UNIVERSIDAD DON BOSCO

VICERRECTORÍA DE ESTUDIOS DE POSTGRADO

TRABAJO DE GRADUACIÓN

ELABORACIÓN DE PROPUESTA DE GUÍA DE IMPLEMENTACIÓN DE SCRUM PARA EMPRESA SALVADOREÑA, UN CASO DE ESTUDIO.

PARA OPTAR AL GRADO DE

MAESTRO EN ARQUITECTURA DE SOFTWARE

ASESOR:

MSC. JUAN CARLOS AQUINO

PRESENTADO POR:

PEDRO JOSÉ FLORES MELARA

JULIO CESAR PORTILLO JOVEL

Antiguo Cuscatlán, La Libertad, El Salvador, Centroamérica.

Enero 2017

Contenido

Contenido 1
1 Resumen 4
2 Introducción 5
3 Marco teórico 7
3.1 Metodologías de desarrollo 8
3.2 Metodologías tradicionales 15
3.3 De la secuencialidad a la iteración 23
3.4 Metodologías ágiles 29
3.4.1 Valores del manifiesto ágil 31
3.4.2 Principios del manifiesto ágil 31
3.5 Scrum 32
3.6 Teoría de Scrum 33
3.7 Valores de Scrum 34
3.8 Principios de Scrum 35
3.9 Roles de Scrum 36
3.9.1 Product Owner (Dueño del Producto) 36
3.9.2 Scrum Master 37
3.9.3 Development Team (Equipo de Desarrollo) 39
3.10 Artefactos 40
3.10.1 Backlog del Producto 40
3.10.2 Items del Backlog del producto (PBIs) 42
3.10.3 Backlog del Sprint 43
3.10.4 Tareas del Sprint 43
3.10.5 Sprint Burn Down Chart 43
3.10.6 Product Release Burn Down Chart 44
3.11 Reuniones 45
3.11.1 Planificación del Sprint 45
3.11.2 Daily Scrum y ejecución del Sprint 47
3.11.3 Reunión de revisión del Sprint 48
3.11.4 Reunión de retrospectiva del Sprint 49
3.11.5 Reunión de refinamiento de Backlog 50
4 Metodología de Investigación 52
4.1 Matriz de congruencia 52

4.2 Objetivos de la investigación 52
4.3 Diseño de la investigación 53
4.4 Instrumentos de Investigación 53
4.4.1 Entrevistas a los expertos 53
4.4.2 Entrevistas a la empresa 56
4.4.3 Síntesis de resultados de la entrevista a la empresa. 57
4.4.4 Síntesis de resultados de la entrevista dirigida a los expertos. 61
5 Caso de estudio 63
5.1 Generalidades 63
5.2 Estructura organizativa 63
5.3 Roles por área 65
5.4 Dominio de complejidad de la empresa 69
5.5 Metodología de trabajo actual 71
5.5.1 Situación actual 72
5.5.2 Problema a solucionar 73
5.5.3 Diagrama de flujo actual 73
5.5.4 Alcances 73
5.5.5 Diseño de solución 73
5.5.6 Planificación de la solución 74
5.5.7 Desarrollo 74
5.5.8 Pruebas internas 74
5.5.9 Pruebas con usuarios 74
5.5.10 Salida en vivo 75
6 Guía de Implementación de Scrum 76
6.1 Problemática actual 76
6.2 Recomendaciones para la implementación 77
6.3 Equivalencias entre roles para la conformación del Scrum Team 79
6.4 Implementación 80
6.5 Organización 80
6.6 Aplicación de la metodología 82
6.7 Preparar el proyecto 82
6.7.1 Ordenar el Backlog 82
6.7.2 Estimar el Product Backlog 84
6.7.3 Planificar el Sprint 85
6.7.4 Estimar las tareas 86

6.7.5 Trabajar en el Sprint 87
6.7.6 Llevar a cabo el Sprint 88
6.7.7 Reuniones de Seguimiento diario. 89
6.7.8 Terminar a Tiempo 90
6.7.9 Asegurar que el Sprint esta hecho 91
6.7.10 Revisar, mejorar y repetir 92
7 Conclusiones 94
8 Recomendaciones 96
9 Bibliografía 98
10 Anexos. 101

1 Resumen

Adaptarse a los cambios de un mundo globalizado no es una tarea fácil, requiere mucha disciplina, sacrificios e inversiones por parte de los interesados, hoy en día si las empresas desean sobrevivir a un ambiente hostil donde predomina "la ley del más fuerte", deben adaptar sus procesos de negocio ante las nuevas necesidades de los clientes y este proceso de adaptación conlleva a nuevas modalidades de trabajo, nuevas necesidades de información. Bajo este contexto donde se encuentra inmersa la empresa en estudio y por ende su departamento de sistemas, es necesario realizar ajustes a sus procesos para poder responder oportunamente a las necesidades emergentes de negocio. Este documento presenta una revisión bibliográfica de las metodologías tradicionales y ágiles, con el objetivo de mostrar el porqué de la transición de lo secuencial a lo ágil, para posteriormente hacer un estudio de caso de una empresa salvadoreña a la cual se le propone una guía de implementación de la metodología ágil Scrum.

2 Introducción

El mundo globalizado al cual se enfrenta las empresas en la actualidad requiere que se encuentren listas para afrontar cualquier reto adaptando sus negocios a las nuevas demandas de los clientes. Estos retos demandan contar con sistemas capaces de obtener información de forma ágil y oportuna, de tal forma que sean capaces de predecir y estar a la delantera de las situaciones que el mercado presenta. Esta situación provoca un efecto dominó donde se ven afectados los equipos de desarrollos, dado que se enfrentan a la necesidad de continuar desarrollando aplicaciones en menor cantidad de tiempo y con la misma demanda de calidad. Ante esta nueva realidad, las organizaciones deben buscar una estrategia que les permita poder responder de forma correcta ante estas demandas, es por ello que se enfrentan a la decisión de continuar realizando sus procesos de desarrollo utilizando las técnicas o metodologías con las cuales han venido trabajando, o hacer el cambio hacia una metodología de desarrollo ágil.

Tomar la decisión de cambiar en una organización la metodología de desarrollo a la que vienen acostumbrados por una metodología de desarrollo ágil, no es una elección que se debe de tomar al azar, antes de iniciar un proceso de cambio de este tipo es necesario realizar una evaluación de los diferentes factores que se encuentran involucrados y la forma en que estos influyen dentro del proceso de desarrollo. Los diversos elementos a evaluar deben de ser todos aquellos que influyen directa e indirectamente durante las fases en que se ejecutan los proyectos de desarrollo, pasando para ello desde la estructura organizativa hasta las características del recurso humano involucrado en este proceso.

Esta situación no es extraña para la empresa en la cual se realiza el presente estudio, ellos se encuentran desarrollando un proceso de migración de su metodología de trabajo que les permita cambiar su esquema de desarrollo actual hacia uno soportado sobre la metodología de

desarrollo ágil Scrum. En el presente documento, encontrara una revisión de los princípiales elementos que se han de tomar en cuenta durante el proceso de implementación de dicha metodología dentro la empresa, así como los elementos y criterios que deberán de tomarse en cuenta al momento de llevar a cabo este proceso.

3 Marco teórico

En un tiempo relativamente corto las tecnologías de información y las comunicaciones han evolucionado a tal grado que gran parte de las rutinas diarias del hombre moderno dependen de éstos. Esta dependencia conlleva a nuevas necesidades de información, haciendo que los sistemas queden obsoletos en poco tiempo. Ante nuevas modalidades de trabajo en contextos extremadamente dinámicos los sistemas de información de las empresas deben adaptarse a las nuevas necesidades de negocio.

Todo nuevo desarrollo o cambio a un producto o servicio conlleva a la realización de un proyecto. Un proyecto es un esfuerzo temporal emprendido para crear un producto, servicio o resultado único. La naturaleza temporal de los proyectos indica que un proyecto tiene un principio y un final definidos. El final se alcanza cuando se han logrado los objetivos del proyecto o cuando se termina el proyecto porque sus objetivos no se podrán cumplir o cuando ya no exista la necesidad del proyecto (Project Managament Institute, 2012).

Según Gómez Fuentes et al. (2012), el origen de un proyecto es el resultado de la competitividad creciente, el constante cambio tecnológico y la globalización, los cuales han obligado a las empresas a reaccionar con rapidez ante los cambios del entorno y es así como uno o la combinación de factores tales como una necesidad de mercado, una necesidad de negocios, el requerimiento de un cliente o proveedor o un avance tecnológico desencadenan en la necesidad de llevar a cabo un proyecto.

Independientemente del motivo por el cual se desarrolla el proyecto, éstos tienen una serie de etapas actividades que son estándar a cualquier proyecto sin importar al rubro que pertenezca.

Beriguete De Leon (2012) explica que la gestión de proyectos se refiera a todas las actividades que se realizan para cumplir con un fin principal definido, en un tiempo establecido utilizando recursos tanto humanos como materiales y para el cual se debe tener presupuestados los costos en que se incurrirán. El objetivo principal es administrar, planificar, coordinar, dar seguimiento y control a todas las actividades y recursos asignados para la ejecución del proyecto de una forma que se pueda cumplir con el alcance en el tipo establecido y los costos presupuestados. Las etapas, actividades o ciclo de vida de la gestión de proyecto son las siguientes: iniciación, planificación, ejecución, control y cierre.

Como desarrollar de una forma óptima cada una de las etapas del proyecto y en base a las necesidades del proyecto dependerá de la metodología seleccionada la cual permitirá alcanzar el equilibrio de la triple restricción de tiempo, costo y alcance.

Saenz Arteaga (2012) expone, el paradigma tradicional dice que el éxito de los proyectos depende de satisfacer la triple restricción en el tiempo, dentro del presupuesto y de acuerdo a las especificaciones. En un mundo dinámico de proyectos relacionados a negocios, sin embargo, lo duradero de la triple restricción no es lo suficientemente grande y un nuevo modelo es necesario.

3.1 Metodologías de desarrollo

Para definir las formas de trabajo para desarrollar software es necesario tocar bases con la ingeniería de software. La ingeniería de software es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software.

S. Pressman (2006), argumenta que la ingeniería de software es una tecnología con varias capas:

Fig. 1. Capas de ingeniería de software *(S. Pressman, 2006, p. 12)*

La Fig. 1 muestra las capas que soporta la ingeniería de software entre ellas se encuentra el proceso.

Singh (s.f), expone un proceso desde el punto de vista del estándar ISO / IEC 12207, la cual se basa en principios de ingeniería general de sistemas que se divide en componentes básicos como el análisis, diseño, fabricación, evaluación, pruebas integración garantía, control de calidad y almacenamiento. La fabricación se interpreta como codificación bajo el contexto de ciclo de vida de software. El desarrollo del estándar inició en Junio de 1989 enfocado en los procesos de ciclo de vida del software y es la base para el estándar ISO / IEC 15504. Ver Tabla 1.

S. Pressman (2006) define "Un proceso es un conjunto de actividades, acciones y tareas que se ejecutan cuando va a crearse algún producto del trabajo. Una actividad busca lograr un objetivo amplio (por ejemplo, comunicación con los participantes) y se desarrolla sin importar el dominio de la aplicación, tamaño del proyecto, complejidad del esfuerzo o grado de rigor con el que se usará la ingeniería de software. Una acción (diseño de la arquitectura) es un conjunto de tareas que producen un producto importante del trabajo (por ejemplo, un modelo del diseño de la arquitectura). Una tarea se centra en un objetivo pequeño pero bien definido (por ejemplo, realizar una prueba unitaria) que produce un resultado tangible" (p. 12).

Un punto importante a cerca de la estructura del proceso es la que describe S. Pressman (2006), el cual dice: "En el contexto de la ingeniería de software, un proceso no es una

prescripción rígida de cómo elaborar software de cómputo. Por el contrario, es un enfoque adaptable que permite que las personas que hacen el trabajo (el equipo de software) busquen y elijan el conjunto apropiado de acciones y tareas para el trabajo. Se busca siempre entregar el software en forma oportuna y con calidad suficiente para satisfacer a quienes patrocinaron su creación y a aquellos que lo usarán (…). La estructura del proceso establece el fundamento para el proceso completo de la ingeniería de software por medio de la identificación de un número pequeño de actividades estructurales que sean aplicables a todos los proyectos de software, sin importar su tamaño o complejidad. Además, la estructura del proceso incluye un conjunto de actividades sombrilla que son aplicables a través de todo el proceso del software (…)" (p. 12).

En un inicio la ingeniería de software adoptó las metodologías de desarrollo de proyectos tradicionales tal cual habían sido implementadas con éxito por otras ramas de la ingeniería, como parte de buenas prácticas para gestionar el desarrollo de software. Hayes y Andrews (2014), exponen su punto de vista el cual dice: "Comparado con hace treinta años hoy en día se tienen computadoras mucho más baratas y rápidas, herramientas de apoyo, lenguajes de programación más potentes, una mejor educación y comprensión de la teoría del software. Internet ha cambiado la forma en cómo trabajan y comunican las personas, además de cambiar las expectativas de cómo el software debe funcionar. El desarrollo de software es análogo a un proceso industrial definido, basado en procesos de ingeniería predictivos que asumen que las personas pueden ser tratadas como recursos abstractos. Es típico adoptar un enfoque de modelado (teórico) definido cuando los mecanismos subyacentes por los que opera un proceso están razonablemente bien entendidos, cuando el proceso es demasiado complicado, el enfoque empírico es la elección adecuada" (p. 5).

En la teoría de control de procesos existen dos clasificaciones de procesos: los procesos definidos y los procesos complejos. “Un proceso definido es aquel que se puede diseñar y ejecutar repetidamente con resultados predecibles. Un proceso complejo es aquel que no puede definirse con precisión suficiente para garantizar resultados predecibles, requiere de un modelo de control empírico que implica inspección frecuente y respuesta adaptativa. El desarrollo de software no es un proceso definido porque las entradas principales del proceso son las personas que definen los requerimientos iniciales” (Hayes y Andrews, 2014, p. 6).

Instituciones internacionales como ISO a través de los años han ido desarrollando y mejorando estándares encaminados a la implementación de buenas prácticas en la gestión de proyectos de software en la industria. Estos estándares contribuyen a los procesos de ciclo de vida del software y gestión de calidad. La Tabla 1 muestra estándares relacionados al desarrollo de software.

Tabla 1

Estándares ISO para el proceso de ciclo de vida del software y gestión de calidad.

ISO	Descripción
ISO 9001	“La norma ISO 9001 es un estándar internacional que especifica los requerimientos para un manejo de control de calidad. Las organizaciones utilizan este estándar para demostrar su habilidad para proveer productos y servicios que se ajuste a los requerimientos del cliente” (Diagnostico y Soluciones, 2007).
ISO 9003	“La ISO 9003 es una norma desarrollada por la Organización Internacional de Normas (OIN) que regula las actividades de inspecciones y pruebas finales. La norma ISO 9003 ya es obsoleta y actualmente no se otorga. Fue reemplazada en el año 2000 por un nuevo conjunto de normas que son las ISO 9001. Estas últimas brindan un marco más amplio que abarca varios aspectos de un sistema administrativo de calidad. Hoy en día, por lo general, las empresas buscan obtener la certificación 9001” (Gosset, s.f).
ISO/IEC 12207	Desarrollada en Junio de 1989, el estándar está enfocado a los procesos del ciclo de vida del software para satisfacer una necesidad crítica. La norma establece una arquitectura de alto nivel del ciclo de vida de software. El ciclo de vida comienza con una idea o una necesidad que puede ser satisfecha total o parcialmente por software Y termina con la retirada del software. La arquitectura se construye con un conjunto de procesos e interrelaciones entre estos procesos (Singh, s.f, p. 1 y 2).

Tabla 1 (Continuación)

Estándares ISO para el proceso de ciclo de vida del software y gestión de calidad

ISO	Descripción
ISO 15504	"ISO/IEC 15504. Es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas de información y productos de software. ISO/IEC 15504, también conocido como Software Process Improvement Capability Determination (SPICE) significa «Determinación de la Capacidad de Mejora del Proceso de Software»" (Ecured, s.f).
ISO/IEC 9126	"Esta norma Internacional fue publicada en 1992, la cual es usada para la evaluación de la calidad de software, llamado "Information technology-Software product evaluation-Quality characteristics and guidelines for their use"; o también conocido como ISO 9126 (o ISO/IEC 9126). Este estándar describe 6 características generales: Funcionalidad, Confiabilidad, Usabilidad, Eficiencia, Mantenibilidad, y Portabilidad" (Borbón Ardila, 2013).
ISO/IEC 14598	"Norma ISO/IEC 14598. En sus diferentes etapas, establece un marco de trabajo para evaluar la calidad de los productos de software proporcionando, además, métricas y requisitos para los procesos de evaluación de los mismos. En particular, es utilizada para aplicar los conceptos descritos en la norma ISO / IEC 9126. Se definen y describen las actividades necesarias para analizar los requisitos de evaluación, para especificar, diseñar y realizar acciones de evaluación y para concluir la evaluación de cualquier tipo de producto de software" (Ecured, s.f).
ISO/IEC 25000	"ISO/IEC 25000, conocida como SQuaRE (System and Software Quality Requirements and Evaluation), es una familia de normas que tiene por objetivo la creación de un marco de trabajo común para evaluar la calidad del producto software. La familia ISO/IEC 25000 es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126, que describe las particularidades de un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software" (ISO 25000, s.f).
ISO/IEC 9004-2	"Norma de gestión avanzado, esta norma internacional proporciona orientación para ayudar a conseguir el éxito sostenido para cualquier organización en un entorno complejo, exigente y en constante cambio, mediante un enfoque de gestión de la calidad. Esta norma internacional proporciona un enfoque más amplio sobre la gestión de la calidad que la norma ISO 9001; trata las necesidades y las expectativas de todas las partes interesadas pertinentes y proporciona orientación para la mejora sistemática y continua del desempeño global de la organización" (Sierra Caballeo, Valencia, y Moralez, s.f).

