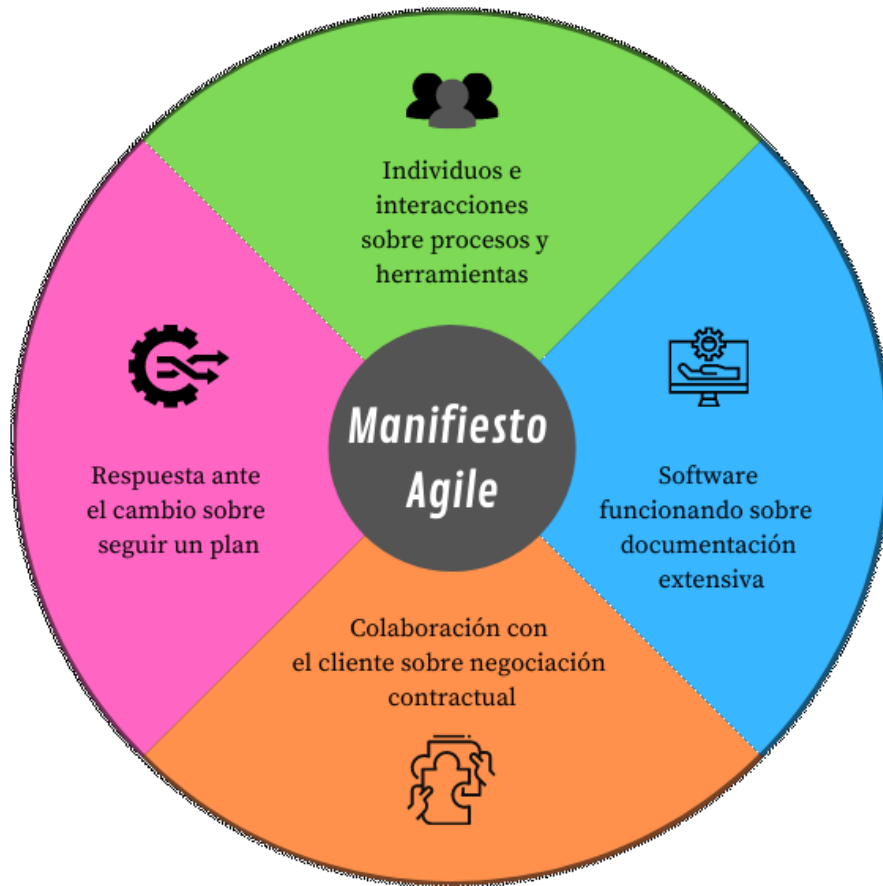


Figura 2 -Valores del manifiesto para el desarrollo de software ágil

### 3.5 Conclusión

En este capítulo se escribió sobre las metodologías ágiles, su origen y evolución, recorriendo su historia y los sucesos fundamentales que la conformaron; así mismo, se escribió sobre la historia del surgimiento del Manifiesto ágil [6] y se describieron y analizaron los valores de éste.

## **CAPÍTULO 4: Scrum**

### **4.1 Introducción**

Uno de los framework más utilizados dentro de las metodologías ágiles es Scrum, cuyo origen se remonta a más de treinta años atrás. En el presente capítulo se detallará su historia, se dará una definición de este, junto a sus valores, roles y los eventos que lo componen.

### **4.2 Historia de Scrum**

Los orígenes de lo que hoy se denomina “Scrum” se remontan al año 1986 en Japón, ese año Hirotaka Takeuchi e Ikujiro Nonaka presentaron un artículo llamado “The New New Product Development Game” [7]. En ese artículo los autores describen cómo empresas como Honda, Canon y Fuji-Xerox producían nuevos productos utilizando un enfoque escalable y basado en equipos integrales. Además, los autores hacen énfasis en la importancia de los equipos auto-organizados haciendo referencia al rugby, en donde la estrategia es definida fuera del equipo pero éste define la táctica. Aquí surge el término Scrum dado a que es una de las formaciones que ocurren durante un partido, en donde todos los miembros se juntan y empujan hacia el mismo lado.

Unos años después de la publicación del artículo, aparece en escena Ken Schwaber, quien empieza a trabajar en las primeras ideas de este método de desarrollo avanzado. Al mismo tiempo, Jeff Sutherland, estaba utilizando un enfoque similar en la compañía Easel Corporation junto a John Scumniotales y Jeff Mckenna.

En el año 1995, Ken Schwaber y Jeff Sutherland se unen y brindan un taller en la

conferencia de programación orientada a objetos, lenguajes y aplicaciones (OOPSLA '95, Object Oriented Programming, Systems, Language and Applications), basándose en “The New New Product Development Game” [7]; publicación antes mencionada, y haciendo público el término “Scrum”.

Más adelante en el tiempo y ya comenzado el nuevo milenio; específicamente en el año 2001, Mike Beedle y Ken Schwaber escriben conjuntamente el libro llamado “Desarrollo ágil de software con Scrum” (Agile Software development with Scrum).

Ese mismo año tanto Ken Schwaber como Jeff Sutherland formaron parte de los 17 autores que firmaron el “Manifiesto para el desarrollo de software ágil”.

Un año después en el 2002 Ken Schwaber junto a Esther Derby y Mike Cohn, crearon Scrum Alliance, la primera organización en ofrecer certificaciones de Scrum.

Luego de que una serie de desacuerdos en base al enfoque excesivamente comercial que se le estaba dando a la Scrum Alliance, Ken Schwaber decide separarse en 2009 y fundar junto con Alex Armstrong la organización Scrum.org con la idea de ofrecer recursos gratuitos y cursos certificados como el de Professional Scrum Master.

Ya en 2010 Ken Schwaber y Jeff Sutherland deciden escribir la guía oficial de Scrum para unificar criterios, dado que cada profesional veía Scrum de una manera distinta y; lo que en su momento eran una serie de reglas, se convirtió en algo que quedaba a la libre interpretación del profesional que lo implementara.

A lo largo de los años, esta guía fue incorporando diferentes actualizaciones y hoy en día se la puede encontrar en varios idiomas.

Entre los cambios que se fueron sucediendo se puede destacar que se eliminaron algunas cosas, como los diagramas de burndown, se reforzó el énfasis en el Daily Scrum como elemento de planificación y también se cambió el nombre de la reunión de Grooming a Refinement.

En 2016 se añadieron los valores de Scrum y en el año 2020 se realizaron varios cambios, como por ejemplo el formato de la reunión diaria de Scrum en donde ya no se realizan las tres clásicas preguntas.

En el siguiente capítulo se describirá más en detalle el contenido de la guía de Scrum y los cambios introducidos los últimos años.

A continuación, se presenta una infografía (Figura 3) de los sucesos previamente descritos:

# HISTORIA SCRUM

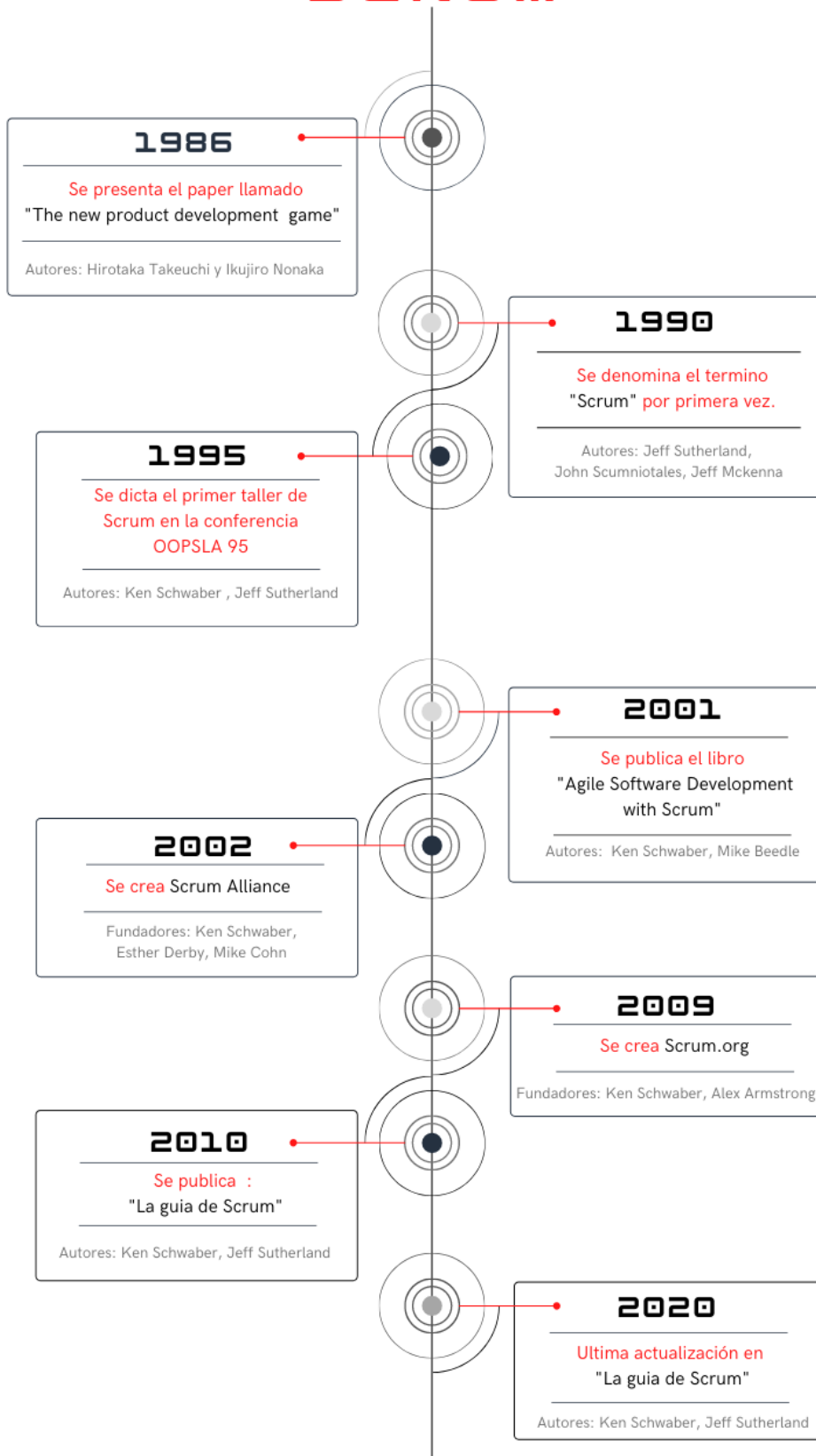


Figura 3 - Cronología de Scrum

### 4.3 Definición de Scrum

Según La guía de Scrum [8].

*“Scrum es un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos. En pocas palabras, Scrum requiere un Scrum Master para fomentar un entorno donde:*

- 1. Un Product Owner ordena el trabajo de un problema complejo en un Product Backlog.*
- 2. El Scrum Team convierte una selección del trabajo en un Increment de valor durante un Sprint.*
- 3. El Scrum Team y sus interesados inspeccionan los resultados y se adaptan para el próximo Sprint.*
- 4. Repita”*

### 4.4 Valores de Scrum

Otro punto importante al cual se hace referencia en “La guía de Scrum” [8] y que es importante mencionar, está relacionado con los valores que debe tener un Scrum para ser exitoso.

La guía establece lo siguiente:

*“El uso exitoso de Scrum depende de que las personas se vuelvan más competentes en vivir cinco valores: Compromiso, Foco, Franqueza, Respeto y Coraje.*

*El Scrum Team se compromete a lograr sus objetivos y a apoyarse mutuamente. Su foco*



*principal está en el trabajo del Sprint para lograr el mejor progreso posible hacia estos objetivos. El Scrum Team y sus interesados son francos sobre el trabajo y los desafíos. Los miembros del Scrum Team se respetan entre sí para ser personas capaces e independientes, y son respetados como tales por las personas con las que trabajan. Los miembros del Scrum Team tienen el coraje de hacer lo correcto, para trabajar en problemas difíciles.”*

#### **4.5 Scrum Team (El equipo Scrum)**

Un Scrum Team está compuesto por tres roles:

- Scrum Master
- Product Owner (Propietario de producto)
- Developers (Desarrolladores)

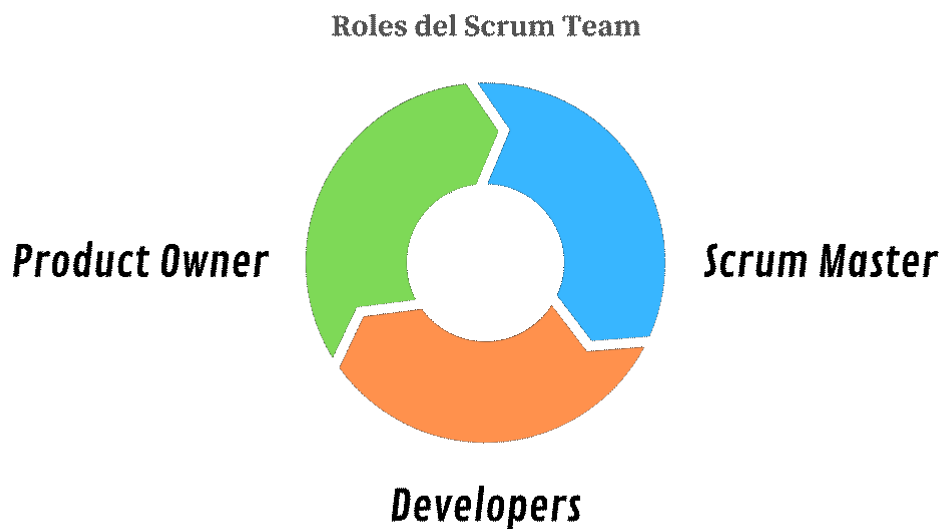


Figura 4 - Roles del Scrum Team

El Scrum Team tiene una cantidad de miembros reducida la cual; por lo general, es de diez o menos personas, buscando además que estos miembros sean auto-organizados; es decir, que entre ellos se coordinen para decidir quién hace qué, cuándo y cómo lo hace.

Un punto importante que menciona “La guía de Scrum” [8], es que dentro de esta distribución no tienen que existir sub-equipos o jerarquías:

*“Es una unidad cohesionada de profesionales enfocada en un objetivo a la vez, el Objetivo del Producto”.*

Una de las características de los Scrum Teams es que sus miembros son multifuncionales, esto quiere decir que todos los miembros tienen todas las habilidades necesarias para lograr el objetivo del Sprint.

#### 4.5.1 Product Owner (Propietario del producto)

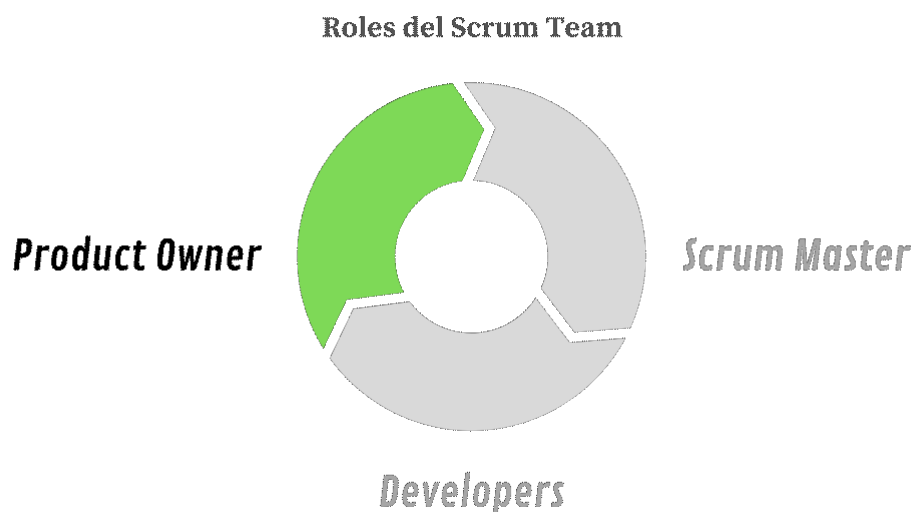


Figura 5 - Roles del Scrum Team - Product Owner

El Product Owner es el encargado de maximizar el trabajo realizado por el equipo, guiándolo en la elaboración de una solución que cumpla con la necesidad del cliente.

Es la persona que se encarga de administrar uno de los artefactos más importantes del Scrum, la pila del producto, más comúnmente conocida por su término en inglés Product Backlog.

Según “La guía de Scrum” [8] las tareas que un Product Owner tiene que desempeñar para una administración eficaz del Product Backlog son:

- *“Desarrollar y comunicar explícitamente el Objetivo del Producto;*
- *Creación y comunicación clara de elementos de trabajo pendiente del producto;*
- *Pedido de artículos de trabajo pendiente del producto;*
- *Asegurarse de que el trabajo pendiente del producto sea transparente, visible y comprendido.”*

Una característica importante de este rol es que el mismo no puede ser desempeñado por un comité, tiene que ser una sola persona.

