

Una guía de trabajo para el fortalecimiento de prácticas colaborativas durante la adopción de Scrum



Trabajo de grado para optar al título de Ingenieros de Sistemas

Ingrith Carolina Muñoz Ordoñez

Daniel Felipe Marín Zúñiga

Director: PhD. Julio Ariel Hurtado Alegría

Codirector: PhD. Cesar Alberto Collazos Ordoñez

Universidad del Cauca

Facultad de Ingeniería de Electrónica y Telecomunicaciones

Línea de investigación de ingeniería de software

Departamento de Sistemas

Grupo IDIS – investigación y Desarrollo en Ingeniería de Software

Popayán

Nota de aceptación

Firma del Jurado

Firma del Jurado

AGRADECIMIENTOS

Quiero agradecer a Dios por brindarme la sabiduría y la perseverancia para permitirme culminar esta carrera.

Agradezco a mi querida familia por todo su apoyo, especialmente a mis padres Henry Muñoz y Sonia Ordoñez, quienes me han brindado su amor y apoyo incondicional a lo largo de mi vida, dándome su ejemplo de superación, tenacidad, humildad y sacrificio. A mis hermanas Fergie y Catherine Muñoz, por su amor, compañía y motivación. Y a mi tía Genny Ordoñez que siempre me brindó su apoyo en cada etapa de mi vida.

A mi novio Felipe Agudelo, que siempre creyó en mí y a su familia por brindarme su compañía y apoyo.

Ingrith Carolina Muñoz Ordoñez

Quiero agradecer a mi familia por estar a mi lado todo este tiempo, en especial a mi madre Luz Alba quien ha sido mi referencia de trabajo y esfuerzo y quien me ha brindado todo su apoyo, a mi hermano Fabio Ricardo por su compañía y a todos los que con su granito de arena aportaron a mi formación y crecimiento.

Daniel Felipe Marín Zúñiga

Agradecimientos especiales:

Queremos agradecer por su confianza y apoyo a nuestro director el PhD. Julio Ariel Hurtado Alegría y a nuestro codirector, el PhD. César Alberto Collazos, por quienes sentimos un gran aprecio, y a quienes admiramos por ser excelentes investigadores, profesores, pero sobre todo grandes seres humanos.

También queremos darles las gracias a las personas que nos acompañaron en este proceso y nos permitieron recolectar la información necesaria para edificar esta propuesta.

A la Universidad del Cauca por brindarnos la oportunidad de aprender y conocer gente maravillosa, a nuestros docentes que nos brindaron su conocimiento y nos ayudaron a formar como profesionales.

TABLA DE CONTENIDO

TABLA DE CONTENIDO	3
LISTA DE FIGURAS	7
LISTA DE TABLAS	8
CAPÍTULO 1	9
INTRODUCCIÓN	9
1.1. PLANTEAMIENTO DEL PROBLEMA	9
1.2. PREGUNTA DE INVESTIGACIÓN	10
1.3. OBJETIVOS	11
1.3.2. OBJETIVOS ESPECÍFICOS	11
1.4. METODOLOGÍA	11
1.5. ORGANIZACIÓN DEL DOCUMENTO	13
CAPÍTULO 2	15
MARCO TEÓRICO Y ESTADO DEL ARTE	15
2.1. MÉTODOS ÁGILES	15
2.1.1. Scrum	15
2.1.2. Adopción de Scrum	15
2.1.3. Problemas de Adopción de Scrum	15
2.2. COLABORACIÓN	16
2.2.1. Ingeniería de la colaboración - Collaboration Engineering (CE)	17
2.2.1.1. Patrones de colaboración	17
2.2.1.2. Thinklets	17
2.2.1.3. Técnicas y métodos de colaboración	17
2.3. TRABAJOS RELACIONADOS	18
2.3.1. Colaboración en métodos ágiles	18
2.3.2. Colaboración en Scrum	19
2.3.3. Problemas y estrategias relacionadas con la colaboración durante adopción de Scrum	20
2.3.4. Estrategias para la adopción de Scrum	21
2.3.5. Estudios empíricos sobre el incremento de la colaboración en Scrum	21

2.3.6.	Tabla comparativa de los diferentes trabajos relacionados	22
CAPÍTULO 3	24
SCRUM	24
3.1.	DEFINICIÓN	24
3.2.	MARCO DE TRABAJO SCRUM	25
3.2.2.	Eventos de scrum.....	25
3.2.3.	Artefactos de scrum	26
3.2.4.	Definición de “HECHO”	27
3.3.	BENEFICIOS DE SCRUM.....	27
CAPÍTULO 4	29
ESTUDIO SOBRE LOS ASPECTOS COLABORATIVOS TENIDOS EN CUENTA POR EQUIPOS DE DESARROLLO DURANTE LA ADOPCIÓN DE SCRUM		29
4.1.	ENTREVISTA SEMIESTRUCTURADA	29
4.1.1.	Diseño.....	29
4.1.1.1.	Objetivo	29
4.1.1.2.	Personas	29
4.1.1.3.	Pregunta de investigación	29
4.1.2.	Planificación	30
4.1.2.1.	Métodos de recolección de datos	30
4.1.2.2.	Protocolo.....	30
4.1.3.	Preguntas	32
4.1.4.	Resultados.....	33
4.1.5.	Análisis de datos	39
4.1.6.	Lecciones aprendidas.....	42
CAPÍTULO 5	43
ESPECIFICACIÓN DE THINKLETS E IMPLEMENTACIÓN STHINKLETS		43
5.1.	ESPECIFICACIÓN DE THINKLETS Y GENERACIÓN DE STHINKLETS	43
5.1.1.	Patrones de Colaboración	43
5.1.2.	Especificación de Thinklets por categoría.....	44
5.2.	STHINKLETS	45

5.2.1.	Proceso de creación de los Scrum Thinklets	45
5.2.2.	FASE 1: Descripción del proceso	47
5.2.4.	FASE 3: Establecimiento de tareas colaborativas	48
5.2.5.	FASE 4: Relación con colaboración	53
5.2.5.2.	Especificación SThinklets	53
CAPÍTULO 6	60
UNA GUÍA DE TRABAJO PARA EL FORTALECIMIENTO DE PRÁCTICAS COLABORATIVAS DURANTE LA ADOPCIÓN DE SCRUM		60
6.1.	ESTRUCTURA DE LA GUIA DE TRABAJO	60
6.2.	ETAPA DE EXPLORACIÓN	62
6.2.1.	Identificar la metodología de trabajo actual	62
6.2.2.	El manifiesto ágil	62
6.2.3.	Establecer el objetivo	62
6.3.	ETAPA DE PREPARACIÓN	63
6.3.1.	Formación en scrum	63
6.3.2.	Equipo scrum	63
6.3.3.	Identificar roles	64
6.3.4.	Herramientas	65
6.3.5.	Product backlog v.0	66
6.3.6.	Duración del sprint	67
6.3.7.	Definición de “Hecho”	67
6.4.	ETAPA DE DESARROLLO	68
6.4.1.	Planificación del sprint	68
6.4.2.	Reuniones diarias	73
6.4.3.	Revisión del sprint	74
6.4.4.	Retrospectiva del sprint	75
6.4.5.	Refinamiento del product backlog	78
6.4.6.	Velocidad del equipo scrum	78
CAPÍTULO 7	80
EVALUACIÓN DE LA GUÍA DE TRABAJO		80
7.1.	DISEÑO DEL ESTUDIO DE CASO	80

7.1.1.	Objetivo del estudio.....	80
7.1.2.	Caso y unidad de análisis.....	80
7.1.3.	Pregunta de investigación.....	80
7.2.	PLANIFICACIÓN.....	80
7.2.1.	Métodos de recolección de datos.....	80
7.2.2.	Teoría del riesgo.....	81
7.2.3.	Protocolo de estudio de caso.....	83
7.2.4.	Consideraciones éticas.....	83
7.3.	EJECUCIÓN DEL CASO Y RECOLECCIÓN DE DATOS.....	83
7.4.	RESULTADOS.....	86
7.5.	ANÁLISIS DE DATOS.....	87
7.6.	LECCIONES APRENDIDAS.....	88
CAPÍTULO 8.....		90
	CONCLUSIONES Y TRABAJOS FUTUROS.....	90
8.1.	RESUMEN.....	90
8.2.	CONCLUSIONES.....	90
8.3.	TRABAJOS FUTUROS.....	92
8.4.	LECCIONES APRENDIDAS.....	92
8.5.	PARTICIPACIONES.....	92
REFERENCIAS.....		93

LISTA DE FIGURAS

Figura 1: Ciclo de vida MCIS [12]	12
Figura 2: Ciclo de Scrum [53].	24
Figura 3: Proceso Scrum	47
Figura 4: Scrum process model [72].....	48
Figura 5: Estructura de la guía de trabajo.....	61
Figura 6: Mapa de calor.....	83
Figura 7: Equipo de desarrollo en la capacitación.	84
Figura 8: Reunión diaria equipo.....	85
Figura 9: Tablero de Scrum - Sprint 1.....	86
Figura 10: Matriz de riesgo preliminar	86
Figura 11: Matriz de riesgo final	87
Figura 12: Comparativa de matrices.....	87

LISTA DE TABLAS

Tabla 1: Comparativa de los trabajos relacionados	23
Tabla 2: Fragmentos de texto por categoría	34
Tabla 3: Frecuencia vs Colaboración	40
Tabla 4: Frecuencia vs Scrum	40
Tabla 5: Actividades vs Frecuencia Colaboración	41
Tabla 6: Actividades vs Frecuencia Scrum	41
Tabla 7: Patrones de Colaboración	43
Tabla 8: Thinklets patrón generación.....	44
Tabla 9: Thinklets patrón Reducción	44
Tabla 10: Thinklets patrón Organización	44
Tabla 11: Thinklets patrón Consenso.	45
Tabla 12: Thinklets patrón Convergencia	45
Tabla 13: Thinklets patrón Divergencia	45
Tabla 14: Thinklets patrón Clarificación.....	45
Tabla 15: Tabla de verbos asociados a cada patrón de colaboración	49
Tabla 16: Estructura Scrum Thinklet.....	53
Tabla 17: SThinklet_Definicion_Objetivo	54
Tabla 18: SThinklet_Seleccion_Elementos_Sprint	55
Tabla 19: SThinklet_Refinar_Elementos_Sprint	56
Tabla 20: SThinklet_Identificar_Mejoras.....	57
Tabla 21: SThinklet_Identificar_bloqueos.....	58
Tabla 22: Identificar roles - recomendación.....	65
Tabla 23: Herramientas - Recomendación	66
Tabla 24: Planificación de sprint - SThinklet_001	69
Tabla 25: Planificación de sprint - SThinklet_002.....	70
Tabla 26: Planificación de sprint - SThinklet_003.....	72
Tabla 27: Retrospectiva de Sprint - SThinklet_004	76
Tabla 28: Retrospectiva de Sprint - SThinklet_005	77
Tabla 29: Escala de probabilidad	82
Tabla 30: Escala de impacto	82

CAPÍTULO 1

INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

Las metodologías ágiles son un enfoque de desarrollo de productos software caracterizadas por su flexibilidad [1] [2]. Una de las metodologías ágiles más utilizadas es Scrum [3], la cual es un marco de trabajo para la gestión ágil de proyectos software; el uso de metodologías ágiles y específicamente de Scrum se ha vuelto cada vez más frecuente en pequeñas organizaciones desarrolladoras de software [4] [5].

La falta de comprensión de Scrum durante su adopción ha derivado en múltiples inconvenientes u obstáculos [4], incluso cuando el equipo ha recibido entrenamiento en este marco de trabajo [6]. Parte de las dificultades durante la adopción es cómo saber si ésta realmente se está logrando, en qué medida se avanza hacia la misma y qué hacer en caso de que ésta no esté funcionando. Debido a, que no hay un método oficial para medir la adopción de Scrum, la comunidad científica ha tratado de establecer los factores determinantes en su adopción [7], con el fin de que sirvan de apoyo en una iniciativa de adopción.

Dentro de los factores críticos asociados con las dificultades de adopción de Scrum, están los relacionados con el comportamiento humano [8] [7]; lo cual es crítico en el contexto ágil considerando que uno de sus principios promulga: “Los individuos y sus interacciones, sobre los procesos y las herramientas” [52]. Esto implica, que se debe dar mayor importancia a factores como la comunicación, interacción, integración, cultura organizacional, compañerismo y colaboración [9]; dado que, la madurez ágil significa fomentar más capacidades blandas, como la atención, la comunicación, el compromiso, el intercambio y la autoorganización [10].

Particularmente, la colaboración es un factor importante en el desarrollo de software, ya que, es un medio eficaz para el logro de los objetivos de negocio y el aumento del desempeño, por esta razón, las prácticas colaborativas adquieren un gran valor hoy en día para la industria local, regional, nacional y mundial [11]. La colaboración está relacionada con la capacidad que un grupo de trabajo tiene para resolver problemas complejos que no lograrían resolver de forma individual bajo las mismas restricciones de tiempo, esfuerzo y costos [11]. Al mismo tiempo, la colaboración fortalece otros aspectos importantes tales como la comunicación y la cohesión del equipo, minimizando la ocurrencia e impacto de errores en la

ingeniería de software [12]. También, la colaboración efectiva debe incluir tareas orientadas a un trabajo grupal interactivo [13] y contar con un entorno de trabajo colaborativo (espacios, infraestructura y tecnologías que apoyen el enfoque y la interacción grupal) [14]. Además, cuando el producto se va realizando a través de actividades de desarrollo de software codependientes de múltiples equipos, también se necesita la colaboración entre equipos. Lograr la colaboración en equipos de ingeniería de software incluye (1) una adecuada comunicación de requisitos, tareas, cambios, entre otros; (2) conciencia de grupo sobre aquello que cada participante está realizando y el estado actual del proyecto, y (3) las interacciones adecuadas para coordinar y realizar las tareas [17].

Pese a que los principios ágiles establecen la colaboración como aspecto clave, no hay garantía de que el equipo termine trabajando colaborativamente [15]. De acuerdo a Kanane [15], cuando un equipo Scrum no logra trabajar de forma colaborativa, se presentan las siguientes dificultades: ausencia de la propiedad colectiva de la información, ausencia de una comunicación efectiva, falta de cooperación y comunicación constante con el cliente, falta de disciplina en las reuniones diarias, falta de participación constante del cliente, falta de comprensión de los valores ágiles, falta de comprensión de los objetivos de las reuniones. Todas estas dificultades ponen en riesgo la adecuada adopción de Scrum [6].

En conclusión, si bien Scrum define aspectos de gestión dentro del contexto ágil, simplifica aspectos relacionados con el trabajo del equipo, suponiendo que la colaboración se dará como consecuencia del proceso, sin proveer recursos metodológicos para lograrla. Un claro ejemplo es que Scrum establece que el equipo debe realizar reuniones, cada una de las cuales tiene una duración aproximada y un objetivo, pero no especifica actividades y/o herramientas para garantizar que se dé la colaboración durante su desarrollo.

1.2. PREGUNTA DE INVESTIGACIÓN

A partir de los problemas reportados en la literatura de la adopción de Scrum asociados a la colaboración, se propone abordar el problema de cómo lograr una adecuada colaboración entre los integrantes de equipos de desarrollo adoptantes de Scrum, por lo que se plantea la siguiente pregunta de investigación: ¿Cómo entender y fortalecer los aspectos colaborativos (comunicación, interacción y conciencia) facilitadores de la adopción de Scrum en un equipo de desarrollo de software?

Como respuesta a esta pregunta, este proyecto propone hacer uso de algunos recursos de la ingeniería de la colaboración, con el fin de brindar apoyo durante la adopción de Scrum para alcanzar la colaboración en el equipo y así disminuir el

riesgo de adopción asociado a este factor en pequeñas organizaciones de la industria del software. Con este fin, se plantea una guía de trabajo para el fortalecimiento de prácticas colaborativas durante la adopción de Scrum para las pequeñas organizaciones de la industria del software, esta ruta de trabajo está compuesta por pasos y sugerencias que debe tener en cuenta para llevar a cabo una correcta adopción de Scrum.

1.3. OBJETIVOS

1.3.1. OBJETIVO GENERAL

Desarrollar una guía de prácticas colaborativas orientadas a disminuir los riesgos de adopción del proceso Scrum relacionados con los aspectos de comunicación, interacción y conciencia.

1.3.2. OBJETIVOS ESPECÍFICOS

- Catalogar los problemas y buenas prácticas sobre el trabajo colaborativo en equipos de desarrollo de software ágiles, particularmente en Scrum.
- Establecer los elementos de la ingeniería de la colaboración que permitan abordar los problemas más relevantes que limitan el trabajo colaborativo de equipos Scrum aprovechando las buenas prácticas identificadas.
- Proponer una guía de prácticas colaborativas para equipos de desarrollo durante la adopción de Scrum y su herramienta de soporte al monitoreo del trabajo colaborativo.
- Evaluar el efecto de las prácticas colaborativas sobre la disminución del riesgo (impacto y probabilidad de ocurrencia) relacionado con los factores colaborativos (comunicación, interacción y conciencia) a través de un estudio de caso práctico: un proyecto de adopción de Scrum en el contexto de equipos de desarrollo de software dentro de un seminario/taller de Scrum.

1.4. METODOLOGÍA

En el desarrollo de este proyecto se ejecutó el método de Hurtado [18], que consta de las fases de exploración, formulación y validación. Para la guía y evaluación de estudios de caso se siguió el método propuesto por Runeson [19].

Hurtado [18] define un método científico inspirado en los patrones de procesos útiles en el desarrollo de software, como un proceso para la realización de un método científico en un dominio de la ingeniería del software.

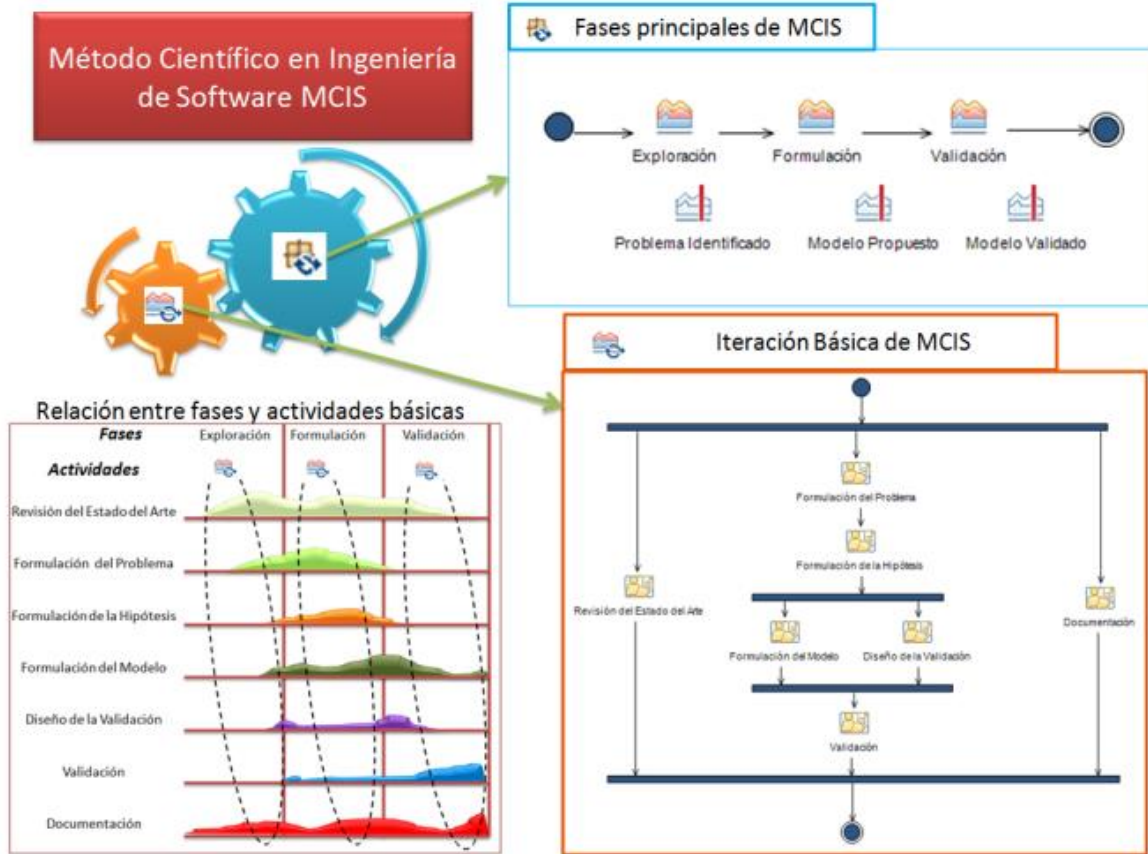


Figura 1: Ciclo de vida MCIS [12]

La Figura 1 representa el ciclo de vida de la investigación de MCIS. El proceso MCIS está organizado por tres fases (e hitos): exploración (problema definido), formulación (modelo definido) y validación (hipótesis validada). Al final de la fase de exploración, se propone un proyecto, finalizando la fase de formulación se completa el modelo y finaliza con la validación de la investigación que se presenta. Las actividades principales a realizar dentro de cada una de las fases (ver en la parte inferior izquierda de la figura 1), se describen a continuación:

La actividad de revisión del estado del arte y documentación, se realizan de forma paralela al resto de las actividades. En la actividad de formulación del problema se logra definir bien el problema, en la actividad de formulación de hipótesis se formula una hipótesis de acuerdo al problema definido y el estado del arte, la formulación del modelo se realiza según la hipótesis y las necesidades, paralelamente se hace la actividad de diseño de la validación donde se definen e implementan los instrumentos para realizar la validación del modelo, enseguida se realiza la actividad de validación.

Por otro lado, Runeson [19] indica la metodología de estudio de casos y da

directrices para realizar estudios de casos e indica que el estudio de caso es una metodología de investigación adecuada para la investigación en ingeniería de software, ya que estudia fenómenos contemporáneos en su contexto natural [19]. Además, muestra cinco pasos del proceso de la realización de los estudios de caso: (i) el diseño del estudio de caso que define los objetivos y se planifica el estudio de caso, (ii) la preparación de la recolección de datos que define los procedimientos y protocolos para la recopilación de datos, (iii) la recolección de evidencia, (iv) el análisis de los datos recogidos y (v) los informes.

Para establecer las métricas del estudio de caso para la evaluación de la guía de trabajo para el fortalecimiento de prácticas colaborativas durante la adopción de Scrum, se usó la metodología Goal Question Metric (GQM) propuesta por Basili [20]. Esta es una metodología orientada por objetivos. La cual sirve para desarrollar y mantener métricas. La medición debe ser realizada siempre y orientada a un objetivo. Se define el objetivo y este es refinado con preguntas y se definen métricas que ayudan a responder a estas preguntas. Se deben considerar preguntas que sean potencialmente medibles, ya que las preguntas ayudan a medir si se alcanza el objetivo.

1.5. ORGANIZACIÓN DEL DOCUMENTO

El presente documento está organizado por capítulos de la siguiente manera:

En el primer capítulo se presenta una introducción del trabajo de grado, donde se contextualiza el problema, se define la propuesta, los objetivos y la metodología.

En el segundo capítulo se presenta el marco teórico donde se muestran algunos conceptos sobre metodologías tradicionales y metodologías ágiles. También se describe los trabajos relacionados en torno a aspectos de adopción de acuerdo con el seguimiento a la metodología, aspectos de adopción asociados al tipo de producto, asociados a los equipos de trabajo y enfoques actuales para adopción de Scrum.

En el tercer capítulo se define el marco de trabajo Scrum, sus roles, artefactos y eventos, además de los beneficios que trae para que la organización realice su adopción.

En el cuarto capítulo se presenta el estudio de la literatura, el estudio de caso exploratorio y la entrevista a expertos para determinar los principales factores problema, retos, riesgos y situaciones que afectan la adopción de Scrum.

En el capítulo cinco se describe el diseño de la ruta de trabajo para la adopción de Scrum en pequeñas organizaciones de la industria del software.

En el capítulo sexto se presenta el estudio de caso que permite evaluar la

efectividad de la ruta de trabajo en una pequeña organización de la industria de software local.

Finalmente, en el capítulo siete, se presentan las conclusiones y los trabajos futuros.

CAPÍTULO 2

MARCO TEÓRICO Y ESTADO DEL ARTE

El objetivo de este capítulo es presentar los conceptos más relevantes para el desarrollo del presente trabajo de grado, con el fin de facilitar una mejor comprensión de los capítulos y secciones posteriores. Además, presentar un análisis de trabajos relacionados a la adopción de Scrum e ingeniería de la colaboración, mediante una revisión de la literatura a partir de bases de datos reconocidas. Con la revisión se busca describir en qué actividades de la metodología Scrum se han incorporado elementos colaborativos como patrones, Thinklets, entre otros.