Los estándares no son rígidos y recomiendan tener en consideración ciertos procesos para poder gestionar los proyectos con éxito, éstos son flexibles y permiten ser adaptados según las necesidades de la empresa.

La ingeniería de software agrupa los métodos, herramientas y procesos que deben ser tomados en cuenta como disciplina para poder generar un producto único o servicios. Sin

embargo ¿qué procesos deben ser tomados en cuenta para desarrollar software?, ¿cómo estructurar, planificar y controlar el proceso de desarrollo de un producto o servicio?.

Dependiendo de factores como la complejidad del proyecto, la cultura organizacional y la estabilidad de los requerimientos, es como puede variar la metodología a utilizar para el desarrollo del proyecto. ¿Pero qué es una metodología? "una metodología de desarrollo de sistemas se refiere al marco que se utiliza para estructurar, planificar y controlar el proceso de desarrollo de un sistema de información. Una gran variedad de estos marcos han evolucionado a lo largo de los años, cada uno con sus propias fortalezas y debilidades reconocidas. Una metodología de desarrollo de sistemas no es necesariamente adecuada para ser usada por todos los proyectos. Cada una de las metodologías disponibles se adapta mejor a tipos específicos de proyectos, basados en diversas consideraciones técnicas" (Centers of Medicare and Medicaid Services, 2008).

Una metodología es "un conjunto integrado de técnicas y métodos que permite abordar de forma homogénea y abierta cada una de las actividades del ciclo de vida de un proyecto de desarrollo. La metodología para el desarrollo de software en un modo sistemático de realizar, gestionar y administrar un proyecto para llevarlo a cabo con altas posibilidades de éxito. Una metodología para el desarrollo de software comprende los procesos a seguir sistemáticamente para idear, implementar y mantener un producto software desde que surge la necesidad del producto hasta que se cumple el objetivo por el cual fue creado" (INTECO: Instituto Nacional de Tecnologías de la Comunicación., 2009, p. 39).

INTECO: Instituto Nacional de Tecnologías de la Comunicación (2009), destaca los elementos que forman parte de una metodología que definen la estrategia global para enfrentarse con el proyecto así como también las ventajas de uso:

Elementos:

- Fases: tareas a realizar en cada fase.
- Productos: E / S de cada fase, documentos.
- Procedimientos y herramientas: apoyo a la realización de cada tarea.
- Criterios de evaluación: del proceso y del producto. Saber si se han logrado los objetivos.

Ventajas del uso de una metodología:

- Desde el punto de vista de gestión.
    - Facilitar la tarea de planificación.
    - Facilitar la tarea del control del seguimiento de un proyecto.
    - Mejorar la relación costo/beneficio.
    - Optimizar el uso de recursos disponibles.
    - Facilitar la evaluación de resultados y cumplimiento de los objetivos.
    - Facilitar la comunicación efectiva entre usuarios y desarrolladores.
- Desde el punto de vista de los ingenieros de software
    - Ayudar a la comprensión del problema.
    - Optimizar el conjunto y cada uno de las fases del proceso de desarrollo.
    - Facilitar el mantenimiento del producto final.
    - Permitir la reutilización de partes del producto.
- Desde el punto de vista del cliente o usuario
    - Garantía de un determinado nivel de calidad en el producto final.
    - Confianza en los plazos de tiempo fijados en la definición del proyecto.

- Definir el ciclo de vida que más se adecue a las condiciones y características del desarrollo.

Balaji y Sundarajan Murugaiyan (2012), proponen que antes de decidir qué modelo de desarrollo se usará, es necesario responder las siguientes preguntas:

1. ¿Qué tan estables son los requerimientos?
2. ¿Quiénes son los usuarios finales?
3. ¿Cuál es el tamaño del proyecto?
4. ¿Dónde están localizados los equipos de proyecto?

3.2 Metodologías tradicionales

Balaji y Sundarajan Murugaiyan (2012), Definen las características de las metodologías tradicionales siendo la más representativa la metodología de cascada. La metodología de cascada es un modelo secuencial, en el que los requisitos deben estar claros antes de pasar a la fase de diseño, las pruebas se llevan a cabo una vez que el código ha sido completado. Cada trabajo, producto o actividad se completa antes de pasar a la siguiente. En esta metodología cada fase del desarrollo procede en orden sin ninguna superposición. La documentación y las pruebas se realizan al final de cada fase, lo que ayuda a mantener la calidad del proyecto. En el modelo de cascada cada paso se congela antes del siguiente paso. Es decir, los requisitos se congelan antes de que empiece el diseño, y una vez que el diseño se congela, la codificación se inicia.

Palmquist et al. (2013), Exponen los puntos principales del enfoque de desarrollo de software basado en métodos tradicionales y la gestión de riesgos:

Enfoque de desarrollo de software:

- Los sistemas se desarrollan en un proceso secuencial (a veces incluso en un solo paso).

- Todos los requisitos se determinan de antemano, con los supuestos duales de que todos pueden ser conocidos por adelantado, y que son y seguirán siendo inmutables.
- El análisis se realiza una vez y precede al diseño.
- El diseño se realiza una vez y precede a la codificación.
- Toda la codificación se realiza una vez y precede a la prueba y la integración.
- Todas las pruebas se realizan una vez seguidas o en conjunción con actividades de integración, y preceden al uso operacional.
- Se requiere una revisión y aprobación formal para pasar de un paso a otro.
- El cliente está más involucrado estableciendo los requisitos, desaparece en su mayor parte durante el diseño del análisis y la codificación, y vuelve a emitirse durante la prueba y la aceptación.
- Se requiere una extensa documentación para cada paso.

Gestión de riesgos:

- La identificación temprana de todos los requisitos es necesaria para apoyar la planificación y los presupuestos.
- La identificación y el bloqueo de los requisitos a principios del programa evita el deslizamiento del alcance.
- La documentación de todos los aspectos del diseño y la toma de decisiones es necesaria para futuras referencias.
- Los exámenes exhaustivos no sólo garantizan que todos los interesados conozcan el progreso, sino que también permiten descubrir cuestiones antes.

- Todos los requisitos necesarios para ser documentados antes de que cualquier trabajo de diseño comienza a asegurar que la arquitectura fuera adecuada.
- Toda la codificación necesaria para completarse antes de la prueba para asegurar que las capacidades de todo el sistema puedan ser evaluadas.
- Todos los requisitos debían cumplirse antes de que el sistema pudiera ser puesto en producción.

Royce (2013), plantea un escenario simple de la implementación de la metodología en cascada, pero que a medida el proyecto es más complejo éste es ajustado de tal forma que pueda obtenerse el producto final deseado. Hay dos pasos esenciales comunes a todo el desarrollo de programas informáticos, independientemente de su tamaño o complejidad. Primero hay un paso de análisis seguido de un paso de codificación representado en la Fig. 2. Este tipo de concepto de implementación es muy simple, de hecho, todo lo que se requiere si el esfuerzo es suficientemente pequeño y si el producto final debe ser operado por aquellos que construyeron eso. Como se suele hacer con programas informáticos para uso interno. Es también el tipo de esfuerzo de desarrollo al que la mayoría de los clientes están dispuestos a pagar, ya que ambos pasos implican un trabajo genuinamente creativo que contribuye directamente a la utilidad del producto final. Sin embargo, un plan de implementación para desarrollar sistemas de software más grandes, y sólo con estos pasos, está condenado al fracaso. Se requieren muchos pasos adicionales de desarrollo, ninguno contribuye directamente al producto final como el análisis y la codificación, y todos aumentan los costos de desarrollo. Normalmente, el personal del cliente preferiría no pagar por ellos, y el personal de desarrollo preferiría no implementarlos. La función principal de la administración es vender estos conceptos a ambos grupos y luego hacer cumplir el cumplimiento por parte del personal de desarrollo.

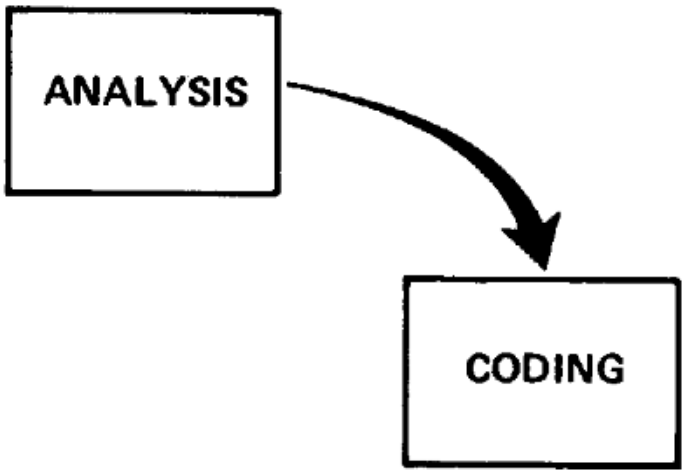

Fig. 2 pasos comunes a todo desarrollo de programas informáticos.

La Fig. 3 muestra los pasos para desarrollo de sistemas con un nivel más alto de complejidad, los pasos de análisis y codificación están todavía en la imagen, pero están precedidos por dos niveles de análisis de requisitos, están separados por un paso de diseño del programa y seguidos por un paso de prueba. Estas adiciones se tratan por separado del análisis y la codificación, ya que son claramente diferentes en la forma en que se ejecutan. Deben ser planificados y empleados de manera diferente para una mejor utilización de los recursos del programa (Royce, 2013).

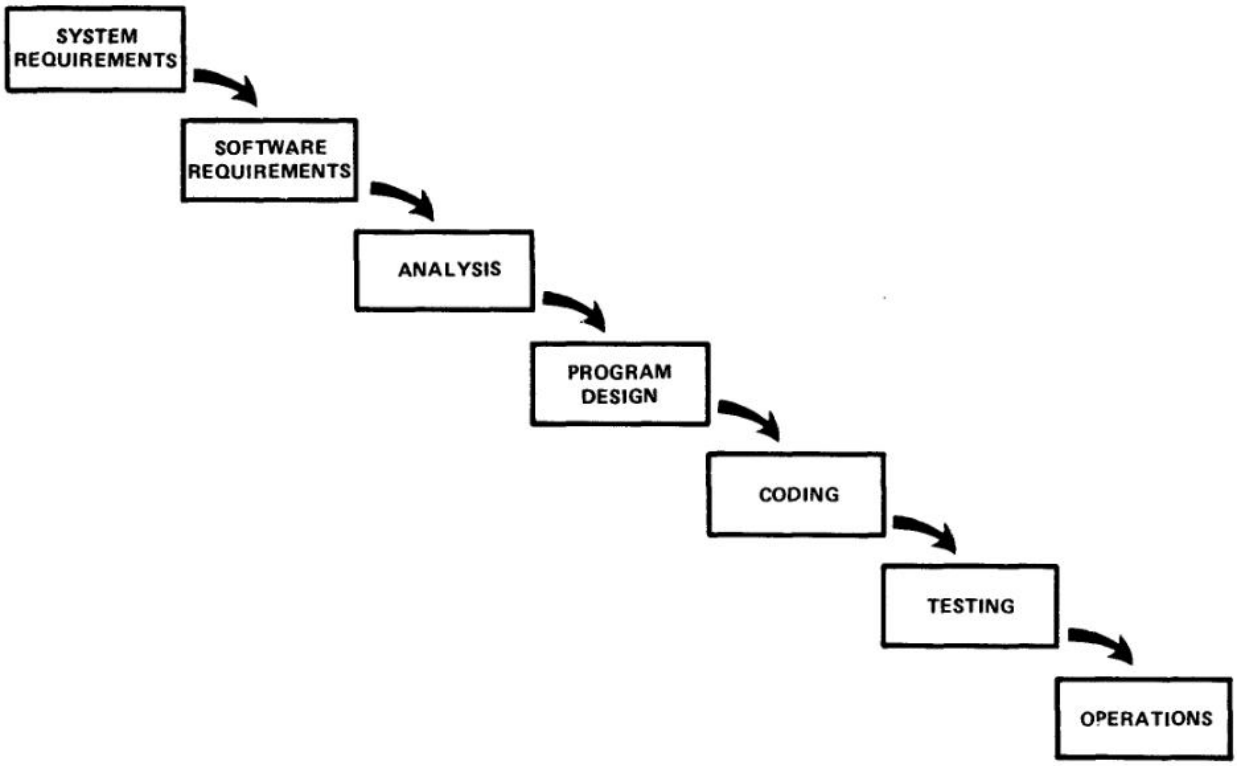

Fig. 3 Pasos para desarrollar un software con un nivel más alto de complejidad.

La Fig. 4 representa la relación iterativa entre las fases de desarrollo. El ordenamiento de los pasos se basa en el siguiente concepto: que a medida que progresa cada paso y se detalla el diseño, hay una iteración con las etapas precedentes y sucesivas pero rara vez con las etapas más remotas de la secuencia. La virtud de todo esto es que a medida que el diseño avanza, el proceso de cambio se alcanza hasta límites manejables. En cualquier momento del proceso de diseño, una vez completado el análisis de los requisitos, existe una base de referencia firme y de primer plano para volver a aparecer en caso de dificultades imprevistas de diseño (Royce, 2013).

La Fig. 5 muestra otro panorama en el que existen puntos de control e iteración en las etapas de requerimientos de software, diseño de programa y pruebas. La fase de prueba que se produce al final del ciclo de desarrollo es el primer evento para el cual se experimentan el

tiempo, el almacenamiento, las transferencias de entrada / salida, etc. Estos fenómenos no son precisamente analizables. Sin embargo, si estos fenómenos no satisfacen las diversas limitaciones externas, entonces invariablemente se requiere un rediseño mayor. Es probable que los cambios de diseño requeridos sean tan perjudiciales que se violen los requisitos de software sobre los que se basa el diseño y que proporciona la razón de ser de todo. Se deben modificar los requisitos o se requiere un cambio sustancial en el diseño (Royce, 2013).