Dependiendo de la organización este rol suele ser ocupado por el cliente, quien es el interesado en el producto o por un representante del mismo. Cabe aclarar que; al ser el representante del cliente, esta persona tiene que tener todo el conocimiento necesario del negocio.

Por esta razón, normalmente no es un perfil técnico y está centrado en el negocio, dado

que se centra en lo que el cliente necesita, más que en los detalles técnicos de cómo se implementa la solución.

#### 4.5.2 Scrum Master

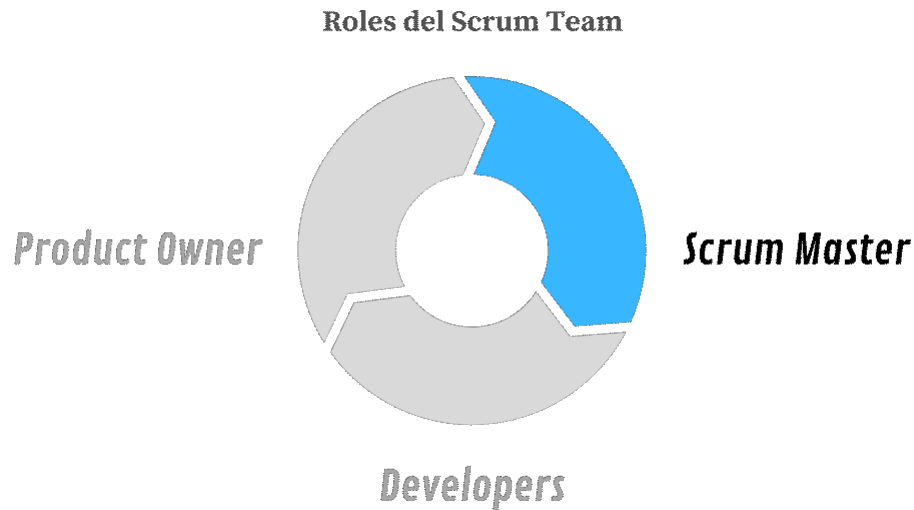


Figura 6 - Roles del Scrum Team - Scrum Master

El Scrum Master es el responsable de velar porque se cumplan los principios establecidos en “La guía de Scrum” [8]

Es la persona encargada de transmitir la teoría y la prácticas de Scrum, tanto al equipo como a otras partes de la organización.

Dentro de las responsabilidades que tiene el Scrum Master para con el Scrum Team, se pueden destacar las siguientes:

- Asesorar y capacitar a los miembros del equipo para trabajar de forma auto-organizada y con responsabilidad.
- Eliminar los impedimentos, para favorecer el progreso del equipo.
- Asegurarse que todos los eventos de Scrum se lleven a cabo, que estos sean

positivos y productivos, y que se respete el tiempo establecido (time-box) para cada uno de ellos.

Dentro de las responsabilidades que tiene el Scrum Master para con la organización la “La guía de Scrum” [8] define lo siguiente:

- *“Liderar, capacitar y mentorizar a la organización en su adopción de Scrum;*
- *Planificar y asesorar sobre la implementación de Scrum dentro de la organización;*
- *Ayudar a las personas y a las partes interesadas a comprender y promulgar un enfoque empírico para el trabajo complejo;*
- *Eliminar las barreras entre las partes interesadas y los equipos de Scrum.”*

#### 4.5.3 Developers (Desarrolladores)

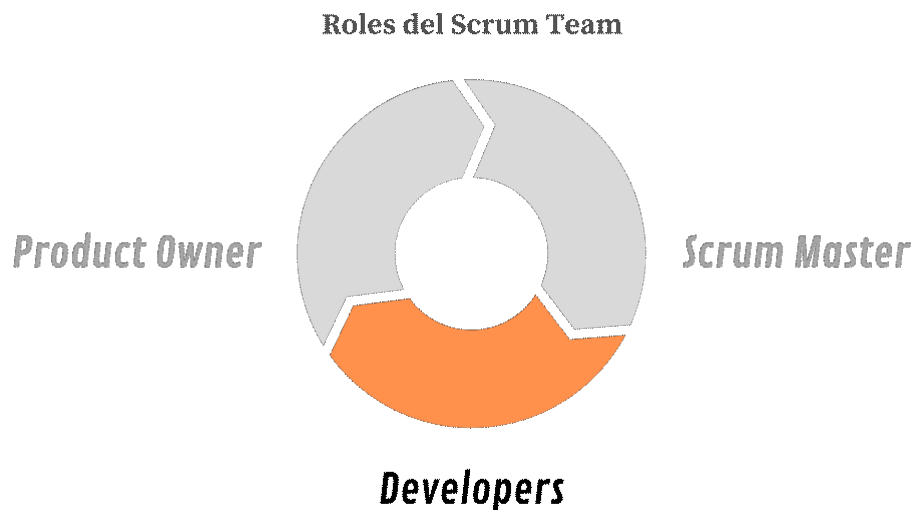


Figura 7 - Roles del Scrum Team - Developers

Los Developers son las personas del Scrum Team que se comprometen a crear un

incremento útil (funcional) en cada Sprint.

Por lo general es un equipo “Cross-functional”, esto quiere decir que tiene que estar compuesto por programadores, testers, diseñadores, etc., los cuales; según “La guía de Scrum” [8], son responsables de las siguientes tareas:

- *“Crear un plan para el Sprint, el Sprint Backlog;*
- *Inculcar calidad al adherirse a una Definición de Terminado;*
- *Adaptar su plan cada día hacia el Objetivo del Sprint; y,*
- *Responsabilizarse mutuamente como profesionales.”*

#### **4.6 Rol por fuera del Scrum: Stakeholder (No definido en “La guía de Scrum”)**

Básicamente son todas las demás personas que están involucradas en el proyecto pero que no forman parte del equipo Scrum. Sin embargo, siguen desempeñando uno de los papeles más importantes: son la fuente de información para el equipo Scrum.

Ejemplos de partes interesadas son los usuarios finales, gerentes, patrocinadores, equipos de IT del cliente y cualquier otra persona que interactúe con el proyecto.

#### **4.7 Eventos de Scrum**

Según se establece en “La guía de Scrum” [8]:

*“el Sprint es un contenedor para todos los eventos de Scrum”.*

Esto quiere decir que todos los eventos de Scrum que se van a mencionar y analizar a continuación se suceden y realizan dentro de un Sprint.

Cada uno de estos eventos se toma como una oportunidad para que el equipo pueda inspeccionar y adaptar los artefactos del Scrum. Buscando además garantizar y promover la transparencia.

La guía define estos eventos para evitar la necesidad de reuniones no definidas por Scrum aconsejando; además, que los mismos sean regulares y se lleven a cabo siempre en los mismos días y horarios, logrando así reducir la complejidad.

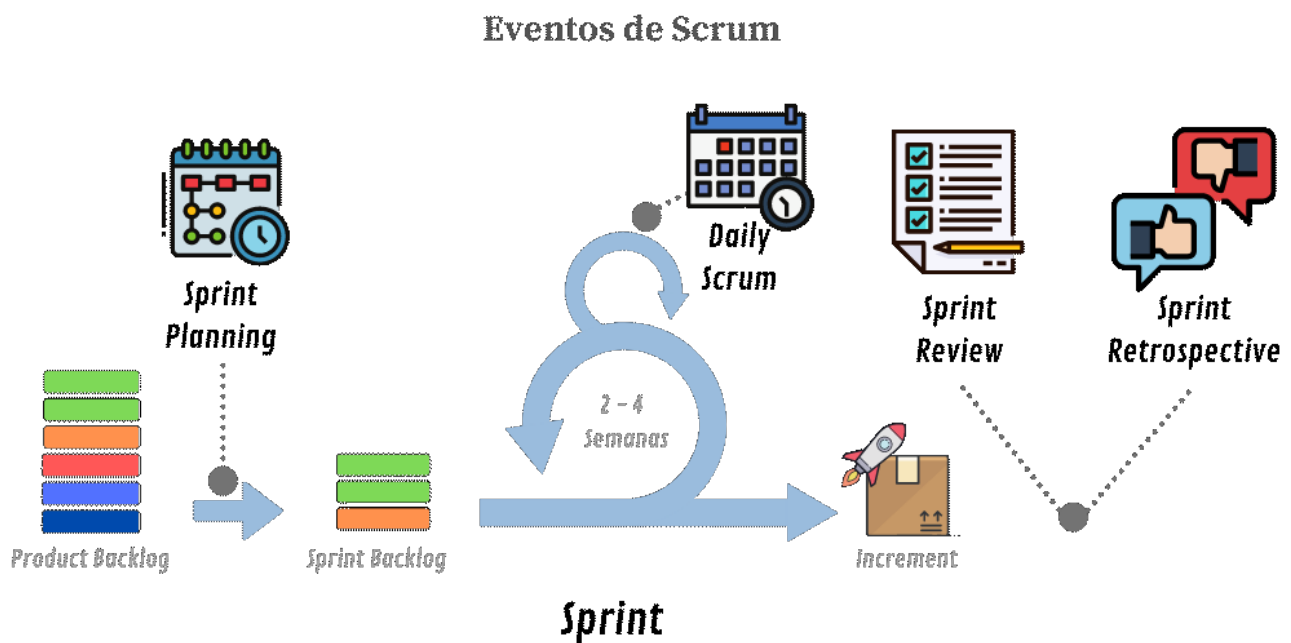


Figura 8 - Eventos de Scrum

#### 4.7.1 El Sprint

El Sprint es un evento que tiene una longitud fija, la cual puede ser de una a cuatro semanas como máximo.

No está establecido formalmente en cual día de la semana debe comenzar y terminar un Sprint, existen equipos que lo implementan de lunes a viernes y otros

que lo hacen de miércoles a jueves. Lo que sí está establecido por “La guía de Scrum” [8] es que; al terminar un Sprint, inmediatamente comienza uno nuevo.

Durante el Sprint, se realizan todas las tareas que nos permiten lograr cumplir con el Objetivo del Producto, incluido los eventos tales como Sprint Planning, Daily Scrum, Sprint Review y Sprint Retrospective.

Otro punto importante a considerar y que también es mencionado en “La guía de Scrum” [8] son las tareas que no se realizan dentro del Sprint, la guía las enumera de la siguiente manera:

*“Durante el Sprint:*

- *No se hacen cambios que pongan en peligro el Objetivo Sprint;*
- *La calidad no disminuye;*
- *El trabajo pendiente del producto se refina según sea necesario;*
- *El alcance se puede clarificar y negociar con el Product Owner a medida que se aprende más. “*

El Product Owner es el único que cuenta con la facultad de cancelar un Sprint, esto puede suceder por diferentes razones, como por ejemplo que el objetivo del Sprint ya no sea válido.

#### **4.7.2 Sprint Planning (Planificación de Sprint)**

Este evento es el primero que se realiza dentro del Sprint y da lugar al comienzo del mismo.



Como resultado, se obtiene un plan de trabajo de cara al Sprint que comienza, esto se logra gracias al trabajo colaborativo de todo el equipo de Scrum.

### 4.7.3 Daily Scrum

La Daily Scrum es un evento cuya duración está estipulada en un máximo de quince minutos y la misma está pensada para los Developers del equipo Scrum.

Con el objetivo de reducir la complejidad, se aconseja realizar este evento siempre en el mismo horario y lugar, todos los días laborables del Sprint.

Según se establece en la versión 2020 de “La guía de Scrum” [8]:

*“El propósito de la Daily Scrum es inspeccionar el progreso hacia el Objetivo Sprint y adaptar el Sprint Backlog según sea necesario, ajustando el próximo trabajo planeado.”*

Los Developers tienen la facultad de elegir qué estructura utilizar para el evento, buscando siempre el progreso hacia el objetivo del Sprint y un plan para el día de trabajo. Se busca además, que el equipo se enfoque y autogestione.

Una de las estructuras de coordinación más utilizadas en la Daily Scrum, consiste en que cada miembro del equipo responda por ejemplo las siguientes tres preguntas:

- ¿Qué hice ayer?
- ¿Qué voy a hacer hoy?
- ¿Veó algún impedimento que haga peligrar el objetivo del sprint ?

Esta técnica es mencionada en versiones anteriores de “La guía de Scrum” [8], pero en la edición del 2020, se establece que el equipo elija como coordinarse de cara al objetivo del Sprint. Como bien se menciona en “La guía de Scrum”[8], este evento ayuda a que el equipo pueda mejorar la comunicación, se identifiquen impedimentos de una manera más eficiente y promueve una rápida toma de decisiones, eliminando además la necesidad de otras reuniones.

#### **4.7.4 Sprint Review (Revision del Sprint)**

En cada fin de Sprint se lleva a cabo este evento con el objetivo de inspeccionar el resultado del trabajo y verificar posibles adaptaciones. Para ello el Scrum Team muestra el trabajo realizado a las partes interesadas y juntos revisan los logros y los cambios; además, se mantiene una discusión acerca del progreso relacionado con el objetivo de producto y los asistentes colaboran para acordar los pasos a seguir. Se intenta que la Sprint Review no sea una simple presentación, este evento es una sesión de trabajo y tiene una duración máxima de cuatro horas para un Sprint de un mes, pero suele ser más corta para Sprints más cortos.

#### **4.7.5 Sprint Retrospective (Retrospectiva del Sprint)**

Este evento tiene como finalidad poder mejorar la calidad de trabajo del equipo en base a la inspección del último Sprint, con respecto a individuos, interacciones, procesos , herramientas y la “Definition of done” utilizados por el equipo.

El equipo utiliza este espacio para analizar qué se hizo bien, que no se hizo tan bien, con qué problemas se encontró y como fueron o no resueltos. Es decir, se realiza lo que se conoce como inspeccionar y adaptar.

Una vez identificado esto, el equipo busca ver cuáles son las mejoras más importantes por realizar. Estas mejoras se pueden especificar en el Sprint Backlog para tenerlas presentes en el próximo Sprint.

Este evento es; además, el último a realizar durante el Sprint y se establece en “La guía de Scrum” [8] que el mismo debe tener una duración máxima de tres horas para Sprints de cuatro semanas, para Sprints más cortos este tiempo se reduce.

#### **4.7.6 Refinement**

El Refinement (anteriormente conocido como Grooming) es un evento que no está establecido en “La guía de Scrum”, pero en los últimos años se incluye como una de las prácticas esenciales a realizar dentro del Sprint.

Este evento tiene el objetivo de ayudar al Product Owner a refinar o mejorar el Product Backlog de cara a los próximos Sprints, permitiendo identificar impedimentos o bloqueos con anticipación.

Al igual que para los demás eventos, se busca que se realice en base a una agenda fija, en el mismo horario y día de la semana. No está estrictamente definida la cantidad de repeticiones durante el Sprint, puede variar dependiendo de las necesidades del equipo.

#### **4.8 Artefactos de Scrum**

Los artefactos de Scrum son elementos utilizados por el equipo de Scrum, los mismos representan el trabajo o valor del Sprint.

Estos garantizan una total transparencia del estado del Sprint, permitiendo que todos los miembros puedan inspeccionar y adaptar en base a la misma información.

Los artefactos mencionados en “La guía de Scrum” [8] son:

- Product Backlog.
- Sprint Backlog.
- Increment.

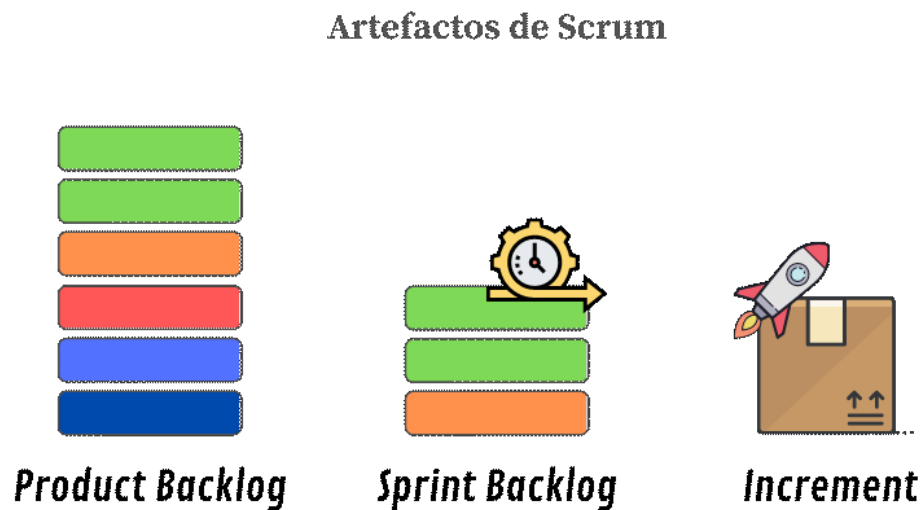


Figura 9 - Artefactos de Scrum

#### 4.8.1 Product Backlog

Se denomina como Product Backlog a la lista ordenada y priorizada de lo que se necesita para cumplir con el Objetivo del Producto, al cual se define como el objetivo a largo plazo del Scrum Team.