2.1. MÉTODOS ÁGILES

Los métodos ágiles son un conjunto de técnicas para la gestión de proyectos y el desarrollo de software. Surgieron como un esfuerzo para mejorar debilidades percibidas y reales de la Ingeniería de Software convencional [9], pero también, se ha extendido a otro tipo de proyectos. Los métodos ágiles cumplen con una serie de principios y valores consignados en el manifiesto ágil [9]. Además, los métodos ágiles están atentos al entorno del proyecto y a las partes interesadas, ofrecen un entorno de trabajo más abierto y un marco mucho más flexible [1] [2].

2.1.1. Scrum

Scrum es una metodología ágil de gestión de proyectos específicamente referida a ingeniería de software [3]. Ya que, posee una adaptación flexible al cambio, se recomienda su uso en proyectos complejos con requisitos cambiantes o poco definidos, donde se necesitan entregas tempranas y de alta complejidad. Scrum se centra en cómo deben funcionar los miembros del equipo para que el sistema sea flexible y se adapte a unas condiciones constantemente cambiantes, por tanto, su funcionamiento está basado en sus roles, eventos, artefactos y reglas asociadas.

2.1.2. Adopción de Scrum

La adopción de Scrum brinda mayor importancia a las personas y sus relaciones, es decir, la adopción debe ser una decisión voluntaria y consciente de interiorizar no sólo el proceso y la metodología Scrum, sino también los valores y principios ágiles, de forma que el pensamiento del equipo cambie, e influir en el cambio en la cultura organizacional. En consecuencia, la adopción genera una visión compartida del proceso, los valores y principios ágiles, y la cultura organizacional.

2.1.3. Problemas de Adopción de Scrum

Scrum es una de las metodologías ágiles más usadas actualmente, sin embargo, la falta de comprensión ha derivado en una serie de problemas u obstáculos que impiden una adopción exitosa [4]. La entrega conjunta implica que los equipos de Scrum deben colaborar [21], por tanto, es importante abordar los problemas de colaboración durante la adopción. Kanane [15], indica que los desafíos que impiden a los miembros del equipo trabajar en equipo se pueden resumir en torno a la gestión de reuniones y la percepción de cómo debe ejecutarse el proceso. Sin embargo, refiere los siguientes problemas:

- Ausencia de la propiedad colectiva: el equipo no se interesa por el producto que está construyendo y no se siente responsable del éxito del producto
- Equipo no alineado: equipos sin la experiencia necesaria para el proyecto.
- Falta de comunicación efectiva: comunicación pobre entre los miembros del equipo.
- Falta de cooperación y comunicación constante con el cliente: poca o nula comunicación e interacción entre desarrolladores y partes interesadas.
- Falta de disciplina en las reuniones diarias: las reuniones duran más del tiempo definido, se llevan a cabo de forma inadecuada o no se realizan.
- Falta de participación constante del cliente: el cliente no está interesado en el proyecto, no está dispuesto a participar o solo siente interés por el resultado al final.
- Falta de comprensión de los valores ágiles: los miembros del equipo tienen diferente comprensión de los valores ágiles.
- Falta de comprensión del objetivo de las reuniones: los miembros del equipo no tienen la misma percepción del propósito de las reuniones de Scrum.
- Tamaño del equipo: tamaño inadecuado que dificulta la comunicación y colaboración.
- Resistencia al cambio: los miembros del equipo no pueden o quieren adoptar en totalidad el marco de trabajo.

2.2. COLABORACIÓN

La colaboración es trabajar en conjunto para lograr una tarea u objetivo que no se lograría de forma individual bajo las mismas restricciones de tiempo, esfuerzo y costos [11]. Un equipo colaborativo es un grupo de personas con habilidades complementarias que comparten tareas, recursos, responsabilidades y liderazgo, para lograr un entendimiento común que les permita alcanzar un objetivo común [13], el cual se logra si y sólo si todos los integrantes alcanzan su objetivo individual [22]. De acuerdo a Kofschoten [23], para que una actividad sea colaborativa, deben darse las siguientes condiciones: (1) Interacción entre un conjunto de personas que trabajan juntos o hacen un esfuerzo conjunto alrededor de la actividad, (2) La actividad está orientada hacia un objetivo o salida común para todos los participantes, y (3) Entendimiento compartido del problema y un estatus común del avance de la actividad y los participantes (conciencia o awareness).

2.2.1. Ingeniería de la colaboración - Collaboration Engineering (CE)

La ingeniería de la colaboración es un área dedicada al diseño y construcción de procesos colaborativos reutilizables con el fin de que las organizaciones puedan usarlos en sus equipos para ejecutar tareas recurrentes que requieren de la colaboración entre sus miembros, sin necesidad del conocimiento especializado en procesos colaborativos, es decir, sin el apoyo continuo de facilitadores profesionales [24] [25]. La actividad principal de CE es el diseño de prácticas de trabajo colaborativo. Por tanto, el diseño debe ser lo suficientemente simple y claro para ser transferible a los miembros del equipo. Los patrones de colaboración que crea la ejecución del diseño deben ser predecibles y deben poder reutilizarse para múltiples instancias de la tarea [11]. Debido a lo anterior, la CE se centra en tareas recurrentes porque el retorno de los recursos dedicados al esfuerzo aumenta cada vez que se ejecuta la práctica de trabajo, y porque si una tarea no se repite, no tiene sentido aprender un método de ejecución para ella [11].

2.2.1.1. Patrones de colaboración

Los patrones de colaboración caracterizan las formas en que las actividades grupales pueden llevar a un grupo hacia un objetivo [11]. Los investigadores de CE identificaron seis patrones de colaboración principales [17] (divergencia/generar, convergencia/reducir, organizar, evaluación, construir en consenso, y clarificar), los cuales permiten recopilar las tareas colaborativas más comunes realizadas dentro de los proyectos, y, además, clasificar las actividades grupales según los cambios de estado que producen.

2.2.1.2. Thinklets

Los ingenieros de colaboración usan Thinklets como bloques de construcción reutilizables para construir diseños lógicos para prácticas de trabajo colaborativo, por lo cual seleccionan entre los Thinklets, en parte, en función de las variaciones que producen [11]. Además, los Thinklets establecen un lenguaje de patrones de procesos de colaboración para prácticas de diseño y ejecución de actividades en forma colaborativa. Los patrones de diseño son descripciones de soluciones conocidas y reutilizables para problemas recurrentes. Por lo tanto, los Thinklets son actividades colaborativas reutilizables y transferibles, que dan lugar a variaciones específicas conocidas de los patrones generales de colaboración, así que, un Thinklet también puede dar lugar a varios patrones generales de colaboración simultáneamente.

2.2.1.3. Técnicas y métodos de colaboración

Algunos autores como Johnson & Johnson et Al [26], definen métodos de cooperación y colaboración, mientras otros como Slavin [27] definen las mismas prácticas como técnicas de colaboración. A continuación, se presenta una lista de algunos métodos o técnicas de colaboración, las cuales serán estudiadas como base para la formulación de la guía para el fortalecimiento de las prácticas

colaborativas en equipos Scrum:

- Learning Together and Alone (LTA) [28]
- Teams-Games-Tournaments (TGT) [27] [29]
- Group Investigation (GI) [29] [30],
- Constructive Controversy [31]
- Jigsaw [29] [32]
- Student Teams Achievement
- Divisions (STAD) [26] [29]
- Complex Instruction (CI) [29] [33]
- Team Accelerated
- Instruction (TAI) [27] [29]
- Cooperative Learning Structures [34] [35]
- Cooperative Integrated Reading & Composition (CIRC) [27] [29].

2.3. TRABAJOS RELACIONADOS

La ingeniería del software está intrínsecamente relacionada con la colaboración, los ingenieros y demás interesados trabajan juntos para desarrollar mejor software [17]. Esto es tan bien entendido, que incluso la mayoría de las herramientas que se utilizan en el ciclo de vida del producto soportan la colaboración, puesto que, permiten eliminar los problemas que conlleva la distancia y construir de forma conjunta artefactos o elementos [12] [36]. Sin embargo, la mayoría de las razones por la que fracasa un proyecto software son causadas por problemas de comunicación e interacción entre desarrolladores y partes interesadas [37]. En este apartado se analizan los trabajos relacionados en los métodos ágiles con foco en los aspectos colaborativos que facilitan adoptar Scrum como metodología de desarrollo.

2.3.1. Colaboración en métodos ágiles

El manifiesto ágil desarrollado por Beck et al [77], promueve los siguientes principios en los equipos de desarrollo de software: 1) Los individuos y su interacción, deben estar sobre los procesos y las herramientas. 2) Es mejor tener software funcionando, a tener una documentación exhaustiva, 3) La colaboración con el cliente es más importante que la negociación contractual, 4) La respuesta al cambio es más efectiva que sólo el seguimiento de un plan [24]. De los valores anteriores el 1 y el 3 se relacionan directamente al factor colaboración, entre los miembros del equipo (Principio 1) y a la colaboración continua con el cliente (Principio 3).

Se puede encontrar evidencia de la influencia positiva de los principios ágiles en diversas áreas de la ingeniería de software, por ejemplo, Noor, Rabiser y Grünbacher [16], adoptan principios ágiles en la planeación de líneas de producto, en especial el principio de la colaboración. Por tanto, se utiliza la ingeniería de la colaboración, específicamente los thinklets, para la creación de un enfoque colaborativo de planificación de línea de productos. López et al. [9] encontraron por medio de una revisión sistemática de la literatura que uno de los problemas en

la adopción de metodologías ágiles, es la falta de colaboración y comunicación con el cliente. Además, algunos artículos reportan el uso de Scrum con la finalidad de mejorar aspectos colaborativos, sin embargo, no mencionan qué aspectos deseaban mejorar, cuáles estrategias o actividades permitieron la mejora en la colaboración e incluso qué aspectos fueron mejorados, de tal forma que solo se limitan a indicar que usando Scrum se mejoró la colaboración, sin un indicador verificable [38].

Irum Inayat y Siti Salwah [39], proponen un marco de trabajo para estudiar la colaboración impulsada por los requisitos en equipos ágiles distribuidos y determinar el impacto de sus patrones de colaboración durante la iteración, definiendo la colaboración en términos de comunicación como el intercambio de información entre los miembros del equipo y la conciencia del conocimiento de los demás. Se hicieron dos estudios de caso, cuya información fue recolectada a través de cuestionarios, entrevistas y observación.

Marqués y Ochoa [40], tratan de mejorar aspectos como la comunicación, la coordinación y la motivación, a través de thinklets que son sugeridos en respuesta a un problema presentado en un equipo de desarrollo del curso de Proyecto software de la Universidad de Chile. En el artículo no se indica la metodología de desarrollo, sin embargo, por sus actividades se puede inferir que se trata de una metodología ágil. El estudio concluye que se pueden transferir habilidades de trabajo en equipo a futuros ingenieros de software de una manera razonable mediante el uso de thinklets. Mayr-Dorn y Laaber [41], proponen un Domain-Specific Language (DSL) para secuencias de comandos de estructuras de colaboración y su evolución, teniendo en cuenta los problemas que traen consigo la ingeniería de software distribuida, tales como seguir el rastro de a quién se va a involucrar y cuándo se involucra, lo que lleva a la falta de comunicación y alternativamente a la sobrecarga de comunicación, lo que a su vez conduce a errores y retrabajo. Los autores muestran los beneficios del DSL en el contexto de una reunión de planificación de iteración en un entorno de desarrollo ágil.

Si bien estos trabajos incluyen aspectos colaborativos en métodos ágiles e incluso establece que la colaboración es un problema recurrente durante la adopción, hasta donde se tiene conocimiento los enfoques carecen de información sobre cómo lograr la colaboración y medir el efecto de la misma en equipos ágiles, particularmente cuando se está adoptando el enfoque ágil.

2.3.2. Colaboración en Scrum

Scrum tiene gran popularidad al centrarse en la colaboración entre el equipo y las partes interesadas, y en considerar el cambio de los requisitos como parte natural del proceso. No obstante, Scrum es considerado un marco de gestión del desarrollo de software, simple pero difícil de adoptar, ya que, algunos interesados y el equipo no suelen comprender y aplicar los principios de Scrum [37] [6]. Slijivar y Gunasekaran [42], proponen hacer uso de la metodología ágil Scrum para facilitar la gestión del diseño de proyecto en empresas operadoras y de servicio,

en industrias petroleras y de gas entre otras, teniendo en cuenta que aspectos como la colaboración con el cliente y la colaboración del equipo pueden mejorar el retorno a la inversión para los clientes. Vibhu Sharma y Vikrant Kaulgud [43], generan un enfoque para la extracción del contexto de los procesos y las actividades ágiles en equipos geográficamente dispersos, para así aumentar la conciencia del equipo, este trabajo considera la conciencia de equipo como elemento clave de la colaboración; aunque existen herramientas como pizarras colaborativas o mensajería instantánea o herramientas de conferencias, los autores afirman que “Estas herramientas proporcionan soluciones a 'cómo colaborar', pero no responder a ¿cuándo debe ocurrir colaboración?”, por lo que proponen la creación de una zona MyContext, con etiquetas de mensajes (actualización, urgente y solicitud de cambio), que de contexto a la información relevante y se muestre a diferentes grupos de interés lo que aumenta la utilidad de la información.

Por otra parte, se encuentran trabajos que resaltan la importancia de la colaboración en Scrum y se formulan estrategias o actividades para mejorar su vinculación en el proceso de desarrollo. Por ejemplo, Judy y Krumins-Beens [44], presentan la idea de la propiedad colectiva del producto en donde el product owner establece una colaboración con el equipo de trabajo basada en el alto rendimiento, el respeto mutuo y la confianza profunda. Es decir, el product owner involucra al equipo de tal forma que este se interese por el producto que están construyendo y se sienta responsable del éxito del producto, de esta forma se establece una propiedad y responsabilidad compartida del producto.

Pese a que estos trabajos reconocen la importancia de la colaboración en Scrum, no brindan un paso a paso de la adopción de la metodología, también algunos de ellos se enfocan en colaboración para roles específicos y finalmente, están mayormente enfocados a aumentar la conciencia del equipo con respecto al proceso. En cambio, la guía propuesta en este trabajo contempla fortalecer la colaboración durante la adopción de Scrum en tres aspectos: comunicación, interacción y conciencia, sin enfocarse en un rol de Scrum en específico, sino que por el contrario se desea brindar un paso a paso que sirva como herramienta para mejorar y/o fortalecer los aspectos colaborativos en todo el equipo de trabajo.

2.3.3. Problemas y estrategias relacionadas con la colaboración durante adopción de Scrum

Scrum establece que el equipo no debe estar compuesto por más de nueve personas que se encuentran en un espacio físico compartido, pero también puede haber varios equipos Scrum distribuidos geográficamente. Debido a lo anterior, Vlietland et. Al. [45] proponen un marco de gobierno para el trabajo entre equipos Scrum codependientes en proyectos de desarrollo en el sector financiero, los cuales pueden presentar dificultades de colaboración entre sí. Inicialmente se identificó el conjunto de acciones de intervención con el objetivo de mitigar los problemas de colaboración entre los equipos de Scrum codependientes. Después, se validó la eficacia de estas acciones de intervención. Además, se triangularon los hallazgos en tres grupos focales. Finalmente se empaquetaron las acciones de

intervención en el marco de gobierno. Lo anterior logró la reducción del tiempo de entrega de 29 días a solo 10 días.

Cabe mencionar que, hasta el momento de la realización de este trabajo este fue el único trabajo relacionado encontrado que asocia la adopción de Scrum y la Colaboración. Sin embargo, a nivel macro se encontraron trabajos relacionados que vinculan métodos ágiles, Scrum y Colaboración.

2.3.4. Estrategias para la adopción de Scrum

Araujo et. al. [6] propone una guía de trabajo para orientar la adopción de Scrum en pequeñas organizaciones de la industria del software con poco o inexistente conocimiento acerca del marco de trabajo Scrum. En la ruta se establecen una serie de pautas a seguir, con el fin de lograr una adopción adecuada de Scrum en equipos de desarrollo. Los hallazgos de Araujo et. al. [6] han permitido establecer los riesgos y problemas de adoptar Scrum. Por otra parte, Pardo et al [46] proponen una guía ágil basada en Scrum donde definen actividades, tareas, roles y criterios para apoyar la gestión global de proyectos de desarrollo de software con múltiples modelos de referencia, teniendo en cuenta aspectos como: la falta de comunicación, falta de coordinación, diversidad cultural, entre otros aspectos que dificultan la gestión de estos tipos de proyectos. Sin embargo, en las guías de Araujo et al y Pardo et al no se involucra elementos de la ingeniería de la colaboración.

En este trabajo de grado se sigue la línea de abordar los problemas y riesgos durante la adopción de Scrum, con foco en los aspectos colaborativos. Por tanto, este trabajo es una continuación de la guía de adopción de Araujo et. al, profundizando en los aspectos colaborativos en Scrum. La guía propuesta pretende establecer pautas concretas para la adopción de Scrum incluyendo elementos propios de la ingeniería de la colaboración. Lo anterior tiene el fin de fortalecer las prácticas colaborativas en el equipo, de tal forma que el riesgo asociado al factor colaboración durante la adopción de Scrum disminuya.

2.3.5. Estudios empíricos sobre el incremento de la colaboración en Scrum

De Vreede [47], a través de dos estudios de caso presenta el enfoque de ingeniería de colaboración para diseñar deliberadamente procesos productivos de colaboración que ofrezcan un valor repetible. Cada estudio de caso realizó el diseño de un único proceso, en este diseño se involucran cinco patrones de colaboración: generate, converge, organize, evaluate y build consensus. Por otra parte, en uno de los casos se realizó el diseño del proceso de planificación del product backlog en una organización que se encontraba adoptando Scrum y que reconocía el valor de la ingeniería de colaboración para la introducción de un desarrollo ágil. Finalmente, el diseño del proceso de planificación del product backlog se abordó por fases, en cada una de las cuales se aplicaron thinklets para facilitar o apoyar tareas. Aileen Vicente, Tiffany Tan y Avin YU [48], proponen una técnica para la integración interdisciplinar en cursos educativos de ingeniería de

software, hacen un enfoque metodológico con el fin de mejorar el enfoque de colaboración interdisciplinar para el desarrollo de proyectos, utilizando como caso de estudio un curso de Ingeniería de Software para estudiantes de Ingeniería de Ciencias de la Computación simulando las condiciones reales de los equipos de desarrollo. Anna Sandberg e Ivica Crnkovic [49], hacen un estudio de caso sobre la colaboración entre la industria y la academia, teniendo en cuenta los desafíos de colaboración que se pueden presentar, utilizan Scrum como base para la colaboración y reportan sus resultados de éxito y desafíos en la colaboración. Se reporta la colaboración en la investigación, que se ha adaptado Scrum durante un período de seis años.

Estos trabajos comprueban que asociar elementos de la ingeniería de la colaboración a Scrum proporciona buenos resultados. Aunque estos trabajos no brindan un camino repetible para adoptar los aspectos colaborativos y tampoco miden el impacto en el trabajo en equipo, brindan un conocimiento empírico valioso para construir una guía de valor para los equipos extraída desde la misma experiencia. La guía de trabajo propuesta en este trabajo de grado, se enfoca en incluir elementos de la ingeniería de la colaboración durante la adopción de Scrum, pero también pretende disminuir el riesgo asociado al factor colaboración y generar una herramienta con valor repetible que sirva como guía durante el proceso de adopción de esta metodología.

2.3.6. Tabla comparativa de los diferentes trabajos relacionados

A continuación, se presenta una tabla comparativa de los trabajos relacionados encontrados en el desarrollo de este trabajo. La tabla indica los trabajos relacionados, el tipo de colaboración que se presenta y en la última columna, se realiza un contraste entre la propuesta y los trabajos relacionados.

Trabajos	Colaboración	Aportes / diferencias
[37], [16], [42]		La guía propuesta aportará un lineamiento que permita fortalecer los aspectos colaborativos no solo del Scrum Master con el cliente [37], o los principios relevantes para la Línea de Producto [16], o el equipo con el cliente [42], sino que tratará de fortalecer las prácticas colaborativas en todo el proceso Scrum.

<p>[47] [44] [43]</p>	<p>[40] [48]</p>	<p>Entre los miembros del equipo</p>	<p>Se presentan prácticas colaborativas cuyos fines son lograr el rendimiento repetible [47], dar contexto a la conciencia colectiva [43], o propiedad colectiva del producto [44], el uso de Thinklets para mejorar la comunicación y colaboración [40] o mitigar el impacto de equipos interdisciplinarios [48]. Sin embargo, estas prácticas se ven limitadas a la interacción entre los miembros de un mismo equipo, nuestra propuesta propone prácticas que contemplan la interacción entre los miembros del mismo equipo y la interacción entre los equipos y sus clientes, sin diferenciar si son equipos de desarrollo profesional o si son equipos de estudiantes.</p>
<p>[39], [36], [38], [41]</p>	<p>Entre equipos de desarrollo</p>	<p>Aunque nuestro trabajo no contempla la colaboración entre equipos de desarrollo por el alcance del mismo, se tendrán en cuenta aspectos relevantes como infraestructura o formas de comunicación que puedan ser utilizados en la colaboración con el cliente o entre un mismo equipo. Además, se indicarán sugerencias para que las prácticas propuestas puedan ser adecuadas a equipos de equipos.</p>	
<p>[12], [36], [50]</p>	<p>Otros</p>	<p>Estos artículos se enfocan en la colaboración en general y brindan una vista de las herramientas colaborativas. Aunque estos trabajos no profundizan en la colaboración en Scrum, las herramientas presentadas fueron consideradas para nuestra propuesta, ya que, se pretende generar una guía para el fortalecimiento de las prácticas colaborativas durante la adopción de Scrum.</p>	

Tabla 1: Comparativa de los trabajos relacionados

CAPÍTULO 3

Debido a, que Scrum es la base sobre la cual se construye la guía de adopción, en este capítulo se presenta la estructura del marco de trabajo Scrum de forma más detallada, con el fin de aumentar la comprensión de este marco de trabajo.

SCRUM

3.1. DEFINICIÓN

Scrum es un marco de trabajo para la gestión de proyectos basado en la construcción evolutiva, se recomienda para proyectos complejos con requisitos cambiantes o poco definidos, donde se necesitan entregas tempranas y la competitividad es alta [51], Scrum está basado en la teoría del empirismo en donde se asegura que el conocimiento procede de la experiencia, de tomar decisiones basándose en el conocimiento y en el pensamiento Lean ¹para enfocarse en lo esencial reduciendo el desperdicio. Además, utiliza un enfoque iterativo para optimizar la predictibilidad y tener más control del riesgo [52]. En la figura 4 se describe la dinámica de Scrum.

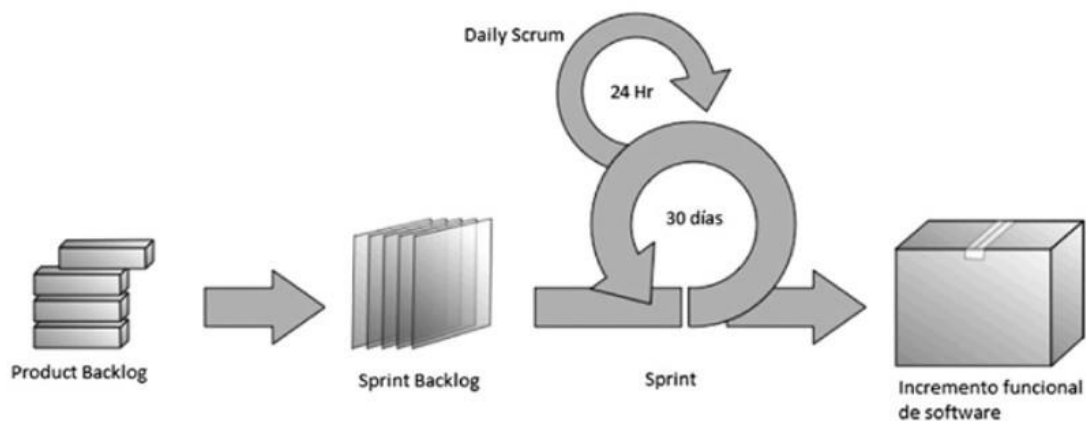


Figura 2: Ciclo de Scrum [53].

¹ El pensamiento "lean" se centra en la identificación y eliminación sistemática de desperdicios en los procesos de transformación y/o entrega de un servicio.

3.2. MARCO DE TRABAJO SCRUM

3.2.1. EQUIPO SCRUM

El equipo Scrum es unidad fundamental de Scrum, consiste en un pequeño equipo de personas que debe ser auto organizado y multifuncional. Los miembros del equipo Scrum tiene la responsabilidad de elegir qué hacer y cómo hacerlo, por lo tanto, no deben ser dirigidos por personas externas al equipo. Además, el modelo del equipo está diseñado para maximizar la creatividad, flexibilidad, productividad y la motivación, el equipo Scrum consiste en un Product Owner, el Scrum master y el Equipo de desarrollo [52].