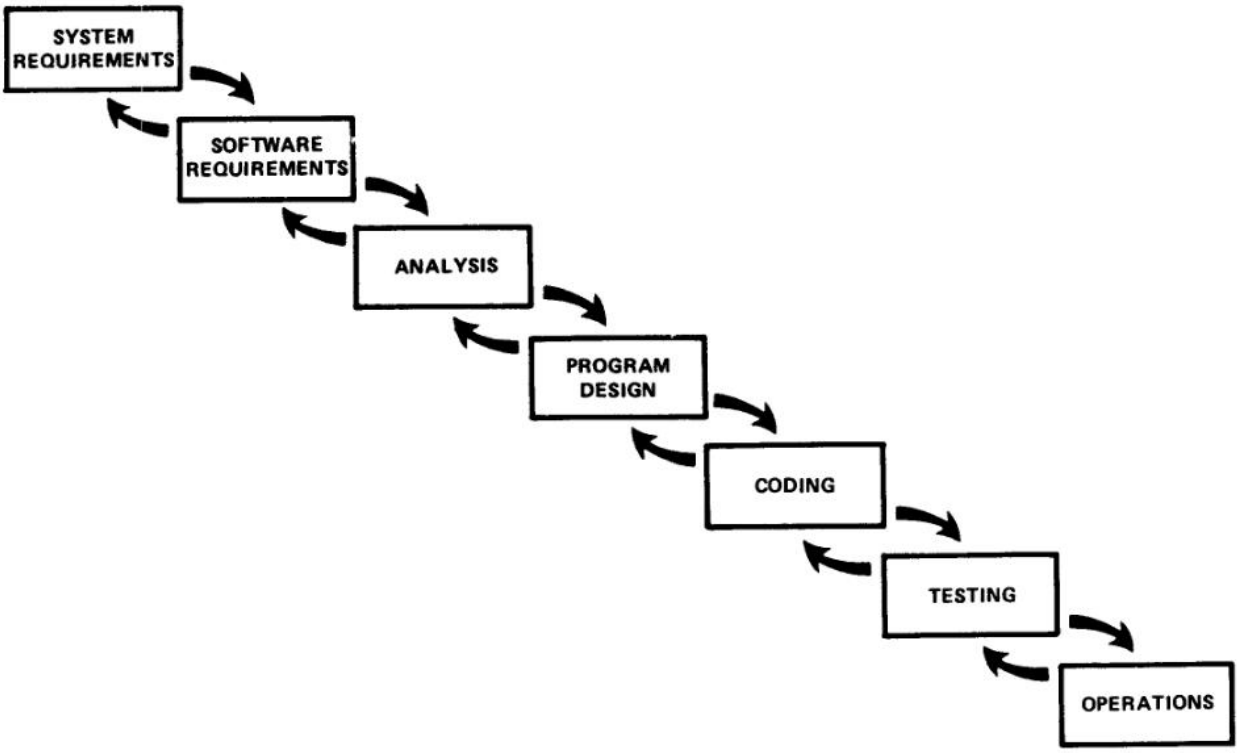

Fig. 4 relación iterativa entre fases sucesivas de desarrollo

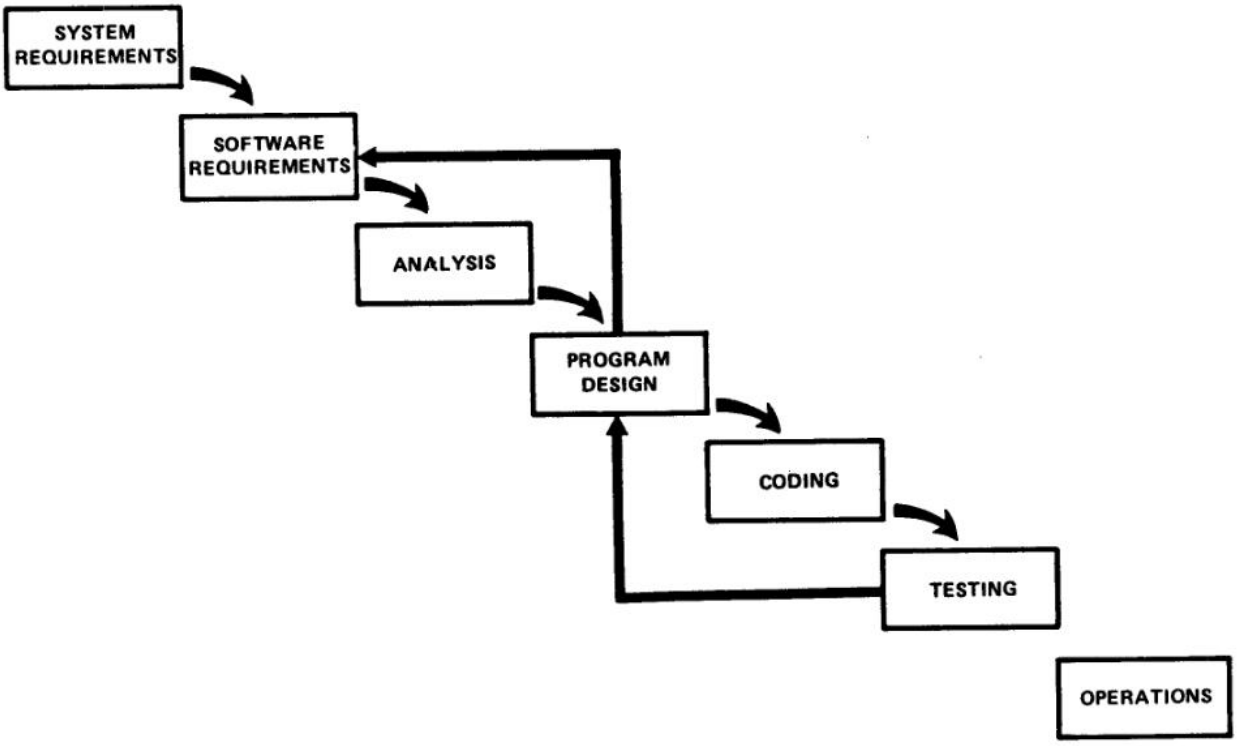

Fig. 5 iteración por puntos de control en etapas de requerimiento de software, diseño de programa y pruebas.

Royce (2013), Resume los cinco pasos necesarios para transformar un proceso de desarrollo arriesgado a uno que proporcione el producto deseado, los cuales son detallados a continuación. Ver Fig. 6.

Completar el diseño del programa antes de que el análisis y la codificación inicien: Una fase de diseño preliminar ha sido insertada entre las fases de generación de requerimientos de software y la fase de análisis. Este procedimiento puede ser criticado sobre la base de que el programador es forzado a diseñar en un relativo vacío de requerimientos de software sin un análisis existente, como resultado el diseño preliminar resultará con errores comparado contra el diseño si se hubiera esperado el análisis completo. Esta técnica puede ser criticada, pero el programador asegura que el software no fallará debido a razones de almacenamiento, sincronización y flujo de datos.

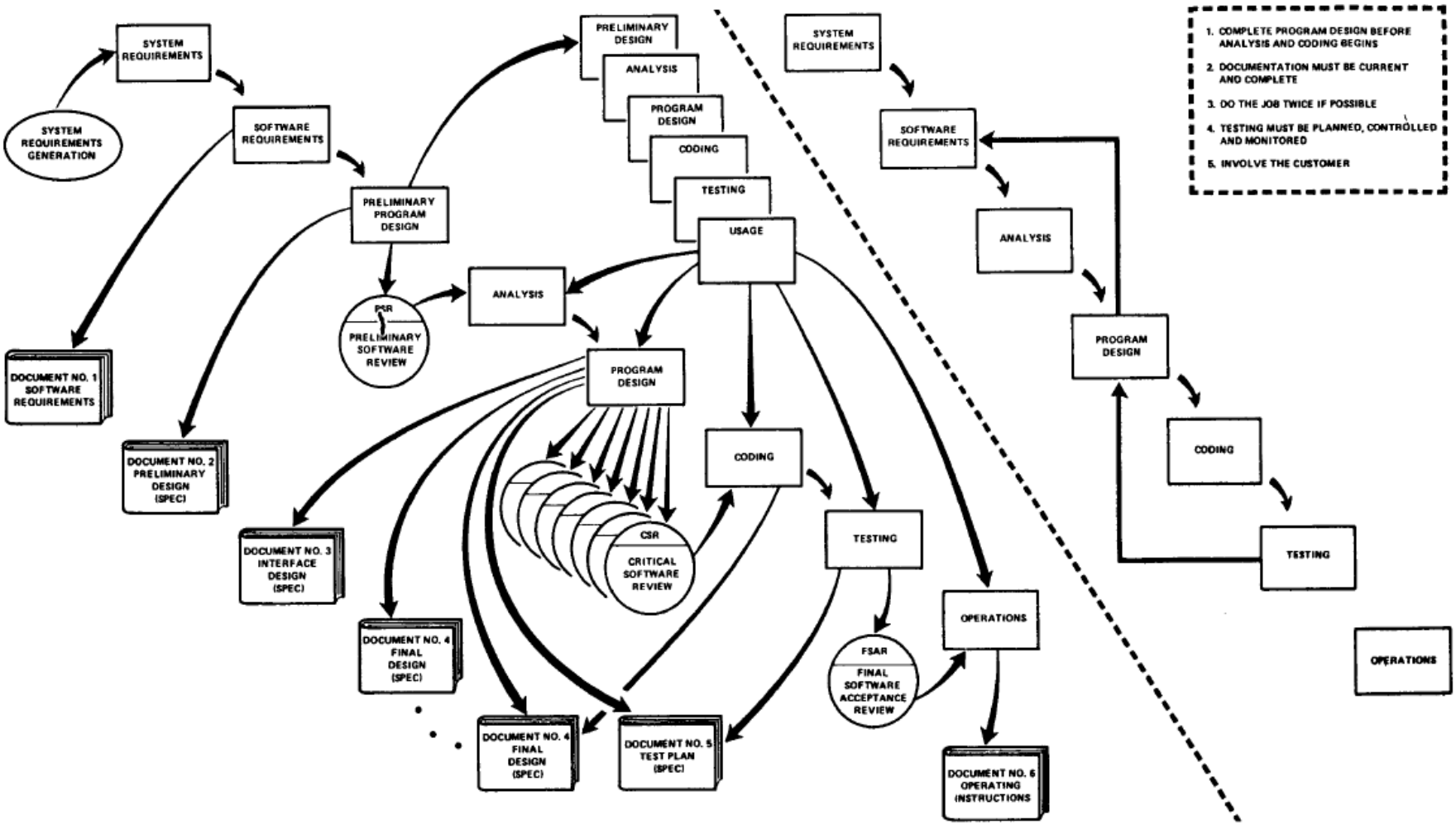

Fig. 6 resume los cinco pasos necesarios para transformar un proceso de desarrollo arriesgado en uno que proporcione el producto deseado.

La documentación debe estar actualizada y completa.

Hacer el trabajo dos veces si es posible: La primera aproximación simulará prácticamente al producto final.

Las pruebas deben ser planeadas, controladas y monitoreadas.

Involucrar al cliente: Por alguna razón lo que un diseño de software va hacer está sujeto a una amplia interpretación, incluso después de un acuerdo previo. Es importante involucrar al cliente de una manera formal para que se vea comprometido en puntos anteriores antes de la entrega final.

Winston W. Royce es considerado como el padre de la metodología de cascada, incluso él es crítico de la metodología ya que a medida los proyectos crecen y se vuelven más complejos, congelar requerimientos y esperar a terminar una fase para seguir con la siguiente representa una desventaja ya que cuando se ha llegado hasta el punto de pruebas es muy probable que los requerimientos del cliente hayan evolucionado a uno más complejo, de tal forma que las bases originales sobre las que se rige el análisis y diseño han cambiado haciendo obsoleta la solución inicial provocando que se plantee un nuevo análisis y diseño. Royce propone métodos iterativos incluso las bases para un modelo ágil.

3.3 De la secuencialidad a la iteración

No todos los proyectos son iguales, pero un atributo común a cada uno de ellos es el grado de complejidad, como abordar cada proyecto identificando su contexto es parte de un marco de trabajo que presenta Snowden y Boone en su artículo “A leader’s framework for decision making”. Identificar el contexto bajo el que nos encontramos es una de las tareas primarias que se deben tomar en cuenta al momento de seleccionar una metodología, la Tabla 2 presenta las diferencias entre metodologías tradicionales y ágiles, mientras que la Tabla 3

identifica los aspectos clave para poder migrar a una metodología ágile. Identificar el contexto, conocer las diferencias entre metodologías y conocer los aspectos clave para migrar a una metodología a otra, constituyen un insumo para la toma de decisión sobre que metodología seleccionar para el desarrollo de un proyecto de software.

Cynefin, permite a los ejecutivos ver las cosas desde nuevos puntos de vista, asimilar conceptos complejos y abordar problemas y oportunidades del mundo real. (Cynefin, pronunciado ku-nev-in, es una palabra galesa que significa los múltiples factores en el entorno y la experiencia que nos influencian de formas que nunca podemos entender). Usando este enfoque, los líderes aprenden a definir el marco con ejemplos desde la historia de su propia organización y escenarios de su posible futuro. Esto mejora la comunicación y ayuda a los ejecutivos a comprender rápidamente el contexto en el que están operando. El marco clasifica las cuestiones que enfrentan los líderes en cinco contextos definidos por la naturaleza de la relación entre causa y efecto. Cuatro de estos: simple, complicado, complejo y caótico. Requieren líderes para diagnosticar situaciones y actuar de manera contextualmente apropiada. El quinto escenario de desorden se aplica cuando no está claro cuál de los otros cuatro contextos es predominante. El uso del marco Cynefin puede ayudar a los ejecutivos a percibir en qué contexto se encuentran para que no sólo puedan tomar mejores decisiones, sino también evitar los problemas que surgen cuando su estilo de gestión preferido les hace cometer errores (Snowden y Boone, 2014).

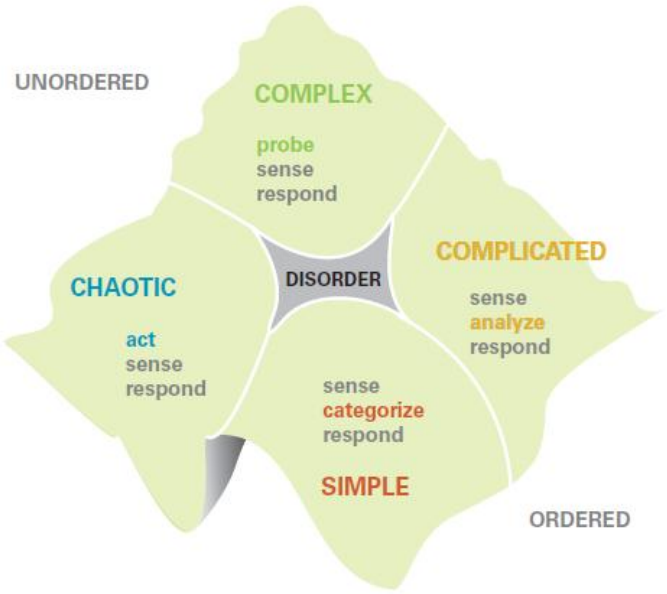

Fig. 7 Dominios del framework Cynefin **(**Snowden y Boone, 2014, p. 7).

Alaimo (2013), describe las características de los dominios de complejidad del marco de trabajo Cynefin, el cual consta de cinco dominios:

Dominio simple: en este dominio se opera con problemáticas simples. Es muy fácil identificar las causas y sus efectos. Por lo general, la respuesta correcta es clara, conocida por todos e indiscutible. En este dominio existen las mejores prácticas, soluciones conocidas para problemas conocidos. Los procesos más eficientes en este dominio son aquellos que especifican una serie lógica de pasos y se ejecutan de manera repetitiva, una y otra vez. Ejemplos de este dominio son la construcción en serie de un mismo producto, la instalación en muchos clientes de un mismo sistema.

Dominio complicado: En este dominio encontramos problemas complejos, buenas prácticas y perfiles expertos. Hay múltiples soluciones correctas para una misma problemática, pero se requiere del involucramiento de expertos para poder identificarlas. Un ejemplo típico de este escenario es la solución de un problema de performance en un

software o base de datos, la sincronización de semáforos en un cruce de tres avenidas, la búsqueda de eficiencia en la distribución logística de mercaderías, etc.

Dominio complejo: Cuando se enfrenta a problemas complejos, los resultados se vuelven más impredecibles. No existen ni mejores ni buenas prácticas catalogadas para las situaciones frente a las cuales se pueden encontrar. Simplemente, no se sabe con anticipación si una determinada solución va a funcionar. Solo se puede examinar los resultados y adaptar. Este es el dominio de las prácticas emergentes. Las soluciones encontradas rara vez son replicables, con los mismos resultados, a otros problemas similares. Para poder operar en la complejidad se necesita generar contextos donde haya lugar para la experimentación y donde el fallo sea de bajo impacto. Se requieren niveles altos de creatividad, innovación, interacción y comunicación. El desarrollo de nuevos productos o la incorporación de nuevas características en productos existentes es un contexto complejo en el que metodologías ágiles como Scrum son utilizadas para actuar, inspeccionar y adaptar las prácticas emergentes de un equipo de trabajo.

Dominio caótico: Los problemas caóticos requieren una respuesta inmediata, en una situación de crisis es necesario actuar de inmediato para restablecer cierto orden. Por ejemplo el sistema de despacho de vuelos en un aeropuerto de alto tráfico deja de funcionar. Aquí se debe actuar de inmediato, alguien debe tomar el control y mover la situación fuera del caos. Por ejemplo: solucionar el problema inmediatamente (sin importar la forma técnica), para luego, fuera del caos, evaluar y aplicar una solución más robusta, de ser necesario. Este es el dominio de la improvisación.

Dominio desordenado: Se presenta cuando no se sabe definir el dominio en el que se encuentra. Se la clasifica como una zona peligrosa, ya que no se puede medir las situaciones

ni determinar la forma de actuar. Es muy típico en estas situaciones que las personas interpreten las situaciones y actúen en base a preferencias personales. El gran peligro del dominio desordenado es actuar de manera diferente a la que se necesita para resolver ciertos problemas. Por ejemplo, mucha gente en el ámbito del desarrollo de software está acostumbrada al desarrollo secuencial, por fases, detalladamente planificado utilizando las mejores prácticas de la industria, y este enfoque, que corresponde al dominio Simple, muchas veces se aplica en el dominio complejo. Si se llegara a encontrar en el espacio desordenado, todo lo que se haga debe estar enfocado netamente a salir de ese espacio hacia uno mejor identificado, para luego actuar de la manera en que dicho dominio lo requiera.

Jan (2016), argumenta que en un entorno simple, es posible - y fácil - crear un plan óptimo. Los proyectos de cascada son un marco perfecto para implementar planes predefinidos. Pero los "Agilistas" generalmente no se preocupan por los proyectos simples. La razón por la que Agile existe es principalmente para servir al dominio caótico. El valor de un plan en el dominio caótico depende de una variable de tiempo desconocida e incontrolable.

Tabla 2

Desarrollo de software tradicional versus agile (Nerur, Mahapatara, y Mangalaraj, 2005, p. 4).

	Tradicional	Agile
Suposiciones fundamentales	Los sistemas son completamente especificables, predecibles y pueden construirse a través de una planificación minuciosa y extensa.	Software de alta calidad y adaptable puede ser desarrollado por pequeños equipos utilizando los principios de mejora continua del diseño y las pruebas basadas en la retroalimentación rápida y el cambio.
Control	Centrado en el proceso	Centrado en las personas
Estilo de gestión	Comando y control	Liderazgo y colaboración
Gestión del conocimiento	Explicito	Tácito
Asignación de funciones	Individual, favorece a la especialización	Equipos auto-organizadores-fomenta el intercambio de roles

Tabla 2 (Continuación)

Desarrollo de software tradicional versus agile (Nerur, Mahapatara, y Mangalaraj, 2005, p. 4).