Un evento que se realiza para organizar, mejorar y dimensionar los elementos que componen el Product Backlog es el ya mencionado Refinement, en el cual se dividen y definen aún más los elementos para que éstos sean más pequeños y precisos.

En este proceso de refinamiento se evalúan además cuáles son los elementos que se

consideran listos para ser seleccionados en el evento de Sprint Planning.

#### **4.8.2 Sprint Backlog**

El Sprint Backlog está compuesto por el conjunto de elementos del Product Backlog que fueron seleccionados para el Sprint, junto con el plan de acción para la entrega de estos.

Es un plan de trabajo realizado por y para los Developers, el mismo permite ver fácilmente lo que planean realizar los Developers para lograr el objetivo del Sprint (Sprint Goal).

Este artefacto se va actualizando a medida que transcurre el Sprint y se busca además que tenga un buen detalle de los elementos que lo componen, para que el progreso pueda ser inspeccionado en el evento Daily Scrum.

#### **4.8.3 Increment (Incremento)**

Un Increment es un paso concreto hacia el Objetivo del Producto, cada uno de estos se van sumando a los Increments anteriores, no sin antes ser verificados o testeados con el objetivo de garantizar que todos los Increments juntos funcionen correctamente.

### **4.9 Conclusión**

Se describió la historia de Scrum y cuáles fueron sus orígenes, también se aportó la definición de Scrum y sus valores según “La guía de Scrum” [8], se dio una descripción de cada uno de los roles que conforman un Scrum Team y se detallaron los eventos y artefactos que conforman Scrum.

## CAPÍTULO 5: Scrum Distribuido

### 5.1 Introducción

Desde la aparición de las metodologías ágiles, el desarrollo de software ha ido cambiado radicalmente, en gran parte gracias al avance de la tecnología y el uso masivo de internet. Hoy en día, las empresas dedicadas al desarrollo de software suelen ampliarse a otros países y tener equipos distribuidos a lo largo de todo el mundo y esto se da por diferentes razones como, por ejemplo:

- **Búsqueda de talento en otras locaciones:** El ampliarse al mundo les permite a las empresas contratar personas con diferentes talentos, que quizás ya no estén encontrando en su país o ciudad de origen.
- **Costos:** Como ha pasado históricamente, las empresas buscan abaratar costos contratando mano de obra más barata. En el desarrollo de software esto se puede encontrar en locaciones como India, Filipinas, China y Latinoamérica. Lo que hace que empresas de; por ejemplo, Estados Unidos, tengan la mayoría de sus equipos (generalmente desarrolladores) en estos países.
- **Tiempo:** Algo que también siempre se buscó es aprovechar la mayor cantidad posible de horas productivas, por lo que se pretende tener equipos con diferentes husos horarios de manera de estar virtualmente trabajando en el producto las veinticuatro horas. Por ejemplo, un equipo de Argentina comienza a trabajar de 8 a.m. hasta las 6 p.m., al terminar su día laboral, un equipo situado en la Filipinas inicia su jornada de trabajo y continúa con el desarrollo. A esta técnica se la denomina “Follow The Sun” (Seguir al sol), dado que cuando un equipo está

durmiendo, el otro continúa la tarea que dejó el primero.

Pero un equipo no solamente puede estar distribuido alrededor del mundo, en estos últimos años se ha afianzado notablemente la idea de que cada persona pueda trabajar desde su casa, siendo un beneficio tanto para la empresa como para el empleado.

Esto genera que el equipo de trabajo esté distribuido incluso si todos viven en la misma ciudad.

Scrum ofrece un marco que actualmente puede resultar útil para trabajar con equipos distribuidos, pero sin embargo es importante ajustar algunas prácticas y utilizar las herramientas que faciliten la interacción.

A lo largo de este capítulo, se van a detallar las consideraciones a tener en cuenta para implementar Scrum de manera distribuida.

## **5.2 Niveles de equipos distribuidos**

Los equipos de desarrollo de software se han ido distribuyendo, y adoptado diferentes niveles los cuales se pueden diferenciar de la siguiente manera:

- Co-locado (Ubicados en el mismo lugar físico)
- Co-locado de manera parcial (Ubicados parcialmente en el mismo lugar)
- Distribuido con horas superpuestas
- Distribuido sin horas superpuestas

En la Figura 10, se puede ver una representación gráfica de estos niveles, la misma está basada en la que se encuentra en el libro “A Practical Guide to Distributed Scrum” [3]

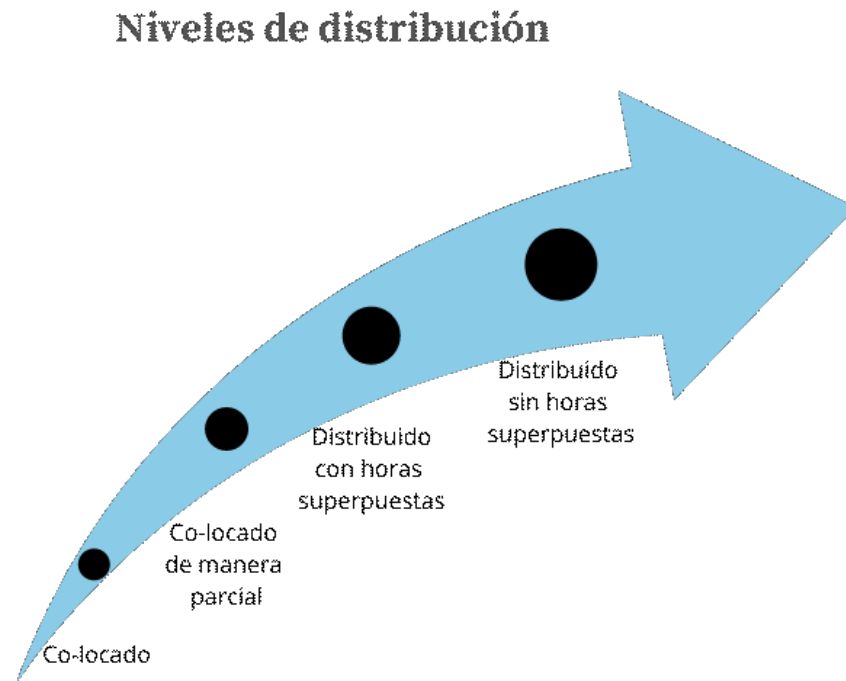


Figura 10 - Niveles distribución de equipos [3]

#### 5.2.1 Equipo co-locado

En este tipo de distribución, todos los miembros del equipo se encuentran en el mismo lugar físico trabajando en una misma oficina, ya sea en una sala de reuniones o cada uno en su escritorio. La principal característica de estos equipos es que tienen la posibilidad de reunirse cara a cara diariamente.



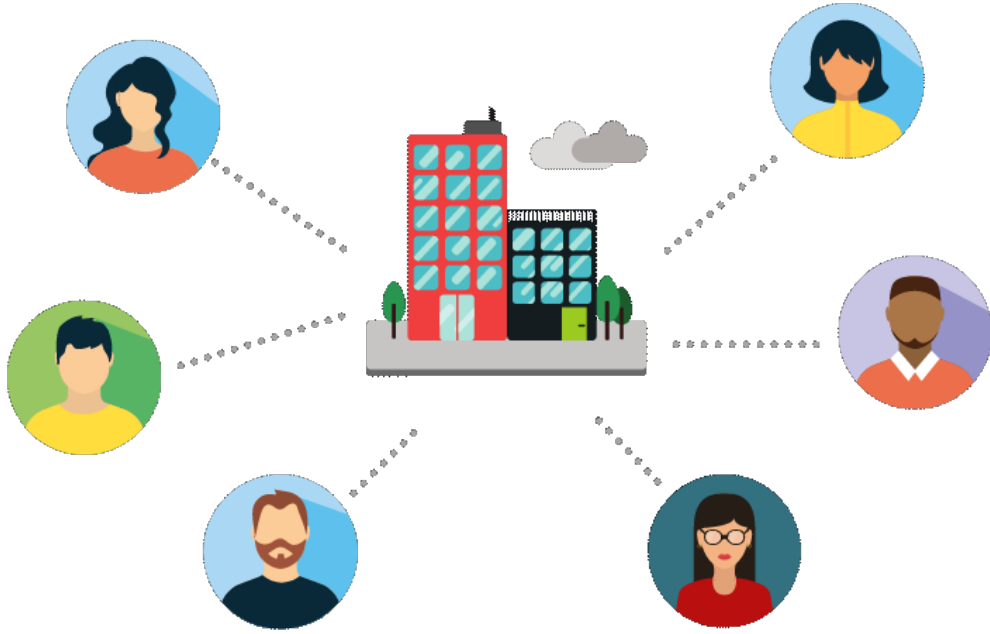


Figura 11 - Equipo co-locado

### 5.2.2 Equipo parcialmente co-locado

En esta distribución, algunos miembros se encuentran en el mismo lugar físico, mientras que otros miembros se encuentran en un lugar diferente.

Es la distribución típica que se genera cuando algunas personas del equipo están en la oficina y otros trabajan desde su casa o un espacio de coworking.



Figura 12 - Equipo parcialmente co-locado

### 5.2.3 Distribuido con horas superpuestas

Distribución en la cual los miembros del equipo interactúan como mínimo una hora durante la jornada laboral. Un ejemplo de este modelo se da cuando el equipo se encuentra distribuido en diferentes países, por ejemplo algunos miembros están en Buenos Aires, Argentina y otros en Chicago, Estados Unidos, teniendo de esta manera siete horas superpuestas en donde el equipo puede interactuar.

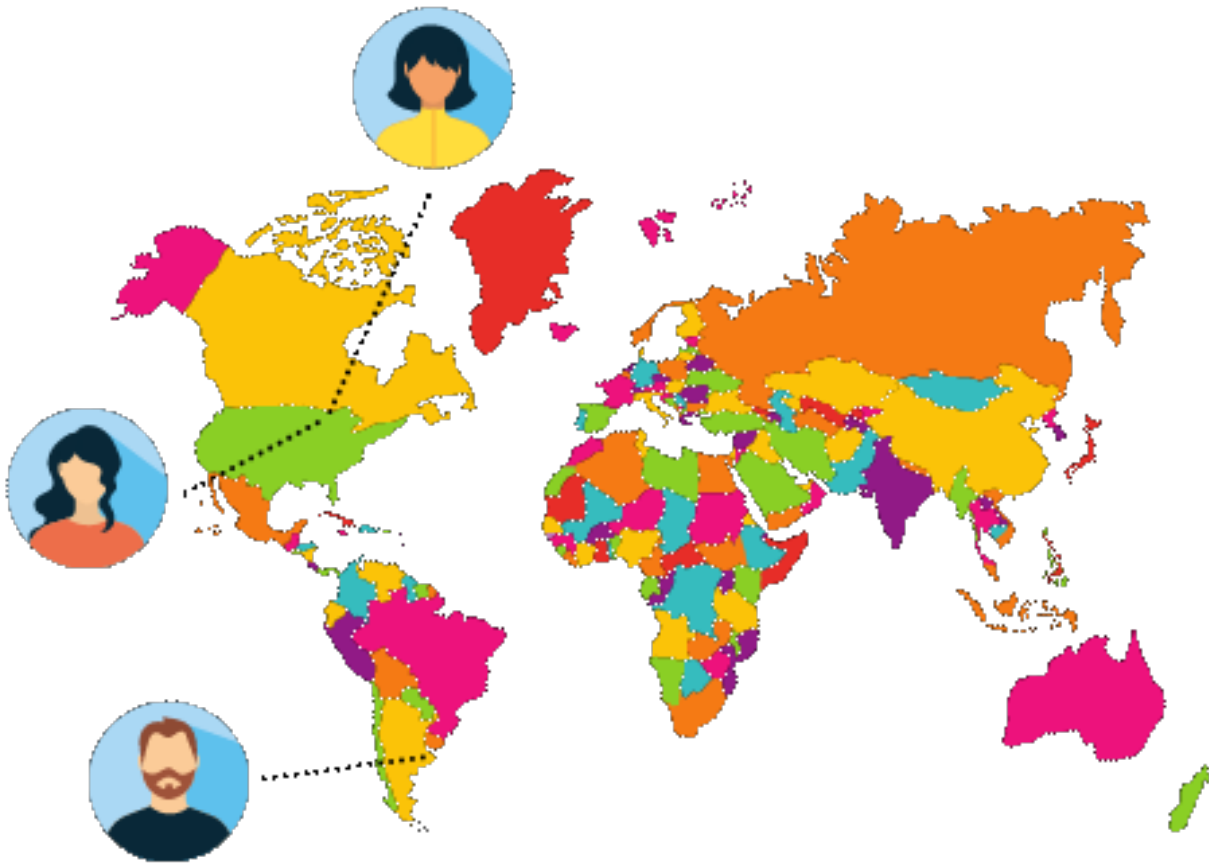


Figura 13 - Distribuidos con horarios de trabajo superpuestos

En la Figura 14, se puede ver representado gráficamente como existen horas superpuestas (en verde) a partir que el equipo de Argentina comienza su día laboral, hasta las 16:00 que es cuando el equipo de Chicago culmina su día.

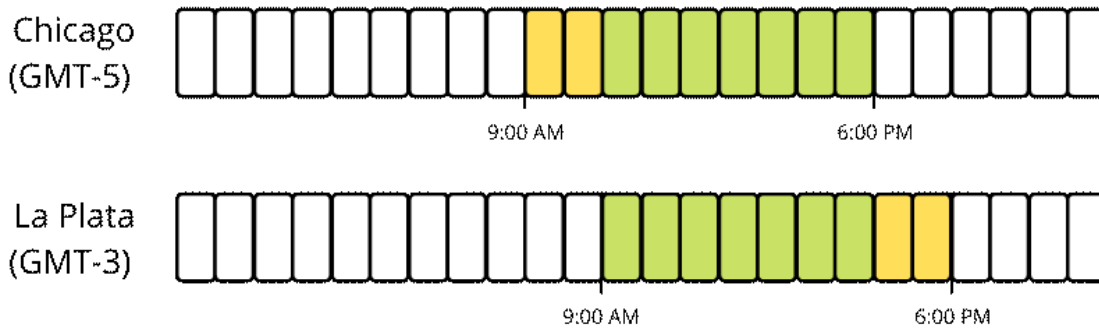


Figura 14 -Horas superpuestas entre Chicago y La Plata

#### 5.2.4 Distribuido sin horas superpuestas

Este nivel de distribución es en el cual los miembros del equipo no tienen la posibilidad de interactuar durante las horas de trabajo estándares.

Este caso se puede dar por ejemplo cuando parte del equipo se encuentra en Argentina y la otra parte en Filipinas, teniendo una diferencia horaria de once horas.

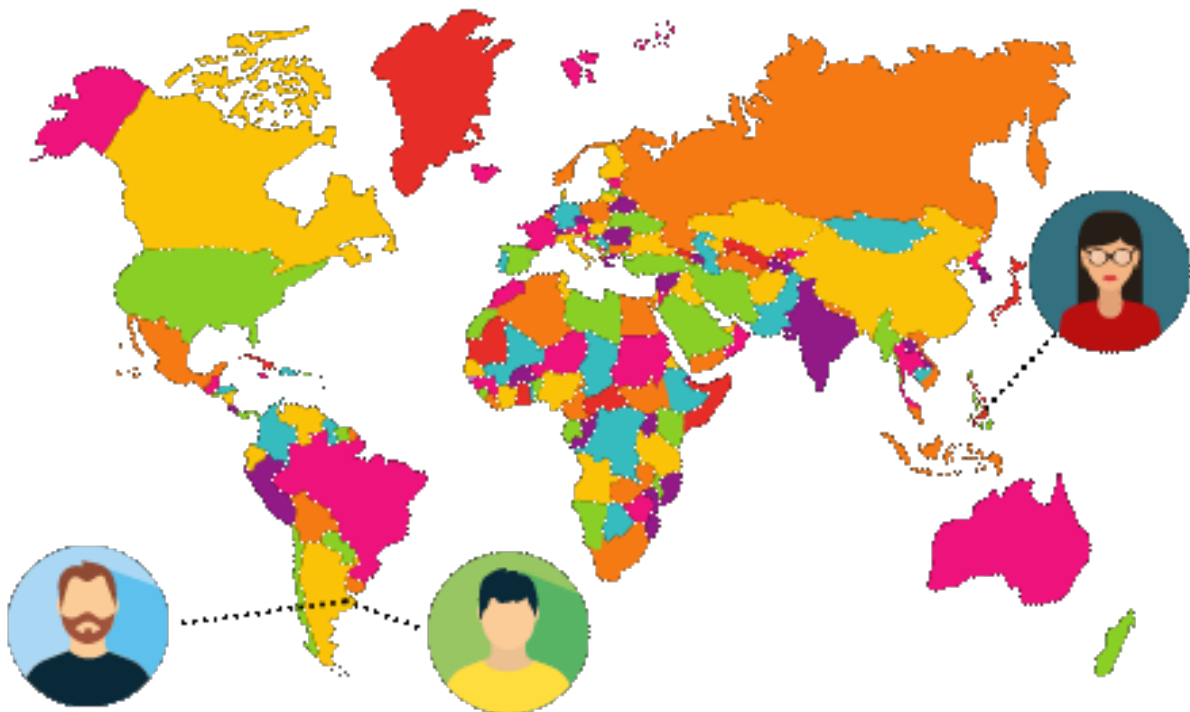


Figura 15 - Distribuido sin horas de trabajo superpuestas

En la Figura 16, se puede observar esto representado gráficamente, y notar como no hay horas en común entre dos ciudades de estos países, por ejemplo, cuando un equipo ubicado en La Plata termina su día, otro ubicado en Manila aún está a tres horas de comenzar.