- **Product Owner:** (Dueño del producto) Es el encargado de maximizar el valor del producto, es una única persona y es el único responsable de la gestión del Product Backlog, toda la organización debe respetar sus decisiones que se ven reflejadas en el product backlog.
- **Scrum Master:** Es un líder al servicio del Product Owner, el Equipo de desarrollo y de la organización, es el encargado de hacer que Scrum sea entendido y adoptado por tanto es quien realiza entrenamientos de Scrum.
- **Development Team:** (*Equipo de desarrollo*) Son los encargados de entregar un incremento del producto [52], el tamaño del Equipo de desarrollo está entre tres y nueve personas, son auto organizados y multifuncionales, deben tener las habilidades necesarias para culminar el trabajo.

3.2.2. Eventos de scrum

Los eventos de Scrum están definidos de forma clara y poseen una duración máxima establecida [52].

- **Sprint Planning:** (*Reunión de Planeación del sprint*) en esta reunión se fija el objetivo del sprint, el plan de trabajo y cómo se va a entregar [53], la reunión consta de dos momentos: el ¿Qué? y el ¿Cómo?, en el primer momento el product Owner comunica el objetivo del sprint y se asegura que todos lo entiendan, el segundo momento es el cómo se va a realizar el trabajo para cumplir con el objetivo, aquí los desarrolladores definen las tareas a realizar. Esta reunión tiene una duración de 8 horas para un sprint de un mes [52].
- **Daily Scrum:** (*Scrum diario*) Reuniones diarias para seguir el progreso del equipo, cada integrante del Equipo de desarrollo responde a tres preguntas ¿Qué hice ayer? ¿Qué voy a hacer hoy? ¿Qué inconvenientes o problemas

he tenido?, no debe tener una duración mayor a 15 minutos y preferiblemente de pie. Se debe hacer a la misma hora y en el mismo lugar para reducir la complejidad [53].

- **Sprint Review:** (*Revisión del sprint*) el último día del sprint se presenta el incremento del producto al cliente y product Owner, es una reunión informal no de seguimiento con el objetivo de inspeccionar, y obtener retroalimentación de información y fortalecer la colaboración, en esta reunión se inspecciona el product backlog y se actualiza de ser necesario [51]. Tiene estipulado una duración de 4 horas para un sprint de un mes [52].
- **Sprint Retrospective:** (*Retrospectiva del sprint*) después de la revisión del sprint y antes de la planeación del sprint el equipo se inspecciona a sí mismo y crea un plan de mejora. Esta reunión debe durar 3 horas para un sprint de un mes, en esta reunión el equipo indica que fue bien y que fue mal durante el sprint pasado, reflexiona y crea un plan de mejora para el próximo sprint [52].

3.2.3. Artefactos de scrum

Los artefactos de Scrum proporcionan transparencia y presentan oportunidad de inspección y adaptación para proporcionar valor. Tienen el propósito de asegurar que todos tengan la misma información [52].

- **Product backlog:** (lista de producto) es una lista que define todo lo necesario en el producto final, esta ordenada por valor, riesgo, prioridad y necesidad, basándose en los conocimientos de ese momento, tiene la característica que se mantiene viva (evoluciona) durante todo el proyecto. Constantemente se le debe hacer refinamiento, mantenerla lo más actualizada posible y será responsabilidad del product Owner encargarse de eso. Los elementos del product backlog están ordenados, a mayor prioridad mayor detalle [52].
- **Sprint backlog:** (Lista de pendientes del sprint) Es el punto de partida para cada sprint, lista de puntos del Product Backlog seleccionados para llevarse a cabo en el próximo sprint [51]. Esta lista es una predicción realizada por el Equipo de desarrollo acerca de que funcionalidades formaran parte del próximo incremento y del trabajo necesario para cumplir el objetivo del sprint. Los elementos de esta lista están a nivel de detalle que se puede calcular el progreso diario, esta lista puede ser actualizada durante el sprint

y pertenece únicamente al Equipo de desarrollo [52].

- **Incremento del producto:** Es la suma de los elementos del producto completados durante un sprint y que cumplen con la definición de terminado [52], si hay elementos no terminados, deben devolverse al product backlog con una alta prioridad para que sean tomados en el siguiente sprint [53].
- **Seguimiento al progreso:** Es importante para la transparencia realizar un seguimiento del progreso, que en cualquier momento se pueda sumar el trabajo restante, medir la productividad o determinar la velocidad del equipo. Existen varias formas de hacer el seguimiento, estas son **el Release Burndown Chart** que es un gráfico donde muestra la curva de avance de los sprints para completar el próximo release, el **task board** (tablero de tareas) que comunica el estado de las mismas durante la ejecución [54] y el Sprint **Burndown Char** que una gráfica para realizar el seguimiento del sprint donde muestra las horas pendientes para terminar el sprint [55].

3.2.4. Definición de “HECHO”

Scrum es un enfoque ágil para el desarrollo de productos y servicios innovadores. Se inicia creando un Product backlog, siempre se trabaja primero el ítem más importante. Cuando se da por terminado un elemento del Product backlog o un incremento todos deben conocer cuando se da por “terminado”, deben tener la misma definición de hecho, con el fin de tener una aproximación de cuántos elementos del Product backlog pueden seleccionar para el siguiente sprint. A medida que el equipo madure se recomienda ampliar la definición de “terminado” para una mayor calidad [52].

3.3. BENEFICIOS DE SCRUM

Las ventajas que se encontraron respecto a la adopción de Scrum como marco de trabajo son las siguientes [55]:

- Entrega constante de resultados:
 - Se genera una gestión de lo esperado por el cliente y se basa en resultados tangibles.
 - El cliente establece sus expectativas indicando el valor que le aporta cada requisito del proyecto y cuando espera que esté completado; y comprueba de manera regular si se van cumpliendo sus expectativas, da feedback, ya desde el inicio del proyecto puede tomar decisiones informadas a partir de resultados objetivos y dirige

- estos resultados del proyecto, iteración a iteración, hacia su meta.
- El cliente puede empezar a utilizar los resultados más importantes del proyecto antes de que esté finalizado por completo.
 - El cliente puede redirigir el proyecto en función de sus nuevas prioridades, de los cambios en el mercado, de los requisitos completados que le permiten entender mejor el producto, de la velocidad real de desarrollo, etc.
 - Cuando el beneficio pendiente de obtener es menor que el coste de desarrollo, el cliente puede finalizar el proyecto.
 - Mitigación sistemática de los riesgos del proyecto. Desde la primera iteración el equipo tiene que gestionar los problemas que pueden aparecer en una entrega del proyecto. Al hacer patentes estos riesgos, es posible iniciar su mitigación de manera anticipada. La cantidad de riesgo a que se enfrenta el equipo está limitada a los requisitos que se puede desarrollar en una iteración. La complejidad y riesgos del proyecto se dividen de manera natural en iteraciones
- EL equipo va mejorando y simplificando su forma de trabajar constantemente, ayudando a mejorar la productividad y calidad.
 - El equipo y el cliente pueden estar en sintonía al poder medir los resultados y el esfuerzo en forma de objetivos y requisitos entregados al negocio.
 - El producto se enriquece con las contribuciones de todos.
 - El equipo está más motivado al poder utilizar su creatividad resolviendo problemas y al poder decidir organizar su trabajo.

CAPÍTULO 4

ESTUDIO SOBRE LOS ASPECTOS COLABORATIVOS TENIDOS EN CUENTA POR EQUIPOS DE DESARROLLO DURANTE LA ADOPCIÓN DE SCRUM

En muchas organizaciones se han encontrado inconvenientes al momento de adoptar Scrum como marco de trabajo en sus proyectos, estos se ven relacionados con factores asociados al comportamiento humano como la cultura, organizacional, la comunicación, la interacción, el compañerismo, la comunicación. Etc. En el presente capítulo, se consolidan los principales problemas y situaciones que de acuerdo con la perspectiva empírica de líderes de equipos de trabajo se presentan durante la adopción de Scrum. Estos problemas permitirán identificar los aspectos colaborativos que el equipo reconoce y cuáles de ellos representan falencias tanto de los factores colaborativos como de la metodología.

4.1. ENTREVISTA SEMIESTRUCTURADA

4.1.1. Diseño

4.1.1.1. Objetivo

Identificar los aspectos colaborativos tenidos en cuenta por equipos de desarrollo que utilizan metodologías ágiles, específicamente Scrum en su fase de adopción.

4.1.1.2. Personas

Las entrevistas están enfocadas a ingenieros de sistemas o afines, pertenecientes a equipos de desarrollo en el contexto de la metodología ágil Scrum en su fase de adopción.

Los participantes fueron tres líderes de equipos de desarrollo que voluntariamente accedieron a participar de la entrevista para profundizar en los aspectos colaborativos de Scrum durante su fase de adopción.

4.1.1.3. Pregunta de investigación

¿Cuáles son los aspectos colaborativos que las empresas de software identifican durante la adopción de Scrum y qué tanto cubren las características relacionadas con la colaboración?

4.1.2. Planificación

4.1.2.1. Métodos de recolección de datos

Los métodos de recolección de datos en este estudio fueron entrevista semiestructuradas. Las entrevistas semiestructuradas son un enfoque intermedio entre encuestas estandarizadas, en su mayoría cerradas de individuos y las sesiones de libre participación, con grupos, son las entrevistas semi-estructuradas [56]. A las entrevistas semi-estructuradas Lewis Dexter, lo llamó entrevista de élites, aunque esa etiqueta puede sugerir erróneamente hablar solo con encuestados de alto estatus [56]. Robert Merton la llamó la entrevista enfocada, aunque esa frase ahora corre el riesgo de confusión con los grupos focales. Por otro lado, los antropólogos culturales hablan más estrechamente de la entrevista etnográfica y los sociólogos a veces se refieren a entrevistas profundas [56].

La entrevista semi-estructurada es conducida en forma conversacional con un encuestado a la vez, empleando una combinación de preguntas cerradas y abiertas, a menudo acompañadas por un seguimiento hacia atrás de las preguntas (por qué y cómo). El diálogo puede hacer zigzag en torno a los temas de la agenda, en lugar de adherirse rígidamente a las preguntas literales como en una encuesta estandarizada, y puede ahondar en cuestiones totalmente imprevistas. Las entrevistas semi-estructuradas son relajadas, atractivas y al ser cara a cara, pueden ser más largas que las encuestas telefónicas, aunque duran tanto como los grupos focales. Una hora se considera una duración máxima razonable de una entrevista estructurada para minimizar la fatiga tanto para el entrevistador como para el entrevistado.

Los pasos prácticos para diseñar y realizar entrevistas semiestructuradas son: seleccionar y reclutar a los encuestados, redactar las preguntas y la guía de entrevistas, aplicar las técnicas recomendadas para este tipo de entrevistas y analizar la información recopilada.

4.1.2.2. Protocolo

Inicialmente se definió el protocolo de la entrevista semiestructurada con el fin de facilitar la réplica y garantizar que todas las entrevistas se llevaran a cabo de la misma forma. A continuación, se encuentra el protocolo utilizado durante este caso de estudio.

Inicialmente se contactó vía correo electrónico a los participantes, explicando el objetivo del estudio, la duración de la entrevista y finalmente solicitando la confirmación de su participación enviando la fecha y hora para realizarla.

Una vez se realizaba el encuentro con el participante se realizaron las siguientes actividades:

- Saludo en voz alta.

- Presentación del entrevistador o entrevistadores.
- Recopilación de información básica (nombre-cargo-ocupación)
- Lectura en voz alta del propósito de la entrevista (Entrevista para recolectar información de utilidad para el desarrollo de la tesis de pregrado “Una guía de trabajo para el fortalecimiento de prácticas colaborativas durante la adopción de Scrum”).
- Explicación del contenido de la entrevista.
- Solicitar consentimiento para grabar la sesión
- Lectura del concepto de Colaboración.
- Compromiso del entrevistado a decir la verdad.
- Preguntar: ¿cuáles son sus labores en la empresa?, ¿cuánto tiempo lleva laborando ahí?, y algunas preguntas libres. Con el fin de establecer una buena relación con el entrevistado, y que este se sienta en confianza (bajar un poco en nivel de ansiedad y el estado defensivo en el que puede estar una persona, previo a una entrevista).
- Presentación de las preguntas.
- Intervenciones permitidas:
 - Facilitador: parafrasear para continuar con la entrevista (ok, si, ya, hmm, uh-huh, ajam, vale, etc).
 - Invitación: preguntas abiertas (cuéntame más al respecto, qué ocurrió, qué hicieron, cómo lo superaron)
 - Invitación complementada: preguntas sobre un hecho específico (Tú mencionaste que [evento, acción, objeto, artefacto] dime más sobre eso, me dijiste que [acción] cuéntame qué pasó después).
 - Invitación a contar algo referente al tema que considere de importancia y que no se encuentre contemplado en la entrevista.
- Agradecimientos en voz alta.

La información recolectada durante las entrevistas fue almacenada en audio. Posteriormente, cada entrevista fue transcrita a un documento digital para luego ser ingresada al software WEFT QDA [57].

4.1.3. Preguntas

La entrevista fue realizada a tres líderes de desarrollo de equipos adoptantes de Scrum y consistió en 22 preguntas, divididas en tres categorías: interacción (6 preguntas), comunicación (5 preguntas) y conciencia (8 preguntas).

- ¿Cuánto tiempo lleva utilizando la metodología ágil Scrum?
- ¿Cuál o cuáles fueron los principales problemas que tuvo durante la adopción de Scrum?
- ¿Qué problemas considera usted que persisten desde la adopción hasta el momento?

Preguntas sobre interacción:

- ¿Podría describir la forma en que desarrolla las actividades diarias del equipo?
- ¿Podría describir la forma en que desarrolla las reuniones?
- ¿Qué beneficios encuentra Usted en la forma en que las actividades se vienen desarrollando?
- ¿Podría contarnos acerca de los problemas relacionados con la interacción entre los participantes durante el desarrollo de las actividades? ¿Cuáles cree que son las causas?
- ¿Podría detallar el procedimiento a seguir al encontrar y reportar un problema?
- En una escala de 1-5 siendo 1 muy poco y 5 mucho ¿Cómo considera usted que es la interacción entre los participantes del equipo?

Preguntas sobre comunicación:

- ¿Cuáles canales y medios digitales utilizan para la comunicación? ¿Cuáles son los más indispensables?
- ¿Qué beneficios encuentra Usted en la forma en que las comunicaciones de su equipo se vienen dando? ¿cuáles son sus fortalezas?
- ¿Cuál o cuáles considera que son los problemas más frecuentes durante la comunicación entre los participantes?
- ¿Cuáles son los problemas de comunicación más comunes durante las reuniones? ¿Cuáles cree que son las causas?
- En una escala de 1-5 siendo 1 muy poco y 5 mucho ¿Cómo considera que es

la comunicación del equipo? ¿Por qué?

Preguntas sobre conciencia:

- ¿Cómo logra el equipo un entendimiento compartido y permanente del problema, el producto y avance del proyecto para todo el equipo?
- ¿Cómo sabe usted en este momento cuál es el estado de avance del proyecto y qué asunto está resolviendo cada miembro del equipo actualmente?
- ¿Cuáles son sus responsabilidades y las de sus compañeros en los proyectos de software?
- ¿Hasta qué punto se siente usted responsable del éxito o fracaso del producto? ¿Por qué?
- ¿Considera que cada miembro del equipo es responsable del éxito o fracaso del producto? ¿Por qué?
- En una escala de 1-5 siendo 1 muy poco y 5 mucho ¿Cómo considera su conocimiento permanente sobre el estado de avance del producto y las tareas actuales que los miembros del equipo están abordando?
- ¿Podría describir problemas relacionados con el desconocimiento de aspectos relacionados con el producto, el proyecto o el trabajo de los integrantes del equipo?
- ¿Podría describir algunas estrategias de solución a los problemas identificados en su equipo de trabajo durante esta entrevista?

4.1.4. Resultados

La herramienta WEFT QDA clasificó fragmentos de texto en 4 categorías:

- **Interacción:** Se abordan y analizan los aspectos relacionados con interacción en el equipo de trabajo, ya sea con el cliente o entre integrantes del mismo equipo desde la perspectiva Scrum.
- **Comunicación:** Se analizan las herramientas, técnicas y/o canales de comunicación que utiliza el equipo de trabajo para tener una comunicación constante entre si desde la perspectiva Scrum.
- **Conciencia:** Se abordan y analizan los aspectos relacionados con la conciencia que se tiene del proyecto, los roles y/o responsabilidades que se tienen dentro de este, desde la perspectiva Scrum.

- Scrum: Se aborda lo relacionado con la metodología Scrum que no pudo entrar en ninguna de las categorías anteriores.

Las tres primeras categorías representan la Colaboración, ya que, como se mencionó anteriormente son necesarias para que una actividad sea colaborativa.

Esta clasificación presento los siguientes resultados:

Tabla 2: Fragmentos de texto por categoría

	Número de fragmentos de texto
Interacción	15
Comunicación	32
Conciencia	53
Scrum	10
Total	110

Interacción (15 extractos de texto)

Se pudo observar que los participantes confundían la interacción con la comunicación en las reuniones:

“... sobre todo con la parte del cliente como tal, esa parte ha sido súper difícil porque es una persona que está en EU entonces digamos que la comunicación no es tan fluida...”

“... se hace el feedback, y con ellos al final del día... estamos revisando lo que se hizo, si se tuvo algún problema, algún retraso, si para el día siguiente se puede hacer algo o qué hay pendiente para hacer al otro día, entonces en nuestro caso ha funcionado así...”

“... la reunión diaria, nosotros siempre lo hacemos a las 8 am, entonces el equipo empieza por iniciativa propia, en otros lados se planifica las personas que van a hacer el daily meeting y acá lo que hacemos es que sea por iniciativa propia entonces cualquiera hace la llamada y en máximo 2 minutos expone lo que es los avances, los problemas y lo que va a trabajar...”

“... yo les he dicho reunión diaria corta, máximo 15 minutos, ¡de pie!, para que sea corta, porque si se sientan en un sillón pues ya no es corta, entonces cada quien debe decir ¿qué hice desde la última reunión? ¿Qué pienso hacer en este día de trabajo? y ¿qué problemas posiblemente tenga?, entonces uno por uno, les cuesta muchísimo eso a ellos.”

“... hacemos una socialización entre todos, y decidimos qué desarrollos se van a hacer, y por ejemplo la parte del análisis lo hacemos entre todos... entonces cuando hay una nueva funcionalidad que requiere diseño nos reunimos entre todos vemos que es lo que hay que hacer, cada uno se pone en su búsqueda de qué opciones hay para hacer y empezamos a

dar ideas de cómo empezar a estructurar una funcionalidad, es muy colaborativo aquí la verdad, si tratamos de pedir la idea de todos.”

Un equipo en particular afirma que los problemas de interacción se veían afectados directamente por el tipo de contratación laboral:

“... allá todos trabajan por prestación de servicios OPS, entonces no les podemos decir hagamos la reunión diaria... y como les digo por ser OPS no hay forma de decirles tienen que venir a las 8 o a las 9 y ellos ya lo han manifestado, alguna vez el director les dijo necesito que me cumplan horario y dijeron a bueno entonces peguen como si fuéramos de planta...”

Por lo tanto, cabe resaltar que varios de los entrevistados relacionan la Interacción directamente con la comunicación al interactuar en las reuniones. Por otra parte, cuando se considera la interacción como la colaboración de un conjunto de personas que trabajan juntas para alcanzar un objetivo, la interacción se hace de forma implícita en las reuniones.

Comunicación (32 extractos de texto)

Los equipos utilizan herramientas como video conferencias, correos, entre otros como medio de comunicación entre sus integrantes. La forma de comunicación más común es la cara a cara y el correo electrónico.

Los problemas más comunes encontrados por los equipos están los relacionados con la información que no queda plasmada durante la comunicación directa:

“... que no quedan registro de que "tú me dijiste que tenía que ser así" o yo le digo una cosa o como no estuvo atenta puede entender otra cosa, son detallitos de profundidad, ahí tenemos ese problemita al no tener una herramienta que nos ayude a gestionar eso, pero sería más tiempo de desarrollo para nosotros, más carga.”

“... podemos concluir que alguien tiene que especificar algo y se nos olvida, o que el implemento una cosa y no era lo que yo entendía entonces eso pasa a veces, eso creo que es lo más crítico.”

“... cuando son comunicaciones directas a veces no queda ese registro o las conclusiones de la comunicación... si alguien te dice te faltó hacer esto o me di cuenta que faltó reportar esto, no hay evidencia de eso... entonces eso se pierde, queda en la conversación.”

“... algo de registrar la comunicación o las conclusiones de reuniones, a veces son muy rápidas, que quede como el registro de si tu dijiste esto o no acordamos esto...”

Otro problema muy común encontrado durante la comunicación es el factor personas:

“...nosotros hacemos un puño de 5, que es de 1 a 5 entonces la gente dice yo le doy un 3 yo le doy un 2, pero eso también es algo que hay que mejorar, y es que también se sesga la evaluación, porque con mi compañero que tenemos más experiencia yo digo 4, el dice 3 y los compañeros nos ven a nosotros y dicen 4 y 3 porque ya están sesgados...”

“... nosotros acá contábamos con la parte de diseño al principio, pero resulta que esa parte de diseño no nos funcionó muy bien porque resulta que el diseñador gráfico tal vez no tenía mucha experiencia... no satisfacía las necesidades del cliente, ni de nosotros mismos.”

“... por ejemplo, la forma de ser de las personas incluso, eso es inherente a cualquier compañía, entonces hay personas que tienen una forma de ser más fuerte y no saben decir las cosas y otros que son más volátiles entonces si se enfrentan esas dos personas pues hay conflicto...”

“... generalmente la forma de ser de las personas, creo que eso es, siempre pasa; entonces también es conocerse, es muy importante, el tema de cuando hay mucho tiempo es que uno ya conoce a las personas y sabe cómo llevarlas...”

“... digamos que no manifiesten en detalle lo que están haciendo, por ejemplo el compañero tenía su historia asignada y la planificó mal, que se demoraba una semana y terminó demorándose mucho más...”

Entrevistador: ¿porque crees que se da eso?

Respuesta: Puede surgir inclusive desde los sesgos cognitivos que tengas con base en la experiencia, entonces con base en la experiencia que él tenía estimó y ya cuando se enfrentó al problema se dio cuenta que era más complejo.”

“... a veces es complejo porque no todos tenemos como esa visión y hay muchas personas con distintos tipos de sensaciones, pensamientos, todos somos diferentes, entonces a unos les molesta y a otros no.”

“... hay uno que es bien difícil trabajar con él, ósea él es un rancho aparte, es su forma de ser, su estilo, entonces él dice: “yo mantengo la base de datos y los demás que hagan la parte web, a mi no me importa nada”. Él no participa en las reuniones, él no colabora, él nada, o sea se pierde un desarrollador ahí...”

“...difícil comunicarse con sujeto X, de pronto también eso se dé en otra organización, cuando hay alguien que tenga una forma de pensar diferente...”

Esta categoría permitió encontrar el factor persona teniendo en cuenta forma de ser, pensar, actuar, entre otros, como problema de comunicación en todos los

equipos y como este factor puede afectar los procesos.

Conciencia (53 extractos de texto)

Respecto a la conciencia todos los equipos utilizan herramientas como Trello, Bitbucket, Jira, Taiga, entre otros. Para tener un conocimiento compartido de la información del proyecto.

Algunos equipos reconocen el conocimiento técnico como factor de crecimiento y/o estancamiento:

“...de hecho cuando hicimos la capacitación de Scrum él no fue, imagínese les dan un curso gratis, con una empresa especializada y no va.”

“...como ya hemos venido cumpliendo con los proyectos... nos dieron una certificación de Oracle y no la pagan totalmente...”

“... en la parte técnica es importante saber cómo están avanzando, y si tienen problemas lo que se hace es enviar un apoyo de alguien que tenga más conocimiento.”

Los equipos son conscientes de sus roles y el alcance que estos conllevan:

“... dependiendo del rol como tal, los desarrolladores ya saben que tienen asignadas unas tareas...”

“... desarrolladores serian 6 y una persona de pruebas, y yo estoy volando entre desarrollo y análisis, es decir, tal persona puede hacer eso. Entrevistador: ¿Entonces las responsabilidades dependen de los roles?”

Respuesta: si, si, totalmente.”

Además, algunas personas reconocen la importancia de cada uno de sus compañeros para el éxito o fracaso de sus actividades y/o del producto:

“... cuando uno trabaja colaborativamente hay responsabilidades individuales y si un elemento falla perjudica todo el proyecto.”