	Tradicional	Agile
Comunicación	Formal	Informal
Papel del cliente	Importante	Crítica
Ciclo de proyecto	Guiada por tareas y actividades	Guiado por características del producto
Modelo de desarrollo	Modelo de ciclo de vida (cascada, espiral o alguna variación)	Modelo de entrega evolutiva
Forma / estructura organizativa deseada	Mecánico (burocrático con alta formalización)	Orgánica (acción social cooperativa flexible y participativa)
Tecnología	No hay restricción	Favorece la tecnología orientada a objetos.

Tabla 3

Aspectos clave para migrar a una metodología agile (Nerur, Mahapatara, y Mangalaraj, 2005, p. 5).

Rubro	Aspecto clave
Gestión y Organización	Cultura organizacional Estilo de gestión Estructura organizativa Gestión del Conocimiento de Desarrollo de Software Sistemas de recompensa
Personas	Trabajar eficazmente en un equipo Alto nivel de competencia Relaciones con los clientes: compromiso, conocimiento, proximidad, confianza y respeto
Procesos	Cambiar de un enfoque centrado en el proceso a un enfoque orientado a las personas y centrado en las personas Desarrollo corto, iterativo, impulsado por pruebas que enfatiza la adaptabilidad Gestión de proyectos grandes y escalables Selección de un método ágil adecuado
Tecnología (herramientas y técnicas)	Adecuación de la tecnología y las herramientas existentes Nuevos conjuntos de habilidades-refactorización, gestión de configuración, JUnit

3.4 Metodologías ágiles

Sinha (2010), argumenta sobre la metodología ágil. Ágil es un enfoque adaptativo que se basa en la filosofía de que los cambios no se pueden evitar. Promueve un ciclo corto de entrega, un análisis justo a tiempo, una colaboración cercana, y una alta visibilidad. Una iteración en general abarca de dos a cuatro semanas y generalmente se completa con una entrega. En general, la primera iteración se usa para realizar el alcance y plan preliminar, y un diseño inicial. En las siguientes iteraciones se hace el desarrollo. Luego de completar una iteración de desarrollo, se hace una demo y se obtiene retroalimentación del cliente. Cualquier cambio necesario en el software que está funcionando se implementará en las iteraciones siguientes. Uno de los beneficios de este modelo es que las modificaciones necesarias se incorporan en el software sin sorpresas de último minuto.

Las metodologías ágiles presentan ventajas, que son la razón por la cual la mayoría de empresas deciden migrar, entre otros factores. Mahanti (2006), lista algunas de ellas:

- Mejor retorno de la inversión ROI: Los clientes proporcionan retroalimentación frecuente sobre cada iteración, lo que permite al equipo de desarrollo crear mejor software, mejorando así el ROI.
- Detección precoz y cancelación de proyectos fallidos: En proyectos tradicionales, los informes optimistas sobre tareas abstractas como análisis y diseño, retrasan los problemas que se identifican en las primeras etapas del proyecto. Los métodos ágiles ejecutan tareas de diseño, análisis e implementación repetidamente en iteraciones cortas que permiten una mayor visibilidad del estado de avance del proyecto. Los patrocinadores pueden cancelar el proyecto anticipadamente si encuentran que no va como se espera y por lo tanto pierden una inversión mínima.

- Mayor calidad de software: el desarrollo basado en pruebas, las iteraciones cortas, el alcance y los comentarios frecuentes de los clientes mejoran la calidad general del software.
- Mejora del control sobre el proyecto: Los procesos ágiles se centran en las personas sobre los procesos. Con menos estrés en la documentación y más atención en la entrega de funcionalidad al final de cada iteración, los equipos ágiles pueden mejorar su velocidad. Las iteraciones cortas, los equipos multidisciplinarios, el intercambio de conocimientos, la integración continua y la retroalimentación permiten un mejor control sobre el proyecto y una mayor visibilidad.
- Reducción de la dependencia de los individuos y mayor flexibilidad: Las prácticas ágiles hacen hincapié en la creación de equipos cohesivos de desarrolladores que analizan colectivamente, diseñan y codifican software. Esto elimina la posibilidad de cualquier cuello de botella en el proceso de desarrollo debido a los desarrolladores que trabajan individualmente en las tareas.

Las metodologías ágiles están regidas por valores y principios, estos nacen en Febrero de 2001, cuando 17 profesionales se reunieron en Utah, Estados Unidos para discutir qué medidas podían tomar para dar solución a los problemas generados por las metodologías tradicionales, y es así como queda definido el manifiesto ágil, el cual está compuesto por cuatro valores (Beck et. al, 2001) y doce principios (Beck et. al, 2016), que son listados literalmente en las siguientes secciones.

3.4.1 Valores del manifiesto ágil

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

3.4.2 Principios del manifiesto ágil

1. Nuestra mayor prioridad es satisfacer al cliente mediante la entrega temprana y continua de software con valor.
2. Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos Ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
3. Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al periodo de tiempo más corto posible.
4. Los responsables de negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
5. Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
6. El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
7. El software funcionando es la medida principal de progreso.
8. Los procesos Ágiles promueven el desarrollo sostenible. Los promotores, desarrolladores y usuarios debemos ser capaces de mantener un ritmo constante de forma indefinida.

9. La atención continua a la excelencia técnica y al buen diseño mejora la Agilidad.
10. La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
11. Las mejores arquitecturas, requisitos y diseños emergen de equipos auto-organizados.
12. A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia.

3.5 Scrum

Scrum es un marco para organizar y gestionar el trabajo. El marco de Scrum se basa en un conjunto de valores, principios y prácticas que proporcionan la base a la cual la organización agregará su implementación única de prácticas de ingeniería relevantes y sus enfoques específicos para realizar las prácticas de Scrum. El resultado será una versión de Scrum que es únicamente de la empresa que lo implemente (Rubin, 2016). La Fig. 8 muestra el marco de trabajo Scrum.

Scrum es un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que entregan productos del máximo valor posible productiva y creativamente. Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo de modo que se pueda mejorar. El marco de trabajo Scrum consiste en los Equipos Scrum y sus Roles, Eventos, Artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es

esencial para el éxito de Scrum y para su uso. Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos (Schwaber y Sutherland, 2016).

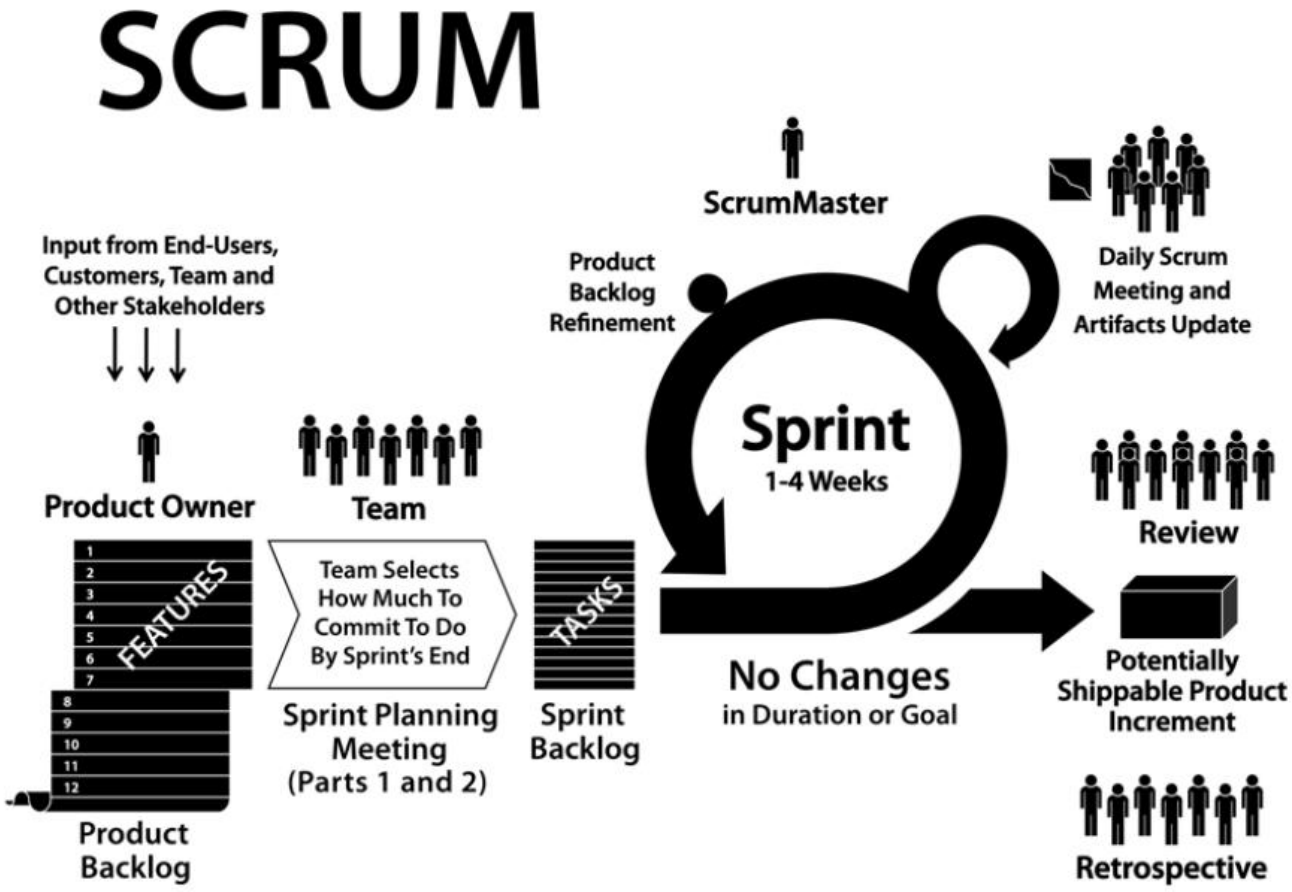

Fig. 8 Scrum marco de trabajo *(Sutherland, 2011, p. 16)*

3.6 Teoría de Scrum

Scrum se basa en la teoría de control de procesos empíricos. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación (Schwaber y Sutherland, 2016).

Transparencia: los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se están viendo. Por ejemplo: Todos los participantes deben compartir un lenguaje común para referirse al proceso; y, aquellos que desempeñan el trabajo y aquellos que aceptan el producto de dicho trabajo deben compartir una definición común de "Terminado" (Schwaber y Sutherland, 2016).

Inspección: hacia un objetivo para detectar variaciones indeseadas. La inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos en el mismo lugar de trabajo (Schwaber y Sutherland, 2016).

Adaptación: si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables y que el producto resultante será inaceptable, el proceso o el material que está siendo procesado deben ajustarse. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores (Schwaber y Sutherland, 2016).

3.7 Valores de Scrum

Cuando el Equipo Scrum incorpora y vivencia los valores de compromiso, coraje, foco, apertura y respeto, los pilares Scrum de transparencia, inspección y adaptación se materializan y fomentan la confianza en todo el mundo. Los miembros del Equipo Scrum aprenden y exploran estos valores a medida que trabajan en los eventos, roles y artefactos de Scrum. El uso exitoso de Scrum depende de que las personas lleguen a ser más virtuosas en la convivencia con estos cinco valores. Las personas se comprometen de manera individual a alcanzar las metas del Equipo Scrum. Los miembros del Equipo Scrum tienen la capacidad para hacer bien las cosas y para

trabajar en los problemas difíciles. Todos se enfocan en el trabajo del Sprint y en las metas del Equipo Scrum. El Equipo Scrum y sus interesados acuerdan estar abiertos a todo el trabajo y a los desafíos que se les presenten al realizar su trabajo. Los miembros del equipo Scrum se respetan entre sí para ser personas capaces e independientes (Schwaber y Sutherland, 2016).

3.8 Principios de Scrum

Alaimo (2013), argumenta que Scrum es el modelo más utilizado dentro de las metodologías ágiles. Muchos de los valores y principios del Manifiesto Ágil tienen su origen en Scrum. A continuación se presentan los principios desde la óptica de Scrum:

1. Individuos e interacciones por sobre procesos y herramientas. Scrum se apoya en la confianza hacia las personas, sus interacciones y los equipos. Los equipos identifican lo que hay que hacer y toman la responsabilidad de hacerlo, removiendo todos los impedimentos que encuentren en su camino y estén a su alcance. Los equipos trabajan en conjunto con otras partes de la organización cuando los impedimentos están fuera de su ámbito de control.
2. Software funcionando por sobre documentación exhaustiva. Scrum requiere que al final de cada Sprint se entregue un producto funcionando. La documentación es entendida, en Scrum, como un producto intermedio sin valor de negocio. Los equipos pueden documentar tanto como crean necesario, pero ninguno de estos documentos pueden ser considerados como el resultado de un Sprint. El resultado de un Sprint es, nuevamente, el producto funcionando. El progreso del proyecto se mide en base al producto funcionando que se entrega iterativamente.

3. Colaboración con el cliente por sobre la negociación de contratos. El Scrum Product Owner es el responsable de la relación que existe con los usuarios finales, stakeholders y áreas de la organización que van a obtener el beneficio del producto. El Scrum Product Owner es parte del Equipo Scrum y trabaja colaborativamente con el resto de los individuos dentro del equipo para asegurarse que el producto construido tenga la mayor cantidad posible de valor al final de cada iteración.
4. Respuesta al cambio por sobre el seguimiento de un plan. Scrum, por diseño, se asegura que todo el mundo dentro de un equipo tenga toda la información necesaria para poder tomar decisiones informadas sobre el proyecto en cualquier momento. El progreso es medido al final de cada Sprint mediante software funcionando y la lista de características pendientes está visible continuamente y para todos los miembros. Esto permite que el alcance del proyecto cambie constantemente en función de la retroalimentación provista por los stakeholders. Fomentar el cambio es una ventaja competitiva.

3.9 Roles de Scrum

3.9.1 Product Owner (Dueño del Producto)

Schwaber y Sutherland (2016), describe al Product Owner (Dueño de Producto) como el responsable de maximizar el valor del producto y el trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos. El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto Product Backlog). La gestión de la Lista del Producto incluye:

1. Expresar claramente los elementos de la Lista del Producto;
2. Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible;
3. Optimizar el valor del trabajo que el Equipo de Desarrollo realiza;
4. Asegurar que la Lista del Producto es visible, transparente y clara para todos y que muestra aquello en lo que el equipo trabajará a continuación; y,
5. Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario.

El Dueño de Producto podría hacer el trabajo anterior o delegarlo en el Equipo de Desarrollo. Sin embargo, en ambos casos el Dueño de Producto sigue siendo el responsable de dicho trabajo.

3.9.2 Scrum Master

Schwaber y Sutherland (2016), describen al Scrum Master como el responsable de asegurar que Scrum se entienda y se adopte. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. Es un líder que está al servicio del equipo Scrum. El Scrum Master ayuda a las personas externas al equipo Scrum a entender qué interacciones con el equipo Scrum pueden ser útiles y cuáles no. Ayuda a todos a modificar estas interacciones para maximizar el valor creado por el equipo Scrum.

El servicio del Scrum Master al Dueño del Producto:

- Encontrar técnicas para gestionar la Lista de Producto de manera efectiva;
- Ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Lista de Producto claros y concisos;
- Entender la planificación del producto en un entorno empírico;

- Asegurar que el Dueño de Producto conozca cómo ordenar la Lista de Producto para maximizar el valor;
- Entender y practicar la agilidad; y,
- Facilitar los eventos de Scrum según se requiera o necesite.

El servicio del Scrum Master al Equipo de Desarrollo:

- Guiar al Equipo de Desarrollo en ser autoorganizado y multifuncional;
- Ayudar al Equipo de Desarrollo a crear productos de alto valor;
- Eliminar impedimentos para el progreso del Equipo de Desarrollo;
- Facilitar los eventos de Scrum según se requiera o necesite; y,
- Guiar al Equipo de Desarrollo en entornos organizacionales en los que Scrum aún no haya sido adoptado y entendido por completo.

El servicio del Scrum Master a la organización:

- Liderar y guiar a la organización en la adopción de Scrum;
- Planificar las implementaciones de Scrum en la organización;
- Ayudar a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto;
- Motivar cambios que incrementen la productividad del Equipo Scrum; y,
- Trabajar con otros Scrum Masters para incrementar la efectividad de la aplicación de Scrum en la organización.

3.9.3 Development Team (Equipo de Desarrollo)

Schwaber y Sutherland (2016), describen al Equipo de Desarrollo como profesionales que realizan el trabajo de entregar un incremento de producto “Terminado” que potencialmente se pueda poner en producción al final de cada Sprint. Solo los miembros del equipo de desarrollo participan en la creación del Incremento. La organización es la encargada de estructurar y empoderar a los equipos de desarrollo para que estos organicen y gestionen su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del equipo de desarrollo. Los equipos de desarrollo tienen las siguientes características:

- Son auto organizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos de funcionalidad potencialmente desplegables;
- Los Equipos de Desarrollo son multifuncionales, esto es, como equipo cuentan con todas las habilidades necesarias para crear un Incremento de producto;
- Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla;
- Scrum no reconoce subequipos en los equipos de desarrollo, no importan los dominios particulares que requieran tenerse en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla; y,
- Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

El tamaño óptimo del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. Tener menos de tres miembros en el Equipo de Desarrollo reduce la interacción y resulta en ganancias de productividad más pequeñas. Los Equipos de Desarrollo más pequeños podrían encontrar limitaciones en cuanto a las habilidades necesarias durante un Sprint, haciendo que el Equipo de Desarrollo no pudiese entregar un Incremento que potencialmente se pueda poner en producción. Tener más de nueve miembros en el equipo requiere demasiada coordinación. Los Equipos de Desarrollo grandes generan demasiada complejidad como para que pueda gestionarse mediante un proceso empírico. Los roles de Dueño de Producto y Scrum Master no cuentan en el cálculo del tamaño del equipo a menos que también estén contribuyendo a trabajar en la Lista de Pendientes de Sprint (Sprint Backlog).