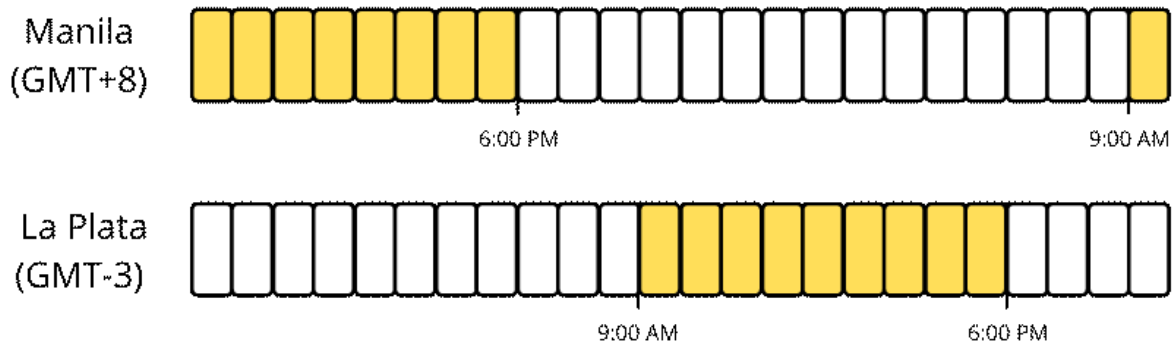


Figura 16 -Horas superpuestas entre Manila y La Plata

### 5.3 Modelos de equipos distribuidos

Además de los diferentes niveles de equipos distribuidos que se han mencionado anteriormente, existen diversas maneras de organizar dichos equipos. Jeff Sutherland describe tres modelos distribuidos en su publicación “Distributed Scrum: Agile Project Management with Outsourced Development Teams”[5], los cuales serán detallados a continuación:

- Scrum asilados
- Scrum de Scrum distribuidos
- Scrum totalmente integrados

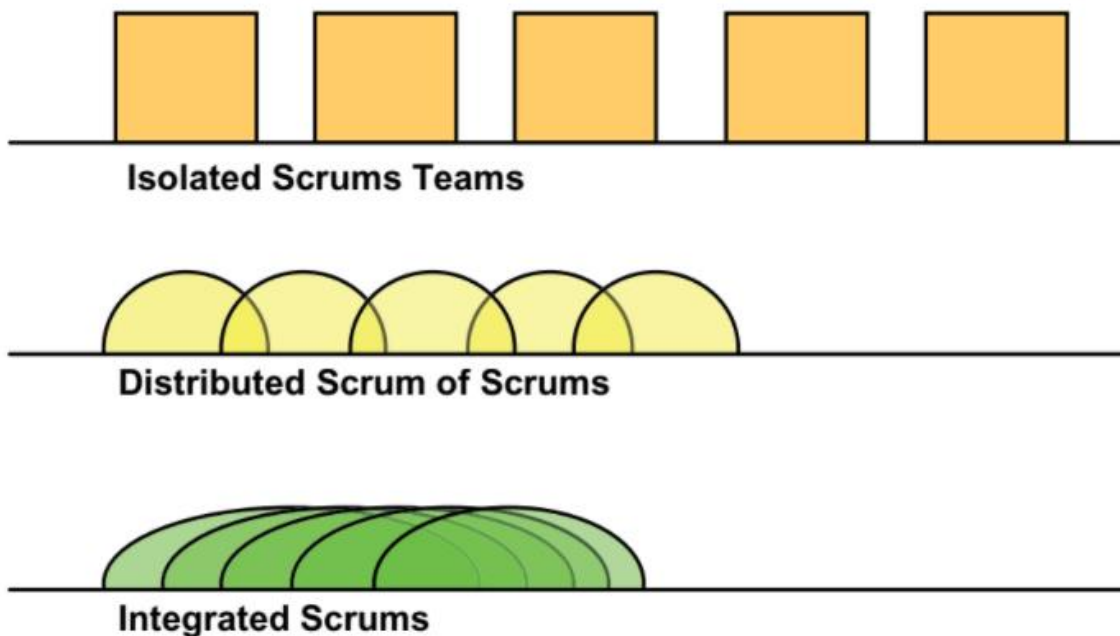


Figura 17 - Modelo de equipos distribuidos, según Jeff Sutherland [5]

### 5.3.1 Scrum aislados

Modelo en donde los equipos de Scrum se encuentran aislados en diferentes zonas geográficas sin tener contacto entre sí. Esto quiere decir que los equipos funcionan de una manera totalmente independiente sin tener contacto entre sí, no es necesario coordinar una colaboración entre los equipos, lo que permite aplicar Scrum de una manera más sencilla.

### 5.3.2 Scrum de Scrum distribuidos

Modelo en el cual los equipos están aislados en zonas geográficas diferentes y se integran a través de reuniones regulares de Scrum of Scrum para coordinar el trabajo entre ellos.

### **5.3.3 Scrum totalmente integrados**

En este modelo, los equipos tienen sus miembros dispersos en diferentes zonas geográficas y usos husos horarios. . Es el modelo recomendado para equipos Scrum experimentados en múltiples ubicaciones. La Daily Scrum y el uso de buenas prácticas de comunicación ayudan a superar desafíos de comunicación y ejecución.

### **5.4 Conclusión**

En este capítulo se describieron las razones por las cuales Scrum puede resultar útil en un marco distribuido. Por otra parte, se detallaron los distintos niveles de distribución y se mencionaron los tres modelos de equipos distribuidos presentados por Jeff Sutherland.

## CAPÍTULO 6: Desafíos de utilizar Scrum Distribuido

### 6.1 Introducción

Es importante reconocer los desafíos que existen al implementar Scrum en equipos distribuidos porque; aunque no se pueden prevenir en su totalidad, se pueden abordar de una manera más efectiva. Usar Scrum en equipos distribuidos plantea una serie de retos a superar, los cuales provienen de la distancia física, las diferencias culturales e idiomáticas, y los horarios particulares de cada equipo y las personas que lo componen.

Los autores consideran que la comunicación es el desafío principal que se destaca sobre los demás, dado que ésta es vital para cualquier proyecto e impacta directamente en su resultado. La forma más eficaz de comunicación se relaciona con las interacciones cara a cara porque les permite a las personas no solo interpretar la comunicación verbal, sino también la comunicación no verbal (el lenguaje corporal, las expresiones faciales, los movimientos de las manos). Pero al tener un mercado en el cual predominan los equipos distribuidos en diferentes locaciones, esta interacción cara a cara se pierde, disminuyendo la capacidad de comunicarse de una manera más natural.

Estos desafíos también son mencionados en el texto “Distributed Scrum: Agile Project Management with Outsourced Development Teams” [5] escrito por Jeff Sutherland, en donde se enumeran cinco categorías en las cuales se pueden englobar los principales problemas o desafíos al utilizar Scrum distribuido.

- **Estratégica:** Dificultad para aprovechar los recursos disponibles, las mejores prácticas a menudo se consideran prohibitivas, consumen mucho tiempo y son difíciles de mantener.



- **Gestión de proyectos y procesos:** Dificultad para sincronizar el trabajo entre los lugares distribuidos.
- **Comunicación:** Falta de mecanismos de comunicación eficaces.
- **Culturales:** Comportamientos, procesos y tecnologías conflictivas.
- **Técnicos:** Formatos de datos, esquemas y estándares incompatibles.
- **Seguridad:** Garantizar la confidencialidad y privacidad de las transmisiones electrónicas.

## 6.2 Desafíos más frecuentes

En la siguiente parte del capítulo, se detallarán los desafíos más comunes a los cuales se enfrenta un equipo distribuido, los cuales están englobados dentro de las cinco categorías antes mencionadas.

### 6.2.1 Diferencias culturales

Las diferencias culturales afectan la comunicación y colaboración dentro de un equipo. Estas diferencias se hacen más evidentes cuando los miembros del equipo Scrum son de países diferentes.

Hay que tener en cuenta y cuidar los comentarios que se realizan, dado que una palabra que es natural en una cultura puede ser ofensiva en otra.

Un ejemplo de diferencia cultural es algo que comentan la mayoría de las personas que han tenido la oportunidad de trabajar con equipos de India. Las personas que provienen de este país suelen tener miedo a decir que no y siempre dicen que si a cualquier pedido por parte del cliente, lo que genera problemas de cara al compromiso del equipo.

### 6.2.2 Diferencias de idioma

Un equipo distribuido en diferentes países puede llegar a enfrentar diferencias de idioma que afectarán el modo de comunicarse. Los desafíos pueden aparecer de varias formas, por ejemplo, cuando alguno o todos los miembros del equipo distribuido no manejan el mismo idioma, o los miembros del equipo pueden no ser capaces de entenderse fácilmente entre sí. Un equipo Scrum en el que sus miembros se comunican continuamente pondrá en evidencia los retos de comunicación.

Un ejemplo de esto puede ser cuando el Product Owner es un cliente que habla en el idioma inglés y el resto del equipo (incluyendo desarrolladores y Scrum Master) hablan en español. Al participar de alguno de los eventos, como por ejemplo la Retrospective, sucede que los miembros que no hablan en inglés o directamente no lo entienden, evitan participar y se quedan callados, sin expresar sus puntos de vista.

### 6.2.3 Diferencia horaria

Cuando un equipo tiene sus miembros distribuidos en diferentes husos horarios, se generan diferentes problemas de coordinación para llevar a cabo los distintos eventos.

Por ejemplo, se dificulta mucho poder pautar la Daily en un horario que satisfaga a todos los miembros del equipo. Si se toma el caso de que algunos miembros se encuentren en Argentina y otros en Estados Unidos, va a suceder que cuando en este último país comiencen a trabajar, los miembros de Argentina ya están cursando casi la mitad de su jornada. Como es sabido, a la Daily deben acudir todos los desarrolladores, entonces se suele setear el horario a primera hora de Estados Unidos, haciendo que sea incómodo

para los de Argentina, teniendo que esperar varias horas para coordinarse con sus compañeros.

#### 6.2.4 Comunicación

A pesar de que todos los miembros del equipo hablen el mismo idioma y no tengan diferencias culturales, también se pueden dar importantes problemas de comunicación. Una gran parte de la comunicación es no verbal, y un gran reto en la comunicación entre los miembros de un equipo distribuido es la pérdida de señales no verbales. En 1959, Edward T. Hall; un renombrado antropólogo social, sostenía que en una conversación normal más del 65% del significado social se produce a través del canal no verbal. En una reunión a distancia, al no disponer de las típicas pistas no verbales de significado, sólo se dispone del 35% del canal de comunicación normal.

Este tipo de problemas se acentúa cuando en las reuniones alguno de los miembros no utiliza video y solo se comunican mediante audio. Esto puede darse debido a un problema técnico, o simplemente porque la persona no se siente cómoda encendiendo la cámara.

#### 6.2.5 Coordinación de calendario

El tener miembros del equipo de diferentes países, también afecta la manera en que se organizan y planifican los días libres o vacaciones de sus miembros dado que; por ejemplo, cuando en Argentina es verano y la mayoría se toma vacaciones, en países como España es invierno. Esto genera que gran parte del equipo esté ausente; o peor aún, que las personas no se puedan tomar vacaciones y tengan que ajustar su calendario para que se adapte al del país del cliente.

### 6.2.6 Desarrollo sobre una misma funcionalidad

Cuando un equipo distribuido tiene que desarrollar una misma funcionalidad utilizando una única línea de control, se puede decir que se está aplicando un enfoque lineal. En dicho enfoque todos los desarrolladores van subiendo las tareas que han terminado a la misma línea del control de versiones.

Trabajar de esta manera tiene un problema muy importante, y es que los errores se propagan más fácilmente o que no se pueda asegurar de que la línea principal tenga una versión estable del código. Por esta razón, es posible que los desarrolladores no suban su código al control de versiones frecuentemente hasta que sea estable, para evitar introducir errores y se propague a otros desarrolladores. Por lo tanto, se infringiría la recomendación de que todo el código esté bajo control de versiones y a menudo integrado.

Por otro lado, un enfoque para la integración continua, podría ser que cuando un desarrollador intente subir su código a una única línea principal, el servidor de integración continua intente compilar su código con el que ya existe en la línea principal y ejecutar los tests. Si todo es un éxito, el código se subirá correctamente al control de versiones, y si no, el desarrollador tendrá que solucionar los errores. Hasta que el servidor no termina, no se sube al control de versiones, por lo que puede que se pase un tiempo sin hacer nada hasta recibir confirmación de que todo ha sido un éxito o no.

### 6.2.7 Tamaño del equipo

El tamaño de un equipo es muy importante; dado que como se mencionó en capítulos anteriores, este debe ser de 10 personas o menos.

Cuando se tiene un equipo distribuido, un número cercano a 10 personas puede dificultar la interacción y comunicación, dado que estas se realizan de manera virtual.

Un equipo grande en donde sus miembros además tienen diferentes usos horarios, dificulta por ejemplo la posibilidad de tener una reunión en un horario cómodo para todos.

En la Figura 18 se puede ver gráficamente cómo al aumentar el número de personas, aumentan los canales de comunicación posibles entre ellas, haciendo más difícil compartir información entre los miembros del equipo.

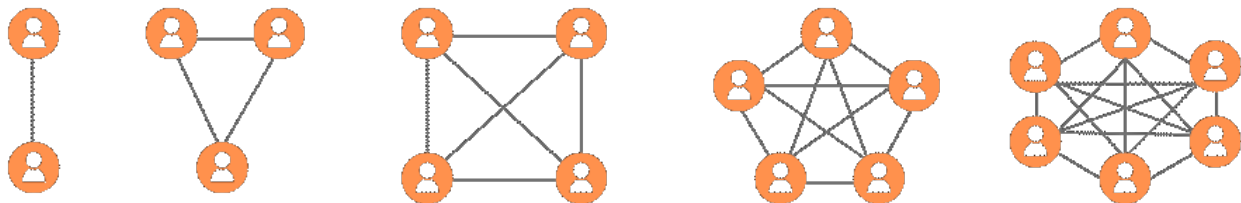


Figura 18- Canales de comunicación [12][13]

### 6.2.8 Inserción de nuevos miembros en un equipo distribuido

Otro desafío de tener equipos distribuidos se presenta cuando ingresa un nuevo miembro al mismo. Al estar todas las personas distribuidas en diferentes locaciones, se hace más difícil poder integrarse y conocerse, sumado también a algunos de los problemas antes descritos, como pueden ser diferencias culturales e idioma.

Por ejemplo, si una persona de Argentina ingresa a un equipo en donde la mayoría son de Estados Unidos y; claramente todos hablan en inglés, se hace muy difícil que esta persona se pueda integrar con facilidad, por lo que lleva bastante tiempo lograr una unión de este nuevo miembro con el equipo.

## 6.3 Eventos de Scrum distribuido y sus desafíos

### 6.3.1 Sprint Planning Distribuida

Cuando se realiza una Sprint Planning de manera distribuida, es necesario una coordinación extra, de manera que todos los miembros puedan reunirse a pesar de que algunos estén de manera remota e incluso con otro huso horario. Por ejemplo, en los equipos cuyos miembros no tengan horas en común, se presenta el problema de que algunas personas se deberán conectar fuera de su horario de trabajo habitual para poder estar en la reunión.

### 6.3.2 Daily Scrum Distribuida

La Daily Scrum es uno de los eventos esenciales del Scrum y cobra aún más valor cuando el equipo se encuentra distribuido. Este evento permite a los miembros distribuidos tener un punto de encuentro en donde se puedan coordinar y colaborar uno con otros para la resolución de problemas.

Cuando se planteó la idea original de este evento en donde el equipo se reúne diariamente, se le dio el nombre de “Daily Stand Up Meeting”, el cual se podría traducir como reunión diaria parados. El objetivo de la reunión era que el equipo se reúna en una sala y estén todos parados, buscando así reducir la duración de la reunión y cumplir con el tiempo de 15 minutos o menos. Al estar todos los miembros de manera distribuida, esta reunión se realiza de forma virtual utilizando alguna herramienta de telecomunicación, sumando además una herramienta que permita al equipo divisar los artefactos de Scrum.