“... yo me siento muy responsable de la calidad de los productos que nos hemos comprometido a sacar, pero yo sé que no depende de mí...”

“... es vital que cada uno sienta que el producto es de ellos y que todos tengan la 10.”

Es importante denotar que algunos equipos consideran el modelo de negocio como un conocimiento adicional al proyecto:

“... estamos con una capacitación entonces va a ver una persona encargada que conoce cómo hacer una liquidación, que no es ingeniero de sistemas, pero es de nuestro modelo de negocios, entonces nosotros solicitamos esa capacitación, pero eso es debido a, que el equipo manifestó que tenía dificultades en entender el modelo de negocio.”

“... estamos desarrollando un producto de gestión documental, y necesitamos un cliente que nos de las reglas del negocio, nosotros no conocemos las reglas del negocio, no sabemos cómo funciona una tabla de retención documental, no sabemos qué elementos debe tener un radicador, no sabemos hasta dónde llega la norma, entonces eso no depende de nosotros; cuando uno trabaja como ingeniero de sistemas no solamente es el software que se va a hacer sino que hay un contexto en un sistema como tal, donde hay personas, actividades, procesos, sobre todo procesos, que hay que empaparse en corto tiempo...”

A pesar de que tienen un alto grado de conciencia de la información del proyecto, debido a, las herramientas anterior mente mencionadas, reconocen que la falta de conocimiento técnico y del modelo de negocio, como un factor importante para el desarrollo correcto de la metodología.

Scrum (10 extractos de texto)

Como se especificó anteriormente en esta categoría se tuvo en cuenta aspectos de Scrum que no pudieron ser clasificados en interacción, comunicación o conciencia.

Algunos equipos consideran la gestión del cambio como un factor importante para el uso de esta metodología:

“... podemos responder al cambio muy rápido, eso es lo principal y para nosotros es fundamental...”

“... el marco de trabajo nos permite identificar los cambios muy rápido porque cada 15 días se está generando releas, ellos miran y van aprobando, nos anticipamos a los cambios prácticamente.”

Y reconocen algunos inconvenientes relacionados con la adopción de esta metodología:

“... sabíamos que empezar a dejar artefactos de las metodologías tradicionales, a empezar a trabajar sin esos artefactos genera un poco de miedo.”

“... no todos en el equipo conocían el marco de trabajo, entonces se empezaron a hacer las capacitaciones, y otro es que no todos estamos certificados”

“El principal problema que tuvimos allá es que la gente utiliza Scrum un momentico y después se olvida, ya no lo utiliza, si no está alguien que está pendiente, que haga las

reuniones diarias, reunión retrospectiva, en el momento menos pensado ya no están utilizando nada de Scrum...”

“... yo creo que uno de los principales retos es la cultura y la concientización, muchas veces las personas hacen algo, pero o porque están obligados o porque alguien les dice que tienen que hacerlo, pero es diferente cuando ya todos saben que eso si funciona...”

“... cuando hay alguien que tenga una forma de pensar diferente, "para mí no me vale Scrum, no me representa nada bueno" entonces va a ser reacio nunca han trabajado con Scrum, entonces imagínese ahora decirle venga hagamos una reunión diaria, él se ríe de eso, ósea no le encuentra sentido.”

Esta sección permitió identificar el conocimiento de la metodología como un factor crítico durante su adopción, y además también permite denotar el comportamiento personal como un factor a tener en cuenta para la adopción de esta metodología.

4.1.5. Análisis de datos

Para el análisis de las entrevistas se utilizó la herramienta de software WEFT QDA [57], herramienta de investigación cualitativa utilizada para el análisis de datos textuales como transcripciones de entrevistas, documentos y notas de campos. Esta herramienta es de uso gratuito y posee una licencia de dominio público además de estar disponible para Windows y Linux.

Esta herramienta permite: 1) guardar los datos de forma organizada a partir de categorías definidas por el investigador; 2) buscar y clasificar los datos (transcripción de entrevistas) en categorías analíticas establecidas por el investigador; 3) establecer relaciones a partir de los datos a través de búsquedas, 4) visualizar las búsquedas en forma de textos o cuadros de doble entrada. [57]

Una vez integrada la información de las entrevistas en la herramienta, se generaron 4 categorías de análisis (Interacción, Comunicación, Conciencia y Scrum) que generan una linealidad entre los objetivos de este artículo y el análisis de resultados.

Producto de la entrevista se identificaron un listado de actividades que fueron agrupadas por frecuencia, es decir, cuantos equipos coincidieron al mencionar la actividad (1, 2 o 3), a su vez las actividades desarrolladas por los equipos se asociaron a actividades colaborativas y a eventos o elementos Scrum con el fin de identificar actividades dentro de la metodología en donde se pueden presentar los problemas especificados por los participantes.

Tabla 3: Frecuencia vs Colaboración

	Actividad (Frecuencia)	Colaboración
Interacción	Reuniones diarias. (3) Reuniones. (3) Ayuda entre los miembros del equipo para la solución de problemas presentados en el proyecto. (3)	Interacción cara a cara. Interacción intergrupala. Resolución de conflictos
Comunicación	Reuniones. (2) Redes Sociales. (2) Videoconferencias. (2) Personalidad. (1)Compromiso.(2)	Canales de comunicación Comunicación directa Comunicación indirecta Resolución de conflictos
Conciencia	Conocimiento de actividades desarrolladas por otros miembros del equipo. (3) El equipo en conjunto logra el avance. (3) Consulta permanente del estado de las tareas. (3) Las habilidades de los miembros del equipo están al servicio del proyecto. (2) Roles. (1) Documentación.(2)	Conciencia de equipo Conocimiento compartido Conciencia del estado del producto y del proyecto.

Tabla 4: Frecuencia vs Scrum

	Actividad (Frecuencia)	Scrum
Interacción	Reunión diaria. (3)	Reunión de planificación Reunión diaria Revisión de Sprint Retrospectiva del Sprint
Comunicación	Reunión diaria. (3) Notificar problemas u obstáculos al equipo. (3)	Reunión de planificación Product backlog Reunión diaria Revisión de Sprint Retrospectiva del Sprint
Conciencia	Reunión diaria. (3) Reunión de planificación. (2) Consulta del tablero de tareas.(3) Roles. (2) Herramientas tecnológicas. (2)	Roles Reunión diaria Reunión de planificación Sprint backlog Release burndown chart Sprint burndown chart Task board

Colaboración

Tabla 5: Actividades vs Frecuencia Colaboración

	Número actividades	Frecuencia por actividad
Interacción	3	3 Actividades con frecuencia 3
Comunicación	5	4 Actividades con frecuencia 2 1 Actividad con frecuencia 1
Conciencia	6	3 Actividades con frecuencia 3 2 Actividades con frecuencia 2 1 Actividad con frecuencia 1

Scrum

Tabla 6: Actividades vs Frecuencia Scrum

	Número actividades	Frecuencia por actividad
Interacción	1	1 Actividad con frecuencia 3
Comunicación	2	2 Actividades con frecuencia 3
Conciencia	5	Actividades con frecuencia 3 Actividades con frecuencia 2

Las actividades con mayor incidencia (frecuencia 3) revelan problemas o factores de riesgo presentes durante la adopción que se pueden presentar de forma frecuente en equipos adoptantes de Scrum. Por otra parte, las actividades con menor incidencia (frecuencia 1) revelan problemas o factores de riesgo menos comunes.

Los participantes identificaron de forma explícita algunos de los problemas que encontraron durante la adopción de Scrum basados en su experiencia. La mayoría de estos problemas están asociados al factor persona que se evidencia en la cultura, las creencias, el lenguaje, la personalidad y la actitud.

Para este trabajo se denomina riesgo a los diferentes factores que afectan la correcta adopción de Scrum. Los principales riesgos identificados en este estudio son:

- El pensamiento de trabajo individual.
- Restar importancia al propósito de cada evento Scrum (falta de conciencia)
- Falta de participación o ausencia durante las sesiones
- El factor persona (personalidad, cultura, creencias, etc) puede afectar el proceso de adopción.
- Ausencia de la propiedad colectiva de la información.

- Ausencia de una comunicación efectiva.
- Falta de cooperación y comunicación constante con el cliente
- Falta de disciplina en las reuniones diarias
- Falta de participación constante del cliente
- Falta de comprensión de los valores ágiles
- Falta de comprensión de los objetivos de las reuniones.

4.1.6. Lecciones aprendidas

Durante del desarrollo del estudio de caso, fueron varias las lecciones aprendidas:

- El estudio de caso basado en entrevistas semiestructuradas permitió que los participantes no sólo se limiten a contestar las preguntas sino también a interactuar de forma abierta, ya que, el entrevistador puede ahondar en temas que considere relevantes por medio de preguntas abiertas, esto permitió obtener información de forma natural sobre la adopción de Scrum.
- El método de recolección de datos permitió conocer la percepción de cada participante.
- Hay una brecha grande entre lo que dice la guía de Scrum y lo que realmente sucede al momento de realizar la adopción, es un camino difícil, ya que se trata de cambiar el pensamiento y valores de las personas más que las herramientas o los procesos.
- La adopción puede verse afectada por actitudes o comportamientos de trabajo individual, ya que, puede perjudicar el trabajo en equipo y el desarrollo del proyecto.
- La adopción puede verse afectada por el tipo de contratación, ya que, por ejemplo, el contrato por prestación de servicios no requiere el cumplimiento de horarios, por tanto, la presencia durante las reuniones de algunos miembros del equipo se puede ver comprometida.
- La capacitación del equipo en Scrum no garantiza una buena adopción, el factor persona hace que la adopción sea exitosa.

CAPÍTULO 5

ESPECIFICACIÓN DE THINKLETS E IMPLEMENTACIÓN STHINKLETS

En este capítulo, se especifican los Thinklets existentes actualmente y se clasifican en categorías mencionadas en mayor proporción en la literatura [58], [59], [60], [61], [62], [63]. Además, se describen Scrum Thinklets (SThinklets) realizados a través de la metodología propuesta por [64] con el fin de mejorar la colaboración durante la adopción de Scrum. Los SThinklets son actividades colaborativas reutilizables y transferibles asociadas al proceso Scrum.

5.1. ESPECIFICACIÓN DE THINKLETS Y GENERACIÓN DE STHINKLETS

Mediante la revisión de literatura se identificaron 34 Thinklets y se asociaron a diversos patrones de la colaboración [64].

5.1.1. Patrones de Colaboración

A continuación, se especifican los patrones de la Colaboración.

Tabla 7: Patrones de Colaboración

Categorías (Patrón de Colaboración)	Objetivo
Divergencia	Pasar de un estado en el que se tienen menos conceptos, a un estado en el que se tienen más. Se busca que un grupo cree conceptos que aún no se han considerado
Convergencia	Pasar de un estado en el que se tienen muchos conceptos, a un estado en el que se tienen pocos conceptos pero que requieren mayor atención. Se busca que un grupo reduzca lo que se denomina “carga cognitiva” al reducir la cantidad de conceptos que deberán considerar.
Organización	Pasar de un estado de menor comprensión a uno de mayor comprensión entre las relaciones de los conceptos.
Evaluación	Pasar de comprender en poca proporción a analizar las posibles consecuencias de los conceptos.
Consenso	Pasar de tener pocos miembros a tener muchos, los cuales estarán de acuerdo sobre los cursos de acción. Se busca que un grupo crítico de partes interesadas en el éxito, lleguen a compromisos aceptables mutuamente y se comprometan también a cumplir objetivos

	específicos.
Generación	Pasar de tener pocos a tener muchos conceptos. Se busca obtener una base conceptual amplia para compartirla al grupo.
Reducción	Pasar de tener una gran cantidad de conceptos a tener una cantidad más

5.1.2. Especificación de Thinklets por categoría

A continuación, se presenta cada uno de los thinklets organizados por cada una de las categorías, es decir, agrupados de acuerdo al patrón de Colaboración [64]. En el documento de anexos (Anexo 5) se encuentra en mayor detalle esta especificación.

Tabla 8: Thinklets patrón generación

CATEGORÍA	NOMBRE THINKLET
Generación	FreeBrainstorm
	DirectedBrainstorm
	LeafHopper
	OnePage
	ComparativeBrainstorm
	BranchBuilder
	DealersChoice
	Plus/Minus/Interesting
	The Lobbyist

Tabla 9: Thinklets patrón Reducción

CATEGORÍA	NOMBRE THINKLET
Reducción	FastFocus
	BroomWagon

Tabla 10: Thinklets patrón Organización

CATEGORÍA	NOMBRE THINKLET
Organización	ThemeSeeker
	PopCornSort
	Evolution
	ChauffeurSort

Tabla 11: Thinklets patrón Consenso.

CATEGORÍA	NOMBRE THINKLET
Consenso	MoodRing
	StrawPoll?
	PinTheTailOnTheDonkey
	CrowBar

Tabla 12: Thinklets patrón Convergencia

CATEGORÍA	NOMBRE THINKLET
Convergencia	DimSum
	GoldMiner
	ExpertChoice
	GarlicSqueezer
	ReviewReflect

Tabla 13: Thinklets patrón Divergencia

CATEGORÍA	NOMBRE THINKLET
Divergencia	OneMinuteMadness
	TopFive

Tabla 14: Thinklets patrón Clarificación

CATEGORÍA	NOMBRE THINKLET
Clarificación	Concentration

5.2. STHINKLETS

En este trabajo se definen los Scrum Thinklets o SThinklets como unidades cohesivas que tienen un propósito y una forma de comportamiento. Están diseñados para integrar la ingeniería de la colaboración y la metodología Scrum, son una guía o recomendación para realizar tareas específicas de Scrum de manera colaborativa.

5.2.1. Proceso de creación de los Scrum Thinklets

El proceso para obtener los SThinklets, parte de la identificación de tareas

específicas pertenecientes a Scrum, que puedan ser realizadas de manera colaborativa. A estas tareas se les asocia elementos de la ingeniería de la Colaboración, en este caso Thinklets para fortalecer la colaboración.

Por lo tanto, para la creación de los SThinklets, se realiza la descomposición del proceso Scrum, y a su vez, de cada una de las actividades que lo conforman. Lo anterior, se hace con base en la metodología proporcionada por [60], [65] la cual está compuesta por 4 fases. En la primera, se realiza un diagnóstico del proceso, técnica o tarea; en la segunda, se realiza la descomposición del proceso; en la tercera se establecen las tareas que requieren colaboración, y en la última, se realiza la asignación de ThinkLets.

El procedimiento para la creación de SThinklets es el siguiente, [60] [65]:

FASE 1: Diagnóstico del proceso

1. Identificar el proceso que requiere descomposición, en este caso, el proceso Scrum.
2. Realizar una descripción del proceso Scrum.
3. Identificar participantes y características relevantes del proceso.

FASE 2: Descomposición del proceso

1. Realizar la descomposición del proceso Scrum para obtener las actividades que lo componen e identificar la secuencia entre ellas.

FASE 3: Establecimiento de tareas colaborativas

1. Realizar la descomposición de cada una de las actividades obtenidas en el paso anterior.
2. Identificar las tareas que se pueden realizar de manera colaborativa.
3. Identificar los roles involucrados en las tareas identificadas en el paso anterior.
4. Asociar uno o más patrones de colaboración a las tareas obtenidas en el paso 2.

FASE 4: Relación con colaboración

1. Asociar a las tareas obtenidas en la fase anterior uno o más thinklets; estos últimos, están relacionados con el o los patrones de colaboración asociados en la fase previa.
2. Realizar un proceso mediante el cual, las tareas colaborativas identificadas sean escritas en términos de Scrum, pero utilizando el thinklet seleccionado de manera implícita.

5.2.2. FASE 1: Descripción del proceso

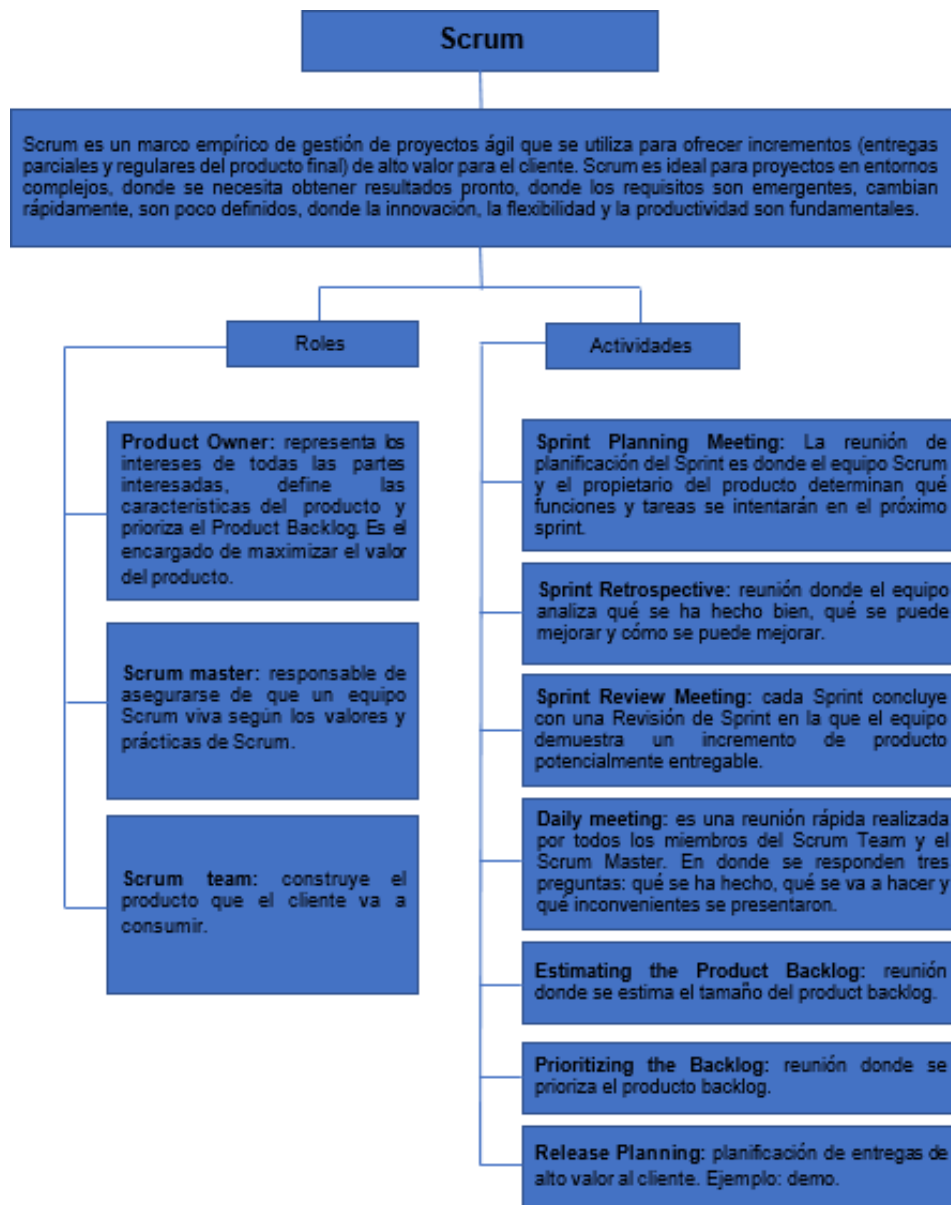


Figura 3: Proceso Scrum

5.2.3. FASE 2: Descomposición del proceso

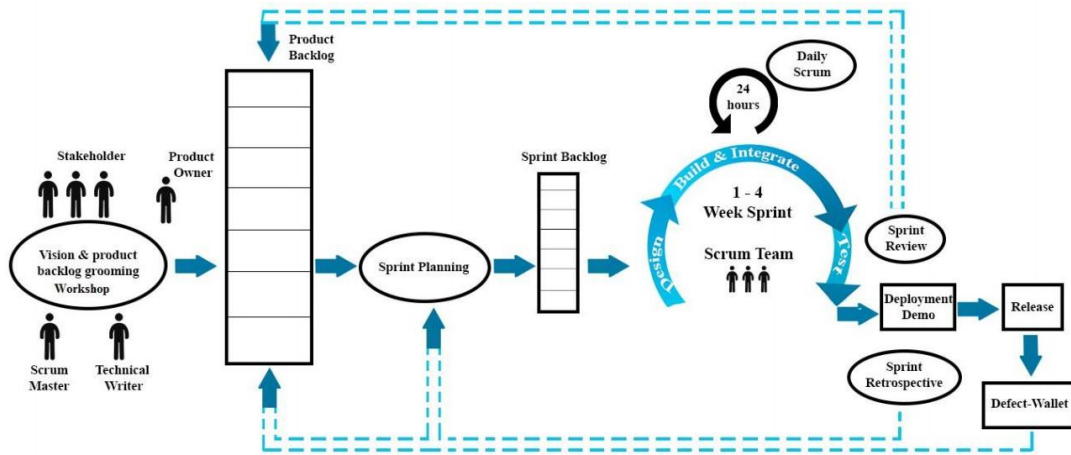


Figura 4: Scrum process model [72]

El proceso Scrum a nivel macro está conformado por las siguientes actividades:

- Sprint planning
- Daily meeting
- Sprint Retrospective
- Sprint Review Meeting
- Estimating the Product Backlog
- Release planning

5.2.4. FASE 3: Establecimiento de tareas colaborativas

A continuación, se realizó una descomposición más minuciosa del proceso Scrum, para ello, las actividades se descomponen en tareas y se identificó los roles asociados a cada actividad.

- Sprint planning
 - Definir el objetivo del sprint.
 - Seleccionan elementos del Product Backlog para el sprint actual.
 - Refinar los elementos del Product Backlog o Sprint backlog si es necesario.
- Daily meeting: Scrum Team - Scrum Master
 - Identificar el progreso hacia el objetivo del sprint -X (no colaborativa, pero es importante para el awareness)
 - Identifican obstáculos.

- Adaptar el sprint Backlog si es necesario
- Actualizar el estado del sprint
- Sprint Retrospective: Scrum Team - Scrum Master - Product Owner
 - Identificar aspectos a mejorar.
 - Identificar los obstáculos durante el sprint.
 - Examinar las causas de dichos obstáculos.
 - Compartir como fueron solucionados (o no) dichos obstáculos.
- Sprint Review Meeting: Scrum Team - Scrum Master - Product Owner
 - Identificar el estado final del sprint
 - Determinar futuras adaptaciones o mejoras.
 - Ajustar el product Backlog si es necesario
 - Presentar los resultados de su trabajo a los interesados
- Estimating the Product Backlog: Scrum Team
 - Socializar la tarea a desarrollar
 - Definir la complejidad de las tareas a desarrollar (planning poker)
- Release planning: Scrum Team - Scrum Master - Product Owner
 - Construir / Actualizar el backlog
 - Priorizar el product Backlog si es necesario /Repriorizar el backlog

Una vez se obtiene una descomposición más detallada, se identifican las actividades que se pueden realizar de forma colaborativa. Además, a cada actividad se le asocia uno o más patrones de colaboración según la siguiente tabla construida para facilitar la selección de los patrones a partir de las acciones o verbos usados para describir cada actividad.

Tabla 15: Tabla de verbos asociados a cada patrón de colaboración

Patrón	Verbos
Divergencia	Descomponer, separar, desvincular, diferenciar, desacoplar, desligar
Convergencia	Clasificar, agrupar, enfocar, catalogar
Organización	Explicar, comprender, exponer
Evaluación	Evaluar
Consenso	Acordar, concertar, votar, determinar
Generación	Investigar, indagar, averiguar, buscar, deducir, ahondar, hallar

Reducción	Limitar, restringir, concretar, enfocar
Clarificación	Precisar, socializar, explicar, definir, clarificar, conceptualizar

A continuación, se encuentran las tareas colaborativas identificadas:

Sprint planning

- Definir el objetivo del sprint.
El Product owner propone alternativas que pueden incrementar valor al producto. Los miembros del Scrum Team seleccionan y definen los objetivos a realizar durante el sprint, asegurándose que los mismos satisfagan las expectativas del Product owner.

Patrón principal recomendado: Convergencia.

Patrones secundarios recomendados: Generación

- Seleccionan elementos del Product Backlog para el sprint actual.
El equipo selecciona los elementos del Product Backlog necesarios para cumplir con los objetivos del Sprint considerando la productividad (velocidad) del equipo.

Patrón principal recomendado: Consenso

Patrones secundarios recomendados: Convergencia

- Refinar los elementos del Product backlog o Sprint Backlog si es necesario.
El Scrum team evalúa si es necesario refinar elementos del Product Backlog y determina si se realizan cambios, modificaciones o no.

Patrón principal recomendado: Evaluación

Patrones secundarios recomendados: Reducción, Generación, Consenso

Daily meeting: Scrum Team - Scrum Master

- Identificar el progreso hacia el objetivo del sprint -X (no colaborativa, pero es importante para el awareness)
- Identifican obstáculos
El Scrum Team expone los obstáculos que se presentaron para desarrollar sus actividades y el Scrum Master toma nota para poder solventar los obstáculos presentados.