3.10 Artefactos

Schwaber y Sutherland (2016), exponen que los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

3.10.1 Backlog del Producto

Schwaber y Sutherland (2016), argumentan que la lista de Producto es una lista ordenada de todo lo que podría ser necesario en el producto y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto (Product Owner) es el responsable de la Lista de Producto, incluyendo su contenido, disponibilidad y ordenación. Una Lista de Producto nunca está completa. El desarrollo más temprano de la misma solo refleja los

requisitos conocidos y mejor entendidos al principio. La Lista de Producto evoluciona a medida de que el producto y el entorno en el que se usará también lo hacen. La Lista de Producto es dinámica; cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Mientras el producto exista, su Lista de Producto también existe. La Lista de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a realizarse sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, el orden, la estimación y el valor. A medida que un producto es utilizado y se incrementa su valor y el mercado proporciona retroalimentación, la Lista de Producto se convierte en una lista más larga y exhaustiva. Los requisitos nunca dejan de cambiar así que la Lista de Producto es un artefacto vivo. Los cambios en los requisitos de negocio, las condiciones del mercado o la tecnología podrían causar cambios en la Lista de Producto. A menudo, varios Equipos Scrum trabajan juntos en el mismo producto.

Schwaber y Sutherland (2016), exponen que para describir el trabajo a realizar sobre el producto se utiliza una única Lista de Producto. En ese caso podría emplearse un atributo de la Lista de Producto para agrupar los elementos. El refinamiento (refinement) de la Lista de Producto es el acto de añadir detalle, estimaciones y orden a los elementos de la Lista de Producto. Se trata de un proceso continuo en el cual el Dueño de Producto y el Equipo de Desarrollo colaboran acerca de los detalles de los elementos de la Lista de Producto. Durante el refinamiento de la Lista de Producto, se examinan y revisan sus elementos. El Equipo Scrum decide cómo y cuándo se hace el refinamiento. Este usualmente consume no más del 10% de la capacidad del Equipo de Desarrollo. Sin embargo, los elementos de la Lista de Producto pueden actualizarse en cualquier momento por el Dueño de Producto o a criterio suyo. Los elementos de la Lista de Producto de orden más alto son generalmente más claros y detallados que los de

menor orden. Se realizan estimaciones más precisas basándose en la mayor claridad y detalle; cuanto más bajo es el orden, menor es el detalle. Los elementos de la Lista de Producto de los que se ocupará el Equipo de Desarrollo en el siguiente Sprint tienen una granularidad mayor, habiendo sido descompuestos de forma que cualquier elemento pueda ser "Terminado" dentro de los límites del bloque de tiempo del Sprint. Los elementos de la Lista de Producto que pueden ser "Terminados" por el Equipo de Desarrollo en un Sprint son considerados "Preparados" o "accionables" para ser seleccionados en una reunión de Planificación de Sprint. Los elementos de la Lista de Producto normalmente adquieren este grado de transparencia mediante las actividades de refinamiento descritas anteriormente. El Equipo de Desarrollo es el responsable de proporcionar todas las estimaciones. El Dueño de Producto podría influenciar al Equipo ayudándoles a entender y seleccionar sus compromisos, pero las personas que harán el trabajo son las que hacen la estimación final.

3.10.2 Items del Backlog del producto (PBIs)

James (2012), describe a los Items del Backlog del producto como la herramienta que especifica el qué más, que el cómo, de una característica centrada en el cliente a menudo escrita en forma de Historia de Usuario tiene una definición de "Terminado" abarcadora de todo el producto para evitar la deuda técnica. Puede tener criterios de aceptación específicos del ítem. El esfuerzo es calculado por el equipo, de preferencia en unidades relativas (por ejemplo, puntos de la historia). El esfuerzo es de aproximadamente 2 ó 3 personas, y entre 2 ó 3 días, o menos para equipos más avanzados.

3.10.3 Backlog del Sprint

James (2012), argumenta que son PBIs comprometidos negociados entre el equipo y el Product Owner durante la Reunión de Planificación del Sprint. El alcance comprometido es fijo durante la ejecución del Sprint. Las tareas iniciales son identificadas por el equipo durante la reunión de planificación del Sprint. El equipo descubrirá las tareas adicionales necesarias para cumplir con el compromiso de alcance fijo durante la ejecución de Sprint. Visible para el equipo y es referenciado durante la Daily Scrum Meeting.

3.10.4 Tareas del Sprint

James (2012), argumenta que las Tareas del Sprint especifican cómo alcanzar el qué del PBI. Requiere aproximadamente un día de trabajo. El esfuerzo restante se re-estima a diario, por lo general en horas. Durante la ejecución de Sprint, una persona de contacto puede ofrecerse para ser el principal responsable de una tarea. Propiedad de todo el equipo, se espera colaboración.

3.10.5 Sprint Burn Down Chart

James (2012), describe las características del Sprint Burn Down Chart. La Fig. 9 muestra un ejemplo de una tabla de tipo Sprint Burn Down Chart.

- Indica el total de horas restantes de las tareas del equipo dentro de un Sprint.
- Es un re-estimado a diario, por lo tanto puede subir antes de bajar.
- Diseñado para facilitar la auto-organización del equipo.
- Algunas variaciones, como agrupar por persona de contacto o agregando tendencias, tiende a reducir su efectividad para fomentar la colaboración.
- Es una buena idea implementarlo en los primeros días de Scrum, pero en la práctica ha sido a menudo mal utilizada como un informe de gestión, invitando a

la intervención. El Scrum Master debe descontinuar el uso de esta tabla si se convierte en un impedimento para la auto organización del equipo.

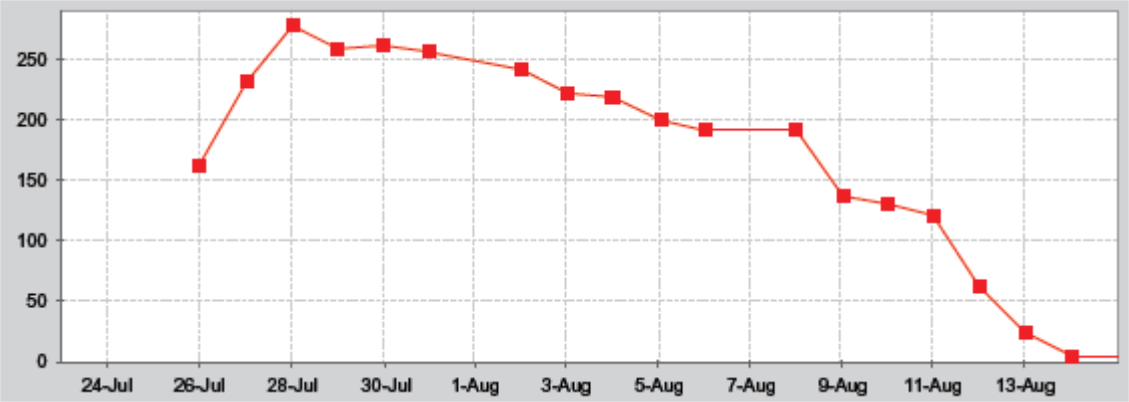

Fig. 9 Ejemplo de Sprint Burn Down Chart.

3.10.6 Product Release Burn Down Chart

James (2012), lista tres características de este artefacto:

1. Realiza un seguimiento del esfuerzo restante del Product Backlog de un Sprint al próximo.
2. Puede utilizar unidades relativas como Puntos de Historia en el eje Y.
3. Utiliza tendencias históricas para ajustar las previsiones.

3.11 Reuniones

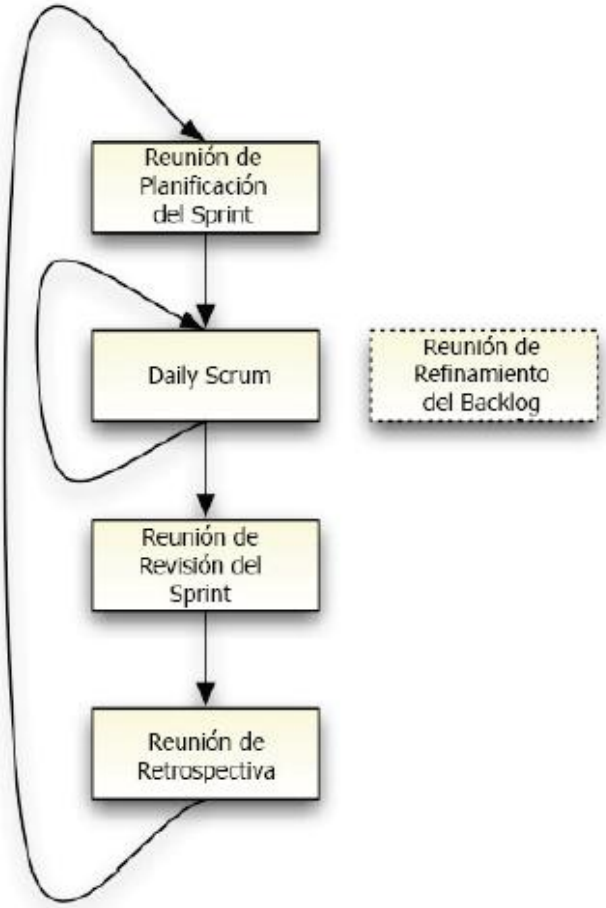

Fig. 10 Diagrama que muestra el flujo de las reuniones en Scrum.

Todas las reuniones de Scrum son facilitadas por el ScrumMaster, quien no tiene autoridad para tomar decisiones en ellas (James, 2012). La Fig. 10 muestra el diagrama de flujo de las reuniones en Scrum.

3.11.1 Planificación del Sprint

James (2012), describe la planificación del Sprint:

- Al comienzo de cada Sprint, el Product Owner y el equipo tienen una Reunión de Planificación del Sprint donde negocian qué ítems del Backlog del Producto intentarán convertir en producto funcionando durante el Sprint. El Product Owner

es el responsable de declarar cuáles son los ítems más importantes para el negocio. El equipo es responsable de seleccionar la cantidad de trabajo que cree que podrán realizar sin acumular deuda técnica. El equipo "toma" el trabajo desde el Product Backlog hacia el Sprint Backlog.

- Cuando los equipos se encuentran frente a un trabajo complejo con una incertidumbre inherente, deben trabajar juntos para intuitivamente estimar su capacidad para comprometerse con los ítems, mientras aprenden de los sprints anteriores. Planificar su capacidad por hora y comparar sus estimaciones con datos reales hace que el equipo simule precisión y reduzca su sentido de responsabilidad con de sus compromisos. A menos que el trabajo sea realmente predecible, se debe descartar esas prácticas dentro de los primeros Sprints o evitarlos por completo.
- Hasta que un equipo haya aprendido a lograr un incremento del producto potencialmente entregable en cada Sprint, se debe reducir la cantidad de funcionalidad comprometida. El fracaso en cambiar viejos hábitos conduce a la deuda técnica y la muerte eventual del diseño.
- Si la parte superior del Backlog del Producto no ha sido refinada, una parte importante de la reunión de planificación se debe dedicar a esto, tal como se describe en la sección de la Reunión de Refinamiento del Backlog.
- Hacia el final de la Reunión de Planificación del Sprint, el equipo desglosa los ítems seleccionados en una lista inicial de tareas del Sprint (Sprint Tasks) y hace un compromiso final para realizar el trabajo. El tiempo máximo asignado

(también conocido como timebox) para la planificación de un Sprint de 30 días, es de ocho horas, reducido proporcionalmente en caso de un Sprint más corto.

3.11.2 Daily Scrum y ejecución del Sprint

James (2012), describe las actividades del Daily Scrum y la ejecución del Sprint:

- Cada día, a la misma hora y en el mismo lugar, los miembros del equipo de desarrollo pasan 15 minutos reportándose entre sí. Cada miembro del equipo resume lo que hizo el día anterior, lo que hará hoy, y qué impedimentos está enfrentando.
- Mantenerse de pie en el Daily Scrum ayudará a que sea breve. Los temas que requieren atención adicional pueden ser discutidos por los interesados después de que cada miembro del equipo haya reportado. Al equipo puede resultarle útil mantener una lista de Tareas del Sprint (Sprint Task List), un Sprint Burndown Chart, y un Listado de Impedimentos. Durante la ejecución del Sprint es común descubrir tareas adicionales necesarias para alcanzar las metas del Sprint. Los impedimentos causados por los problemas que escapan al control del equipo se consideran impedimentos organizacionales.
- Para el Product Owner, casi siempre es útil asistir a la Daily Scrum. Pero cuando alguno de los asistentes también es el jefe del equipo, el efecto del arma invisible impide la auto-organización y el liderazgo emergente. Las personas que carecen de la experiencia real de la auto organización no verán este problema, al igual que los peces no son conscientes del agua. Por el contrario, un equipo que necesita experiencia adicional en los requisitos del producto se beneficiará de una mayor participación del Product Owner, incluyendo la asistencia a la Daily Scrum.

- La Daily Scrum tiene la intención de alterar los viejos hábitos del trabajo por separado. Los miembros deben permanecer atentos para detectar signos de la antigua estrategia. Por ejemplo, mirar sólo en el Scrum Master cuando se habla es un síntoma de que el equipo no ha aprendido a operar como una entidad auto-organizativa.

3.11.3 Reunión de revisión del Sprint

James (2012), describe la revisión del Sprint:

- Después de la ejecución del Sprint, el equipo mantiene una reunión de revisión del Sprint para mostrar un incremento del producto al Product Owner y a todos los demás interesados.
- La reunión debe ofrecer una demostración en vivo, no un informe.
- Después de la demostración, el Product Owner revisa los compromisos contraídos en la Reunión de Planificación del Sprint y declara qué ítems considera terminados. Por ejemplo, un ítem de software que es meramente "código completado" no se considera terminado, porque el software no probado no es entregable. Los elementos incompletos son devueltos al Backlog del Producto y clasificados de acuerdo a las prioridades del Product Owner como candidatos para futuros Sprints.
- El Scrum Master ayuda al Product Owner y a los stakeholders a convertir su feedback en los nuevos Ítems del Product Backlog o Product Backlog Items (PBIs) para su priorización por el Product Owner. A menudo, descubrir nuevo alcance supera la tasa del equipo de desarrollo. Si el Product Owner determina

que el alcance recién descubierto es más importante que las expectativas originales, el nuevo alcance desplaza el viejo alcance en el Product Backlog.

- La reunión de revisión de Sprint es el encuentro apropiado para los stakeholders (incluso a los usuarios finales). Es la oportunidad de revisar y adaptar el producto a medida que emerge, y de forma iterativa refinar la comprensión de cada uno de los requisitos. Nuevos productos, especialmente productos de software, son difíciles de visualizar en el vacío. Muchos clientes necesitan ser capaces de reaccionar ante una pieza de software que funcione para descubrir lo que realmente quieren. El desarrollo iterativo, un enfoque dirigido por el valor, permite la creación de productos que no podrían haber sido especificados por adelantado en un enfoque dirigido por planes.

3.11.4 Reunión de retrospectiva del Sprint

James (2012), lista puntos importantes respecto a la Reunión de retrospectiva del Sprint:

- Cada Sprint finaliza con una retrospectiva. En esta reunión, el equipo reflexiona sobre su propio proceso. Inspeccionan su comportamiento y adoptan las medidas para adaptarlo en los futuros Sprints.
- Los Scrum Masters dedicados encontrarán alternativas a las reuniones obsoletas, temidas, que todo el mundo espera.
- Una retrospectiva en profundidad requiere de un ambiente de seguridad psicológica que no se encuentra en la mayoría de las organizaciones. Sin seguridad, la discusión retrospectiva o bien evitará los problemas incómodos, o se deteriorará en la culpa y la hostilidad. Un impedimento común para la plena transparencia en el equipo es la presencia de personas que realizan evaluaciones

de desempeño. Otro impedimento para una retrospectiva profunda es la tendencia humana a sacar conclusiones y proponer acciones con rapidez. Un tercer impedimento a la seguridad psicológica es la distribución geográfica. Equipos dispersos geográficamente por lo general no colaboran, así como aquellos que comparten una sala de equipo.

- Las retrospectivas a menudo exponen impedimentos organizacionales. Una vez que un equipo ha resuelto los obstáculos dentro de su influencia inmediata, el Scrum Master debe trabajar para expandir esa influencia.
- El Scrum Master debe utilizar una variedad de técnicas para facilitar retrospectivas, incluyendo la redacción en silencio, líneas de tiempo, e histogramas de satisfacción. En todos los casos, el objetivo es lograr una comprensión común de múltiples perspectivas y desarrollar acciones que llevarán al equipo al siguiente nivel.