Al organizar el evento, se debe tener en cuenta que puede suceder que no todos los miembros del equipo se encuentren bajo el mismo huso horario, ocasionando que algunas personas tengan la Daily Scrum en un horario laboral normal y otras se encuentren en uno muy avanzado en el día.

El idioma también puede ser un problema, dado que algunos miembros que por ejemplo no tienen un buen inglés, comenten su avance de una manera muy acotada, teniendo incluso miedo de reportar un impedimento, para evitar así tener que responder preguntas.

### **6.3.3 Review del Sprint distribuida**

La Review del Sprint de manera distribuida, puede resultar desafiante en términos de comunicación, horarios y calidad de conexión. Esto ocurre porque al momento en que el equipo muestre el trabajo realizado al Product Owner, se pueden presentar malos entendidos que en una reunión de manera presencial no se tienen, por ejemplo: error de conexión al momento de compartir pantalla, ruidos externos mientras se está presentando el trabajo, que el Product Owner no pueda escuchar o ver claramente lo que se está mostrando, etc.

### **6.3.4 Sprint Retrospective distribuida**

Este evento es uno de los más desafiantes para realizar de manera distribuida, dado que es esencial tener una buena comunicación y poder ver cómo se expresan las personas al dar un feedback. Como se mencionó anteriormente el no tener una comunicación cara a cara, afecta notablemente el modo de comunicarse y más aún cuando la persona está dando un feedback.

Otros puntos importantes a tener en cuenta son la diferencia cultural y el idioma. En el caso de las diferencias culturales, cada persona deberá considerar y tener cuidado en los modos, dado que para una cultura pueden ser normales, pero para otra no. En cuanto al idioma, se genera el mismo problema que en otros eventos dado que puede pasar que; si el idioma utilizado no es el nativo de la mayoría de los integrantes, estos se van a ver limitados al momento de realizar comentarios o de expresar cómo fue el Sprint.

## 6.4. Home office en Scrum Distribuido

### 6.4.1 Por qué home office?

El home office es una forma de trabajo distribuido en el que los miembros del equipo trabajan desde su casa, parcial o totalmente. Es decir, que los equipos que implementan home office pueden tener diferentes distribuciones, la más típica es cuando algunos miembros se encuentran en la oficina y otros en sus respectivos hogares, como se puede observar en la siguiente figura:

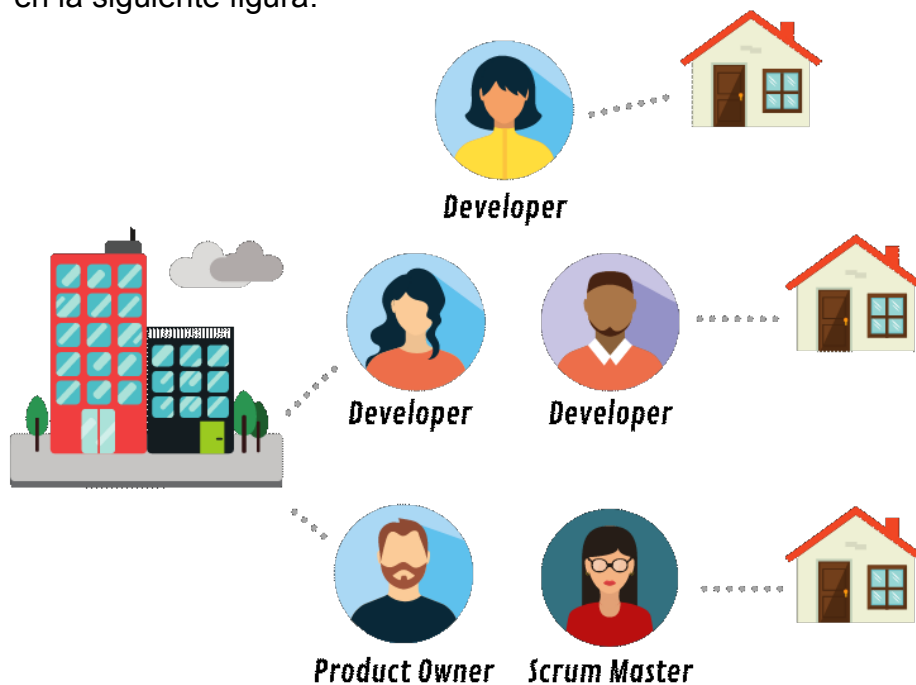


Figura 19- Equipo distribuido entre la oficina y el home office



Últimamente es muy común encontrar equipos en donde todos sus miembros trabajan desde el hogar; incluso desde otras ciudades, y ni siquiera se conocen en persona.

Dentro de las cosas positivas de implementar home office, se pueden destacar las siguientes:

- Disminuye o elimina las horas que las personas pierden para ir y venir a las oficinas.
- Disminuye la contaminación.
- Incrementa la calidad de vida.
- Permite que las personas puedan trabajar desde otras ciudades, disminuyendo así la población de las grandes capitales.
- Flexibilidad horaria.
- Menores costos de mantenimiento de oficinas.
- Más tiempo con la familia y mascotas.

Por otro lado, al ser una forma de trabajo distribuida, se tienen diferentes dificultades al trabajar en equipo, la mayoría de estas fueron mencionadas en el capítulo anterior, como por ejemplo falta de comunicación, lenguaje, cultura, etc.

#### **6.4.2 Retos implementar Scrum con home office**

Cuando un Scrum Team, se encuentra distribuido en su totalidad con la modalidad home office, es cuando se hace totalmente lo contrario a lo que establece uno de los principios del Manifiesto ágil [6], el cual expresa la importancia de la comunicación cara a cara.

Entre los retos que cada miembro del equipo puede encontrar en su casa se destacan:

- **Hijos en casa:** Uno de los retos más grandes para los padres que realizan home office, es justamente el cuidado de los hijos. Es posible que ambas actividades se crucen, los horarios se solapen y esto genere agotamiento físico y mental (tanto para los padres como para los hijos).
- **Calidad de conexión:** En algunos países del primer mundo no es común que la conexión falle, pero en Argentina se tiene el problema de que la conexión es bastante inestable, esto puede ser un gran problema dado que la persona se puede quedar sin internet en el medio de un evento importante, como por ejemplo la Sprint Review.
- **Lugar para trabajar cómodos:** No siempre se encuentra un lugar cómodo para trabajar desde el hogar, dado que no todas las casas tienen una habitación dedicada en donde montar una oficina personal, con lo cual la persona se puede encontrar con diversos ruidos, interrupciones o gente pasando por detrás en el medio de una reunión utilizando cámara.
- **Distracciones:** Los ruidos ambientales, las tareas del hogar, los familiares, las mascotas, las fallas técnicas y hasta las mismas herramientas tecnológicas son algunas de las distracciones que enfrentan las personas al trabajar desde casa, en especial quienes residen en grandes ciudades.
- **Estructuración del día:** Al estar trabajando desde su casa, las personas suelen perder la noción del tiempo y es muy común que tomen el almuerzo a cualquier hora y que se queden trabajando más tiempo del habitual.

## 6.5 Conclusión

Se plantearon los desafíos generales más frecuentes, así mismo se escribió sobre los eventos y desafíos específicos en cada uno de ellos, como por ejemplo las diferencias culturales y de idioma. En un apartado especial se escribió sobre los motivos que llevan a miembros de un equipo a trabajar con la modalidad home office , y los desafíos que se presentan al implementar Scrum en dicha modalidad.

## **CAPÍTULO 7: Encuesta sobre Scrum Distribuido**

### **7.1 Introducción**

Con la finalidad de poder analizar diferentes factores relacionados al día a día de las personas que trabajan utilizando el framework Scrum; y además lo hacen de manera distribuida, se decidió generar una encuesta.

Entre los factores analizados se pueden mencionar; por ejemplo, la distribución de los equipos, qué valor les dan a los eventos que realizan de forma distribuida, con qué desafíos se encontraron los encuestados y cuáles son las cosas que valoran positivamente de esta modalidad de trabajo.

### **7.2 Descripción de la población.**

Como se mencionó anteriormente, esta encuesta se realizó a personas que trabajan; o trabajaron, con el framework Scrum de manera distribuida, con la particularidad además de que los encuestados son del mundo IT.

Este grupo estuvo compuesto en su mayoría por actuales y ex compañeros de trabajo de los autores de la presente tesis y se obtuvieron un total de cincuenta encuestas completadas.

Dentro de las características de la población se puede destacar lo siguiente:

- Más del 40% de los encuestados, hace más de cinco años que trabaja con la metodología ágil.
- De los 50 encuestados, 24 cumplen el rol de Developer, 4 el de Product Owner y 5 el de Scrum Master. El resto se divide entre Agile Coach, Project Manager y testers.

- El 66% prefiere trabajar desde su casa, con la modalidad home office, el 7% prefiere desde la oficina. En igual porcentaje hay quienes optan por un mix entre oficina y home office.
- Con respecto a compartir el lugar de trabajo al realizar home office, se obtuvo que el 38% no lo comparte con ninguna persona y otro 38% lo comparte con su pareja. Otros comentaron además que comparten su lugar de trabajo con sus mascotas.
- Los equipos de los cuales forman parte los encuestados, están compuestos el 50% por desarrolladores localizados en Argentina, el 17% se encuentra desde Estados Unidos y un 14% se encuentra en India. El resto está distribuido entre España, México, Paraguay y China.
- De los encuestados, el 70% trabaja en equipos cuyo Scrum Master está en Argentina, el 16% mencionó que el Scrum Master está en Estados Unidos y el resto que el Scrum Master que se encuentra en Holanda, Reino Unido, Irlanda o España.
- En cuanto a las herramientas, la mayoría de los encuestados utiliza Azure DevOps, Jira y Trello.

### **7.3 Desafíos de Scrum Distribuido**

Gran parte de los encuestados expresó que se ha enfrentado a los desafíos que fueron planteados anteriormente en esta tesis, como por ejemplo diferencias culturales, idioma, huso horario, etc.

Por otro lado, los autores de la tesis aprovecharon la oportunidad para solicitar un breve testimonio sobre otros desafíos con los que se hayan encontrado los encuestados al

trabajar con Scrum Distribuido. De dichos testimonios, se pueden destacar las siguientes respuestas:

- *“Trabajando de forma remota, se dificulta controlar la motivación de las personas, las dificultades que tenga (depende mucho de cuan introvertido/extrovertido la persona sea), el sentimiento de pertenencia a un equipo y la relación interpersonal que se obtiene de interactuar todos los días con un grupo de personas.”*
- *“Lograr socializar o generar empatía con compañeros que se suman desde diferentes locaciones”*
- *“Participación de algunos de los integrantes del equipo en las distintas reuniones de manera más activa.”*
- *“Como desafío puedo citar el poder encontrar una comunicación efectiva con los miembros del equipo, cuando son gente novata, por ejemplo, no se animan a levantar la mano ante un impedimento. Al estar distribuido eso muchas veces no se ve, en cambio en la oficina es más simple para ellos el solicitar ayuda.”*
- *“En las Planning virtuales, se pierde mucho el foco y algunas veces no todos están prestando atención.”*
- *“Me encontré con el desafío de que no conozco al cliente en persona y no llegamos a establecer un buen vínculo.”*

## 7.4 Beneficios de Scrum Distribuido

En lo que respecta a los beneficios de trabajar usando Scrum Distribuido, 45 de los 50 encuestados eligieron el trabajo remoto como uno de los principales beneficios, 36 valoraron la posibilidad de tener flexibilidad horaria y 30 la calidad de vida que brinda el poder trabajar de manera remota.

Pero también se aprovechó la oportunidad para consultar sobre otros beneficios con los que se hayan encontrado al trabajar con Scrum Distribuido y de lo cual se pueden destacar las siguientes respuestas:

- *“Conocer otras culturas”*
- *“Poder trabajar en empresas que antes requerían presencialidad obligatoria.”*
- *“Aumento de productividad al estar en un ambiente menos estructurado y personalizado a cada persona, también debido a la reducción de estrés que conlleva el día a día en una oficina.”*
- *“Poder pasar más tiempo con mis hijos, menos horas de viaje ida y vuelta a la oficina. Aunque disfruto de ir.”*
- *“Poder hacer uso del "home office" desde donde lo necesite, sin necesidad de un espacio físico fijo.”*

## 7.5 Conclusión

La hipótesis que los autores tenían antes de realizar esta encuesta era que la mayoría de las personas iban a considerar que los eventos de Scrum realizados de manera

presencial son más efectivos, pero los resultados indicaron lo contrario. Los encuestados eligieron porcentajes muy similares en cuanto a la efectividad de los eventos de Scrum que se realizan de forma virtual y los que se realizan de forma presencial. Hay eventos que para los encuestados incluso son más efectivos si se hacen de manera virtual y distribuida, estos son la Review y la Daily.

Los autores consideran que esto se debe en gran parte, a los avances tecnológicos que se han sucedido en los últimos años, los cuales permitieron acceder a múltiples herramientas de comunicación online y de trabajo colaborativo. Sumado; además, a que los encuestados tienen una buena capacidad de adaptación a estas herramientas.

No es un dato menor que esta encuesta fue realizada luego de una pandemia que afectó a toda la población y llevó a que muchas personas pasaran a trabajar de manera distribuida desde sus respectivos hogares.

Por lo tanto, los autores concluyen en base al análisis de los datos recolectados, que la mayoría de los encuestados valora positivamente trabajar de manera distribuida, participando en gran parte de los eventos de Scrum de manera virtual.



## CAPÍTULO 8: Experiencias personales de los autores

### 8.1 Introducción

A lo largo de los años, los autores de la tesis han tenido diversas experiencias utilizando Scrum distribuido, a continuación, se brindarán detalles de las mismas, mencionando los diferentes tipos de equipos, la distribución de los mismos y los desafíos a los cuales se han enfrentado.

### 8.2 Proyectos distribuidos en los que trabajó Pablo:

Pablo trabaja en IT desde el año 2011 y en el año 2015 comenzó a trabajar con Agile, principalmente utilizando Scrum. Ha tenido la oportunidad de participar en diversos proyectos distribuidos; siempre como desarrollador, a continuación, se detallan los cuatro más importantes:

#### 8.2.1 Primer proyecto

**Proyecto:** Caterpillar - Catálogo de maquinaria.

**Rol dentro del proyecto:** Developer Junior (front-end).

**Cantidad de equipos distribuidos en el proyecto:** Un solo equipo.

**Tamaño del equipo del cual formó parte:** El equipo estaba compuesto por siete personas, entre ellas cinco desarrolladores, un Scrum Master y un Product Owner.

**Distribución del equipo del cual formó parte:** El Scrum Master y el Product Owner se encontraban en Chicago, dos desarrolladores en la oficina de Buenos Aires, dos desarrolladores en Chicago, y Pablo desde una oficina en La Plata.

**Desafíos encontrados en el proyecto:** Al estar solo en la oficina, Pablo tenía muy poco contacto en persona con sus compañeros de Buenos Aires, con decir que solo los vio una o dos veces en un año.

Todas las Sprint Review eran de forma virtual, salvo dos que se realizaron en la oficina de Buenos Aires desde una sala con un dispositivo llamado Round Table, en donde se conectaron los compañeros de Argentina, el Scrum Master y el resto de los desarrolladores que estaban en Chicago. En ese contexto, mantener el diálogo fue muy difícil, debido a que el sonido no era bueno, además el idioma complicaba mucho a Pablo para entender a los demás. Prácticamente no tomó parte del evento, solo al momento de presentarse.

### 8.2.2 Segundo proyecto

**Proyecto:** Desarrollo de una herramienta para uso interno de la empresa.

**Rol dentro del proyecto:** Developer Full Stack.

**Cantidad de equipos distribuidos en el proyecto:** Dos equipos, un equipo de desarrollo y otro de testing.

**Tamaño del equipo del cual formó parte:** El equipo de desarrollo estaba compuesto por siete personas, entre ellos cinco desarrolladores, un Scrum Master y un Product Owner.

**Distribución del equipo:** El Scrum Master se encontraba en Madrid, el Product Owner en Chicago, los desarrolladores estaban distribuidos entre Madrid y La Plata. El equipo de testing estaba distribuido entre España, Filipinas y China.

### **Desafíos encontrados en el proyecto:**

La comunicación entre los equipos de La Plata y de China solo era por mail debido a las once horas de diferencia que entre ellos existía. Por lo general se tenía como intermediario de estos al equipo que se encontraba en España. Prácticamente el contacto con el equipo de China era nulo, el problema surgía cuando alguien de parte de ese equipo debía testear algo implementado por alguien de La Plata, por ejemplo.

Las dificultades con el idioma era algo muy común, dado que era muy difícil poder entender a los compañeros de China hablando en inglés.