Patrón principal recomendado: Clarificación

Patrones secundarios recomendados: NA

- Adaptar el sprint Backlog si es necesario (puede aplicar el mismo del product backlog)
Scrum Team evalúa si es necesario refinar elementos del Sprint Backlog y determina si se realizan cambios, modificaciones o no.

Patrón principal recomendado: Evaluación

Patrones secundarios recomendados: Reducción, Generación, Consenso

- Actualizar el estado del sprint
El Scrum team se encarga de actualizar el estado de sus actividades para poder hacer seguimiento al avance del sprint.

Patrón principal recomendado: Organización

Sprint Retrospective: Scrum Team - Scrum Master - Product Owner

- Identificar aspectos a mejorar.
El Scrum team identifica aspectos a mejorar para los siguientes Sprints.
Patrón principal recomendado: Generación.
Patrones secundarios recomendados: Reducción, Evaluación, Consenso

Compartir con el equipo los obstáculos presentados (se propone un STHINKLET que abarca las siguientes 3 actividades)

- Identificar los obstáculos durante el sprint.
El Scrum team identifica los obstáculos que se presentaron durante el sprint

Patrón principal recomendado: Generación

Patrones secundarios recomendados: Reducción, Evaluación, Consenso

- Examinar las causas de dichos obstáculos.
El Scrum team evalúa la causa de los obstáculos y se hace una clarificación de los mismos para lograr una mejor comprensión

Patrón principal recomendado: Evaluación

Patrones secundarios recomendados: Clarificación, Organización

- Compartir como fueron solucionados (o no) dichos obstáculos.
El Scrum team comparte el conocimiento adquirido a partir de los obstáculos y evalúa las soluciones o posibles soluciones.

Patrón principal recomendado: Clarificación

Patrones secundarios recomendados: Evaluación

Sprint Review Meeting: Scrum Team - Scrum Master - Product Owner

- Identificar el estado final del sprint
A partir del estado de las actividades del Scrum team se identifica y se comparte lo que se alcanzó a cumplir del sprint.

Patrón principal recomendado: Organización

Patrones secundarios recomendados: Clarificación

- Determinar futuras adaptaciones o mejoras.
El Scrum team y el product owner evalúan si es necesario refinar elementos del Product Backlog y determina si se realizan cambios, modificaciones o no.

Patrón principal recomendado: Generación

Patrones secundarios recomendados: Reducción, Evaluación, Consenso

- Ajustar el product Backlog si es necesario
El Scrum team y product owner evalúa si es necesario refinar elementos del Product Backlog y determina si se realizan cambios, modificaciones o no.

Patrón principal recomendado: Evaluación

Patrones secundarios recomendados: Reducción, Generación, Consenso

- Presentar los resultados de su trabajo a los interesados (No colaborativa)

Patrón principal recomendado: Clarificación

Patrones secundarios recomendados: Generación

Estimating the Product Backlog: Scrum Team

- Socializar la tarea a desarrollar
Patrón principal recomendado: Clarificación
Patrones secundarios recomendados: Organización
- Definir la complejidad de las tareas a desarrollar (planning poker)
Patrón principal recomendado: Consenso
Patrones secundarios recomendados: Evaluación

Release planning: Scrum Team - Scrum Master - Product Owner

- Construir / Actualizar el backlog
Patrón principal recomendado:
Patrones secundarios recomendados:

- Priorizar el product Backlog si es necesario /Repriorizar el backlog
Patrón principal recomendado:
Patrones secundarios recomendados:

5.2.5. FASE 4: Relación con colaboración

5.2.5.1. Estructura Scrum Thinklet

La tabla que se presenta a continuación, contiene la estructura de un SThinklet, esta estructura fue obtenida con base en los ThinkLets existentes, con el fin de garantizar que la especificación de cada uno de ellos se realice forma adecuada.

Tabla 16: Estructura Scrum Thinklet

IDENTIFICADOR
NOMBRE STHINKLET
ACTIVIDAD DE SCRUM
TAREA DE SCRUM
PATRÓN DE COLABORACIÓN SUGERIDO
THINKLETS ASOCIADO SUGERIDOS
ROLES INVOLUCRADOS
OBJETIVO/PROPÓSITO
¿CÓMO USAR?
ARTEFACTOS DE ENTRADA SUGERIDOS
ARTEFACTOS DE SALIDA SUGERIDOS
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS

5.2.5.2. Especificación SThinklets

Cada Thinklet creado, fue especificado según la estructura definida en la sección anterior. Por lo tanto, cada una de las siguientes tablas hace referencia a un SThinklet, los cuales, tienen como sugerencias posibles patrones de colaboración y Thinklets asociados, sin embargo, no hay restricción y pueden ser utilizados otros Thinklets que no se encuentren ahí especificados. Se debe tener en cuenta que estos SThinklets no son obligatorios y consisten en una sugerencia para la mejora del proceso, por lo tanto, pueden ser adaptados a las necesidades del equipo.

Tabla 17: SThinklet_Definicion_Objeto

IDENTIFICADOR:	ST_001
NOMBRE STHINKLET:	SThinklet_Definicion_Objeto
ACTIVIDAD DE SCRUM	Sprint planning
TAREA DE SCRUM	Definir el objetivo del sprint
PATRÓN DE COLABORACIÓN SUGERIDO:	Convergencia, Generación
THINKLETS ASOCIADO SUGERIDOS:	ExpertChoice, Plus/Minus/Interesting
ROLES INVOLUCRADOS:	Product Owner, Scrum Team
OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como propósito que el Product owner pueda acordar con el Scrum team cuál va a ser el objetivo del Sprint.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. El Product Owner propone alternativas para agregar valor al producto y realiza una breve socialización de cada una de ellas. 2. Una vez se presenten los posibles objetivos del Sprint, el Product Owner debe conducir una votación para reducir el número de alternativas. 3. De forma individual los participantes deben expresar las ventajas y desventajas de abordar cada propuesta en el Sprint, con el fin de determinar cuál de las propuestas es más viable de realizar y aporta mayor valor al producto. 4. Después del paso anterior, el Productowner dirige una nueva votación para seleccionar el objetivo del Sprint. 5. En caso de empate, el Product Owner decide el objetivo del Sprint (Entre las propuestas resultantes de la primer votación – Paso 2).
ARTEFACTOS DE ENTRADA SUGERIDOS:	Lista de alternativas para el objetivo del Sprint
ARTEFACTOS DE SALIDA SUGERIDOS:	Objetivo del Sprint
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	Thinklets sugeridos: <ul style="list-style-type: none"> • ExpertChoice: se usa de manera implícita, en

	<p>el momento cuando el Product Owner dirige la votación para organizar y reducir las alternativas presentadas.</p> <ul style="list-style-type: none"> • Plus/Minus/Interesting: se usa de manera implícita, cuando se solicita a los miembros del Scrum team que expresen las ventajas y desventajas de cada una de las alternativas de objetivo de Sprint presentadas por el Product Owner.
--	--

Tabla 18: SThinklet_Seleccion_Elementos_Sprint

IDENTIFICADOR:	ST_002
NOMBRE STHINKLET:	SThinklet_Seleccion_Elementos_Sprint
ACTIVIDAD DE SCRUM	Sprint planning
TAREA DE SCRUM	Seleccionar elementos del product backlog para el Sprint actual
PATRÓN DE COLABORACIÓN SUGERIDO:	Consenso, Convergencia
THINKLETS ASOCIADO SUGERIDOS:	ExpertChoice , MoonRing
ROLES INVOLUCRADOS:	Scrum team
OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como objetivo que los miembros del Scrum team puedan seleccionar las historias de usuario y las tareas que se van a abordar en el sprint en concordancia con el objetivo del Sprint.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. Un miembro del Scrum team debe tomar el liderazgo de esta actividad, lo cual se realizará por turnos garantizando que todos los miembros participen como líderes de esta actividad al menos una vez. 2. El líder expresa de forma clara el objetivo del Sprint. 3. El líder nombrará uno por uno los elementos del Product Backlog. 4. Los miembros del Scrum team postulan los elementos del Product Backlog que consideran acordes al objetivo del Sprint levantando la mano o expresando

	<p>claramente estar de acuerdo con incluir el elemento en el Sprint actual.</p> <p>El líder conduce una votación para determinar si el elemento debe ser incluido, si la mayoría del Scrum team vota a favor se incluye en el Sprint actual.</p> <p>5. Si los elementos incluidos rebasan la productividad del Scrum team, se debe someter a votación los elementos que fueron agregados al Sprint actual (de tal forma que sea acorde a la productividad). Los elementos restantes deberán regresar al Product backlog con mayor prioridad.</p>
ARTEFACTOS DE ENTRADA SUGERIDOS:	Elementos del Product backlog, Objetivos del sprint
ARTEFACTOS DE SALIDA SUGERIDOS:	Sprint Backlog
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	<p>Thinklets sugeridos:</p> <ul style="list-style-type: none"> • ExpertChoice: se usa de manera implícita, en el momento cuando se designa un miembro del Scrum team para dirigir la votación. • MoonRing: se usa de manera implícita, en el momento que se discuten las actividades necesarias para el cumplimiento del objetivo del sprint

Tabla 19: SThinklet_Refinar_Elementos_Sprint

IDENTIFICADOR:	ST_003
NOMBRE STHINKLET:	SThinklet_Refinar_Elementos_Sprint
ACTIVIDAD DE SCRUM	Sprint planning
TAREA DE SCRUM	Refinar los elementos del Product Backlog o Sprint backlog si es necesario
PATRÓN DE COLABORACIÓN SUGERIDO:	Evaluación, Reducción, Generación, Consenso
THINKLETS ASOCIADO SUGERIDOS:	StrawPoll, MoodRing
ROLES INVOLUCRADOS:	Scrum Team

OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como objetivo que el Scrum team pueda refinar el Product backlog, es decir, dividir y definir aún más los elementos del Product Backlog en elementos más pequeños y Precisos.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. El Scrum Team evalúa si es necesario refinar elementos del Product Backlog. 2. Los miembros del Scrum team indican los elementos del Product backlog que deben ser refinados. 3. Se revisa uno a uno los elementos seleccionados para ser refinados y se discute hasta qué grado deben ser divididos. 4. Se definen los elementos del Product backlog resultantes.
ARTEFACTOS DE ENTRADA SUGERIDOS:	Product backlog
ARTEFACTOS DE SALIDA SUGERIDOS:	Product backlog refinado
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	<p>Thinklets sugeridos:</p> <ul style="list-style-type: none"> • StrawPoll: se presenta de forma implícita al determinar que se deben refinar elementos del Product backlog. • MoodRing: se presenta de forma implícita cuando los miembros del Scrum team dan su opinión y discuten acerca de los elementos del Product backlog.

Tabla 20: SThinklet_Identificar_Mejoras

IDENTIFICADOR:	ST_004
NOMBRE STHINKLET:	SThinklet_Identificar_Mejoras
ACTIVIDAD DE SCRUM	Sprint Retrospective
TAREA DE SCRUM	Identificar aspectos a mejorar
PATRÓN DE COLABORACIÓN SUGERIDO:	Generación, Reducción, Evaluación, Consenso
THINKLETS ASOCIADO SUGERIDOS:	LeafHopper, DirectedBrainstorm, Plus/Minus/Interesting,
ROLES INVOLUCRADOS:	Scrum Team, Scrum Master, Product Owner

OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como objetivo que Scrum team, scrum master y product owner expongan aspectos que el equipo puede mejorar, con el fin de establecer acuerdos o estrategias de mejora.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. Un miembro del equipo indica un aspecto de mejora. Además, puede o no indicar la solución que crea conveniente. 2. Los miembros del equipo pueden aportar ideas de cómo lograr la mejora. 3. Los miembros del equipo pueden discutir las ventajas y desventajas de las propuestas de mejora. 4. Después de debatir las propuestas, el equipo establece un acuerdo acerca de la mejora y como se debe implementar.
ARTEFACTOS DE ENTRADA SUGERIDOS:	Idea de mejora
ARTEFACTOS DE SALIDA SUGERIDOS:	Estrategia de mejora.
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	<p>Thinklets sugeridos:</p> <ul style="list-style-type: none"> • LeafHopper: se presenta de forma implícita al solicitar a los miembros del equipo expresar las ideas de mejora que el equipo puede implementar. • DirectedBrainstorm: se presenta de forma implícita cuando se genera una lluvia de ideas de cómo realizar una mejora. • Plus/Minus/Interesting: se presenta de forma implícita cuando el equipo debate cuál es la estrategia de mejora más viable o conveniente a implementar.

Tabla 21: SThinklet_Identificar_bloqueos

IDENTIFICADOR:	ST_005
NOMBRE STHINKLET:	SThinklet_Identificar_bloqueos
ACTIVIDAD DE SCRUM	Identificar los obstáculos durante el sprint.
TAREA DE SCRUM	Sprint Retrospective
PATRÓN DE COLABORACIÓN SUGERIDO:	Generación, Reducción, Evaluación, Consenso, Clarificación, Organización,

THINKLETS SUGERIDOS:	ASOCIADO	LeafHopper, DirectedBrainstorm, Plus/Minus/Interesting
ROLES INVOLUCRADOS:		Scrum team
OBJETIVO/PROPÓSITO:		Este STHINKLET tiene como objetivo que el Scrum team comparta los obstáculos presentados durante el sprint, sus causas y como fueron solucionados.
¿CÓMO USAR?		<ol style="list-style-type: none"> 1. Cada miembro del Scrum team expone los obstáculos que se presentaron durante el desarrollo de sus tareas. 2. Examinar las causas del obstáculo. 3. Compartir la solución del obstáculo, para que el equipo logre tener conocimiento compartido tanto del problema como de la solución.
ARTEFACTOS DE ENTRADA SUGERIDOS:		
ARTEFACTOS DE SALIDA SUGERIDOS:		Lista de obstáculos con sus respectivas soluciones
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:		<p>Thinklets sugeridos:</p> <ul style="list-style-type: none"> • LeafHopper: se presenta de forma implícita al solicitar a los miembros del equipo expresar las ideas de mejora que el equipo puede implementar. • DirectedBrainstorm: se presenta de forma implícita cuando se genera una lluvia de ideas de cómo realizar una mejora. • Plus/Minus/Interesting: se presenta de forma implícita cuando el equipo debate cuál es la estrategia de mejora más viable o conveniente a implementar.

CAPÍTULO 6

UNA GUÍA DE TRABAJO PARA EL FORTALECIMIENTO DE PRÁCTICAS COLABORATIVAS DURANTE LA ADOPCIÓN DE SCRUM

En este capítulo se propone una guía de trabajo que sirva como guía para orientar el proceso de adopción de Scrum en una pequeña organización. Para ello se consideró utilizar una guía de adopción existente y complementar con elementos de la ingeniería de la Colaboración que sugieran herramientas o acciones para fortalecer la colaboración y lograr una mejor adopción.

¿Cómo se hace? Basados en la investigación de la literatura sobre adopción de Scrum, en la experiencia propia y tomando las lecciones aprendidas y sugerencias. Además, tomando como base la ruta de trabajo propuesta por Araujo et. al. [6] a la cual se adiciona los aportes encontrados durante el desarrollo de este trabajo.

¿Por qué Scrum? Porque *Scrum* es un marco de trabajo que no solo se aplica para proyectos de software si no para cualquier otro tipo de proyecto. Además, es una metodología ampliamente utilizada a nivel regional, nacional y mundial [40].

¿A quiénes está dirigida la guía de trabajo? Esta propuesta de trabajo sirve de guía para las pequeñas organizaciones que quieran adoptar Scrum como marco de trabajo. Está orientada a pequeñas organizaciones debido al alcance del proyecto y porque su validación se llevó a cabo en organizaciones con esta característica. La guía también está orientada para equipos adoptantes que quieran fortalecer la colaboración con el fin de mejorar su proceso de adopción.

¿Qué contiene? En esta guía se brindan sugerencias y actividades que pueden ayudar en el proceso de adopción.

6.1. ESTRUCTURA DE LA GUIA DE TRABAJO

Para construir la guía de trabajo se determina una estructura. La guía propuesta sigue la misma estructura con la cual está construida la guía de adopción de Scrum que se desea complementar incluyendo elementos de la ingeniería de la Colaboración. A partir de las actividades, tareas y recomendaciones de la guía se irán identificando elementos de la ingeniería de la colaboración que pueden mejorar los problemas referidos por los líderes de equipo en las entrevistas semiestructuradas.

En [6] se propuso una guía de trabajo para la adopción de Scrum, la ruta de trabajo consta de las etapas de: exploración, preparación y desarrollo.

En la etapa de exploración, se presentan las prácticas ágiles que la organización puede empezar a desarrollar. También se encuentran algunas actividades para ayudar a las personas a comprender los principios y valores ágiles.

En la etapa de preparación es donde las personas aprenden sobre el marco de trabajo Scrum, cuáles son sus roles, artefactos y eventos. Se realiza actividades que les ayudan entender el marco de trabajo de Scrum y se prepara el ambiente y las herramientas necesarias para poder empezar los Sprints.

En la etapa de desarrollo se empiezan a usar todos los eventos, roles y artefactos que establece el marco de trabajo Scrum en sus proyectos. Aquí se explican técnicas sugeridas, para realizar cada evento y artefactos, se dan pautas para que el equipo pueda realizar una mejora continua y así lograr una adopción completa del marco de trabajo.



Figura 5: Estructura de la guía de trabajo

A continuación, se va a detallar cada una de las etapas:

6.2. ETAPA DE EXPLORACIÓN

Esta etapa tiene como objetivo llevar a la organización a la adopción de Scrum, entendiendo que las personas son una parte fundamental durante la adopción de esta metodología, ya que, deben interiorizar los valores y principios ágiles. Se debe tener conciencia de que para ser ágil es importante conocer y dominar las prácticas y valores del manifiesto ágil, entender por qué son importantes y los beneficios que se obtienen al llevarlas a la práctica tanto en lo individual como en lo colectivo.

Esta etapa se enfoca en lograr que se brinde entrenamiento sobre valores y principios ágiles y que sean entendidos.

6.2.1. Identificar la metodología de trabajo actual

Reconocer cuál es la metodología de trabajo que se usa actualmente ayuda a que se tenga entendimiento de las diferencias entre la metodología actual y la metodología que se desea adoptar.

Una forma de extraer el proceso de trabajo, es hacer a través de un taller en el cual los asistentes, por parejas o individualmente, esquematizan su proceso actual sin tener en cuenta una nomenclatura formal de procesos. Deben hacer el dibujo de la forma que mejor entiendan dicho proceso. Después, se pide que cada uno explique el proceso dibujado. Luego, entre todos generar un solo modelo. Este ejercicio sirve para identificar oportunidades de mejora y que todos sean conscientes de cuál es la metodología de trabajo actual.

6.2.2. El manifiesto ágil

El manifiesto ágil surge en el 2001 a partir de una reunión de 17 expertos en ingeniería de software. Se trata de cuatro valores y 12 principios en los que se fundamenta el agilísimo, es muy importante, ya que, es el eje central sobre el que se fundamentan todas las metodologías ágiles existentes.

Antes de realizar la adopción de cualquier marco de trabajo ágil es importante conocer, comprender y llevar a la práctica este manifiesto. Teniendo claro, que ser ágil no significa leer el manifiesto ágil, ser ágil significa interiorizar los valores y principios, es un cambio de pensamiento y cultura en las personas, por tanto, ser ágil debe evidenciarse en la práctica. Una forma de enseñar y practicar los valores y principios ágiles es mediante los juegos ágiles [Anexos].

6.2.3. Establecer el objetivo

La organización establece cuál es el objetivo por el que quiere realizar la

adopción, por qué y para qué se adopta Scrum. Es importante tener este objetivo y hacerlo transparente para que todos sepan cual es el propósito o que se quiere lograr con el cambio de paradigma.

6.3. ETAPA DE PREPARACIÓN

Es importante que el equipo y los interesados conozcan el marco, cómo funciona y qué beneficios obtienen al adoptarlo, para ello se sugiere una etapa de preparación.

Esta etapa es un tiempo donde la organización recibe entrenamiento o capacitación sobre Scrum, se identifican los roles, se capacita el *Equipo de desarrollo* sobre las herramientas a utilizar para desarrollar el proyecto y se crea la versión inicial del Product Backlog, al igual que se comunica la definición de “lo hecho” y la duración de los sprints. Al finalizar esta etapa el equipo debe tener todas las herramientas que se necesitan para empezar las iteraciones de Scrum.

El propósito de esta etapa es evitar experimentos tempranos inmaduros y que el equipo tenga el entrenamiento adecuado.

6.3.1. Formación en scrum

Todas las personas de la organización, incluyendo los interesados, deben conocer Scrum, su funcionamiento, eventos, artefactos y terminología. Es esencial que se realice una capacitación de dicho marco de trabajo para conocer sobre Scrum con el objetivo de que se pueda trabajar sobre el mismo sentido.

Existen diferentes enfoques en los que se puede realizar la formación de Scrum [66]:

- **Método tradicional:** Consiste en clases magistrales en las que se explica teóricamente los conceptos de Scrum.
- **Simulación de proyectos:** Consiste en dividir la enseñanza en dos partes, una teórica y otra práctica. En la parte teórica se describen los aspectos generales de Scrum (roles, eventos, artefactos). En la segunda parte se lleva a cabo un proyecto simple, donde se distribuyen los roles, se ejecutan los eventos de Scrum y se desarrollan los artefactos.
- **Juegos educativos:** Son juegos de corta duración en donde identifican roles, se desarrollan artefactos y se llevan a cabo los eventos de Scrum para construir productos pequeños.

6.3.2. Equipo scrum

Ser ágil es trabajar en conjunto para lograr pequeños objetivos en pequeños

periodos de tiempo, por tanto, el equipo debe tener la capacidad de trabajar de forma colaborativa y auto-organizada, esto implica tener buena comunicación y motivación para lograr los objetivos individuales y colectivos [52].

Scrum dice que el equipo deberá tener los siguientes valores: compromiso, coraje, foco, apertura y respeto [52]. El equipo Scrum debe trabajar en mejorar las habilidades y valores que se requieren para el agilísimo, para ello se recomienda que se tenga en cuenta dentro del plan, actividades que ayuden a mejorar las habilidades blandas como: comunicación, autoorganización, autogestión, capacidad de cambio, colaboración, etc. Este es un esfuerzo que debe hacer la organización de manera conjunta para potenciar las habilidades en todos los miembros del equipo, debe realizarse a través de talleres o actividades específicas para este fin.

6.3.3. Identificar roles

Equipo de desarrollo, Product Owner y Scrum Master son los roles de *Scrum* y conforman el *Equipo Scrum*. Es necesario identificar y establecer la persona idónea para representar cada rol, cada persona que representa un rol debe tener claridad de los objetivos y responsabilidades del dicho rol. Si la persona no tiene el entendimiento del rol, entonces es recomendable tener entrenamiento de cada uno de los roles [67].

- El *Equipo de desarrollo* debe tener un tamaño entre 3 y 9 personas, ya que, un equipo con menos de tres personas puede no tener las habilidades necesarias para culminar un sprint, generando sobrecarga e incrementos posiblemente muy pequeños [52]. En un equipo con más de 9 personas se dificulta la coordinación y se torna compleja la autogestión. El *Product Owner* y el *Scrum Master* no se cuentan como integrantes del equipo a menos que ellos contribuyan al desarrollo de los elementos del *Product Backlog*. El *Equipo de desarrollo* debe tener entrenamiento del marco de trabajo Scrum, pero también debe contar con el conocimiento técnico para poder realizar el desarrollo del *Product Backlog*.
- El *Product Owner* su principal responsabilidad es la gestión del *Product Backlog* y que todos tengan claro su contenido. También, es necesario que establezca estrategias de comunicación para interactuar con clientes, interesados, usuarios finales o el mercado para poder identificar las necesidades reales y transformarlas en elementos del *Product Backlog*.
- El *Scrum Master* es la persona encargada de verificar que realmente se esté llevando adecuadamente Scrum. Es el encargado aprender y transmitir el conocimiento sobre Scrum al resto del equipo (por medio de capacitaciones, talleres, charlas, etc.). Además, se asegura de que cada

una de las personas que representa un rol, entienda bien sus responsabilidades. El *Scrum Master* es quien motiva al equipo Scrum por lo tanto debe interesarse en mejorar sus habilidades de comunicación, proactividad, animador y de gestión de equipos. Si la organización no cuenta con un *Scrum Master* y no tiene la forma de conseguir uno, se recomienda que se identifique una persona con estas habilidades. El *Scrum Master* no es ni el líder de proyectos, ni el jefe, el *Scrum Master* sirve de soporte al Equipo.