3.11.5 Reunión de refinamiento de Backlog

James (2012), argumenta los siguientes puntos respecto a las reuniones de refinamiento de Backlog:

- La mayoría de los PBIs inicialmente deben refinarse, ya que son grandes y poco comprendidos. Los equipos han encontrado útil tomar un tiempo de la ejecución de cada Sprint para preparar el Backlog del Producto para la próxima Reunión de Planificación del Sprint.
- En la reunión de refinamiento del Backlog, el equipo estima la cantidad de esfuerzo que se debe invertir para completar los Ítems del Backlog del Producto y proporciona información técnica para ayudar al Product Owner a priorizarlos.

- Los grandes ítems se dividen y se clarifican, teniendo en cuenta temas tanto de negocio como técnicos. A veces, un subconjunto del equipo, junto con el Product Owner y los stakeholders, descomponen y dividen los ítems del Backlog del Producto antes de involucrar a todo el equipo en la estimación.
- Un Scrum Master experimentado puede ayudar al equipo a identificar finas rebanadas verticales de trabajo que tienen valor comercial, mientras promueve una definición rigurosa de "Terminado" que incluye las pruebas adecuadas y refactorización.
- Es común escribir los PBIs en forma de Historias de Usuario. En este enfoque, los PBIs de gran tamaño son llamados Epics. El desarrollo tradicional rompe las características en tareas horizontales (que se asemeja a fases en cascada) que no pueden ser priorizadas de manera independiente y que carecen de valor del negocio desde la perspectiva del cliente. Este hábito es difícil de cambiar.
- La agilidad requiere aprender a dividir las Historias de Usuario que representan características más pequeñas del producto.
- Como la mayoría de los clientes no utilizan la mayoría de las características de los productos, es conveniente dividir las Epics para entregar las historias más importantes primero. Aunque la entrega de características de menor valor más adelante es probable que incluya algún re trabajo, la repetición es mejor que la ausencia de trabajo.
- La reunión de refinamiento carece de nombre oficial y también es llamada "Backlog Grooming", "Mantenimiento del Backlog" o "Story Time".

4 Metodología de Investigación

4.1 Matriz de congruencia

Tabla 4

Matriz de congruencia que detalla, el problema general, objetivos específicos, metodología a emplear variables y dimensiones.

Elaboración de propuesta de una guía de implementación de Scrum para una empresa salvadoreña, un caso de estudio.					
Problema General	Objetivos específicos	Variables y dimensiones		Metodología a emplear	
Construir una propuesta de guía que sirva para implementar la metodología de desarrollo ágil SCRUM en una empresa salvadoreña.	• Realizar una revisión bibliográfica para comprender el por qué de la transición de las metodologías tradicionales hacia las metodologías ágiles. • Estudiar los dominios de complejidad y clasificar el contexto de la empresa a partir de las categorías existentes. • Documentar y establecer las características de la metodología ágil SCRUM. • Perfilar el caso de empresa salvadoreña, conocer su nivel de organización, situación y necesidades actuales. • Proponer una guía de implementación de la metodología ágil Scrum y optimizar su proceso actual de desarrollo de software.	*Independiente*	Metodología Scrum	*Alcance de la investigación*	Descriptivo
		Dependiente	Perfil de la empresa	*Tipo de investigación*	Cualitativo
		Intervinientes	Cultura organizacional de la empresa	*Técnicas de análisis*	Inductivo
				Técnicas de recopilación de información: • Entrevistas semi-estructuradas • Observación • Revisión literaria	

4.2 Objetivos de la investigación

1. Realizar una revisión bibliográfica para comprender el porqué de la transición de las metodologías tradicionales hacia las metodologías ágiles.
2. Estudiar los dominios de complejidad y clasificar el contexto de la empresa a partir de las categorías existentes.

3. Documentar y establecer las características de la metodología ágil SCRUM.
4. Perfilar el caso de empresa salvadoreña, conocer su nivel de organización, situación y necesidades actuales.
5. Proponer una guía de implementación de la metodología ágil Scrum y optimizar su proceso actual de desarrollo de software.

4.3 Diseño de la investigación

El diseño de la investigación está compuesto de una revisión bibliográfica, que va desde las metodologías tradicionales, la transición de lo secuencial a lo iterativo incremental, metodologías ágiles hasta concluir con el marco de trabajo Scrum. Posteriormente se realiza un estudio enmarcado a la empresa salvadoreña, con enfoque cualitativo, exploratorio y descriptivo para determinar su situación y problemas actuales, conocer los puntos de vista de los entrevistados mediante entrevistas semi-estructurada y las necesidades de optimización de los procesos actuales de desarrollo de software.

4.4 Instrumentos de Investigación

4.4.1 Entrevistas a los expertos

Los objetivos que se persiguen al entrevistar a los expertos en la metodología son los siguientes:

1. Conocer las buenas prácticas de implementación de la metodología Scrum en base a sus experiencias.
2. Validar la guía de implementación propuesta.
3. Retroalimentar la guía mediante las recomendaciones hechas por parte de los expertos.

La estructura de la entrevista semi-estructurada está compuesta por tres rubros:

1. Buenas prácticas para implementar la metodología.
2. Validación de la metodología.
3. Preguntas a los expertos.

Perfil de Entrevistados:

1. Ingeniero en Sistemas o Industrial (o carreras afines).
2. Cuatro o cinco años en la dirección de proyectos.
3. Haber participado en proyectos de implementación de la metodología Scrum como Product Owner o Scrum Master.

El resumen de los resultados de la entrevista con los expertos se muestra en la Tabla 8.

Tabla 5

Preguntas de la entrevista dirigida a los expertos Scrum.

Rubro	Pregunta	Objetivo
Buenas prácticas para implementar la metodología.	¿Cuáles considera que son factores de éxito para implementar Scrum?	Proponer una guía de implementación de la metodología ágil Scrum y optimizar su proceso actual de desarrollo de software.
	¿Qué estrategias ha implementado para abordar el tema de la cultura organizacional y la renuencia al cambio, respecto a la metodología Scrum?	
	¿Cuáles considera que son los factores de éxito que pueden influir positivamente en el desarrollo de proyectos con Scrum?	Documentar y establecer las características de la metodología ágil SCRUM.
	En base a su experiencia, ¿qué factores negativos han influenciado para que las tareas de desarrollo no puedan finalizar en el tiempo estimado?	
	¿Cuál ha sido su estrategia para planificar las reuniones con el equipo de desarrollo y el Product Owner, sin que afecte negativamente el progreso de las actividades?	
	¿Qué estrategias de comunicación han sido exitosas con los clientes e interesados durante la ejecución del proyecto?	

Tabla 5 (Continuación)

Preguntas de la entrevista dirigida a los expertos Scrum.

Rubro	Pregunta	Objetivo
Validación de la metodología	¿Qué tipos de empresa pueden adoptar la metodología Scrum? ¿La metodología SCRUM puede ser aplicable a cualquier tipo de proyecto de desarrollo de software, pequeño, mediano o grande?	Estudiar los dominios de complejidad y clasificar el contexto de la empresa a partir de las categorías existentes.
	¿Cómo perfilaría al recurso humano, para otorgar los roles: Product Owner, Scrum Master y Desarrollador?	Proponer una guía de implementación de la metodología ágil Scrum y optimizar su proceso actual de desarrollo de software.
Recomendaciones	¿Cuál es la cantidad mínima de recurso humano que debe existir en un departamento de desarrollo de sistemas, para la implementación de Scrum? ¿Considera importante la implementación de una herramienta que permita la gestión de proyectos ágiles? Scrum valora el Software en funcionamiento sobre la documentación, pero durante las fases del ciclo implementadas, ¿qué puntos son importantes a documentar? En base a su experiencia, ¿cuáles han sido beneficios al implementar Scrum y cuál ha sido su estrategia para comunicar los resultados a la gerencia de la empresa e interesados? Ante múltiples proyectos en la organización, ¿cómo configuraría los equipos Scrum?	Proponer una guía de implementación de la metodología ágil Scrum y optimizar su proceso actual de desarrollo de software.
	Para describir los requerimientos, ¿considera necesario solo el uso de Historias de Usuario o Historias de Usuarios complementadas con Casos de Uso? ¿Qué estrategia han utilizado en sus Equipos de Desarrollo para realizar una estimación de tiempos apegada a la realidad?	Documentar y establecer las características de la metodología ágil SCRUM.
	Desde su punto de vista, ¿Considera que los roles de Scrum deben ser estáticos y pertenecer a una estructura organizativa, o deben ser dinámicos e independientes de la estructura organizativa de la empresa?	Proponer una guía de implementación de la metodología ágil Scrum y optimizar su proceso actual de desarrollo de software

4.4.2 Entrevistas a la empresa

Los objetivos que se persiguen al entrevistar a los empleados del departamento de desarrollo de la empresa son los siguientes:

1. Conocer la situación actual desde la óptica de los desarrolladores y coordinadores del área de desarrollo de software.
2. Identificar problemas actuales con la forma de trabajo que realizan en el departamento
3. Explorar el nivel de conocimiento de la metodología.

La estructura de la entrevista semi-estructurada está compuesta de tres rubros:

1. Metodología de desarrollo actual.
2. Problema actual.
3. Nivel de conocimientos de Scrum.
4. Expectativas de la metodología.

Perfil de Entrevistados

Esta entrevista ha de realizarse a todos los miembros que conforman el equipo de desarrollo, a los diferentes niveles de la estructura organizativa que tiene la empresa. Los entrevistados pertenecen a dos grupos: coordinadores de desarrollo y los desarrolladores de sistemas. En total se contabilizan cuatro coordinadores de desarrollo y nueve desarrolladores de software.

El resumen de los resultados de la entrevista con los miembros de la empresa se muestra en la Tabla 7.

Tabla 6

Preguntas de entrevista dirigidas a los empleados de la empresa.

Rubro	Pregunta	Objetivo
Metodología de desarrollo actual	¿Qué tipo de metodología de desarrollo de software utilizan?	Perfilar el caso de empresa salvadoreña, conocer su nivel de organización, situación y necesidades actuales
	Al momento de iniciar un proyecto de desarrollo nuevo, ¿cuentan con una metodología de trabajo establecida, podría describirla?	
	¿La metodología actual responde a las necesidades de la empresa?	
Problema actual	¿Considera que la metodología actual involucra a todos los miembros del equipo de forma integral en todas las fases del proyecto?	
	Describa cuales son las ventajas y desventajas de la metodología actual	
	¿Considera acertada la estimación de tiempos al planificar las actividades de desarrollo?	
	¿En qué fases del desarrollo del proyecto se concentra la documentación?	
	¿Cómo gestionan el cambio de requerimientos una vez el proyecto está en marcha?	
Nivel de conocimientos de Scrum	¿Qué conocimientos posee de la metodología Scrum?	
	¿Ha participado en proyectos de desarrollo, ejecutados bajo la metodología Scrum?	
	¿Qué rol desempeñó y cuáles son sus experiencias al respecto?	
	¿Posee algún diplomado o certificación sobre la metodología Scrum?	
Expectativas de la metodología	¿Cuáles son sus expectativas al adoptar esta metodología?	
	¿Considera que la organización se encuentra preparada para implementar esta metodología? Describa porque si o por qué no	

4.4.3 Síntesis de resultados de la entrevista a la empresa.

Perfil del desarrollador de sistemas.

La empresa describe las metas y competencias de la siguiente manera:

1. Brindar soporte a sistemas de la organización local y regional.
2. Desarrollar mejoras a los sistemas informáticos.

3. Realizar desarrollo de software.
4. Realizar investigación de tecnología nueva.
5. Identificar requerimientos.
6. Implementar sistemas informáticos.
7. Monitoreo de aplicaciones realizadas
8. Brindar capacitaciones a usuarios y supe usuarios.
9. Generar modificaciones en aplicaciones para que se apeguen a cambios de procesos.
10. Innovación de aplicaciones para aumentar la optimización de procesos manuales de producción.

Perfil del coordinador de desarrollo de sistemas.

La empresa describe las metas y competencias de la siguiente manera:

1. Trabajar en mesas de procesos para el área correspondiente, en equipo con los directores, gerentes y usuarios para optimizar el desarrollo e implementación de sistemas informáticos, en función de los objetivos de la empresa.
2. Identificar y promover oportunidades de mejora en los procesos de las diferentes áreas de la compañía.
3. Planificar, organizar y controlar el desarrollo e implementación de sistemas informáticos.
4. Evaluar el avance de los proyectos y asegurar que los mismos cumplan con la metodología de desarrollo de proyectos.
5. Atender y resolver consultar de los usuarios, sobre el desarrollo o implementación de los sistemas informáticos.

6. Supervisar al personal y las actividades del área bajo su responsabilidad.
7. Promover la capacitación del personal con relación a la operación del área correspondiente.
8. Evaluar el desempeño del personal a su cargo.
9. Brindar apoyo al área asignada a cargo.

Tabla 7

Muestra el resumen del resultado de las entrevistas con los miembros de la empresa.

Rubro	Resumen de resultados
Metodología de desarrollo actual	• La metodología actual que utilizan es una mezcla entre el enfoque de cascada e iterativo. • Descripción de la metodología actual: Todo inicia con los requerimientos de un área de la empresa, este requerimiento puede ser una nueva funcionalidad que pretende optimizar un proceso manual, o un cambio a una aplicación existente se define la situación actual. El analista de procesos es quien se encarga de realizar una caracterización de proyectos donde se establecen las reglas del negocio y alcances, una vez finalizado este documento es enviado a un comité de desarrollo quienes son los encargados de priorizar los requerimientos. Esta lista priorizada es enviada al coordinador de desarrollo y en conjunto a los desarrolladores se hace un análisis y diseño de la solución, una vez la solución es planteada mediante un documento, este es planificado y las tareas son asignadas a los desarrolladores de sistemas. Cuando el requerimiento es finalizado en su totalidad, este es probado por una unidad de testeo interna, posterior a esa prueba se realiza otra con el usuario final y si cumple con la funcionalidad requerida es aprobado y pasa a producción, de lo contrario se documentan las observaciones y es retornado al coordinador de desarrollo para volver analizar el requerimiento y repetir los pasos desde ese punto. • La metodología actual responde a las necesidades de la empresa parcialmente ya que no es tan rápido el proceso como esperan, sin embargo cada etapa es importante y no se pueden obviar tareas administrativas como planificación, análisis, y pruebas. Dependiendo de la complejidad del requerimiento este es entregado en un tiempo establecido, el proyecto no recibe retroalimentación por parte del cliente o usuario final ya que éste puede ver su funcionalidad hasta el final del desarrollo del requerimiento, esto conlleva a atrasos con los planes de implementación ya que no puede salir en vivo si este no es aprobado.

Tabla 7 (Continuación)

Muestra el resumen del resultado de las entrevistas con los miembros de la empresa.

Rubro	Resumen de resultados

Problema actual	• La metodología actual no involucra a todos los interesados integralmente, ya que el cliente o usuario solo participa en la definición del requerimiento y se vuelve a contactar hasta la etapa de pruebas.
	• Ventajas y desventajas de la metodología actual:
	  o Ventajas:
	  ▪ Los requerimientos se congelan mientras están en proceso de desarrollo.
	  ▪ Existe un comité que prioriza los requerimientos.
	  ▪ La documentación de requisitos es amplia y detallada a nivel de procesos.
	  o Desventajas:
	  ▪ Si existe un cambio de última hora en los requerimientos este es tomado en cuenta hasta el final de las pruebas, muy probablemente el diseño original ha cambiado dejándolo obsoleto.
	  ▪ No se recibe retroalimentación del cliente ya que este es contactado hasta en las etapas de prueba.
	  ▪ La estimación de tiempos no es acertada ya que no todos los desarrolladores poseen la misma experiencia, existen programadores junior y senior.
	  ▪ La documentación es concentrada en las etapas de especificación de requerimientos y pruebas.
	  ▪ Los requerimientos entrantes son gestionados en la etapa de definición de requerimientos y es priorizado por el comité de desarrollo, pero deben esperar a que el requerimiento original haya pasado por las pruebas para dictaminar nuevas observaciones donde es incluido los nuevos cambios a los requerimientos originales.

Nivel de conocimientos	• Actualmente los coordinadores de desarrollo y desarrolladores de sistemas han recibido un diplomado de la metodología Scrum por parte de la universidad Don Bosco en el año 2015. • Ninguno de los integrantes del departamento ha participado en proyectos con la metodología ágil Scrum, toda su experiencia se encuentra a nivel teórico.
Expectativas de la metodología	• Una de las expectativas común por parte del equipo de desarrollo y coordinadores es poder agilizar la incorporación de nuevos requerimientos ya que estos cambios pueden afectar negativamente en el diseño original, provocando retrasos en los tiempos originalmente estimados para la salida en vivo o implementación. • Actualmente la organización tiene dentro de sus valores empresariales el lema “simple y rápido”, por tanto respaldan la agilización de los procesos de desarrollo para generar valor a los objetivos gerenciales mediante la entrega oportuna de tecnología informática a cada una de las áreas de la empresa que lo necesite.