### **8.2.3 Tercer proyecto**

**Proyecto:** Sotheby's - Catálogo de obras de arte.

**Rol dentro del proyecto:** Developer (UI).

**Cantidad de equipos distribuidos en el proyecto:** Tres equipos, un equipo de desarrollo de front-end, un equipo de desarrollo de back-end, un equipo de testing, y para los tres equipos un Scrum Master y una Product Owner.

**Tamaño del equipo del cual formó parte:** El equipo de desarrollo de front-end estaba compuesto por siete personas, entre ellos cinco desarrolladores, un Scrum Master y una Product Owner.

**Distribución del equipo:** El Scrum Master se encontraba en Colombia, la Product Owner en Nueva York. Los desarrolladores del equipo se encontraban distribuidos entre Buenos Aires y Colombia. El equipo de testing se encontraba en Colombia e India.

**Desafíos encontrados en el proyecto:** El problema en este proyecto fue que no se utilizó Scrum de manera correcta, dado que no había Sprint Review formales, las Daily no eran diarias, solo se tenían tres semanales y tampoco había Retrospective.

El trabajo que se realizó en front-end era la actualización de la tecnología, ya que se había desarrollado en una versión de Angular muy antigua, la misma tenía un par de años desde su uso y no había documentación o alguien del equipo original que ayudara. Por otra parte, el equipo fue sufriendo bajas de desarrolladores Sr y altas de desarrolladores juniors.

#### 8.2.4 Cuarto proyecto

**Proyecto:** Examity - Herramienta de supervisión de exámenes online

**Rol dentro del proyecto:** Developer Full Stack.

**Cantidad de equipos distribuidos en el proyecto:** Un equipo de desarrollo.

**Tamaño del equipo del cual formó parte:** El equipo de desarrollo estaba compuesto por seis personas, entre ellos cuatro desarrolladores, un tester y un Product Owner. Se presentaba la particularidad de no tener Scrum Master.

**Distribución del equipo:** El líder de desarrollo y el Product Owner se encontraban en Boston, el resto de los desarrolladores en La Plata y Santa Fe, el tester en Mar del Plata.

**Desafíos encontrados en el proyecto:** Se detectaron varios problemas en este proyecto, las Daily solían durar más de quince minutos(solo eran seis personas) y generalmente se extendían por culpa de charlas o discusiones extras.

Otro problema bastante común era que el anterior Product Owner constantemente cambiaba los horarios de los eventos(Daily y Refinement). Esto se solucionó un poco con la llegada de un nuevo Product Owner.

No había Scrum Master, el líder del equipo de desarrollo se encargaba de todo lo relacionado a dicho rol, por ende, no dedicaba mucho tiempo a desarrollo.

Las Sprint Review eran en conjunto con otros equipos dedicados a otros proyectos del cliente y a veces solían ser muy largas y aburridas.

En el equipo de desarrollo existían demoras en las revisiones de Pull Requests, y eso hacía también que se demore el testing, era muy frecuente tener que corregir algo un día antes de la Sprint Review debido a estas demoras.

Solo se contaba con un tester, el cual estaba siempre muy cargado de responsabilidades, lo que hacía que los desarrolladores dejaran de lado sus tareas para poder ayudarlo.

Otro problema fue que algunos de los equipos estaban trabajando en bloques de código del mismo sistema. Se empezaron a presentar errores con varios tests rotos, dado que la configuración del proyecto no estaba hecha para controlar el código que se subía al repositorio. El backlog no estaba totalmente definido, y se tenían constantemente eventos de Refinement.

### **8.3 Proyectos distribuidos en los que trabajó Joaquin:**

Joaquin trabaja en IT desde el 2012 y en el año 2016 comenzó a trabajar con Agile, primero como desarrollador y luego como Scrum Master, ha tenido la oportunidad de participar en diversos proyectos distribuidos, a continuación, se describen los cuatro más importantes:

### 8.3.1 Primer proyecto

**Proyecto:** Desarrollo de una herramienta para uso interno de la empresa.

**Rol dentro del proyecto:** Developer Full Stack.

**Cantidad de equipos distribuidos en el proyecto:** Tres equipos, un equipo de desarrollo de back-end, un equipo de desarrollo de front-end y otro de testing, y para los tres equipos un Scrum Master y un Product Owner.

**Tamaño del equipo del cual formó parte:** El equipo estaba compuesto por cinco personas, entre ellos tres desarrolladores, un Scrum Master y un Product Owner.

**Distribución del equipo:** El Scrum Master y el Product Owner se encontraban en España. El equipo que realizaba desarrollo de back-end se encontraba en Madrid, el equipo que realizaba desarrollo de front-end estaba distribuido entre La Plata y Madrid. Por último, el tercer equipo; encargado de testing, se encontraba en India.

**Desafíos encontrados en el proyecto:** El principal desafío en este proyecto se encontraba del lado del código, ya que al haber dos equipos trabajando sobre una misma funcionalidad, los mismos tenían que estar muy atentos al momento de subir cambios y deployar.

Por ejemplo, cuando el equipo de back-end subía algún cambio en un servicio, y dicho cambio afectaba la forma en que el equipo de front-end consumía el mencionado servicio, pasaba frecuentemente que las cosas dejaban de funcionar porque el cambio no era oportunamente informado.

En los primeros eventos de Retrospective se dio un caso particular, algunas personas del equipo de Argentina se sentían incómodas por el modo en que las personas de España expresan las cosas, elevando la voz.

### 8.3.2 Segundo proyecto

**Proyecto:** Desarrollo de aplicación interna de la empresa.

**Rol dentro del proyecto:** Scrum Master.

**Cantidad de equipos distribuidos en el proyecto:** Cinco equipos, tres equipos de desarrollo front-end y dos equipos de desarrollo back-end. Un Scrum Master por equipo y un Product Owner para todos los equipos.

**Tamaño del equipo del cual formó parte:** El equipo estaba conformado por seis personas, entre ellos cuatro desarrolladores, un Scrum Master y un Product Owner.

**Distribución del equipo:** El Product Owner se encontraba en España, el Scrum Master en La Plata, los desarrolladores de back-end se encontraban en España y los desarrolladores de front-end se encontraban distribuidos en Buenos Aires y La Plata.

**Desafíos encontrados en el proyecto:** Al contar solo con un Product Owner para los cinco equipos, la comunicación se tornaba muy difícil ya que su tiempo estaba muy limitado.

Se complicaba poder agendar eventos, dado que podía llegar a tener hasta cuatro Planning en un mismo día. Debido a la falta de tiempo, el Product Owner siempre demoraba en responder , y en algunas oportunidades el equipo se quedaba sin trabajo hasta esperar dicha respuesta. Resultaba difícil el poder coordinar la dependencia de

trabajo y de historias; es decir, más allá del trabajo de los Scrum Master, se generaban dependencias entre equipos en un mismo Sprint que desencadenaban en historias de usuario sin poder ser completadas.

### 8.3.3 Tercer proyecto

**Proyecto:** Desarrollo de aplicación interna de la empresa.

**Rol dentro del proyecto:** Scrum Master.

**Cantidad de equipos distribuidos en el proyecto:** Un equipo.

**Tamaño del equipo del cual formó parte:** Siete personas, entre ellos cinco desarrolladores, un Product Owner y un Scrum Master.

**Distribución del equipo:** El Product Owner se encontraba en Buenos Aires, el Scrum Master y el equipo de desarrollo en La Plata.

**Desafíos:** No se generaron muchos desafíos, pero se puede citar situaciones en las cuales casi todos los miembros del equipo estaban en la oficina, y unos pocos desde su casa, con lo cual se realizaba la Daily de manera virtual.

En un principio se reservó una sala para las Daily, se usaba un dispositivo llamado Round Table, el cual hacía que fuese más natural escuchar a los que estaban desde su casa, pero se perdía mucho tiempo en ir a la sala, además de imprevistos relacionados a la configuración del dispositivo, entre otros.



### 8.3.4 Cuarto proyecto

**Proyecto:** Desarrollo de aplicación interna de la empresa.

**Rol dentro del proyecto:** Scrum Master.

**Cantidad de equipos distribuidos:** Dos equipos, uno de desarrollo front-end y otro de desarrollo back-end. Un Scrum por equipo y un Product Owner para los dos equipos.

**Tamaño del equipo del cual formó parte:** Cinco personas, entre ellos tres desarrolladores, un Product Owner y un Scrum Master.

**Distribución del equipo:** El Product Owner se encontraba en Buenos Aires, mientras que el Scrum Master y los desarrolladores de front-end estaban en La Plata, y los desarrolladores de back-end se encontraban en Buenos Aires y en Rosario.

**Desafíos encontrados en el proyecto:** Era un proyecto complejo en cuanto a lo técnico, varios desarrolladores eran nuevos, y esto dificultó mucho las estimaciones.

Para este proyecto, se implementó la técnica de Pair Programming, los desarrolladores preferían no tomar el día de home office, para poder hacerlo en la oficina ya que podían sacar sus dudas más rápido.

Uno de los desarrolladores del equipo de back-end estaba solo en la ciudad de Rosario, esta persona no tenía la posibilidad de ver a sus compañeros y en varias oportunidades manifestó que se sentía aislado y sin ayuda. Además de esto, al realizar algunos eventos de manera presencial, él debía viajar desde su ciudad y volver en el día, dado que el proyecto no contaba con presupuesto como para que se quedara en un hotel.

La coordinación de los equipos fue bastante difícil dado que se generaban demasiadas dependencias entre ellos.

## **8.4 Conclusión**

En el presente capítulo se describieron algunas de las experiencias vividas por los autores a lo largo de los años, muchos de estos proyectos ayudaron a los mismos a aprender de los errores y poder experimentar con diferentes planteos de soluciones que luego se aplicaron en futuros proyectos.

## **CAPÍTULO 9: Planteo de soluciones**

### **9.1 Introducción**

Este capítulo apunta a brindar diversos consejos, “tips” o recomendaciones que pueden ser útiles al momento de afrontar los desafíos planteados anteriormente en el capítulo 6 de la tesis.

Muchos de los consejos aquí mencionados están basados en la experiencia que los autores han podido recabar a lo largo de los años trabajando con la metodología ágil y principalmente con Scrum de manera distribuida.

La primera parte se centra en las posibles soluciones a los problemas más frecuentes, para luego entrar un poco más en detalle de las mejores prácticas para cada evento en particular.

### **9.2 Posibles soluciones para los desafíos más frecuentes**

#### **9.2.1 Diferencias culturales**

Existen diversos consejos que se pueden tener en cuenta para reducir el impacto de las diferencias culturales y lograr que los miembros de equipos con culturas diferentes conozcan un poco más del otro, a continuación se menciona un listado de buenas prácticas:

- Fomentar reuniones anuales entre los miembros de diferentes países para que se puedan ver cara a cara.

- Cada cierto tiempo se aconseja rotar algunos miembros del equipo entre locaciones para que puedan trabajar por ejemplo una semana con sus compañeros en otra ciudad y así conocerse más, aprovechando esto y reunirse luego del horario laboral para conocer más su cultura.
- Realizar eventos virtuales con actividades que permitan conocerse más.
- Mantener un ambiente laboral en donde todos se sientan cómodos al expresarse.
- Respetar los feriados nacionales de cada locación.
- Utilizar correctamente las herramientas de comunicación
- Generar un espacio en donde las personas puedan compartir intereses en común, hobbies, películas, música, etc.

### 9.2.2 Diferencias de idioma

Un equipo distribuido, en el cual sus miembros utilicen diferentes lenguajes, muy posiblemente se enfrente a diferencias de idioma que afecten su capacidad para comunicarse de una manera eficiente.

Para disminuir el impacto de estas diferencias se recomienda aplicar alguno de los siguientes consejos.

- Mantener una comunicación con frases simples y concisas, evitando utilizar palabras complejas para expresarse.

- Acompañar las reuniones con alguna facilitación gráfica que permita identificar de qué se está hablando, puede ayudar a las personas que no logran traducir rápidamente lo que el interlocutor está diciendo.
- Tratar de evitar modismos o palabras propias de una región.
- Evitar utilizar abreviaturas que no son conocidas por todos los miembros del equipo, por ejemplo “BRB” en vez de decir “Be Right Back” (Regreso enseguida).
- Utilizar el chat en las reuniones, para que las personas que no se animan a hablar en otro idioma, puedan tener ese espacio para hacer preguntas.
- Puede ser de ayuda que alguno de los miembros del equipo de cada locación domine el idioma, para poder transmitir lo que sus compañeros quieren expresar.
- Cuando se realiza un acuerdo o un pedido, es aconsejable confirmar que todos los miembros del equipo entendieron correctamente, incluso se puede llegar a pedir que expresen con sus palabras cuál fue el acuerdo al que se llegó.

### 9.2.3 Diferencias horaria

Por lo general los equipos que están ubicados de manera distribuida, trabajan bajo diferentes husos horarios y esto afecta el tiempo de interacción que existe entre ellos.

Para intentar reducir estos problemas se pueden utilizar diferentes estrategias, las cuales se detallan a continuación:

- Establecer; al inicio del proyecto, cuál va a ser la zona horaria a utilizar para los eventos. Esto se puede complementar incluyendo una descripción al momento de enviar la invitación a las reuniones, comentando cual es el huso horario elegido y su equivalente en los diferentes países. Por ejemplo, si el horario elegido para realizar una Daily es a las 9:00 AM horas de Chicago (GMT-5), se puede aclarar que en horario de Buenos Aires es 11:00 AM horas (GMT-3)
- Mantener actualizado el calendario personal en Outlook (o similar), reflejando además los horarios de almuerzo. Esto puede ayudar, dado que cada país tiene diferentes costumbres en cuanto a la hora del almuerzo. Tener esto registrado permite que no se agenden reuniones en ese horario.
- Utilizar la técnica que se denomina “Share the pain” (Compartir el dolor). En algunas ocasiones un evento puede quedar seteado en un horario que puede resultar cómodo para un equipo, pero incómodo para otro (fuera del horario laboral, excesivamente temprano, etc.), en estos casos se recomienda ir alternando el horario y por ejemplo en un Sprint realizar el evento en un horario cómodo para el equipo A y en otro en un horario cómodo para el equipo B. Esto permite que no sea siempre el mismo equipo el que tenga que “sacrificarse” para poder asistir al evento.

## 9.2.4 Comunicación

Al trabajar en un equipo distribuido se debe realizar un esfuerzo extra para mantener la comunicación, dado que se pierde algo muy importante que es el poder comunicarse cara a cara con los compañeros de equipo.

Dentro de las estrategias que se pueden seguir para mejorar la comunicación de manera distribuida se destacan las siguientes:

- En caso de tener equipos en diferentes países, se aconseja que los Scrum Master de estos se comuniquen al menos una vez por semana, para poder estar al tanto de cualquier bloqueo que afecte a uno de los equipos. Esto se puede realizar por ejemplo en la Scrum Of Scrum.
- Crear canales de comunicación entre los miembros del equipo (Chat grupal), para que puedan comunicarse por ese medio y que todos estén al tanto. Esto se puede extender a un canal con los miembros de otros equipos.
- Es aconsejable que; en la medida que sea posible, se utilice la cámara web en las reuniones y principalmente en los eventos de Scrum, dado que la comunicación visual es esencial, para poder entender al otro de una manera más clara. Esto puede evitar además los malentendidos, dado que alguien puede escribir algo por chat y otra persona puede llegar a interpretarlo de una manera diferente, cosa que no ocurre si la persona ve y escucha al otro cuando lo dice.

- Crear una lista de correo/email con todos los miembros del equipo, de esta manera al enviar un correo se puede poner en copia a todo el equipo fácilmente, evitando así que algunos miembros queden fuera.
- En el caso de que los miembros del equipo se encuentren en la misma ciudad, pero trabajan de forma remota desde su casa, se puede implementar que con cierta frecuencia se reúnan en la oficina o en un espacio de coworking, y así permitir que se integren mejor entre ellos y tener un espacio de comunicación más natural.
- Los Sprint cortos ayudan a detectar y resolver más rápidamente problemas de comunicación, dado que brindan la posibilidad de poder discutirlos en el evento de Retrospective y pensar en conjunto qué tareas llevar a cabo para mejorar la comunicación.
- Organizar eventos integradores fuera del horario laboral, para permitir que los miembros del equipo se conozcan más.

### 9.2.5 Coordinación de calendario

Similar a lo que ocurre con el huso horario, al tener equipos cuyos miembros trabajan desde diferentes países, éstos tienen diferentes calendarios, ya sea por los días festivos de cada país o por la etapa del año en la cual es más normal tomarse vacaciones.

Dentro de las estrategias que se pueden seguir para afrontar esta diferencia, se destacan las siguientes:



- Es aconsejable que, al principio de cada Sprint, se consulte cuáles son los días festivos para las próximas semanas, y de esta manera tenerlos en cuenta para la planificación. Lo mismo se puede extender a largo plazo y marcar los feriados al inicio de un proyecto.
- En base a mejorar la comunicación y acercar más a los miembros del equipo, es una buena práctica el mantener un calendario de cumpleaños de los miembros de los equipos, para poder saludarlos, o darle un presente.
- En la medida que sea posible, planificar las vacaciones con tiempo, para poder cargarlas en el calendario y poder organizar mejor los Sprints.