Tabla 22: Identificar roles - recomendación

Elemento de la ingeniería de la colaboración recomendado: Thinklet ExpertChoice y StrawPoll?
Justificación: El uso del Thinklet ExpertChoice puede ser utilizado para seleccionar al Scrum Master y al Product Owner basándose en las opiniones o votos de todo el equipo. Se puede usar de forma implícita el Thinklet: StrawPoll? para llegar a un consenso en cuanto al rol de los miembros del equipo, ya que, se puede dar el caso de cambiar de opinión. Se debe asegurar que todos los miembros del equipo emitan su opinión y presenten su voto, de esta forma se garantiza una comunicación efectiva y se aumenta el sentido de grupo.

6.3.4. Herramientas

Se debe seleccionar las herramientas que darán apoyo los artefactos de Scrum: Product Backlog, Sprint Backlog, incremento del producto y tablero de tareas. La organización decide qué herramientas utilizar, lo importante es que las herramientas seleccionadas sean en común acuerdo con el equipo.

Para un equipo que se encuentra en un solo espacio, se recomienda tener un tablero de Scrum en físico, ya que, influye en las emociones estimulando el sentido de pertenencia del equipo. Además, el tablero brinda información que ayuda a fomentar el pilar de la transparencia en Scrum, porque las personas que pasen cerca del tablero pueden acceder a esta información. El equipo puede diseñar su propio tablero lo que ayuda a fomentar la creatividad.

Algunas herramientas para apoyar el seguimiento del marco de trabajo son:

Scrumtool: Herramienta gratuita que permite la creación de proyectos, historias

de usuario, tareas que pueden ser gestionadas en el Product Backlog. Cada Sprint tiene un tablero con columnas flexibles, es una herramienta intuitiva pero no permite la comunicación entre el equipo ni tampoco el progreso del proyecto.

iceScrum: Herramienta Scrum y Kanban. Ofrece las opciones de operación, consulta y estimación de historias de usuario. Permite añadir historias de usuario a la pila de producto, dividir el tiempo en Sprints y mover estas historias de la pila de producto a cada uno de los Sprint. Posee la técnica de Planning Poker para la estimación y paneles virtuales. iceScrum ofrece varios planes de pago mensual o anual, donde el precio varía según el número de integrantes del equipo, tamaño en la nube y número de proyectos.

ScrumDo: Herramienta paga, es fácil de usar, permite gestionar listas de tareas, historias de usuario e iteraciones, obtener gráficos de avance “burndown”, soporta la planeación con Planning Poker.

Taiga.io: Taiga es un gestor de proyecto open source y gratuito, diseñado para nuevas empresas, basada en metodologías de desarrollos ágiles, con una interfaz fácil y un diseño amigable. Permite archivos adjuntos, llevar la planificación de cada sprint y el uso de kanban, estimación lineal de dos en dos o fibonacci, permite que se personalice la herramienta agregando plugins o quitando módulos. En cuanto a la privacidad taiga permite que se cree un proyecto privado, pero a partir del segundo proyecto son públicos.

Tabla 23: Herramientas - Recomendación

Elemento de la ingeniería de la colaboración recomendado: Patrón Reducción
Justificación: Las herramientas deben ser seleccionadas en común acuerdo con el equipo, se debe considerar varias herramientas, considerar sus características y determinar las ventajas y desventajas. Posterior a ello, la lista inicial debe reducirse a pocas opciones, las cuales pueden someterse a votación para determinar finalmente la herramienta a utilizar.

6.3.5. Product backlog v.0

Es importante generar el Product Backlog inicial antes de realizar la planeación del primer sprint, ya que, este es la entrada para cada planificación de sprint. La gestión del Product Backlog es únicamente responsabilidad del *Product Owner*,

quien debe asegurarse de mantener actualizado el *Product Backlog*, que sea de fácil acceso para el resto del equipo y priorizado de acuerdo a lo que genere más valor al cliente.

El *Product Owner*, con colaboración de los interesados, empieza a definir los requerimientos que se transformarán en elementos del *Product Backlog*, El *Product Owner* los escribe y pregunta cuantas veces sea necesario a los interesados, hasta tener unas necesidades más detalladas. Finalmente se hace un refinamiento con el *Equipo de desarrollo*, donde se reúnen, se leen cada una de las historias de usuario y se despejan las dudas.

Previamente se define un formato para definir el nivel de detalle de un elemento del *Product Backlog*. Por lo general un elemento del *Product Backlog*, contiene lo siguiente: una descripción en lenguaje claro de fácil entendimiento para el cliente y el equipo, en este punto solo se describe la necesidad sin involucrar términos técnicos, ni posibles soluciones, debe estar ordenado de acuerdo a la prioridad del cliente y cada elemento tener una estimación de esfuerzo.

El *Product Owner* se encarga de realizar la priorización de los elementos del *Product Backlog* de acuerdo al valor para el cliente (lo que le cliente quiere ver primero). El *Equipo de desarrollo* realiza una estimación del esfuerzo requerido para cada uno de los elementos del *Product Backlog*. El *Product Backlog* priorizado servirá siempre como insumo para la planificación de los sprints. El *Product Backlog* inicial estará listo cuando se tenga elementos que pueden ser “Terminados” por el *Equipo de Desarrollo* en un *Sprint*, estos son considerados “preparados” o “accionables” para ser seleccionados en una reunión de *Planificación de Sprint* [52], en este momento se puede empezar con la primera iteración de *Scrum*.

6.3.6. Duración del sprint

La duración del sprint se determina junto con el equipo, no debe ser mayor de cuatro semanas [52] y se sugiere que se intente con varias duraciones hasta encontrar la adecuada [68]. El equipo debe converger a una duración, ya que, es importante que todos los sprints tengan un mismo tiempo de duración, para medir el progreso y determinar la velocidad del equipo, esto se hace con la cantidad de puntos de historias (estimación) de usuario que se realizan en cada sprint.

6.3.7. Definición de “Hecho”

Definición de “hecho” o definición de terminado (*Definition of Done*) es el conjunto de características que una *Historia de Usuario* debe cumplir para que el *Equipo de desarrollo* pueda determinar si ha terminado de trabajar en ella [69]. Antes de empezar a desarrollar algún elemento del *Product Backlog* es importante que todo

el equipo, cliente e interesados, tengan claro esta definición de hecho. Se construye en conjunto con el Equipo Scrum. Se pueden definir criterios que debe cumplir para dar por terminado algunos criterios comunes son [69]:

- Todos los criterios de aceptación fueron cumplidos.
- Todos los archivos fuentes están en el repositorio de código.
- El *Product Owner* dio su visto bueno de la funcionalidad construida.

El incremento que se va a construir en cada sprint debe cumplir con la definición de terminado.

La actualización de la definición de hecho se debe realizar en la reunión de retrospectiva, cuando el equipo lo considere necesario.

Es importante la definición de hecho, ya que, esta garantiza que el trabajo sea realizado con un nivel de calidad uniforme.

6.4. ETAPA DE DESARROLLO

En esta etapa se quiere explicar de qué forma se puede poner en funcionamiento el marco de trabajo Scrum. Es en esta etapa donde se empiezan a usar todos los eventos, roles y artefactos que establece el marco de trabajo Scrum.

Adoptar Scrum como marco de trabajo no es un camino sencillo lleva mucho tiempo y esfuerzo y no se puede esperar ver resultados inmediatos, puede que al inicio sea muy difícil, que las personas muestren resistencia al cambio, pero a medida que se van viendo los resultados, la motivación aumenta. En esta etapa se conocerá en la práctica cómo funciona Scrum.

En esta etapa se trabaja en lograr que se dé la participación de las personas en los eventos en los que deben estar, que la duración de los sprint logre ser estable y con el trabajo necesario, que se muestren los avances en las reuniones diarias y al finalizar el sprint en la reunión de revisión, que el equipo aprenda de una forma de trabajo diferente a las metodologías en cascada, que el progreso en la ejecución del sprint sea visible y que la formación en Scrum sea sobre la práctica.

6.4.1. Planificación del sprint

Teniendo como insumo el Product Backlog se realiza la reunión de planificación de sprint (sprint planning), en esta reunión se determina qué elementos del Product Backlog se va a realizar en el sprint y cómo se realizarán. La duración de la reunión de planificación de sprint no debe durar más de 8 horas para un sprint de 4 semanas. Para un sprint más corto esta reunión probablemente tendrá una menor duración [52]. Es importante que, al empezar, el *Product Owner* defina el

objetivo del sprint, ya que sin un objetivo se trabajará en un sprint sin propósito. Estas reuniones de planificación no deben iniciar sin la presencia del *Product Owner*. El *Product Owner* debe presentarse a la reunión preparado e informado. Solamente el equipo de trabajo debe decidir cuántos elementos del *Product Backlog* se va a desarrollar, ni el *Scrum Master* ni el *Product Owner* pueden tomar esta decisión.

Para iniciar la reunión de planificación del sprint, el *Product Backlog* debe estar totalmente actualizado por el *Product Owner*.

El artefacto generado por la planificación es el sprint backlog, que es un documento o tablero como el equipo decida. Preferiblemente se puede usar un tablero en el que indique las historias de usuario a desarrollar, puestas en orden de prioridad de negocio y técnico, también se muestran las tareas de cada historia de usuario y el responsable de cada tarea, las tareas también se ponen de acuerdo a una prioridad técnica. Acompañado del tablero puede ir una gráfica de avance, la cual es gestionada día a día por cada uno de los miembros del equipo, esta gráfica indica en cualquier momento la cantidad de trabajo restante y la cantidad de trabajo terminado.

Tabla 24: Planificación de sprint - SThinklet_001

IDENTIFICADOR:	ST_001
NOMBRE STHINKLET:	SThinklet_Definicion_Objetivo
ACTIVIDAD DE SCRUM:	Sprint planning
TAREA DE SCRUM	Definir el objetivo del sprint
PATRÓN DE COLABORACIÓN SUGERIDO:	Convergencia, Generación
THINKLETS ASOCIADO SUGERIDOS:	ExpertChoice, Plus/Minus/Interesting
ROLES INVOLUCRADOS:	Product Owner, Scrum Team
OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como propósito que el Product owner pueda acordar con el Scrum team cuál va a ser el objetivo del Sprint.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. El Product Owner propone alternativas para agregar valor al producto y realiza una breve socialización de cada una de ellas. 2. Una vez se presenten los posibles

	<p>objetivos del Sprint, el Product Owner debe conducir una votación para reducir el número de alternativas.</p> <ol style="list-style-type: none"> 3. De forma individual los participantes deben expresar las ventajas y desventajas de abordar cada propuesta en el Sprint, con el fin de determinar cuál de las propuestas es más viable de realizar y aporta mayor valor al producto. 4. Después del paso anterior, el Product owner dirige una nueva votación para seleccionar el objetivo del Sprint. 5. En caso de empate, el Product Owner decide el objetivo del Sprint (Entre las propuestas resultantes de la primer votación – Paso 2).
ARTEFACTOS DE ENTRADA SUGERIDOS:	Lista de alternativas para el objetivo del Sprint
ARTEFACTOS DE SALIDA SUGERIDOS:	Objetivo del Sprint
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	<p>Thinklets sugeridos:</p> <ul style="list-style-type: none"> • ExpertChoice: se usa de manera implícita, en el momento cuando el Product Owner dirige la votación para organizar y reducir las alternativas presentadas. • Plus/Minus/Interesting: se usa de manera implícita, cuando se solicita a los miembros del Scrum team que expresen las ventajas y desventajas de cada una de las alternativas de objetivo de Sprint presentadas por el Product Owner.

Tabla 25: Planificación de sprint - SThinklet_002

IDENTIFICADOR:	ST_002
NOMBRE STHINKLET:	SThinklet_Seleccion_Elementos_Sprint
ACTIVIDAD DE SCRUM	Sprint planning
TAREA DE SCRUM	Seleccionar elementos del product backlog para el Sprint actual
PATRÓN DE COLABORACIÓN SUGERIDO:	Consenso, Convergencia

THINKLETS ASOCIADO SUGERIDOS:	ExpertChoice , MoonRing
ROLES INVOLUCRADOS:	Scrum team
OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como objetivo que los miembros del Scrum team puedan seleccionar las historias de usuario y las tareas que se van a abordar en el sprint en concordancia con el objetivo del Sprint.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. Un miembro del Scrum team debe tomar el liderazgo de esta actividad, lo cual se realizará por turnos garantizando que todos los miembros participen como líderes de esta actividad al menos una vez. 2. El líder expresa de forma clara el objetivo del Sprint. 3. El líder nombrara uno por uno los elementos del Product Backlog. 4. Los miembros del Scrum team postulan los elementos del Product Backlog que consideran acordes al objetivo del Sprint levantando la mano o expresando claramente estar de acuerdo con incluir el elemento en el Sprint actual. 5. El líder conduce una votación para determinar si el elemento debe ser incluido, si la mayoría del Scrum team vota a favor se incluye en el Sprint actual. 6. Si los elementos incluidos rebasan la productividad del Scrum team, se debe someter a votación los elementos que fueron agregados al Sprint actual (de tal forma que sea acorde a la productividad). Los elementos restantes deberán regresar al Product backlog con mayor prioridad.
ARTEFACTOS DE ENTRADA SUGERIDOS:	Elementos del Product backlog, Objetivos del sprint
ARTEFACTOS DE SALIDA SUGERIDOS:	Sprint Backlog
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	<p>Thinklets sugeridos:</p> <ul style="list-style-type: none"> • ExpertChoice: se usa de manera implícita, en el momento cuando se designa un miembro del Scrum team para dirigir la votación. • MoonRing: se usa de manera implícita, en el momento que se discuten las

	actividades necesarias para el cumplimiento del objetivo del sprint
--	---

Tabla 26: Planificación de sprint - SThinklet_003

IDENTIFICADOR:	ST_003
NOMBRE STHINKLET:	SThinklet_Refinar_Elementos_Sprint
ACTIVIDAD DE SCRUM	Sprint planning
TAREA DE SCRUM	Refinar los elementos del Product Backlog o Sprint backlog si es necesario
PATRÓN DE COLABORACIÓN SUGERIDO:	Evaluación, Reducción, Generación, Consenso
THINKLETS ASOCIADO SUGERIDOS:	StrawPoll, MoodRing
ROLES INVOLUCRADOS:	Scrum Team
OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como objetivo que el Scrum tem pueda refinar el Product backlog, es decir, dividir y definir aún más los elementos del Product Backlog en elementos más pequeños y Precisos.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. El Scrum Team evalúa si es necesario refinar elementos del Product Backlog. 2. Los miembros del Scrum team indican los elementos del Product backlog que deben ser refinados. 3. Se revisa uno a uno los elementos seleccionados para ser refinados y se discute hasta qué grado deben ser divididos. 4. Se definen los elementos del Product backlog resultantes.
ARTEFACTOS DE ENTRADA SUGERIDOS:	Product backlog
ARTEFACTOS DE SALIDA SUGERIDOS:	Product backlog refinado
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	Thinklets sugeridos: <ul style="list-style-type: none"> • StrawPoll: se presenta de forma implícita al determinar que se deben refinar elementos del Product backlog.

	<ul style="list-style-type: none">• MoodRing: se presenta de forma implícita cuando los miembros del Scrum team dan su opinión y discuten acerca de los elementos del Product backlog.
--	--

6.4.2. Reuniones diarias

Durante el sprint se deben hacer las reuniones diarias. Las reuniones diarias son para realizar seguimiento y también para sincronizar las actividades de las personas del equipo, no tienen como objetivo informar al *Scrum Master* sobre el avance individual del sprint. Esta reunión no debe tener una duración mayor de 15 minutos. A esta reunión no es obligatoria la presencia del *Product Owner*, pero en caso de que se presente no debe responder a las preguntas, su presencia en esta reunión es pasiva [52].

¿Quién facilita la reunión? *Scrum Master* debe facilitar la reunión, pero se debe llegar a que en un momento el equipo por si solo realice la reunión, esto quiere decir que debe ser quien cuide de que la reunión no extienda su tiempo. Controlar cuando las personas hablan mucho o se salen del tema o hablen poco. Además, es quien escribe todos los inconvenientes que el *Equipo de desarrollo* informe en la reunión.

¿Cómo se hace? El equipo debe autoorganizarse para hacer la reunión diaria por eso es importante que las personas sepan en qué momento debe participar, se pueden establecer reglas como determinar que la última persona en llegar a la reunión es la primera que va a empezar a hablar y para decidir quien sigue se puede hacer la técnica del round robin, lanzar un objeto al azar, o pedirle a una persona que levante un número y ese número determina quién es el siguiente. El nivel de energía durante la reunión debe ser alto, para ello el volumen del habla y la distancia pueden ayudar. En lo posible hacer la reunión visualizando el tablero de tareas, para que cada persona hable de cada tarjeta que está en el tablero, de las novedades y actualice el tablero de Scrum, de esta forma logra una sincronización y la participación de todos. Al terminar la reunión se debe hacer en una nota alta para que el equipo termine motivado.

¿Qué puede suceder en esta reunión?

- Hay personas que van a querer solucionar los problemas apenas se comenten, esto se debe evitar y hacer énfasis en que se resolverán tan pronto termine la reunión.
- Demasiada proximidad puede causar incomodidad y estar muy separados puede inhibir la participación a aquellas personas que no les gusta hablar

en voz alta.

- Las reuniones de pie pueden afectar en las respuestas de las personas ya que van a querer responder de forma rápida para terminar la reunión.
- Si las reuniones no duran mucho, pararse todos los días puede convertirse en un ritual innecesario, sucede por lo general en grupos pequeños.
- Las personas pueden empezar a hablar y mirar solo al facilitador y no a los demás, para evitar esto el facilitador puede estar en constante movimiento o simplemente romper el contacto visual [70].
- No se debe esperar a la reunión diaria para informar los inconvenientes que no permitan avanzar al *Equipo de desarrollo*, estos inconvenientes deben ser informados al *Scrum Master* inmediatamente ocurran.
- Puede llegar un momento en que las personas ya no quieran hacer reuniones diarias o bajar la participación, se debe analizar los factores que están produciendo esas situaciones y tomar acciones para ello se debe hacer seguimiento de las reuniones para identificar posibles mejoras, una forma es medir los tiempos que toma cada reunión.

6.4.3. Revisión del sprint

Antes de realizar la revisión del sprint, se recomienda hacer una planificación previa. En esta preparación, se identifican cuáles elementos del Product Backlog son los que se van a presentar. Se prepara una pequeña demostración paso a paso y los datos necesarios para cada caso. De ser posible, preparar todo lo necesario en el computador en el que se va a realizar dicha presentación.

Un elemento del Product Backlog que no cumpla con la definición de hecho, es un elemento que no se terminó y por tanto no se tendrá en cuenta para presentar en la revisión del sprint. Para estos elementos se prepara una breve explicación del por qué no se logró terminar.

La reunión puede empezar informando a todos los asistentes cuáles elementos son los que se van a presentar y cuáles no, contando la razón por la cual no se van a presentar. Esto se hace con el fin de que se bajen las expectativas de la reunión desde el principio.

Los asistentes deben comprender que esta reunión no es para que el *Product Owner* o los interesados acepten el incremento construido. Se trata de poder brindar una retroalimentación del producto que se está mostrando, con el fin de enriquecerlo en conjunto con el equipo Scrum y los interesados. También, el equipo debe ir preparado para escuchar comentarios no tan positivos y no tomar nada personal. No se trata de entregar software funcionando porque si, se trata de hacer entregas que generen valor.

6.4.4. Retrospectiva del sprint

La retrospectiva es una oportunidad que tiene el equipo para examinar métodos, prácticas, herramientas, trabajo en equipo y proponer acciones de mejora. La retrospectiva se realiza después de la revisión del sprint y antes de la siguiente reunión de planificación de sprint [52]. No debe pasar mucho tiempo entre la revisión del sprint y la retrospectiva, ya que puede que el equipo no recuerde lo sucedido en el sprint y se puede perder información valiosa. La duración de la retrospectiva para un sprint de un mes que no sobrepase las tres horas y para sprints de menor duración se reserva un tiempo menor.

Antes de realizar la retrospectiva el equipo debe tener claro que no se trata de juzgar o encontrar culpables, se trata de encontrar oportunidades de mejora como equipo, para ello se recomienda que al iniciar la retrospectiva se lea la primera directiva de las retrospectivas, que es la siguiente:

“Independientemente de lo que descubramos, entendemos y verdaderamente creemos que todos hicieron el mejor trabajo posible, con base en lo que se sabía en el momento, los recursos disponibles y la situación dada.”

(Norm Kerth)

El *Scrum Master* es el responsable de que todos entiendan el propósito de la retrospectiva, de que todos tengan la oportunidad de participar y generar un ambiente positivo que motiva al equipo a mejorar.

El *Scrum Master* puede ser el moderador, aunque se sugiere que la responsabilidad sea rotativa entre el grupo Scrum. El moderador no debe tomar parte activa de una discusión y es el responsable de la dinámica de la reunión, debe asegurarse que las personas que tienen algo para decir, tengan la oportunidad de decirlo. También que no solo hable una persona, sino que se pueda escuchar otras opiniones y generar oportunidades de hablar para las personas que no participan.

La retrospectiva puede tener la siguiente estructura [71]:

- Preparar el escenario: lograr que las personas se focalicen en los objetivos de la reunión, en el tiempo estipulado y con una dinámica productiva.
- Recabar datos: lograr una visión común de la situación a analizar, tanto con datos objetivos como subjetivos. Se recomienda elaborar una línea de tiempo, de lo que sucedió en el sprint.
- Generar entendimiento profundo: entender el porqué, tanto de lo que estuvo mal como de lo que estuvo bien. Indagar, para encontrar las causas profundas que hay que mejorar o cambiar.

- Decidir qué hacer: teniendo una lista de posibles experimentos que el equipo puede realizar para mejorar, aquí se puede realizar una priorización en caso que se tenga una gran cantidad de acciones de mejora y tomar solo las que estén en la cima de la pila.
- Cierre: finalizar la retrospectiva enfatizando en los compromisos definidos y cerrar de forma energética.

Es importante que se realice seguimiento a la retrospectiva, para que no solo se quede en los compromisos. Para realizar el seguimiento se sugiere hacer un tablero tipo Scrum donde se especifique la acción de mejorar, el responsable, la fecha de compromiso, el resultado esperado y el resultado obtenido; además, en lo posible hacer un backup de las retrospectivas.

Muy a menudo las retrospectivas se vuelven rutinarias y las personas dejan de participar, para ello se recomienda variar la técnica de la retrospectiva, para no caer en la monotonía. También se debe tener en cuenta que se pueden generar conflictos personales, para evitar estos conflictos se recomienda que el equipo tenga en cuenta las pautas para una comunicación efectiva.

Tabla 27: Retrospectiva de Sprint - SThinklet_004

IDENTIFICADOR:	ST_004
NOMBRE STHINKLET:	SThinklet_Identificar_Mejoras
ACTIVIDAD DE SCRUM	Sprint Retrospective
TAREA DE SCRUM	Identificar aspectos a mejorar
PATRÓN DE COLABORACIÓN SUGERIDO:	Generación, Reducción, Evaluación, Consenso
THINKLETS ASOCIADO SUGERIDOS:	LeafHopper, DirectedBrainstorm, Plus/Minus/Interesting,
ROLES INVOLUCRADOS:	Scrum Team, Scrum Master, Product Owner
OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como objetivo que Scrum team, scrum master y product owner expongan aspectos que el equipo puede mejorar, con el fin de establecer acuerdos o estrategias de mejora.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. Un miembro del equipo indica un aspecto de mejora. Además, puede o no indicar la solución que crea conveniente. 2. Los miembros del equipo pueden aportar ideas de cómo lograr la mejora. 3. Los miembros del equipo pueden discutir las ventajas y desventajas de las propuestas de mejora.

	4. Después de debatir las propuestas, el equipo establece un acuerdo acerca de la mejora y como se debe implementar.
ARTEFACTOS DE ENTRADA SUGERIDOS:	Idea de mejora
ARTEFACTOS DE SALIDA SUGERIDOS:	Estrategia de mejora.
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	<p>Thinklets sugeridos:</p> <ul style="list-style-type: none"> • LeafHopper: se presenta de forma implícita al solicitar a los miembros del equipo expresar las ideas de mejora que el equipo puede implementar. • DirectedBrainstorm: se presenta de forma implícita cuando se genera una lluvia de ideas de cómo realizar una mejora. • Plus/Minus/Interesting: se presenta de forma implícita cuando el equipo debate cuál es la estrategia de mejora más viable o conveniente a implementar.