4.4.4 Síntesis de resultados de la entrevista dirigida a los expertos.

Tabla 8

Muestra el resumen del resultado de las entrevistas con los expertos

Rubro	Resumen de resultados
Buenas prácticas para implementar la metodología	• El departamento debe tener apoyo por parte de la gerencia informática así como de los altos mandos para poder implementar y empoderar a los equipos con los roles y desempeño de actividades definidas por el marco de trabajo Scrum. • En primer lugar debe existir el apoyo por parte de los altos mandos, posteriormente debe existir un plan de capacitación hacia el departamento que desea implementar la metodología, contemplar un plan de comunicación con los usuarios finales para explicarles la nueva modalidad de trabajo mostrando sus bondades y como va a contribuir a la agilización del proceso de desarrollo. • Un factor que influye positivamente es el ordenamiento del Backlog agregando prioridades en conjunto al Equipo de Desarrollo por posibles dependencias a nivel técnico. También la comunicación y colaboración interna entre los integrantes del Equipo de Desarrollo ante situaciones donde exista estancamiento de avances por x motivos, documentar casos particulares en una base de conocimientos para Sprints posteriores. • Factores que pueden influir negativamente es dejar a los desarrolladores de sistemas junior que estimen tiempo sin supervisión de todo el Equipo de Desarrollo. En esta etapa de estimación de tiempos es importante la participación de todo el equipo. • Es importante que nadie externo a las reuniones diarias sea participe mas solamente sea observador, solo deben participar los involucrados para evitar discusiones que no aporten valor a la reunión. Dejar que el equipo se auto gestione y exprese todas sus dificultades, no permitir que oculte algún tema relevante por temor a ser señalado, de esta forma se previene el riesgo de alguna actividad que permita el estancamiento y evite el progreso de desarrollo de las tareas por hacer para un ítem del Backlog del Sprint. • El Product Owner juega un papel importante, ya que es quien da la cara por los intereses de los clientes o demás interesados, este rol es quien debe mantener comunicación de cada avance obtenido en cada Sprint y gestionar el Backlog para un posible cambio de prioridades. Se recomienda comunicar los avances a medida los demos del producto finalizado van saliendo del Sprint ya sea mediante reuniones en persona, video conferencia o correos electrónicos.

Tabla 8 (Continuación)

Muestra el resumen del resultado de las entrevistas con los expertos

Rubro	Resumen de resultados
Validación de la metodología	• Cualquier empresa puede adaptar le metodología Scrum y utilizar los procesos que más se adecuen a su contexto, siempre y cuando se respeten los roles, y artefactos. • Cualquier tipo de proyecto pequeño, mediano y grande, pero dependerá de la complejidad de los requerimientos. • Perfilar el recurso humano queda a discreción de la organización y departamento de informática, lo único que se requiere es respetar los roles definidos por la metodología. Se requiere de personal altamente colaborativo, responsable, conocedor del negocio y la metodología. En el caso del Equipo de Desarrollo a parte de los atributos previamente comentados, se requiere que posea experiencia en las tecnologías involucradas para desarrollar el software.
Recomendaciones	• Para el caso del Equipo de Desarrollo es ideal que este conformado por seis personas, el mínimo recomendado es de tres y máximo ocho. Los equipos pueden ser remotos. Scrum crece añadiendo equipos Scrum conformados por Product Owner, Scrum Master y Desarrolladores. • No es indispensable tener una herramienta, basta con tener una pizarra o pared libre, marcadores y post-it para llevar el control del back log del Sprint. Pero si se desea utilizar alguna herramienta puede implementarla. • La documentación como los Requerimientos base, las Historias de Usuario y Casos de uso son importantes, también agrega valor llevar un control de lecciones aprendidas para tomar en cuenta en el desarrollo de siguientes Sprint o proyectos futuros. • El beneficio más importante es la comunicación entre el equipo, la visibilidad constante del progreso de las actividades y el empoderamiento del Equipo de Desarrollo, dentro de sus valores está potenciar a las personas sobre los procesos, son las personas quienes generan valor agregado y calidad, siempre y cuando estén motivadas e inspiradas. Los resultados son palpables con cada Item finalizado, pero una recomendación puede ser comunicar a todos los interesados el avance al finalizar cada Sprint. • Ante múltiples proyectos, lo recomendable es que se disponga de múltiples equipos Scrum para que puedan ser atendidos óptimamente. • Las historias de usuario son importantes porque describen algo que el usuario quiere que su sistema sea capaz de hacer, pero no define la forma procedimental de cómo llegar a ese objetivo, es aquí donde entran los casos de uso. Ambas herramientas se complementan, queda a discreción del equipo implementar ambas o solo una de ellas. Una historia de usuario descrita como proceso, después puede ser reescrita como un Caso de Uso. • Para estimar tiempos es importante que todo el Equipo Scrum esté involucrado, ya que mediante la experiencia colectiva tanto de programadores senior y junior se puede estimar con mayor precisión los tiempos de desarrollo. • Los roles son dinámicos y pueden delegarse según sean las necesidades del proyecto, no se recomienda que sean estáticos y formen parte de una estructura organizativa rígida ya que la disponibilidad del recurso no siempre estará al cien por ciento por factores humanos.

5 Caso de estudio

5.1 Generalidades

La compañía en la cual ha de realizarse el caso de estudio, es una empresa industrial con amplia trayectoria en el desarrollo de productos a ser utilizados para el área de la construcción, posee en la actualidad más de 50 años de experiencia en la fabricación y distribución de sus productos, tanto a nivel nacional como Centroamericano. Para la distribución de sus productos, la empresa posee una línea de comercialización a través de diversos distribuidores así como una cadena de establecimientos propios en los que comercializa sus productos.

Para el registro y control de sus operaciones la empresa tiene la política de desarrollo interno de sus sistemas, ya que con ellos les permite poseer mejor control y flexibilidad de sus aplicaciones, adaptándolas de forma más ágil a las necesidades cambiantes que como organización pueda poseer. Dicha estrategia de trabajo ha llevado a centralizar las actividades de desarrollo en El Salvador, en donde se da solución a los requerimientos y necesidades de las diferentes oficinas con que cuenta a nivel regional.

5.2 Estructura organizativa

Para la atención de los requerimientos de desarrollo de los diferentes departamentos, la Gerencia de Tecnología cuenta con la estructura organizativa siguiente ver Fig. 11:

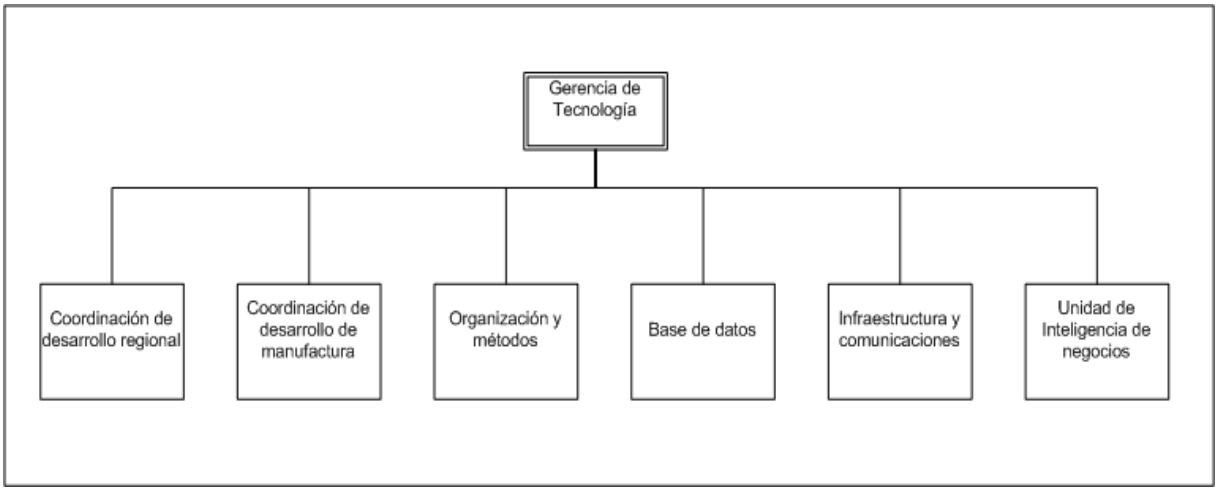

Fig. 11 Estructura organizativa de la gerencia de tecnología.

Gerencia de Tecnología: Área encargada de dirigir conforme a los objetivos estratégicos de la empresa, coordina y delega actividades especializadas a las jefaturas que están bajo su mando, gestiona y crea alianzas con proveedores de servicios tecnológicos, con el fin de brindar un servicio integrado, de alta calidad y disponibilidad de los diversos sistemas con los que cuenta la organización.

Coordinación de Desarrollo Regional: Área que planifica y coordina actividades de desarrollo de software para las áreas administrativas, comerciales, financieras a nivel local como regional. Brinda capacitaciones y soporte de los sistemas que posee la organización a los usuarios finales.

Coordinación de Desarrollo Manufactura: Área que planifica y coordina actividades de desarrollo de software para las áreas de producción. Brinda capacitaciones y soporte de los sistemas a los usuarios finales.

Organización y métodos: Área encargada de desarrollo de proyectos, recopilación, actualización y documentación de procesos en coordinación de los responsables de las diferentes

direcciones de la empresa. El área es la primera línea de comunicación con los usuarios de otras áreas, y las áreas de coordinación de desarrollo de sistemas.

Bases de Datos: Área encargada de administrar, gestionar y garantizar la disponibilidad de las bases de datos que contiene la información que se genera a través de los diferentes sistemas transaccionales de la organización.

Infraestructura y Comunicaciones: Proveer la infraestructura y plataforma tecnológica de comunicación necesaria para el buen funcionamiento de los diferentes sistemas de la empresa, así como garantizar la disponibilidad de los sistemas.

Unidad de Inteligencia de Negocios: Área encargada de desarrollar proyectos de inteligencia de negocios. Ésta área trabaja conjuntamente con las coordinaciones de desarrollo de sistemas, bases de datos, infraestructura y comunicaciones y organización y métodos para generar cubos de información para la toma de decisiones de la alta gerencia.

5.3 Roles por área

Tabla 9

Descripción de puestos de la organización del departamento de informática.

Área	Rol	Descripción de funciones
Gerencia de tecnología	Director de tecnología	• Mantener actualizada a la empresa sobre cambios tecnológicos en cuanto hardware y software. • Responsable de conocer y entender todos los sistemas existentes de la compañía. • Revisar y aprobar análisis, diseños de desarrollo, documentación técnica, mantenimiento de sistemas que son realizados en el área informática. • Evaluar y autorizar los diferentes cambios o adiciones que los coordinadores locales soliciten. • Coordinar el desarrollo e implementación de los proyectos de informática a nivel local y regional. • Elaborar especificaciones para la compra de Hardware y Software. • Asegurar que la empresa tenga el soporte técnico en términos de seguridad en redes LAN y WAN, administración de servicios de correo electrónico y navegación segura. • Realizar mediciones de indicadores de utilización de equipos para proponer actualizaciones planificadas

Tabla 9 (Continuación)

Descripción de puestos de la organización del departamento de informática.

Área	Rol	Descripción de funciones
Gerencia de tecnología	Director de tecnología	• Facilitar información de tipo gerencial de fácil manipulación y acceso. • Presentar informes de avance sobre cambios o nuevos desarrollos de software y cuál será su impacto en los objetivos empresariales.
Unidad de inteligencia de negocio	Coordinador de desarrollo	• Coordinar soluciones de inteligencia de negocio para la toma de decisiones. • Realizar Data Warehouse • Coordinar desarrollo de rutinas de extracción y carga de datos de sistemas transaccionales. • Coordinar el desarrollo de cubos de planeación y gastos de ventas. • Coordinar el desarrollo de métricas por departamento • Capacitar al personal en el uso de herramientas desarrolladas, para la toma de decisiones.
	Desarrollador de sistemas	• Desarrollar soluciones de inteligencia de negocio para la toma de decisiones. • Desarrollo de Datawarehouse. • Desarrollo de rutinas de extracción y carga de datos de los sistemas transaccionales. • Desarrollar cubos de planeación y gastos de ventas.
Organización y Métodos	Analista de procesos	• Liderar el proceso de identificación, recopilación y documentación de procesos y coordinación de todos los responsables de las diferentes direcciones de la empresa. • Gestiona el desarrollo y revisión de proyectos mediante reuniones del área asignada, mediante el análisis de datos y recolección de información que faciliten la toma de decisiones estratégicas de la empresa. • Elaborar y presentar mapas de procesos incluyendo los diferentes escenarios que podrían visualizarse para la evaluación de proyectos. • Realizar trabajo operativo y de campo necesario para el estudio de procesos. • Revisar y validar procesos y procedimientos implementados en los sistemas de la empresa. • Capacitar a los usuarios en los sistemas desarrollados por la empresa.

Tabla 9 (Continuación)

Descripción de puestos de la organización del departamento de informática.

Área	Rol	Descripción de funciones
Coordinación de desarrollo regional	Coordinador de desarrollo	• Planificar, ejecutar y controlar la implementación de sistemas informáticos para el área administrativa y comercial. • Trabajar en mesas de procesos del área, en equipo con los directores, gerentes y usuarios finales para optimizar el desarrollo en función a los objetivos empresariales. • Identificar y promover oportunidades de mejora de los procesos para las áreas de la empresa. • Planificar, ejecutar y controlar el desarrollo de sistemas de información de las áreas a cargo. • Evaluar el avance de los proyectos y asegurar que los mismos cumplan con la metodología de desarrollo implementada en el departamento. • Atender y resolver consultas de los usuarios, sobre el desarrollo e implementación de sistemas. • Supervisar al personal y las actividades del área bajo su responsabilidad. • Evaluar el desempeño del personal a su cargo.
	Desarrollador de sistemas	• Desarrollar sistemas para optimizar procesos dentro de la empresa, administrando y desarrollando proyectos de tecnología en el área comercial y administrativa con el fin de facilitar el trabajo de los colaboradores de la empresa. • Brindar soporte a los usuarios sobre los sistemas de la organización. • Desarrollar mejoras a los sistemas informáticos. • Realizar construcción de software. • Realizar investigación de tecnología nueva. • Brindar soluciones a los diferentes usuarios. • Identificar requerimientos. • Implementar sistemas informáticos. • Monitoreo y control de aplicaciones realizadas. • Cumplir fechas de entrega • Brindar capacitación a usuarios y supe usuarios. • Brindar colaboración en actividades relacionadas a la dirección. • Modificar las aplicaciones para que se apeguen a los cambios de los procesos.

Tabla 9 (Continuación)

Descripción de puestos de la organización del departamento de informática.

Área	Rol	Descripción de funciones
Coordinación de desarrollo de manufactura	Coordinador de desarrollo	• Planificar, ejecutar y controlar la implementación de sistemas informáticos para el área de manufactura. • Trabajar en mesas de procesos del área, en equipo con los directores, gerentes y usuarios finales para optimizar el desarrollo en función a los objetivos empresariales. • Identificar y promover oportunidades de mejora de los procesos para las áreas de la empresa. • Planificar, ejecutar y controlar el desarrollo de sistemas de información de las áreas a cargo. • Evaluar el avance de los proyectos y asegurar que los mismos cumplan con la metodología de desarrollo implementada en el departamento. • Atender y resolver consultas de los usuarios, sobre el desarrollo e implementación de sistemas. • Supervisar al personal y las actividades del área bajo su responsabilidad. • Evaluar el desempeño del personal a su cargo.
	Desarrollador de sistemas	• Desarrollar sistemas para optimizar procesos dentro de la empresa, administrando y desarrollando proyectos de tecnología en el área de manufactura con el fin de facilitar el trabajo de los colaboradores de la empresa. • Brindar soporte a los usuarios sobre los sistemas de la organización específicamente a los de manufactura. • Desarrollar mejoras a los sistemas informáticos de manufactura. • Realizar construcción de software para manufactura. • Realizar investigación de tecnología nueva. • Brindar soluciones a los diferentes usuarios. • Identificar requerimientos. • Implementar sistemas informáticos para manufactura. • Monitoreo y control de aplicaciones realizadas. • Cumplir fechas de entrega • Brindar capacitación a usuarios y supe usuarios. • Brindar colaboración en actividades relacionadas a la dirección. • Modificar las aplicaciones para que se apeguen a los cambios de los procesos.

Tabla 9 (Continuación)

Descripción de puestos de la organización del departamento de informática.

Área	Rol	Descripción de funciones
Base de datos	Administrador de base de datos	• Administras las bases de datos de la empresa, y velar por la calidad de la información de toda la empresa. • Realizar instalaciones de bases de datos para los sistemas de producción. • Planificar y generar copias de respaldo de las bases de datos periódicamente. • Monitoreo de servicios de bases de datos. • Realizar recomendaciones a los desarrolladores de aplicaciones para aumentar el performance de las operaciones sobre las bases de datos. • Aplicar políticas de seguridad y acceso a la información para usuarios de todo nivel de la organización.
Infraestructura y comunicaciones	Administrador de sistemas de producción	• Administrar sistemas de seguridad en toda la red de la organización. • Administrar los servidores de producción de la empresa. • Brindar colaboración con otras actividades relacionadas con área de tecnología. • Configurar y mantener la infraestructura de red virtual • Administrar todos los enlaces de datos de Internet en tiendas y oficinas. • Administración de sistemas operativos de los servidores de producción. • Velar por la continuidad de negocio.
	Administrador de redes	• Mantener la operatividad y seguridad de las redes de la empresa. • Verificar el cumplimiento de políticas y normas a través de herramientas especializadas. • Colaborar con los equipos de desarrollo en el análisis y diseño de proyectos estratégicos de la empresa. • Monitoreo del rendimiento de la red. • Administrar correos de la organización.