### 9.2.6 Desarrollo de una misma funcionalidad

El trabajo distribuido puede generar diferentes escenarios a la hora del desarrollo, un ejemplo claro suele darse cuando diferentes desarrolladores trabajan juntos en una misma funcionalidad, o parte del desarrollo involucra la modificación de algún bloque de código compartido.

A partir de esto hay que tener especial cuidado en qué momento se sube el código, y controlar si el cambio no afecta a otro equipo.

Existen varias estrategias que se pueden seguir para afrontar dificultades técnicas, los autores destacan principalmente tres; Integración Continua, Test Automation y TDD (Desarrollo controlado por pruebas).

### 9.2.7 Tamaño del equipo

El tamaño del equipo es un factor importante, dado que como se describió anteriormente en el capítulo 6.2.7: a mayor cantidad de miembros, mayor cantidad de canales de comunicación. De la misma manera, al tener más miembros, también se puede dar que haya más personas distribuidas en diferentes países, complejizando lo anteriormente mencionado acerca del huso horario, calendario, idioma, etc.

Los autores consideran que un tamaño ideal para un equipo puede ser de siete personas, conformado de la siguiente manera:

- Cinco desarrolladores (incluyendo back-end, front-end, testing)
- Un Product Owner dedicado al equipo
- Un Scrum Master dedicado al equipo.

Se hace particular referencia a un Product Owner y Scrum Master dedicado al equipo, dado que existen diversas “configuraciones” de equipos distribuidos en donde se suele tener un Product Owner dedicado a múltiples equipos y esto complejiza el normal desarrollo del Sprint, debido a que esta persona tiene que estar atento a más de un Backlog, su día a día es más complejo y puede no llegar a tomar parte de todos los eventos.

En el caso de tener más de un equipo distribuido, es ideal poder tener un punto de contacto, es decir, reuniones diarias o semanales entre los Scrum Master de los diferentes equipos y de igual manera entre los Product Owner.

Al tener diversos equipos distribuidos, generalmente sucede que varios estén trabajando sobre la misma funcionalidad como se comentó anteriormente, a partir de esto hay que

tener especial cuidado en qué momento se sube el código, si ese cambio no afecta a otro equipo, etc.

Otro punto que puede ser tenido en cuenta al contar con varios equipos distribuidos, es justamente la distribución. Sería aconsejable en la medida de lo posible que los miembros de cada equipo estén bajo el mismo huso horario y en un caso ideal, en el mismo país.

### **9.2.8 Inserción de nuevos miembros en un equipo distribuido**

La inserción de un nuevo miembro es un factor importante, dado que como se describió anteriormente en el capítulo 6.2.8, puede suceder que éste se sienta aislado del resto, y que por lo general no llegue a conocer a sus compañeros en persona, más aún si están en otros países, afectando su rendimiento y comunicación.

En este sentido el Scrum Master, puede ayudar a que los nuevos miembros se integren de una manera más rápida, facilitando alguna actividad integradora, fomentando además que todos los miembros hablen en los eventos que se suceden en el Sprint.

Un gran punto de contacto, por ejemplo, es la Daily, en donde cada uno de los integrantes tiene su momento para hablar y esto los hace sentir más integrados como equipo.

### **9.3 Consejos para llevar adelante eventos distribuidos**

En esta parte del capítulo los autores mencionan diferentes aspectos a tener en cuenta al llevar adelante la implementación de Scrum de manera distribuida, considerando cada evento en particular.

### 9.3.1 Sprint Planning

Al momento de realizar una Sprint Planning de manera distribuida existen diversas estrategias que se pueden tener en cuenta para que la misma sea más efectiva, considerando además que los participantes se sientan cómodos, dado que es uno de los eventos que más tiempo toma dentro del Sprint.

A continuación, se mencionan una serie de buenas prácticas para llevar adelante el evento.

- Cuando se tienen miembros del equipo que están en husos horarios diferentes y además no se tienen muchas horas en común, una buena práctica puede ser dividir la Planning en dos sesiones, con la finalidad de que al durar menos esta pueda generarse sin que los miembros tengan que excederse de su horario habitual de trabajo.
- Cuando más semanas tenga un Sprint, más trabajo se puede realizar en el mismo, pero esto implica que el equipo tenga más cosas para planificar y por ende tener una Planning larga. En estos casos es oportuno realizar sesiones previas de refinamiento, que permitan ir planificando y conociendo el trabajo de los futuros Sprint con anterioridad.
- Cuando un evento de Planning tiene una duración de dos horas o más, siempre es oportuno realizar una pausa cada cuarenta y cinco minutos aproximadamente, el Scrum Master también puede facilitar alguna actividad que permita distender al equipo.

- El Scrum Master tiene que estar atento a que todas las personas del equipo puedan sacar sus dudas al momento de planificar y que todas tengan la oportunidad de brindar su opinión. Éste cobra más valor cuando el evento se realiza en una sala de reuniones con personas que están presentes físicamente y otras de modo remoto.
- Similar a lo anterior, se debe tener especial atención en que las personas no se superpongan al hablar y se escuchen unas a otras. Para esto se puede establecer alguna regla simple, por ejemplo, que para hablar antes levanten la mano “virtualmente” utilizando alguna funcionalidad de la herramienta del chat.
- Al realizar la reunión de manera virtual, se puede presentar el problema de que algunas personas no estén prestando atención o estén realizando multitasking mientras toman parte de la reunión. En estos casos una posible solución es que luego de que el Product Owner presentó un elemento del Backlog, el equipo tenga que transmitir que fue lo que entendieron, para poder chequear que todos entendieron lo mismo y estaban atentos.
- En los casos en los que sea posible, tener las cámaras activadas durante el evento de Planning, ayuda a que todos estén más focalizados y evita en cierta medida el multitasking.
- Otra posibilidad que brindan las herramientas con las que se realizan los eventos remotos, es la de grabar el evento. En el caso de la Planning tiene diversas ventajas. Por ejemplo, permite que los miembros que se encuentran en otro huso horario y no pudieron asistir, tengan la posibilidad de ver la grabación y estar al tanto de lo que se habló. Otra ventaja se produce del lado del idioma, dado que, si la misma se

realizó en Inglés y algunos de los miembros del equipo no se sienten familiarizados con el mismo, estos pueden ver varias veces la grabación para escuchar las cosas que se perdieron durante el evento.

### 9.3.2 Daily Scrum

El objetivo de este evento es facilitar la transferencia de información y la colaboración entre los miembros del equipo para aumentar su productividad.

Cuando se lleva a cabo en equipos distribuidos con diferencia horaria, con miembros que hablan diferentes idiomas, etc., es muy importante evaluar en qué momento debe realizarse el evento, dependiendo del huso horario que se tome de referencia.

Como ya se mencionó en el capítulo 5.2 de la presente tesis, existen cuatro niveles de equipos distribuidos; en tres de estos niveles, algunos o todos los miembros participan del evento de forma remota. Existen prácticas para dichos niveles, que tratan de minimizar los obstáculos e intentar así respetar el espíritu ágil y de Scrum:

#### **Equipos co-locados de manera parcial:**

- Reunión presencial para los miembros del equipo colocado, con algún dispositivo que los conecte a los miembros que estén remotos.

#### **Distribuidos con horas superpuestas:**

- Es aconsejable encontrar un momento en el que las horas de trabajo coincidan.

- Es deseable que los miembros ubicados en el mismo lugar físico se reúnan cara a cara, usando además algún dispositivo que los conecte a los miembros que estén remotos.
- Se puede utilizar el chat para dar claridad cuando la comunicación verbal (o mismo el idioma) es un problema.

#### **Distribuidos sin horas superpuestas:**

- Seleccionar una hora estándar de reunión en la que todos puedan asistir (la ya mencionada técnica “Share the pain”).
- Alternar los horarios de las reuniones, ya que permite que el equipo se desconecte en horas que sean más cómodas.

Cabe destacar que, independientemente del enfoque que el equipo seleccione, el mismo se puede reevaluar al final de cada Sprint, siempre intentando buscar la mejora continua.

#### **9.3.3 Sprint Review**

Al momento de realizar una Sprint Review de manera distribuida existen diversas estrategias que se pueden tener en cuenta para que la misma sea más efectiva, considerando que a este evento puede ser el que más personas asistan de manera distribuida, dado que no solo toman parte los miembros del equipo Scrum, sino que también los Stakeholders y cualquier otra persona que ellos consideren oportuna.

A continuación, se mencionan una serie de buenas prácticas para llevar adelante el evento.

- Es aconsejable planificar la agenda con anterioridad, estableciendo qué se va a mostrar, en qué orden y quienes serán los encargados de hacerlo. Esto permitirá que la presentación del trabajo realizado a los interesados, fluya de una manera más simple.
- Siempre es aconsejable que una persona esté al tanto de toda la agenda y preparada para tomar el lugar de un compañero.
- Tratar de que sean dos personas las que estén encargadas de mostrar el trabajo realizado, dado que todo se realiza de manera virtual y existe la posibilidad de que el presentador tenga algún problema técnico, como por ejemplo quedarse sin luz o internet al momento de mostrar el trabajo. Es en ese caso que la otra persona asignada puede tomar su lugar y continuar con la presentación.
- Una buena práctica en general es que el facilitador del evento, es decir el Scrum Master, presente cuál va a ser la agenda y luego ceda la palabra a los desarrolladores, para que ellos presenten el trabajo realizado.
- En los casos en que los miembros del equipo no tienen el mismo huso horario, se aconseja alternar los horarios del evento cada cierta cantidad de Sprints. De esta forma algunas Review se llevarán a cabo en el horario normal de una parte del equipo Scrum y fuera de horario para otros, pero en otras será al revés. Este



concepto es el que se menciona en otras partes de esta tesis como “Share the pain” o “compartir el dolor”.

### 9.3.4 Sprint Retrospective

Al momento de realizar una Sprint Retrospective de manera distribuida existen diversas estrategias que se pueden tener en cuenta para que la misma sea más efectiva, considerando que, al ser distribuida, se realizará de manera remota y es esencial utilizar buenas herramientas que permitan al equipo interactuar más durante el evento.

- Comenzar el evento con un “Ice breaker” o actividad integradora, esto permitirá que los miembros del equipo que toman parte del evento se encuentren más distendidos y se predispongan de mejor manera.
- En caso de utilizar una herramienta virtual, para recolectar datos durante el evento, se recomienda enviar el link de ésta junto con la invitación al evento. De esta manera se permite que los asistentes puedan familiarizarse con la herramienta con tiempo.
- Tener en cuenta al utilizar herramientas de manera virtual, que todos los asistentes puedan utilizarla de manera correcta, se puede dedicar una parte del evento a demostrar cómo se usa la misma y darles un tiempo para que prueben por ejemplo mover post-it.
- En los casos en los que sea posible, es buena práctica tener al menos una vez cada cierta cantidad de Sprints el evento de manera presencial. Por ejemplo, si las

personas del equipo están en la misma ciudad, pero trabajan en modalidad Home Office, se puede pautar un lugar de coworking en donde realizar el evento.

- Realizar el evento de Retrospective inmediatamente después de efectuado el de Review, de esta manera en caso de que el equipo se pueda reunir físicamente en un lugar, se aprovecha para tener los dos eventos el mismo día, tomando ventaja de la presencialidad de los integrantes. En el caso en que los miembros tengan diferentes husos horarios, se puede sacar ventaja de las horas en común, para realizar los dos eventos uno seguido del otro.
- Dado que se está trabajando con un equipo distribuido, es aconsejable aprovechar este evento para evaluar los desafíos que se mencionaron en capítulos anteriores, por ejemplo, el grado de comunicación que tiene el equipo, como se sienten en cuanto a los horarios de los eventos, la duración de estos, etc.
- Considerando que el equipo puede estar formado por miembros de diferentes culturas, hay que tener en cuenta cómo y qué actividades se realizan durante la Sprint Retrospective, es aconsejable, por ejemplo, que se tengan herramientas en donde se puedan escribir tarjetas (post it) en caso de que alguien no quiera expresarse verbalmente. No forzar a las personas que no quieren hablar o que no se sienten cómodas. También se pueden realizar actividades integradoras en el momento del “Ice breaker”, para conocer las diferentes culturas de los miembros del equipo.

### 9.3 Recomendaciones en general

Como parte final de este capítulo, se mencionan diferentes consejos que pueden ser útiles al momento de implementar Scrum de manera distribuida. No solo al llevar adelante un evento, sino también a lo largo de todos los Sprints.

- Tener en cuenta la diferencia horaria, generar un punto de contacto entre los miembros que están conectados de manera distribuida.
- Participar en las decisiones relativas a las herramientas que se utilizan para facilitar la comunicación.
- En los casos en que los miembros del equipo distribuido tengan la posibilidad de reunirse en la oficina cada determinado tiempo, se podría tomar ventaja de eso y realizar algunos eventos el mismo día.
- Cuando hay varios equipos distribuidos en el proyecto, es aconsejable tener Sprints más pequeños, para poder sacar más provecho de las Review y poder tener una mejora continua más eficiente, dado que el equipo va a estar analizando qué cosas se pueden mejorar cada menor longitud de tiempo. Los eventos van a durar menos tiempo, lo que va a favorecer a los equipos que los toman fuera de su huso horario.
- Los eventos de Scrum of Scrum, son esenciales cuando se tienen varios equipos distribuidos.
- Es aconsejable que los miembros del equipo Scrum, tengan un buen conocimiento de las herramientas que se utilizan, por ejemplo, Azure DevOps, Jira, etc. Dado que las herramientas son de gran ayuda cuando se trabaja de manera distribuida porque facilitan la coordinación y comunicación del equipo.
- Asimismo, es aconsejable siempre tener el software de gestión correcto, en el caso de que no lo sea, se pueden proponer opciones de cambio, las cuales los miembros del equipo evaluarán, siempre buscando la mejoría.

- Elegir herramientas que permitan la colaboración en tiempo real.
- Tener reuniones técnicas en donde los desarrolladores puedan discutir implementaciones posibles para ciertos problemas del desarrollo de software puede ser beneficioso.
- Mantener actualizado el Product Backlog es esencial al trabajar en un equipo distribuido, dado que aporta transparencia y permite que todos los miembros del equipo estén al tanto de las prioridades y en que está trabajando cada compañero.
- Evitar conversaciones secundarias, las cuales resultan problemáticas para los miembros no involucrados, pero especialmente para los que participan de forma remota.
- Puede resultar de utilidad el tener notas disponibles en una ubicación común, los denominados wikis, que permiten a varios usuarios escribir al mismo tiempo. Esto podría ser un buen repositorio
- Garantizar la presencia de todos en el horario acordado, con los medios y las herramientas correspondientes
- Evitar los cambios reiterados en los horarios de las reuniones, esto resulta un problema grande si existen diferencias horarias.
- Tener una buena configuración en los proyectos en lo que refiere a la fusión de código, para evitar errores en el mismo, test rotos, comportamientos indeseados, etc.
- En los diferentes eventos, el Scrum Master puede hacer periódicamente chequeos para asegurar que todo el mundo sigue escuchando bien y pueda hablar sin problema
- Enfatizar y fomentar; más de lo habitual, la comunicación abierta para aprovechar los beneficios del Scrum en entornos distribuidos.
- El facilitador de los eventos (generalmente el Scrum Master) debe intentar que los participantes no se pisen entre sí, aprovechando las ventajas de las herramientas que se utilicen.

- Organizar encuentros informales fuera del horario laboral.

## **9.5 conclusión**

Como conclusión, hay que mencionar que más allá de los consejos que los autores dejan plasmados en este capítulo, la ventaja que brinda Scrum es que siempre existe la posibilidad de inspeccionar y adaptar. Esto permite que independientemente de la solución que el equipo adopte, siempre se podrá reevaluar la decisión al final del Sprint y adaptarla para el próximo.

## **CAPÍTULO 10: Implementación de la propuesta en un equipo**

### **10.1 Introducción**

Dado que los autores de la tesis están en contacto con diferentes equipos que utilizan Scrum de manera distribuida, se decidió identificar un equipo que esté afrontando alguno de los desafíos mencionados en el capítulo 6, para ayudarlos y tratar de implementar alguna de las soluciones propuestas en el capítulo 9.

A continuación, se describe el equipo, cuáles son los desafíos y que técnicas se implementaron para disminuirlos.

### **10.2 El equipo y sus desafíos**

El equipo elegido se encuentra distribuido con horas superpuestas, de la siguiente manera:

- Cuatro desarrolladores en la ciudad de La Plata.
- Un desarrollador en la ciudad de Junín.
- Un tester en la ciudad de Madrid.
- Scrum Master en la ciudad de La Plata.
- Product Owner en la ciudad de Chicago.