Tabla 28: Retrospectiva de Sprint - SThinklet_005

IDENTIFICADOR:	ST_005
NOMBRE STHINKLET:	SThinklet_Identificar_Bloqueos
ACTIVIDAD DE SCRUM	Identificar los obstáculos durante el sprint.
TAREA DE SCRUM	Sprint Retrospective
PATRÓN DE COLABORACIÓN SUGERIDO:	Generación, Reducción, Evaluación, Consenso, Clarificación, Organización,
THINKLETS ASOCIADO SUGERIDOS:	LeafHopper, DirectedBrainstorm, Plus/Minus/Interesting
ROLES INVOLUCRADOS:	Scrum team
OBJETIVO/PROPÓSITO:	Este STHINKLET tiene como objetivo que el Scrum team comparta los obstáculos presentados durante el sprint, sus causas y como fueron solucionados.
¿CÓMO USAR?	<ol style="list-style-type: none"> 1. Cada miembro del Scrum team expone los obstáculos que se presentaron durante el desarrollo de sus tareas. 2. Examinar las causas del obstáculo. 3. Compartir la solución del obstáculo, para que el equipo logre tener conocimiento compartido tanto del problema como de la solución.

ARTEFACTOS DE ENTRADA SUGERIDOS:	
ARTEFACTOS DE SALIDA SUGERIDOS:	Lista de obstáculos con sus respectivas soluciones
ASPECTOS DE LA INGENIERÍA DE LA COLABORACIÓN USADOS:	Thinklets sugeridos: <ul style="list-style-type: none"> • LeafHopper: se presenta de forma implícita al solicitar a los miembros del equipo expresar las ideas de mejora que el equipo puede implementar. • DirectedBrainstorm: se presenta de forma implícita cuando se genera una lluvia de ideas de cómo realizar una mejora. • Plus/Minus/Interesting: se presenta de forma implícita cuando el equipo debate cuál es la estrategia de mejora más viable o conveniente a implementar.

6.4.5. Refinamiento del product backlog

Es muy importante mantener el Product Backlog actualizado, de esto depende que el *Equipo de desarrollo* trabaje primero en lo más importante para el negocio y no realice trabajo que no es necesario, para ello se requiere que, durante el sprint, el *Product Owner* dedique aproximadamente un 10% de su carga, en refinar el Product Backlog. Él debe preocuparse por mantenerlo lo más actualizado posible, poniendo principal atención a las historias de usuario con mayor prioridad y para el refinamiento debe apoyarse en el *Equipo de desarrollo* para realizar la estimación de los elementos.

6.4.6. Velocidad del equipo scrum

Al finalizar el sprint es importante determinar la velocidad del equipo para tener en cuenta al realizar la planificación del siguiente sprint, esta velocidad se calcula de la siguiente forma:

$$\text{Velocidad} = \text{Trabajo realizado} / \text{Tiempo}$$

El trabajo realizado se mide en puntos, los cuales son los que inicialmente se estimaron a cada actividad al momento de realizar la planificación del sprint.

Así por ejemplo en un sprint que se planificó que en 2 semanas se construirían un total de 120 puntos, la velocidad ideal es:

$$\text{Velocidad ideal} = 120 / 2 \text{ semanas}$$

Pero al finalizar las 2 semanas solo se completaron 100 puntos entonces la velocidad = $100/2$ semanas.

Las unidades pueden cambiar, por ejemplo, en vez de semanas, usar días u horas laborables. Es importante que todos los sprints tengan la misma duración, así podrán ir comparando la velocidad obtenida en cada sprint y seleccionar los puntos que realmente se alcanzaran a realizar en un sprint, y evitar que al final del sprint quede mucho trabajo sin realizar.

CAPÍTULO 7

EVALUACIÓN DE LA GUÍA DE TRABAJO

En este capítulo se presenta el estudio de caso para la evaluación de la efectividad de la guía de trabajo para el fortalecimiento de prácticas colaborativas durante la adopción de Scrum aplicado en un grupo de desarrollo de software.

Para realizar la evaluación de la efectividad de la guía de trabajo, se mide el nivel de riesgo al iniciar el estudio de caso y al finalizar el estudio, con el fin de observar si se presenta un cambio en el nivel de riesgo al aplicar la guía. Además, se monitorea la percepción del riesgo asociado a cada aspecto colaborativo de cada uno de los integrantes del equipo Scrum a través de encuestas.

7.1. DISEÑO DEL ESTUDIO DE CASO

7.1.1. Objetivo del estudio

Evaluar la efectividad de la guía de trabajo para el fortalecimiento de prácticas colaborativas durante la adopción de Scrum en un equipo de desarrollo de software que provee soluciones a la medida.

7.1.2. Caso y unidad de análisis

Para la selección de la unidad de análisis se tuvo en cuenta que la empresa estuviera interesada en adoptar Scrum y que tuviera la disponibilidad de hacerlo. La unidad de análisis seleccionada por el criterio de disponibilidad y por ser un caso típico para la propuesta corresponde a un grupo de desarrollo de software. El grupo está compuesto por 5 personas, un líder y 4 desarrolladores que no tienen conocimiento de la metodología Scrum.

7.1.3. Pregunta de investigación

Este estudio de caso busca resolver la siguiente pregunta, ¿La guía de trabajo logra orientar a un equipo en el proceso de adopción de Scrum como marco de trabajo y fortalece las prácticas colaborativas combatiendo los principales problemas identificados en esta investigación?

7.2. PLANIFICACIÓN

7.2.1. Métodos de recolección de datos

Los métodos de recolección de datos en este estudio son de tipo directo a través

de encuestas definidas a partir de métricas cuantitativas y cualitativas. Las encuestas se llevaron a cabo durante un Sprint y fueron realizadas a cada miembro del equipo por el Scrum Master quien es la persona encargada de monitorear la guía de trabajo.

El propósito de la encuesta es que cada vez que se ejecute, brinde un diagnóstico del estado de la adopción con respecto a los aspectos colaborativos y permita realizar una comparación a medida que se efectúen más encuestas. La encuesta consta de 13 preguntas y se puede encontrar en el documento de anexos (Anexo 4).

7.2.2. Teoría del riesgo

Todas las actividades de las organizaciones implican riesgo. Los riesgos son eventos causados por incertidumbres, que pueden tener un efecto positivo o negativo en los objetivos del proyecto. Por lo tanto, la gestión de riesgos es una parte importante de un proyecto e implica identificar los posibles riesgos y analizar su potencial para responder y controlar las amenazas y oportunidades más significativas de los proyectos [73]

El análisis de riesgos es un proceso de evaluación que se realizó en dos etapas. Inicialmente, se utilizan métodos cualitativos para examinar, categorizar y determinar los principales problemas o riesgos asociados a la colaboración identificados, que son relevantes para la evaluación cuantitativa. El riesgo se define tradicionalmente como una función de probabilidad e impacto [74]. La probabilidad es la posibilidad de que ocurra un evento y el impacto es en qué medida el proyecto se ve afectado por un evento, las consecuencias. Al combinar la probabilidad y el impacto, se puede determinar el Nivel de Riesgo.

Un método comúnmente utilizado para la evaluación de riesgos es la Matriz de impacto y probabilidad que puede tomar valores tanto cualitativos como numéricos. La Matriz de Impacto y Probabilidad es un método simple y fácil de entender para priorizar riesgos y asignar recursos.

Probabilidad

La probabilidad de riesgo, o probabilidad, es la posibilidad de que ocurra un evento de riesgo. La probabilidad se puede expresar tanto de manera cualitativa como cuantitativa. Cuando se habla de probabilidad de manera cualitativa, se utilizan términos como frecuente, posible, raro, etc. También es posible describir la probabilidad de forma numérica. Esto se puede hacer utilizando puntuaciones, porcentajes y frecuencias definidas por las organizaciones en función de la descripción relativa [74].

Impacto

Los impactos a menudo se definen como las consecuencias o los efectos de un

evento de riesgo en los objetivos del proyecto. Estos impactos pueden ser tanto beneficiosos como perjudiciales para los objetivos [74].

La escala de Impacto puede variar, pero la escala más común es la escala de cinco puntos. Por lo general, los impactos se describen de forma relativa; como muy bajo, bajo, moderado, alto y muy alto, pero a menudo también se define mediante escalas numéricas. Dependiendo del objetivo, las escalas reciben una descripción de lo que implica el impacto [75].

Matriz de probabilidad e impacto

La Matriz de Probabilidad e Impacto es uno de los métodos de evaluación cualitativa más utilizados. Se basa en los dos componentes del riesgo, la probabilidad de ocurrencia y el impacto en el(los) objetivo(s) si ocurre. La matriz es una cuadrícula bidimensional que mapea la probabilidad de ocurrencia de los riesgos y su efecto sobre los objetivos del proyecto [75]. La puntuación de riesgo, a menudo denominada nivel de riesgo o grado de riesgo, se calcula multiplicando los dos ejes de la matriz.

$$\text{Riesgo} = \text{Impacto} \times \text{Probabilidad}$$

Las calificaciones de probabilidad e impacto se realizan utilizando opiniones recopiladas de entrevistas [73]. El resultado de estas matrices de riesgo se utiliza para priorizar los riesgos, planificar la respuesta al riesgo, identificar los riesgos para la evaluación cuantitativa y orientar las asignaciones de recursos [76].

Tabla 29: Escala de probabilidad

Escala	Valor	Descripción
Muy alta	5	Ocurre siempre
Alta	4	Ocurre casi siempre, muy frecuente
Media	3	Ocurre en alguna ocasiones
Baja	2	Ocurre rara vez
Muy baja	1	Casi nunca ocurre

Tabla 30: Escala de impacto

Escala	Valor	Descripción
Muy alto	5	Impacto muy significativo sobre la adopción
Alto	4	Impacto significativo sobre la adopción

Medio	3	Algún impacto sobre un elemento clave de la adopción
Bajo	2	Impacto menor sobre la adopción
Muy bajo	1	Impacto leve sobre la adopción

Mapa de calor

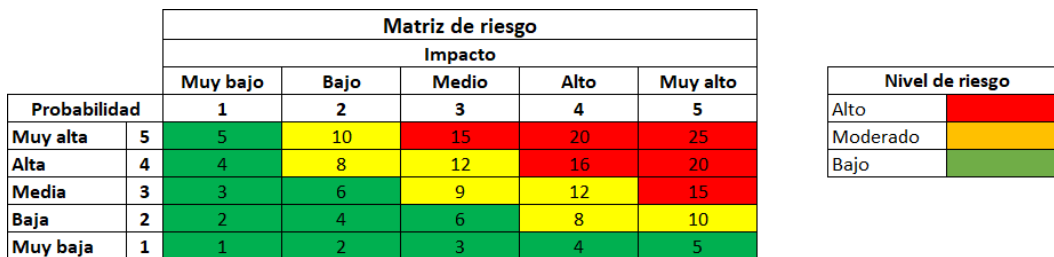


Figura 6: Mapa de calor

7.2.3. Protocolo de estudio de caso

Al iniciar el estudio de caso el grupo seguía una metodología de desarrollo tradicional, el grupo trabaja 8 horas diarias y no se encuentran en el mismo espacio físico debido a la pandemia de Covid-19. Se realiza la capacitación de Scrum y la guía de trabajo es entregada a la persona que desempeña el rol de Scrum Master del equipo de trabajo. La recolección de datos por parte de los investigadores se realiza al finalizar el primer Sprint de Scrum. Durante el proceso se hizo un acompañamiento y se tomó captura de algunas reuniones. Los artefactos generados se almacenaron en OneDrive y en la herramienta taiga.io [78].

7.2.4. Consideraciones éticas.

Los artefactos de la metodología e información sobre el proyecto que se está trabajando están bajo un acuerdo de confidencialidad verbal.

7.3. EJECUCIÓN DEL CASO Y RECOLECCIÓN DE DATOS

Los miembros del equipo empezaron a tener interés por las metodologías ágiles, ya que, es ampliamente utilizado a nivel regional. El caso se inició realizando una presentación general de 4 horas en donde se abordó metodologías ágiles, enfocada en Scrum, a esta reunión asistieron todos los miembros del equipo, esta presentación se realizó de forma dinámica donde se presentó parte teoría combinado con actividades. Posteriormente, se llevó a cabo una capacitación con una duración de 4 horas, en donde se trató la colaboración como uno de los pilares fundamentales de la metodología Scrum incluyendo juegos ágiles y

actividades de grupo, y enfocada en el funcionamiento del ciclo de Scrum.

Posterior a la capacitación se realizó la entrega de la guía de trabajo al líder del grupo. En la figura 6 se muestra al equipo recibiendo la capacitación sobre los conceptos básicos de Scrum.

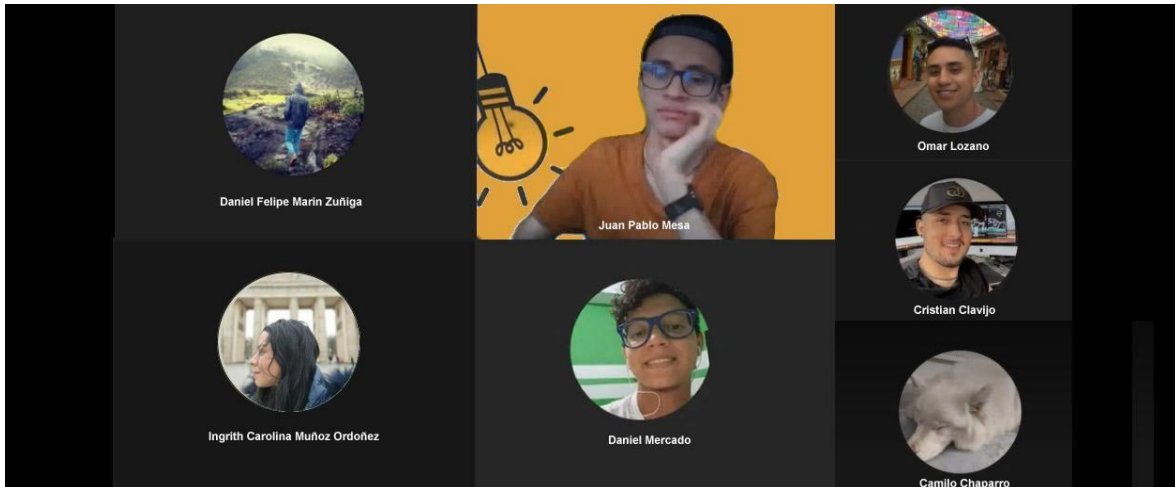


Figura 7: Equipo de desarrollo en la capacitación.

El grupo de desarrollo está compuesto por cinco personas, de las cuales una persona cumple la función de Líder y los otros desarrolladores de rango junior. El grupo sigue el proceso de desarrollo con un modelo tradicional el cual consiste en evaluación, diseño, construcción y pruebas.

Al iniciar el nuevo proyecto, se realizaron reuniones a fin de construir la primera versión del Product Backlog, se solucionan dudas con el cliente, se definió el tamaño del sprint (2 semanas) y la definición de hecho.

El equipo mediante una lluvia de ideas propuso quienes podrían asumir el rol de Scrum Master y Product Owner, la postulación se realizó indicando las habilidades y explicando porque se debería designar a la persona. Todos los miembros del equipo debían postular a las personas que consideran idóneas para estos roles, y quienes fueron seleccionados por medio de una votación.

Una vez se seleccionó al Scrum master le fue entregada la guía de trabajo para el fortalecimiento de prácticas colaborativas durante la adopción de Scrum. Además, como encargado del proceso Scrum se brindó capacitación acerca de la guía y se le solicitó diligenciar la matriz de riesgo con el fin de obtener un diagnóstico del riesgo asociado al factor colaboración previo al uso de la guía. Adicionalmente, se le hizo entrega de la encuesta para determinar la frecuencia de los riesgos que debía realizar a cada miembro del equipo. El Scrum master realizó esta encuesta cada 3 días, dando un total de 5 veces por Sprint.

El Product Owner se encargó de solucionar dudas al equipo sobre el producto. El Scrum Master estuvo representado por el líder del grupo y el Equipo de desarrollo estuvo conformado por 4 personas, el Scrum Master participó activamente en el desarrollo.

El *Equipo Scrum* empezó el primer sprint del proyecto asignado, realizando el Sprint planning, el Product Owner propuso y explicó alternativas de objetivos del Sprint, el equipo discutió y definió el objetivo del Sprint por votación. Además, se seleccionaron las historias de usuario y tareas acordes al objetivo seleccionado a realizar durante el Sprint.

Durante todo el experimento se realizaron reuniones diarias, siempre a las 8:15 am utilizando google meet, en la figura 16 se muestra al equipo realizando una reunión diaria. El equipo llevó el progreso en un tablero de Scrum en taiga.io, en la figura 7 se muestra el tablero Scrum al término del primer Sprint. También utilizaron taiga.io y Excel para el product backlog, documentación que eligieron para el proyecto. El avance del tablero Scrum se registró en su mayoría durante las reuniones diarias.

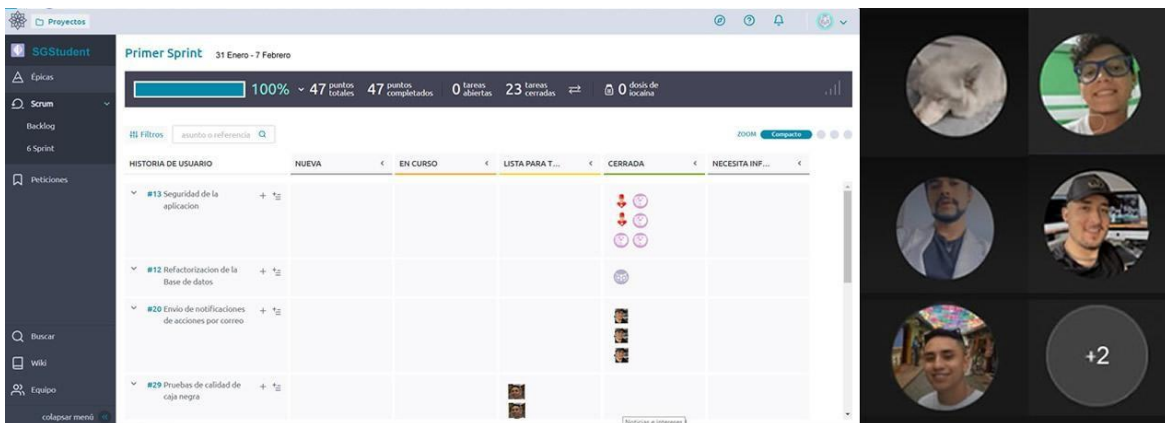


Figura 8: Reunión diaria equipo

Al término de las dos semanas de duración del sprint se realiza la Sprint review con duración de 2 horas, en donde se presenta la demo, se explica la funcionalidad desarrollada durante el Sprint y se discute con los interesados sobre lo que se abordará en el siguiente Sprint. Posteriormente, se realiza la Sprint Retrospective de 1 hora en donde el Scrum Team analizó qué salió bien durante el Sprint, qué problemas encontró y cómo se resolvieron (o no) esos problemas, durante esta reunión el Scrum Team destacó la colaboración y apoyo que se dio entre todos.

Cabe mencionar que el día previo a los eventos y reuniones el Scrum Master seleccionaba los elementos de la Ingeniería de la Colaboración a implementar al día siguiente.

Finalmente, se le solicitó al Scrum master diligenciar la matriz de riesgo con el fin

de obtener un diagnóstico del riesgo asociado al factor colaboración posterior al uso de la guía.

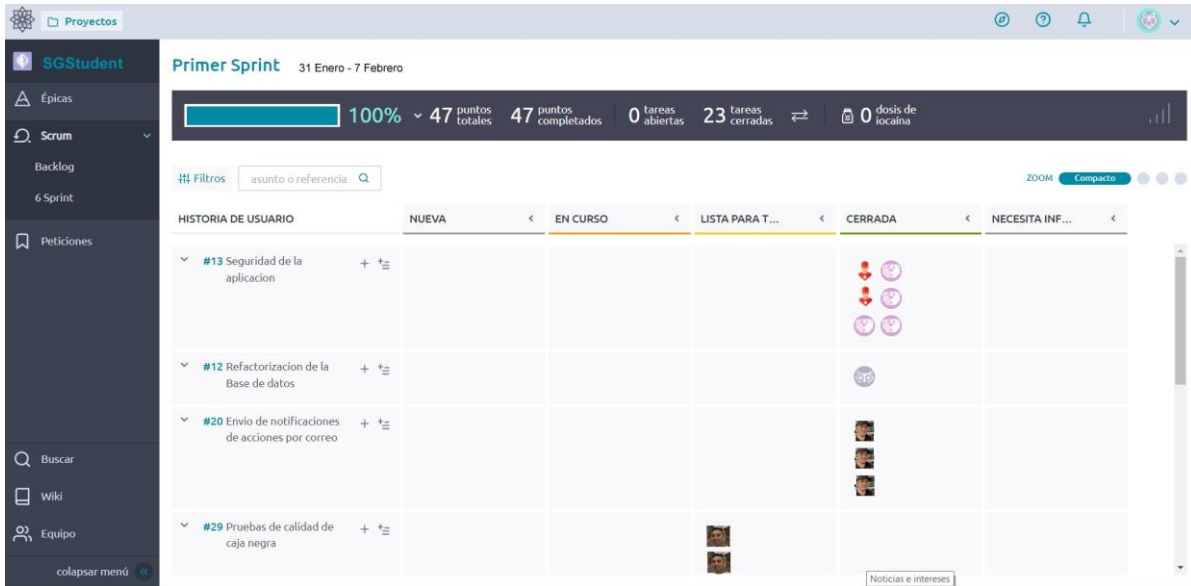


Figura 9: Tablero de Scrum - Sprint 1

7.4. RESULTADOS

Al concluir el primer Sprint del proyecto se dio por finalizado el estudio de caso, se le pidió al equipo que respondieron las encuestas preparadas y se procedió a analizar los resultados. Las encuestas fueron respondidas por el Scrum Master y el Equipo de desarrollo.

Matriz de riesgo estudio preliminar

La matriz de riesgo presentada a continuación consiste en un diagnóstico del riesgo asociado al factor colaboración previo a utilizar la guía. Los valores de probabilidad se obtuvieron al aplicar una encuesta al equipo y en cuanto al impacto, se obtuvo a partir de la literatura y la metodología.

Factor de riesgo	Impacto	Probabilidad	Nivel de riesgo
El pensamiento de trabajo individual.	Alto	Media	Riesgo moderado
Restar importancia al propósito de cada evento Scrum (falta de conciencia)	Muy alto	Media	Riesgo alto
Falta de participación o ausencia durante las sesiones	Alto	Baja	Riesgo moderado
El factor persona (personalidad, cultura, creencias, etc) puede afectar el proceso de adopción	Muy alto	Muy alta	Riesgo alto
Ausencia de la propiedad colectiva de la información.	Medio	Media	Riesgo moderado
Ausencia de una comunicación efectiva.	Alto	Baja	Riesgo moderado
Falta de cooperación y comunicación constante con el cliente	Alto	Alta	Riesgo alto
Falta de disciplina en las reuniones diarias	Alto	Muy baja	Riesgo bajo
Falta de comprensión de los valores ágiles	Muy alto	Media	Riesgo alto
Falta de comprensión de los objetivos de las reuniones.	Alto	Media	Riesgo moderado
Falta de comunicación entre los integrantes del equipo	Alto	Muy baja	Riesgo bajo

Figura 10: Matriz de riesgo preliminar

Matriz de riesgo final

La siguiente matriz de riesgo consiste en los valores obtenidos al realizar la misma encuesta (de la matriz preliminar) al terminar el primer Sprint. No obstante, esta vez, el equipo reportó diferentes valores en la probabilidad.

Factor de riesgo	Impacto	Probabilidad	Nivel de riesgo
El pensamiento de trabajo individual.	Alto	Baja	Riesgo moderado
Restar importancia al propósito de cada evento Scrum (falta de conciencia)	Muy alto	Media	Riesgo alto
Falta de participación o ausencia durante las sesiones	Alto	Baja	Riesgo moderado
El factor persona (personalidad, cultura, creencias, etc) puede afectar el proceso de adopción	Muy alto	Alta	Riesgo alto
Ausencia de la propiedad colectiva de la información.	Medio	Baja	Riesgo bajo
Ausencia de una comunicación efectiva.	Alto	Baja	Riesgo moderado
Falta de cooperación y comunicación constante con el cliente	Alto	Media	Riesgo moderado
Falta de disciplina en las reuniones diarias	Alto	Muy baja	Riesgo bajo
Falta de comprensión de los valores ágiles	Muy alto	Baja	Riesgo moderado
Falta de comprensión de los objetivos de las reuniones.	Alto	Baja	Riesgo moderado
Falta de comunicación entre los integrantes del equipo	Alto	Muy baja	Riesgo bajo

Figura 11: Matriz de riesgo final

Al realizar el estudio preliminar se identificaron 4 factores de riesgo con nivel de riesgo alto, 5 factores de riesgo con nivel de riesgo moderado y 2 factores de riesgo con nivel de riesgo bajo. En cuanto a la matriz de riesgo obtenida al final del Sprint se identificaron 2 factores de riesgo con nivel de riesgo alto, 6 factores de riesgo con nivel de riesgo moderado y 3 factores de riesgo con nivel de riesgo bajo.