Desafíos que se le presentan al equipo:

- **Interacción entre los miembros del equipo:**

El primer desafío que se presenta en el equipo es que, al estar la mayoría en la misma ciudad, estaban acostumbrados a verse al menos una vez a la semana en la oficina, pero desde la pandemia no tuvieron más contacto y no se han reunidos en el mismo lugar físico desde marzo del 2020.

- **Inserción de nuevos miembros en el equipo:**

Un poco relacionado con el primero, hace unos meses se sumó un nuevo miembro al equipo, uno de los autores tuvo la oportunidad de hablar con él y esta persona le comentó que si bien se siente cómodo y considera que sus compañeros lo ayudan mucho, le falta integrarse, y tener más charlas sobre temas generales e informales (por ejemplo, de qué equipo de futbol son o que música escuchan).

- **Diferencia de idioma:**

El tercer desafío se encuentra en el idioma, dado que el Product Owner es de Chicago, por lo cual los eventos como Sprint Planning, Sprint Review y Refinamiento, se realizan en inglés, esto complica a gran parte del equipo, dado que no todos manejan el lenguaje fluidamente. En la Planning se generan muy pocas preguntas y solo hablaba una persona del equipo, los demás suelen tener varias dudas, pero no siempre se animan a preguntar.

### 10.3 Soluciones planteadas

- **Interacción entre los compañeros del equipo e inserción de nuevos miembros:**

Se pensó en un evento recreativo para que el equipo se pueda conocer más, mejorar la comunicación y permitir que los nuevos miembros se sientan más cercanos al equipo.

El evento se realizó de manera virtual mediante una reunión en donde se invitó a los miembros del equipo a realizar diferentes actividades.

La primera actividad fue un “Ice breaker” en donde cada uno tenía 5 minutos para buscar un objeto en sus casas que consideraran que era extraño y que nadie más tenía. Esto permitió primero que se movieran y empezaran a distenderse, luego que cada uno empiece a conocer un poco más del otro, dado que tenían que mostrar el objeto y contar un poco sobre este. Entre los objetos que mostraron se encontraban, por ejemplo, un sacacorchos automático, unas monedas de colección, una entrada al recital de Soda Stereo, etc.

Como segunda actividad se presentó un tablero, que tenía varios cuadrados, cada cuadrado tenía diferentes títulos como, por ejemplo, “Practican un deporte”, “Tienen un hobby”, “Tocan un instrumento musical”. La actividad consistía en que cada uno de los miembros del equipo pongan su nombre en los tableros que los identificaban. Luego se pasó a leer los nombres de cada tablero y pedirles que contarán que deporte practicaban, hace cuanto, etc.

Como última actividad se realizó un juego virtual, que en este caso el elegido fue el clásico “Pictionary” en donde alguien dibuja y los demás tienen que adivinar la palabra.



#### - **Diferencia de idioma:**

Para afrontar las diferencias de idioma del equipo se implementaron diferentes soluciones.

Lo primero que se implementó fue una idea para mejorar las Sprint Planning, dado que varios miembros del equipo no se encuentran muy familiarizados con el idioma inglés y esto genera que no realicen muchas preguntas.

La idea consiste en que el Product Owner comparta con el equipo uno o dos días antes, cuáles son las historias de usuario que se van a ver en la Sprint Planning. Con esta información, el equipo realiza una reunión previa en donde analizan las historias y en caso de tener dudas o preguntas, el Scrum Master las anota y la realiza durante la Sprint Planning.

Relacionado con esto, también se habló con el Product Owner, la posibilidad de que puedan realizar preguntas en castellano en el chat de la reunión y el Scrum Master o algún miembro del equipo las traduzca.

En cuanto a la Retrospectiva, se dividió en dos partes, primero se realiza el evento entre los desarrolladores y el Scrum Master que facilita la misma, luego se realiza una reunión más corta en donde se suma al Product Owner para mostrarle todo lo que se habló y ver que comentarios le gustaría agregar.

#### **10.4 Resultados obtenidos**

El evento recreativo fue todo un éxito y se logró que los miembros del equipo se pudieran conocer mucho más unos a otros, algo que por ahí suele darse de modo más natural en

un modo presencial, por ejemplo, cuando se va a almorzar en la oficina o se comparte una charla cerca de la máquina de café, pero con el trabajo remoto y distribuido se pierde.

Todos contaron acerca de sus hobbies e incluso dos personas se dieron cuenta que tenían el mismo gusto por Star Wars y no lo sabían.

En cuanto a las soluciones implementadas para la diferencia de idioma, los miembros del equipo expresaron que ahora se sienten mucho más seguros en la Planning, dado que tienen la oportunidad de discutir con anterioridad sobre las historias de usuario.

En cuanto a la Retrospective, si bien es extraño que no esté el Product Owner, el equipo habló mucho más que en otras ocasiones, surgieron más acciones y se animaron a expresar qué cosas no les gustaban del desarrollo del Sprint. El tablero o herramienta que se usa para la Retrospectiva es totalmente transparente y el Product Owner puede acceder en cualquier momento, es por esta razón que el Scrum Master traduce las tarjetas al inglés, para que él pueda ver todo lo que se escribió.

Al momento de escribir este capítulo, el equipo continúa implementando estas soluciones luego de tres Sprints y expresaron que lo van a seguir haciendo.

## CAPÍTULO 11: Conclusiones

Inicialmente en esta tesis se escribió sobre el origen de las metodologías ágiles y la evolución de estas hasta el día hoy, concentrándose principalmente en el framework Scrum y en su adaptación e implementación en entornos distribuidos.

Los autores consideraron oportuno remarcar que la guía de Scrum es el último eslabón de una cadena de sucesos y buenas prácticas que se fueron publicando a lo largo de los últimos 50 años y llevaron al éxito que hoy en día tiene Scrum.

En este mismo sentido, se mencionó el gran cambio que ha tenido la forma de desarrollar software y como se inclinó hacia un desarrollo distribuido a lo largo y ancho de todo el mundo, en dónde la mayoría de las personas trabajan en equipos cuyos miembros están localizados en diferentes ciudades.

Se describieron además los diferentes modelos y niveles de equipos distribuidos, haciendo foco a algo que hace mucho se utiliza, pero gracias a la pandemia de estos últimos años está en boca de todas las personas: el trabajo remoto desde casa.

Todos estos modelos y niveles de equipos se enfrentan a múltiples desafíos, los cuales fueron analizados y descriptos en detalle en base a la investigación que realizaron los autores sobre la variada literatura existente, pero sumando además un análisis profundo de casos reales de personas que trabajan hace tiempo con Scrum Distribuido. Dichos casos fueron plasmados gracias a una encuesta a más de cuarenta personas, quienes contaron sus experiencias, sus desafíos y como afrontan o afrontaron los mismos.

Luego de recopilar toda la información necesaria acerca de la metodología, del framework Scrum, del trabajo distribuido, de evaluar y analizar las experiencias de

múltiples personas, los autores consideraron oportuno dejar evidencia de sus propias experiencias a lo largo de los mas de cinco años en los cuales han trabajado utilizando Scrum de manera distribuida en diferentes equipos y proyectos. Pero; además, aportando un beneficio al lector al plantear diferentes soluciones a los problemas más comunes dentro de un equipo que utiliza Scrum de manera distribuida.

Por último y con la finalidad de ver el real alcance de las soluciones planteadas, los autores decidieron identificar un equipo que utilizara Scrum Distribuido y que se enfrentara a alguno de los desafíos más comunes descritos, con la finalidad de acompañarlos y trabajar con ellos en la implementación de alguna de las soluciones planteadas en la tesis y realizando además un análisis detallado del caso y los resultados obtenidos, los cuales fueron gratificadamente exitosos.

Como fue detallado en la primer parte de la tesis, la idea de tener; por ejemplo, equipos auto-organizados no nació en el 2010 con la publicación de la guía de Scrum, sino que fueron temas que varios autores fueron publicando en diferentes papers a partir de la década del 80, como por ejemplo “The New New Product Development Game” en donde se menciona el termino Scrum por primera vez

En la actualidad el home office se ha convertido en una realidad a partir de la pandemia del 2020. Luego de analizar la encuesta, los autores pudieron comprobar que a pesar de que la mayoría de las personas disfrutan de esta modalidad, sienten que están mas aisladas de sus compañeros y no tienen el “sentimiento de pertenencia”. Muchas personas extrañan poder compartir un café o el almuerzo o tan solo saludarse en el pasillo.

A pesar de que los desafíos de Scrum en equipos distribuidos son variados, la mayoría de los que fueron mencionados por las personas encuestadas, se centran en cuatro en particular: Idioma, Cultura, Comunicación, Diferencia horaria.

Otra cosa a destacar mencionar de la encuesta es que antes de realizarla los autores pensaban que la mayoría iba a preferir los eventos de forma presencial (dado que es uno de los principios de Agile: el estar cara a cara en los eventos), pero el resultado fue parejo, se puede marcar que los encuestados en su mayoría prefieren específicamente que la Daily y la Sprint Review sean virtuales.

El conjunto de eventos y herramientas que proporciona Scrum permite afrontar de mejor forma los desafíos de trabajar en equipos distribuidos. En este sentido se puede destacar el rol del Scrum Master, en lo que refiere a motivar a los miembros, ser el nexo con el Product Owner, mantener un buen clima y buenas practicas para que el equipo esté comunicado de mejor manera

Luego de entrevistar a un equipo Scrum distribuido, los autores el compromiso de realizar un acompañamiento, plantear soluciones y actividades para afrontar los desafíos que el equipo estaba teniendo. Los autores continúan en contacto con el equipo para verificar como fluye todo y es muy gratificante saber que luego de varios Sprints los miembros se sienten mas seguros en cuanto a la diferencia de idioma, la relación entre todos los miembros es mucho mejor (y así mismo con el cliente) y todos se sienten mas seguros y motivados.

Los autores tuvieron desafíos al realizar la tesis de manera distribuida: falta de constancia, organización y motivación, pero realizarla utilizando Scrum los ayudó muchísimo a organizarse, estructurarse y a tener constancia.

## Referencias

- [1] Esquivel, S. (2006). *Scrum en entornos geográficamente dispersos*.
- [2] Raúl Herranz, N. M. (2011). *Scrum Distribuido*.
- [3] Woodward, E., Surdek, S., & Ganis, M. (2010). *Practical Guide to Distributed Scrum, A (IBM Press)*. IBM Press.
- [4] Paasivaara, M., Durasiewicz, S., & Lassenius, C. (2008). *Using Scrum in a Globally Distributed Project: A Case Study*.
- [5] Sutherland, J., Viktorov, A., Blount, J., & Puntikov, N. (2007). *Distributed Scrum: Agile Project Management with Outsourced Development Teams*. 40th Hawaii International Conference.
- [6] *Agile Manifesto*. (2001). Recuperado el 2021, de <https://agilemanifesto.org>
- [7] Takeuchi, H., & Nonaka, I. (1986). Obtenido de The New New Product Development Game: <https://hbr.org/1986/01/the-new-new-product-development-game>
- [8] Schwaber, K. & Sutherland, J. (2020). Obtenido de Scrum Guide [scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf](https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-European.pdf)
- [9] Cockburn, A. (2004). *Crystal Clear: A Human Powered Methodology for Small Teams*. Addison-Wesley.
- [10] Fowler, M. (1999). *Refactoring: Improving the Design of Existing Code*. Addison-Wesley.

[11] Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley.

[12] Lalsing, V., Kishnah, S., & Pudaruth, S. (2012). *People factors in agile software development and project management*. International Journal of Software Engineering & Applications (IJSEA).

[13] Schmula (2008). Lost in translation in large teams. Obtenido de <https://www.shmula.com/lost-in-translation-in-large-teams/470/>



## Anexo

### Tesis realizada usando Scrum Distribuido.

Teniendo en cuenta que los autores de la presente tesis se encontraban distribuidos en locaciones diferentes; y además ambos poseen experiencia trabajando con Scrum, se decidieron a probar la idea de desarrollar la presente tesis utilizando dicho framework, ya que consideraron que esto aportaba valor.

La distribución del equipo era 100% remota, dado que cada uno de los autores se encontraba en su hogar, y la mayor parte del tiempo en diferentes provincias (Buenos Aires y Entre Ríos). Cabe mencionar que el medio de comunicación era mediante la herramienta online llamada Google Meet.

Lo primero que se plantearon fue tener Sprints dedicados al desarrollo de los temas de los capítulos. Los Sprints eran inicialmente de tres semanas, pero en una Retrospective se decidió establecer la duración en dos semanas, lo cual se mantuvo hasta el final del desarrollo, llegando a tener veintitrés Sprints.

Cabe destacar que no en todos los Sprints se trabajó con la misma velocidad, incluso en los Sprints que estuvieron dentro de los meses de diciembre, enero y febrero prácticamente no hubo avance en el desarrollo.

Cada Sprint tuvo su respectiva Sprint Planning en donde se analizaba el Product Backlog que estaba compuesto por historias de usuario relacionadas con partes de un capítulo.

En este evento se estimaba cuantas historias de usuario se podían desarrollar (escribir) en dos semanas y en base a eso se generaba el Sprint Backlog. Siempre teniendo en cuenta factores como tiempo de búsqueda e investigación, traducción de textos al español, correcciones, etc.

Las Daily se realizaban de dos maneras posibles, de preferencia mediante una videollamada en la cual cada uno compartía cuales fueron los avances y en qué parte del capítulo estaba trabajando. Pero en caso de no ser posible por tiempo o trabajo, la comunicación era mediante chat.

Se utilizó además una especie de validación, la cual consistía en que, al finalizar una parte del capítulo, el escritor de éste se lo pasaba al compañero, para que éste pueda leerla y corregir errores de tipeo, sintaxis, completar textos, etc.

En un inicio, la Sprint Review se realizaba sólo entre los autores para hacer un repaso y releer todo lo escrito durante el Sprint. Ya más avanzada la tesis, la Sprint Review se realizaba en conjunto con la directora, para mostrarle cuáles eran los avances y las ideas propuestas para el siguiente Sprint, además de recibir su feedback.

Las Retrospective se realizaban de manera más informal, dado que solo eran dos personas, pero fue algo que contribuyó mucho a la mejora y adaptación en la forma de escribir la tesis.

Como se mencionó anteriormente, uno de los temas planteados en la Retrospective; y que se fue modificando a lo largo del trabajo, fue la longitud de los Sprints, ya que al principio se tuvieron Sprints de tres semanas, pero dicha duración resultó muy extensa (se tenían días en los cuales no se avanzaba).

Se pasó entonces a Sprints de dos semanas de duración, que permitieron tener un avance más constante y menos días ociosos.

Los autores de esta tesis se apoyaron en la herramienta Azure DevOps para crear y mantener el backlog, lo que les permitió tener una estructura y orden acerca del trabajo realizado, y el pendiente.

A continuación, se puede ver en la Figura 20 una parte de la estructura del backlog que los autores utilizaron, en donde se pueden ver épicas (tema), features (capítulo), user stories (parte del capítulo) y tareas para desarrollar los mismos.

Order	Work Item Type	Title	State	Iteration Path
2	Epic	> Scrum	Active	Tesis Agile
3	Epic	> Scrum Distribuido	Active	Tesis Agile
	Feature	> Capitulo - Scrum Distribuido	Active	Tesis Agile
+	User Story	> Scrum distribuido	Closed	Tesis Agile\Sprint 20
	Task	✓ Buscar y analizar información sobre Scrum Distribui...	Closed	Tesis Agile\Sprint 15
	Task	✓ Desarrollar un documento sobre Scrum Distribuido	Closed	Tesis Agile\Sprint 16
	Task	✓ Revisar y mejorar el documento	Closed	Tesis Agile\Sprint 18
	Task	✓ Agregar el documento a la estructura del capitulo d...	Closed	Tesis Agile\Sprint 20
	User Story	> Niveles de Scrum Distribuido	Closed	Tesis Agile\Sprint 20
	User Story	> Modelos de equipos distribuidos	Closed	Tesis Agile\Sprint 20
	Feature	> Capitulo - Desafíos de Scrum Distribuido	Active	Tesis Agile
	User Story	> Desafíos de Scrum Distribuido - Intro	Closed	Tesis Agile\Sprint 21
	User Story	> Desafíos mas frecuentes de Scrum Distribuido	Closed	Tesis Agile\Sprint 21
	User Story	> Desafíos en eventos de Scrum Distribuido	Closed	Tesis Agile\Sprint 21
	Feature	> Capitulo - Home Office y Scrum Distribuido	Active	Tesis Agile\Sprint 21
	User Story	> Home Office	Closed	Tesis Agile\Sprint 21
	Feature	> Capitulo - Experiencia de los autores	Active	Tesis Agile\Sprint 21
4	Epic	> Soluciones Propuestas	Active	Tesis Agile

Figura 20 – Product Backlog de la Tesis en Azure DevOps

Para concluir, los autores consideran que utilizar Scrum los ayudó a poder avanzar de una manera más orgánica, permitiendo además estar más enfocados en el día a día y motivados con el proceso.