Nivel de riesgo	Matriz de riesgo preliminar	Matriz de riesgo final
Alto	4	2
Moderado	5	6
Bajo	2	3

Figura 12: Comparativa de matrices

7.5. ANÁLISIS DE DATOS

La guía de trabajo ayudó a reducir el nivel riesgo de los problemas o factores de riesgo identificados durante este estudio, aunque se evidencian resultados favorables es importante reconocer que aún hay margen de mejora durante el proceso de adopción.

Pese a que el nivel de riesgo del factor “El pensamiento de trabajo individual” se mantuvo, el equipo comprendió la importancia del trabajo en equipo lo que generó

que la probabilidad de ocurrencia disminuyera.

Al igual que el anterior factor de riesgo, el factor “El factor persona (personalidad, cultura, creencias, etc) puede afectar el proceso de adopción” mantuvo el nivel de riesgo. No obstante, la probabilidad de ocurrencia es baja, debido a que los miembros del equipo comprendieron que sus acciones pueden afectar de forma positiva/negativa el proceso.

En cuanto al factor de riesgo “Ausencia de la propiedad colectiva de la información” el nivel de riesgo asociado bajo de “moderado” a “bajo”. Ya que, el equipo comprendió la importancia de conocer las actividades y el estado del proceso de desarrollo, admitiendo que el trabajo individual es una contribución para lograr un objetivo mayor.

El equipo entendió que la comunicación con el cliente es importante e intentó involucrar más al cliente. Sin embargo, el cliente no aumentó su participación durante el proceso.

Pese a que el nivel de riesgo de “Falta de disciplina en las reuniones diarias” se mantuvo y no se presentaron cambios en su probabilidad de ocurrencia. Cabe resaltar que al inicio la probabilidad de ocurrencia fue “muy baja” debido a que las reuniones diarias se desarrollaron de forma obligatoria. No obstante, los miembros del equipo manifestaron comprender y estar de acuerdo con el objetivo de las mismas.

La disposición del equipo durante el proceso de adopción contribuyó a que aumentara el nivel de comprensión e interiorización de los valores y principios ágiles, logrando que el factor de riesgo “Falta de comprensión de los valores ágiles” cambiara a nivel de riesgo moderado.

Aunque aumentó la comprensión de los objetivos de las reuniones, el equipo aún no logra un entendimiento compartido sobre ellas. Algunos miembros aún desconocen el propósito de algunas reuniones.

Algunos niveles de riesgo no se vieron afectados, debido a, que el equipo se conocía y tenía una dinámica de comunicación que prevaleció a lo largo del Sprint. Además, la comprensión del proceso irá aumentando a medida que los integrantes del equipo entiendan el valor y el propósito de cada una de las actividades del proceso Scrum.

7.6. LECCIONES APRENDIDAS

- Capacitarse en la metodología Scrum no garantiza una buena adopción.

- Fue evidente que la motivación del equipo juega un papel fundamental en la adopción de esta metodología.
- El papel del Scrum Master en la motivación y en la elección de los elementos de colaboración sugeridos fue vital para que el equipo continuará con el proceso de adopción.
- El cliente demuestra mayor interés en participar después de observar un avance funcional del producto.

CAPÍTULO 8

CONCLUSIONES Y TRABAJOS FUTUROS

8.1. RESUMEN

En este proyecto se generó y evaluó una guía de trabajo para el fortalecimiento de prácticas colaborativas durante la adopción de Scrum, incorporando elementos de la ingeniería de la Colaboración que conllevo a la disminución de los riesgos asociados a la colaboración durante la adopción.

Inicialmente se identificaron los factores problema asociados a la colaboración que afectan la adopción de Scrum. Para ello primero se estudió la literatura, después se realizaron entrevistas semiestructuradas a líderes de equipos adoptantes de Scrum para identificar los principales riesgos asociados a la adopción de Scrum. Luego, se realizó un estudio de caso en un equipo de desarrollo de software y finalmente se realizaron encuestas para determinar si al aplicar la guía de trabajo el riesgo disminuyó.

8.2. CONCLUSIONES

Un Sprint es un periodo de tiempo corto para que el equipo logre disminuir el nivel de riesgo de todos los factores de riesgo presentes durante una adopción.

Para responder a la pregunta de investigación ¿Cómo entender y fortalecer los aspectos colaborativos (comunicación, interacción y conciencia) facilitadores de la adopción de Scrum en un equipo de desarrollo de software?, realizada en este trabajo, se formuló como respuesta la guía de trabajo la cual sugiere incorporar elementos de la Ingeniería de la Colaboración, específicamente Thinklets en el proceso de adopción de la metodología Scrum, con el fin de mejorar la colaboración y mitigar el riesgo asociado.

Que el equipo de trabajo se conociera previamente a la implementación de la metodología facilitó las actividades de comunicación e interacción entre ellos. Sin embargo, en el caso de estudio preliminar se observó que el factor persona (personalidad, cultura, creencias, etc) puede llegar a impactar directamente en el desarrollo de las actividades, ya que, se puede afectar la comunicación.

Actualmente existen herramientas como Trello, Bitbucket, Azure DevOps, Taiga entre otros, que permiten gestionar la información correspondiente al proyecto de

manera transparente a todos los integrantes del equipo. El uso de estas herramientas facilita el acceso a la información del proyecto, lo que puede contribuir a aumentar el nivel de conciencia del equipo, aunque no lo garantiza.

Existen herramientas como Skype, Google, correos electrónicos, incluso Whatsapp, que permiten una comunicación constante entre los integrantes del equipo. Sin embargo, éstas no ofrecen una guía para gestionar las comunicaciones, lo que sumado al factor persona puede llegar a generar problemas de comunicación, a pesar de que se tenga el canal para hacerlo.

La transición de una metodología tradicional a una ágil puede generar miedo e incertidumbre al equipo de desarrollo, ya que, muchos de los artefactos (documentación) pasan a un segundo plano en las metodologías Ágiles.

Los equipos ven necesaria la capacitación de Scrum de sus integrantes para una mejor adopción de la metodología. Sin embargo, la adopción debe ser una decisión voluntaria y consciente de interiorizar no sólo el proceso y la metodología Scrum, sino también los valores y principios ágiles.

La comunicación juega un papel muy importante para disminuir el riesgo del factor persona, una comunicación efectiva puede llevar a un mejor entendimiento de los integrantes del equipo y puede llevar a una mejor cohesión.

El Scrum master notó que se presentó un incremento en la participación de los miembros del equipo en las diversas reuniones, debido a, las recomendaciones para la participación brindadas en la guía.

Los integrantes del equipo tuvieron gran nivel de aceptación de las recomendaciones realizadas, ya que, el Scrum master no se vio en la necesidad de obligar a ningún miembro del equipo a participar.

Las encuestas realizadas a los miembros del equipo permiten recolectar la percepción de todos los miembros y realizar un diagnóstico de los riesgos que se presentan en el momento de su realización.

Realizar la medición de la adopción con base a los aspectos colaborativos, permite al Scrum master monitorear el estado de la adopción e implementar las acciones necesarias para mitigar los riesgos asociados a la Colaboración.

Aunque se evidencian resultados favorables con respecto a la implementación de la guía, es importante reconocer que aún hay margen de mejora durante el

proceso de adopción.

La adopción de metodología Scrum implica interiorizar y hacer propios los valores y principios ágiles. Por lo tanto, la propuesta realizada en este trabajo puede servir de apoyo para lograr mejorar la adopción, específicamente permitiendo el fortalecimiento de la colaboración.

8.3. TRABAJOS FUTUROS

- Adicionar prácticas para fortalecer valores y principios ágiles.
- Ampliar la guía para incluir elementos de la ingeniería de la Colaboración teniendo en cuenta la localización del equipo. Puesto que, la ubicación (remota o física) puede ocasionar diferencias en los resultados de la aplicación de las prácticas sugeridas en esta guía.
- Adoptar y aplicar la guía de trabajo en pequeñas organizaciones que no pertenezcan a la industria del software, ya que, Scrum es un marco de trabajo que no va únicamente relacionado al desarrollo.
- Generar herramientas para involucrar al cliente en el proceso Scrum, para generar conciencia del proceso y entienda la importancia de su rol.
- Adoptar y aplicar la guía de trabajo en grandes organizaciones.
- Adaptar la guía para Scrum de Scrum, es decir, adopción de Scrum por parte de un equipo grande que se divide en varios equipos Scrum.

8.4. LECCIONES APRENDIDAS

- Dado que las metodologías ágiles se basan en las personas y no en los procesos, los aspectos humanos y colaborativos, influyen directamente en la adopción, ya que, por ejemplo, tener un equipo motivado conlleva a que de alguna manera las personas sean más felices trabajando y estén más dispuestos a la adopción de nuevas metodologías y/o herramientas.
- La ingeniería de la Colaboración brinda herramientas importantes que pueden generar el comportamiento esperado en algunas situaciones, por lo que es de gran ayuda en actividades que involucran a varias personas.
- La alta cohesión, motivación y disposición del equipo juega un papel muy importante durante la adopción de una metodología.
- Pese a que la participación activa del cliente es importante, esta participación se encuentra restringida al tiempo y disponibilidad del cliente.

8.5. PARTICIPACIONES

- Se socializó la idea de trabajo en universidades alemanas. Noviembre de 2021.

REFERENCIAS

- [1] G. Papadopoulos, «Moving from traditional to agile software development methodologies also on large, distributed projects,» *Procedia - Social and Behavioral Sciences.*, vol. 175, pp. 455-463, 2015.
- [2] C. Rodríguez y R. Dorado, «¿Por qué implementar Scrum?,» *Revista Ontare*, vol. 3, pp. 125-144, 2015.
- [3] Schwaber, *SCRUM Development Process*, London: Springer, London, 1997.
- [4] J. D. Yepes, C. Pardo y O. Gómez, «Revisión sistemática acerca de la implementación de metodologías ágiles y otros modelos en micro, pequeñas y medianas empresas de software,» *Revista Tecnológica ESPOL*, vol. 28, nº 5, pp. 464-479, 2015.
- [5] VersionOne, «VersionOne 10th Annual State of Agile Report,» *VersionOne*, vol. 6, nº 2, pp. 2-8, 2006.
- [6] L. V. Araujo y A. N. Castrillon, «Una ruta de trabajo para la adopción de Scrum en pequeñas organizaciones en la industria del software,» Universidad del Cauca, Popayán, Cauca, 2017.
- [7] . A. Ali, . M. Rehman y M. Anjum, «Framework for Applicability of Agile Scrum Methodology: A Perspective of Software Industry,» *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, nº 9, 2017.
- [8] N. Khan, N. Ikram y S. Imtiaz, «SCRUM Adoption: A Solution to Backlog Problems,» de *Fifth Asian Conference on Information Systems*, At Krabi, Thailand, 2016.
- [9] J. López, R. Juárez, C. Huertas, S. Jiménez y C. Guerra, «Problems in the Adoption of Agile-Scrum Methodologies: A Systematic Literature Review,» de *2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)*, Puebla, México, 2016.
- [10] R. Mantovani, I. Mantovani, P. A. Da Rosa, S. Reinehr y A. Malucelli, «Processes versus people: How should agile software development maturity be defined?,» *Journal of Systems and Software*, vol. 97, pp. 140-155, 2014.
- [11] R. Owen Briggs, G. Kolschoten, G.-J. De Vreede y D. L. Dean, «Defining Key Concepts for Collaboration Engineering,» de *Americas Conference on Information Systems Proceedings*, Acapulco, Mexico, 2006.
- [12] J. Whitehead, «Collaboration in Software Engineering: A Roadmap,» *IEEE Computer Society Washington, DC, USA*, pp. 214-225, 2007.
- [13] D. Mishra, A. Mishra y S. Ostrovska, «Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation,» *Information and Software Technology*, vol. 54, nº 10, p. 1067–1078, 2012.
- [14] J. Heerwagen, K. Kampschroer, K. Powell y V. Loftness, «Collaborative knowledge work environments,» *BUILDING RESEARCH & INFORMATION*, vol. 32, nº 6, p. 510–528, 2004.

-
- [15] A. Kanane, «Challenges related to the adoption of Scrum. Case study of a financial IT company,» UMEA University, Department of informatics. IT management master program, Umea, Suecia, 2014.
 - [16] M. Noor, R. Rabiser y P. Grünbacher, «Agile product line planning: a collaborative approach and a case study,» *Journal of Systems and Software*, vol. 81, nº 6, pp. 868-882, 2008.
 - [17] C. Restrepo, L. Jiménez y J. A. Hurtado, «Integrating the Collaboration Engineering with Software Process Models: A Visual Approach,» de *Interacción '17 Proceedings of the XVIII International Conference on Human Computer Interaction*, Cancun, Mexico, 2017.
 - [18] J. A. Hurtado, «Toward a Scientific Method in Software Engineering (Position Paper),» Universidad del Cauca, Popayán, Colombia.
 - [19] P. Runeson y M. Höst, «Guidelines for conducting and reporting case study research in software engineering,» *Empirical Software Engineering*, vol. 14, nº 2, pp. 131 - 164, 2009.
 - [20] C. G. Basili Victor, «Goal Question Metric paradigm,» *Encycl. Softw. Eng.* vol. 2, no. 1-54004–8, 1994.
 - [21] J. Vlietland y H. Van Vliet, «Towards a governance framework for chains of Scrum teams,» *Information and Software Technology*, vol. 57, nº 1, pp. 52-65, 2014.
 - [22] C. Collazos, S. Ochoa y J. Mendoza, «Collaborative evaluation as a mechanism for improving evaluation of classroom learning,» *Ingeniería e Investigación*, vol. 27, nº 2, pp. 72-76, 2007.
 - [23] G. Kolfschoten, «Theoretical foundations for collaboration engineering,» Delft University of Technology, Delft, Zuid-Holland, Netherlands, 2007.
 - [24] R. Úbeda, «Métodos ágiles para el desarrollo de software,» Universidad Politécnica de Cataluña, Barcelona, España, 2009.
 - [25] S. Werner Knoll, M. Hörning y G. Horton, «Applying a ThinkLet and ThinXel based Group Process Modeling Language: A Prototype of a Universal Group Support System,» de *Proceedings of the 42nd Hawaii International Conference on System Sciences*, Hawai, 2009.
 - [26] D. W. Johnson, T. Johnson y M. B. Stanne, «Cooperative learning methods: A meta-analysis,» University of Minnesota., Minneapolis, 2000.
 - [27] R. E. Slavin, «Cooperative Learning,» *Review of educational research*, vol. 50, nº 2, pp. 315-342, 1980.
 - [28] D. W. Johnson y R. T. Johnson, «Learning together and alone: Overview and meta-analysis,» *Asia Pacific Journal of Education* , vol. 22, nº 1, pp. 95-105, 2002.
 - [29] Z. Jianhua y K. Akahori, «Web-based collaborative learning methods and strategies in higher education,» de *Proceedings of 2nd International Conference on Information Technology Based Higher Education and Training*, Kumamoto, Japón, 2001.
 - [30] Y. Sharan y S. Sharan, *Expanding cooperative learning through group investigation*, Nueva York, Estados Unidos: Teachrs College Press, 1992.

-
- [31] D. Tjosvold, «Constructive controversy for management education: Developing committed, open-minded researchers,» *Academy of Management Learning & Education*, vol. 7, nº 1, pp. 73-85, 2008.
 - [32] F. Pozzi, «Using Jigsaw and Case Study for supporting online collaborative learning,» *Computers & Education*, vol. 55, nº 1, pp. 67-75, 2010.
 - [33] E. G. Cohen, R. Lotan, B. A. Scarloss y A. R. Arellano, «Complex Instruction: Equity in Cooperative Learning Classrooms,» *Theory Into Practice - THEORY PRACT*, vol. 38, nº 2, pp. 80-86, 1999.
 - [34] Dotson, Jeanie M., «Cooperative learning structures can increase student achievement,» Kagan Publishing, Kagan Online Magazine, San Clemente, Estados Unidos, 2001.
 - [35] M. Karnes, D. Collins, L. Maheady y G. F. Harper, «Using cooperative learning strategies to improve literacy skills in social studies,» *Reading and Writing Quarterly*, vol. 13, nº 1, pp. 37-51, 1997.
 - [36] F. Lanubile, C. Ebert, R. Prikadnicki y A. Vizcaíno, «Collaboration tools for global software engineering,» *IEEE Software*, vol. 27, nº 2, pp. 52 - 55, 2010.
 - [37] M. Perkusich, H. Oliveira de Almeida y A. Perkusich, «A Model to Detect Problems on Scrum-based Software Development Projects,» de *Proceeding SAC '13 Proceedings of the 28th Annual ACM Symposium on Applied Computing*, Coimbra, Portugal, 2013.
 - [38] M. T. Hansen y H. Baggesen, «From CMMI and isolation to Scrum, Agile, Lean and collaboration,» de *2009 Agile Conference*, Chicago, Estados Unidos, 2009.
 - [39] I. Inayat y S. S. Salim, «A framework to study requirements-driven collaboration among agile teams: Findings from two case studies,» *Computers in Human Behavior*, vol. 51, pp. 1367-1379, 2015.
 - [40] M. Marques y S. F. Ochoa, «Improving Teamwork in Students Software Projects,» de *2014 IEEE 27th Conference on Software Engineering Education and Training (CSEE&T)*, Klagenfurt, Austria, 2014.
 - [41] C. Mayr-Dorn y C. Laaber, «A Domain-Specific Language for Coordinating Collaboration,» de *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, Vienna, Austria, 2017.
 - [42] I. Slijivar y A. Gunasekaran, «Agile-Scrum for Facility Design Project Management,» de *SPE Western Regional Meeting*, California, Estados Unidos, 2018.
 - [43] V. S. Sharma y V. Kaulgud, «An Automated Contextual Collaboration approach for Distributed Agile Delivery,» de *2015 30th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW)*, Lincoln, NE, Estados Unidos, 2015.
 - [44] K. H. Judy y I. Krumsins-Beens, «Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership,» de *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, Waikoloa, HI, Estados Unidos, 2008.

-
- [45] J. Vlietland, R. v. Solingen y H. v. Vliet, «Aligning codependent Scrum teams to enable fast business value delivery: A governance framework and set of intervention actions,» *Journal of Systems and Software*, vol. 113, pp. 418-429, 2015.
- [46] C. J. Pardo, D. Viveros, R. Chilito y F. J. Pino, «Scrum+: An agile guide for the global software development (GSD) multi-model project management,» de IEEE Colombian Conference On Communications and Computing (COLCOM), Medellín, Colombia, 2018.
- [47] G.-J. de Vreede, «Two Case Studies of Achieving Repeatable Team Performance through Collaboration Engineering,» *MIS Quarterly Executive*, vol. 13, nº 2, pp. 115-129, 2014.
- [48] T. Tan, A. J. Vicente y A. R. Yu, «Collaborative approach in software engineering education: an interdisciplinary case,» *Journal of Information Technology Education: Innovations in Practice*, vol. 17, nº 1, pp. 127-152, 2018.
- [49] A. Sandberg y I. Crnkovic, «Meeting industry - Academia research collaboration challenges with agiles methodologies,» de 2017 IEEE/ACM 39th International Conference on Engineering: Software Engineering in Practice Track (ICSE-SEIP), Buenos Aires, Argentina, 2017. Software
- [50] J. A. Hurtado y C. Collazos, «Estudios de Caso en la Determinación y evaluación de procesos colaborativos en el contexto del desarrollo de software,» IDIS Research Group - Universidad del Cauca, Popayán, Colombia.
- [51] D. E. L. Proyecto and J. M. Sell, «Universitat Politècnica de Catalunya Proyecto Final de Carrera Métodos ágiles para el desarrollo de software,» Management, 2009.
- [52] K. Schwaber and J. Sutherland, «La Guía de Scrum,» OOPSLA. pp. 1–21, 2013.
- [53] A. N. Cadavid, J. D. Fernández Martínez, and J. Morales Vélez, «Revisión de metodologías ágiles para el desarrollo de software A review of agile methodologies for software development,» *Univ. Icesi*, vol. 11 No. 2, pp. 30–39, 2013.
- [54] B. Boehm and W. J. Hansen, «Spiral Development: Experience, Principles, and Refinements Spiral Development Workshop,» *Softw. Eng. Inst.*, 2000.
- [55] K. Brito Acuña, «Selección De Metodologías De Desarrollo Para Aplicaciones Web En La Facultad De Informática De La Universidad De Cienfuegos,» p. 148, 2009.
- [56] Laura Díaz-Bravo, Uri Torruco-García, Mildred Martínez-Hernández, Margarita VarelaRuiz. «La entrevista, recurso flexible y dinámico» Universidad Nacional Autónoma de México, México D.F., México, 2013.
- [57] University of Surrey in 2004.[Online]. Disponible en: <http://www.pressure.to/qda/>
- [58] G.-J. Vreede, G. Kolfshoten, R. O. Briggs y J. Appelman, «ThinkLets as Building Blocks for Collaboration Processes: A Further Conceptualization,» *Lecture Notes in Computer Science*, vol. 3198, pp. 137-152, 2004.
- [59] Y. Méndez, J. Jiménez, C. Collazos, T. Granollers y M. González, «Thinklets: Useful Device for Designing Collaborative Usability Evaluation Methods,» vol. 5, nº 2, 2008.
- [60] A. Solano, METODOLOGÍA PARA LA EVALUACIÓN COLABORATIVA DE LA USABILIDAD DE SISTEMAS SOFTWARE INTERACTIVOS, 2015.

- [61] A. Solano, Y. Méndez y C. Cóllazos, «THINKLET: Elemento clave en la generación de métodos colaborativos para evaluar usabilidad de software,» 2010.
- [62] G.-J. d. Vreede y R. O. Briggs, «Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks,» de Proceedings of the 38th Annual Hawaii International Conference on System Sciences, Big Island, HI, USA, 2005.
- [63] G.-J. d. Vreede, P. G. Koneri, D. L. Dean y A. L. Fruhling, «The Design and Fiel Evaluation of a Repeatable Collaborative Software Code Inspection Process,» de Groupware: Design, Implementation, and Use: 11th International Workshop, CRIWG 2005, Porto de Galinhas, Brazil, September 25-29, 2005, Proceedings , Brazil, Springer, 2005, pp. 325-338.
- [64] Daniela Jácome, Juan Páez, J. Hurtado, C. Collazos “UN CATÁLOGO DE THINKLETS PARA INCREMENTAR LA COLABORACIÓN EN LA INGENIERÍA DE REQUISITOS”, Universidad del Cauca, 2019
- [65] G. Kofschoten y G.-J. d. Vreede, «The Collaboration Engineering Approach for Designing Collaboration Processes,» 2007.
- [66] R. Pena, M. Suhit, R. M. Pena, M. Adrián, S. Dirección, Á. Soria, and I. E. Scott, “Virtual Scrum Lego: Un Ambiente Virtual para la enseñanza de Scrum con Lego. Alumnos Co-dirección,” Universidad Nacional del Centro de la Provincia de Buenos Aires , 2016.
- [67] M. Moreira, “Agile Adoption Roadmap,” The agile journa, november edition, 2010.
- [68] V. Eloranta, K. Koskimies, and T. Mikkonen, “Exploring ScrumBut — An empirical study of Scrum anti-patterns,” Inf. Softw. Technol., vol. 74, pp. 194–203, 2016.
- [69] D. M. Alaimo, Proyectos ágiles con #Scrum: flexibilidad, aprendizaje, innovación y colaboración en contextos complejos. 2013.
- [70] J. Yip, “It’s Not Just Standing Up : Patterns for Daily Stand-up Meetings,” in Pattern languages of programming conference, 2006.
- [71] D. Esther and D. Larsen, “Agile retrospectives - Making Good Teams Great,” J. Prod. Innov. Manag., vol. 24, no. 5, pp. 505–506, 2007.
- [72] Sara Ashraf, Shabib Aftab “IScrum: An Improved Scrum Process Model”, Department of Computer Science, Virtual University of Pakistan, 2017
- [73] Maylor, H. “Project Management, Fourth Edition”, 2010
- [74] Curtis, P. & Carey, M. “Risk Assessment in Practice. Deloitte & Touche LLP”, 2012
- [75] Project Management Institute, Inc. “A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Fifth Edition”, Inc. 2013
- [76] Cox, L. “What’s Wrong with Risk Matrices? Risk analysis: an official publication of the Society for Risk Analysis”, 2008.
- [77] BECK, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, C., Mellor, S., Schwaber, K., Sutherland, J., & Thomas, D. “Manifiesto for agile software development”. 2001.
- [78] FLOSSYSTEMS. “Taiga”. [Online]. Disponible en: <https://www.taiga.io/es>