5.4 Dominio de complejidad de la empresa

La Tabla 9 presenta las características de los dominios de complejidad del framework Cynefin, en combinación con la frecuencia de ocurrencia de éstos en el contexto de la empresa. Los porcentajes de frecuencia son rangos cualitativos y ha quedado a discreción de los miembros del departamento valuar la ocurrencia de sucesos de ésos tipos.

Tabla 9

Presenta las características de los dominios de complejidad y la frecuencia con que ocurren en el contexto de la empresa, mediante un intervalo cualitativo que va de cero a cien por ciento.

Características de los dominios	0 – 20%	20 – 40%	40 – 60%	60 – 80%	80 – 100%
Dominio Simple					
Problemas Simples	-	X	-	-	-
Fácil de identificar causas	-	X	-	-	-
Existen mejores prácticas	-	X	-	-	-
Problemas conocidos a soluciones conocidas	-	X	-	-	-
Especifican una serie de lógica de pasos y se ejecutan de manera repetitiva	X	-	-	-	-
Dominio Complicado					
Problemas complejos	-	-	-	X	-
Perfiles expertos	-	-	X	-	-
Múltiples soluciones correctas a una misma problemática.	-	X	-	-	-
Se requiere involucramiento de expertos para poder identificar.	-	X	-	-	-
Dominio Complejo					
Problemas complejos	-	-	-	X	-
Resultados impredecibles.	-	-	-	X	-
No existen ni mejores ni buenas prácticas catalogadas.	-	-	-	X	-
No se sabe con anticipación si una solución va a funcionar.	-	-	-	X	-
Solo se puede examinar resultados y adaptar	-	-	-	-	X
Dominio Caótico					
Los problemas requieren de una repuesta inmediata	-	-	X	-	-
Es necesario actuar de inmediato para restablecer cierto orden.	-	-	X	-	-
Solucionar el problema inmediatamente, sin importar la forma técnica, para luego fuera del caos aplicar a una solución más robusta.	-	X	-	-	-

Tabla 9 (Continuación)

Presenta las características de los dominios de complejidad y la frecuencia con que ocurren en el contexto de la empresa, mediante un intervalo cualitativo que va de cero a cien por ciento.

Características de los dominios	0 – 20%	20 – 40%	40 – 60%	60 – 80%	80 – 100%
Dominio Desordenado					
Se presenta cuando no se sabe definir el dominio en el que se encuentra.	-	X	-	-	-
No se pueden medir las situaciones, ni determinar la forma de actuar.	X	-	-	-	-
Se actúa en base a preferencias personales.	X	-	-	-	-

Al interpretar los resultados obtenidos, podemos concluir que la empresa se encuentra en un dominio complejo en el que es conocido que se puede implementar metodologías ágiles como Scrum, que son utilizadas para actuar, inspeccionar y adaptar prácticas emergentes de un equipo de trabajo (Alaimo, 2013).

5.5 Metodología de trabajo actual

Para la realización de las actividades de desarrollo, la empresa ha desarrollado una metodología de desarrollo en cascada, sustentada cada una de las fases del desarrollo sobre una política de trabajo que consta de 10 etapas a realizar como parte del ciclo de vida de desarrollo de los sistemas.

En algunos casos, según la magnitud y tamaño de los sistemas se toma la estrategia de aplicar una estrategia de desarrollo evolutivo-iterativo, fraccionando las funcionalidades de la aplicación de acuerdo a módulos que la integran, empleando para el desarrollo de cada una de estas etapas ciclos cortos de desarrollo en cascada. La Fig. 12 muestra el esquema de trabajo de la metodología actual.

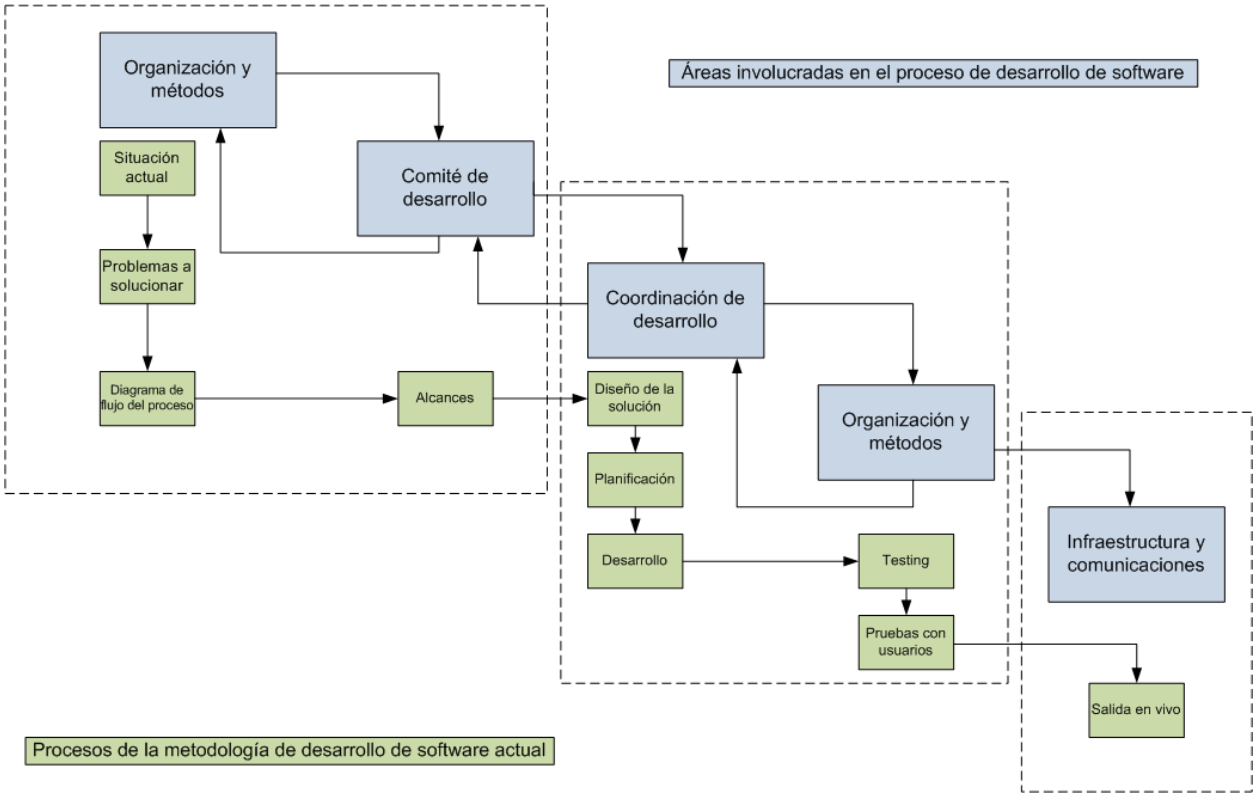

Fig. 12 Esquema de trabajo actual y las áreas involucradas.

5.5.1 Situación actual

En esta etapa del proceso de la metodología de desarrollo participa el área de organización y métodos en conjunto con un comité de desarrollo conformado por coordinadores de desarrollo, directores y usuarios finales quienes exponen su caso sobre el proceso que desean automatizar o bien agregar una mejora al proceso automatizado que ya está produciendo.

El analista de procesos, presenta una breve descripción de la forma en cómo se realizan los procesos actualmente, detallando la secuencia de pasos y procesos que se llevan a cabo para realizar la actividad planteada por el comité de desarrollo.

5.5.2 Problema a solucionar

Durante esta etapa el comité de desarrollo expone las problemáticas a las que se enfrenta con la forma en que se realizan los procesos y así como de los factores que los ocasionan. El analista de procesos se encarga de documentar esas problemáticas.

5.5.3 Diagrama de flujo actual

Posterior al planteamiento del problema a solucionar, el analista de procesos realiza un diagrama de flujo que representa gráficamente la situación actual. Esta herramienta le permite mostrar visualmente el proceso al comité de desarrollo para confirmar que todas las partes entienden claramente cómo trabaja el proceso.

5.5.4 Alcances

Una vez todas las partes involucradas entienden el proceso actual, el comité de desarrollo se encarga de detallar las expectativas de los usuarios interesados, así como los alcances que se buscan lograr con la implementación de las mejoras en el proceso. En esta etapa se detallan los alcances y limitantes encontradas en un documento.

5.5.5 Diseño de solución

Conocido el problema a solucionar, los alcances y limitaciones, la coordinación de desarrollo del área comercial o de manufactura que está conformada por jefe de desarrollo y desarrolladores de sistemas, plantea un diseño para la solución mediante un diagrama de casos de uso y modelo del dominio, en caso de ser una modificación se retoman los diseños actuales sobre los cuales se hará las modificaciones, se mide el impacto que éste tendrá sobre la arquitectura actual. La solución y su detalle son planteados a los demás interesados del comité de desarrollo.

5.5.6 Planificación de la solución

Definida la solución por parte de la coordinación de desarrollo, se planifican las actividades y tareas a realizar, se estima el tiempo de desarrollo y se asignan los desarrolladores a dichas actividades. El detalle del plan de desarrollo es presentada a los demás interesados del comité de desarrollo donde se plantean los tiempos estimados así como el impacto que éste requerimiento tendrá sobre la arquitectura actual a nivel técnico. En esta etapa se redefinen nuevamente las prioridades y alcances de los requerimientos solicitados.

5.5.7 Desarrollo

Durante esta etapa los requerimientos son "congelados" y se ejecutan las actividades de programación de la solución por parte del equipo de desarrollo.

5.5.8 Pruebas internas

Una vez está concluido el desarrollo la aplicación es desplegada en un ambiente de desarrollo para que el usuario probador en este caso el analista de procesos, se encarga de probar la funcionalidad de la aplicación realizando observaciones sobre el cumplimiento de los requerimientos acordados. En caso de que la funcionalidad no sea la esperada, el listado de observaciones son enviadas al coordinador de desarrollo y estas son analizadas en conjunto con el equipo de desarrollo. En caso de que la funcionalidad esté acorde a las expectativas de los usuarios, se realiza una prueba piloto con los usuarios finales.

5.5.9 Pruebas con usuarios

En esta etapa, se presenta y entrega a los usuarios finales el producto desarrollado, a fin que realice las pruebas respectivas y valide que la aplicación produzca los resultados esperados. En esta etapa los usuarios realizan las observaciones sobre la funcionalidad de la aplicación. En caso de que existan observaciones sobre el cumplimiento de los requerimientos, estas son

notificadas al analista de procesos quien posteriormente las envía a la coordinación de desarrollo para realizar ajustes sobre el diseño de la solución y planificación de actividades. En caso de que los usuarios finales estén satisfechos con la solución, ésta es aprobada por parte de los usuarios y el analista de organización y métodos.

5.5.10 Salida en vivo

Una vez se han superado las pruebas y observaciones de los usuarios finales, así como de los interesados se planifica y ejecuta el “Go Live” de las aplicaciones, así como la ejecución de las etapas de capacitación al resto de usuarios.

Cada una de estas etapas son debidamente documentadas y ejecutadas durante las realización de los diferentes proyectos. La aplicación de cada una de estas etapas implica el consumo de una gran cantidad de tiempo y provoca la impaciencia de los usuarios ante la necesidad de obtener resultados de manera más ágil, lo cual es necesario para la realización de las soluciones informáticas que permitan a la empresa estar a la vanguardia de innovación en los productos y servicios que provee a sus clientes.

6 Guía de Implementación de Scrum

Antes de iniciar con el proceso de implementación de la metodología es importante definir la estrategia de implementación, quienes van asumir los roles, cuál será la dinámica de las reuniones y que artefactos serán utilizados para poder implementar Scrum.

6.1 Problemática actual

Según la información recopilada en las entrevistas, la metodología actual no involucra integralmente a todos los interesados en el proyecto de software, ya que el usuario final solo participa en el planteamiento de la situación actual, problemática a solucionar y éstos son contactados nuevamente hasta las etapas de pruebas.

Los requerimientos se “congelan” mientras están en proceso de desarrollo, existe un comité de desarrollo que prioriza cuales son los requerimientos que deben ser trabajados con mayor grado de urgencia. Si existe un cambio de último momento en los requerimientos, estos son tomados en cuenta hasta el final de las pruebas, es probable que transcurrido todo ese tiempo el diseño original cambie debido a los nuevos requerimientos, haciendo que el diseño actual quede obsoleto, lo que involucra un gran esfuerzo en tareas de refactorización.

Es importante que el usuario final esté involucrado en todo momento ya que el software al ser un elemento intangible no es fácil de describir con mayor detalle su funcionalidad solo con palabras, es necesario el elemento visual de una interfaz de usuario y que pueda implementar la funcionalidad deseada, de tal forma que se puedan abarcar más detalles de lo que realmente el usuario desea.

La estimación de tiempos no es del todo acertada ya que no todos los desarrolladores poseen la misma experiencia, en la empresa existen desarrolladores junios y senior.

6.2 Recomendaciones para la implementación

Apoyo al departamento de informática: Según las recomendaciones realizadas por los expertos, el departamento debe tener apoyo por parte de la gerencia informática así como de los altos mandos para poder implementar y empoderar a los equipos con los roles y desempeño de actividades definidas por el marco de trabajo Scrum.

Capacitaciones: Debe existir un plan de capacitación sobre la metodología antes de implementarla. Esta actividad ya está superada, dado que según los entrevistados actualmente todos los miembros han recibido un diplomado por parte de la universidad Don Bosco sobre la metodología Scrum.

Priorización de Backlog: Priorizar el Backlog de manera conjunta con el equipo de desarrollo ayuda a encontrar dependencias a un nivel técnico.

Estimación de tiempos: Es importante que la estimación de tiempos para cada una de las tareas del Backlog del Sprint sea discutida con todos los miembros del Equipo de Desarrollo ya que en el caso de la empresa existen desarrolladores junior y senior. Si la estimación de tiempos se deja a criterio individual es muy probable que se estén obviando detalles que un miembro del equipo más experimentado sepa de antemano. Es importante que todos los miembros participen y discutan para tener un mayor criterio sobre las tareas a realizar para la actividad asignada.

Reuniones diarias a puerta cerrada: Nadie externo a las reuniones diarias debe ser participe mas solamente sea observador, solo deben participar los involucrados para evitar discusiones que no aporten valor a la reunión. Dejar que el equipo se autogestione y exprese todas sus dificultades, no permitir que oculte algún tema relevante por temor a ser señalado, de esta forma se previene el riesgo de alguna actividad que permita el estancamiento y evite el progreso de desarrollo de las tareas por hacer para un ítem del Backlog del Sprint.

Comunicación de avances: El Product Owner es quien da la cara por los intereses de los clientes o demás interesados, este rol es quien debe mantener comunicación de cada avance obtenido en cada Sprint y gestionar el Backlog para un posible cambio de prioridades. Se recomienda comunicar los avances a medida los demos del producto finalizado van saliendo del Sprint ya sea mediante reuniones en persona, video conferencia o correos electrónicos.

Cantidad de miembros del equipo de desarrollo: Es ideal que este conformado por seis personas, el mínimo recomendado es de tres y máximo ocho. Los equipos pueden ser remotos. Scrum crece añadiendo equipos Scrum conformados por Product Owner, Scrum Master y Desarrolladores.

Documentación: Los Requerimientos base, las Historias de Usuario y Casos de uso son importantes, también agrega valor llevar un control de lecciones aprendidas para tomar en cuenta en el desarrollo de siguientes Sprint.

Múltiples Equipos Scrum: Según la cantidad y complejidad de proyectos, dependerá el número de equipos Scrum. Actualmente en la empresa existen tres líneas de desarrollo de productos de software, existe la línea administrativa-comercial, la línea de manufactura y la línea de inteligencia de negocios. Inicialmente se recomienda que existan dos equipos Scrum atendiendo las líneas de desarrollo administrativa-comercial y de manufactura. Quedará a discreción de los implementadores, incrementar los equipos Scrum según sea la cantidad o complejidad de los proyectos entrantes.

6.3 Equivalencias entre roles para la conformación del Scrum Team

La Tabla 10 presenta la conformación de equipos Scrum.

Tabla 10

Muestra las equivalencias de roles de la empresa versus los roles Scrum y el por qué los primeros deberían implementar los segundos.

Rol empresa	Rol Scrum que implementa	Fundamento
Analista de procesos	Product Owner	• El analista de procesos es la primera línea de comunicación entre usuarios finales y el equipo técnico desarrollador de software. • Liderar el proceso de identificación, recopilación y documentación de procesos y coordinación de todos los responsables de las diferentes direcciones de la empresa. • Debe expresar claramente los elementos de la Lista del Producto. • Debe asegurar que la Lista del Producto sea visible, transparente y clara para todos. • Debe asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario.
Coordinador de desarrollo	Scrum Master	• Es el responsable de asegurar que Scrum se entienda y se adopte. • Es quien trabaja en mesas de procesos del área, en equipo con los directores, gerentes y usuarios finales para optimizar el desarrollo en función a los objetivos empresariales. • Es un líder que está al servicio del equipo Scrum. Ayuda a las personas externas al equipo Scrum a entender qué interacciones con el equipo. • Identifica y promueve oportunidades de mejora de los procesos para las áreas de la empresa. • Elimina impedimentos para el progreso del Equipo de Desarrollo.

Tabla 10 (Continuación)

Muestra las equivalencias de roles de la empresa versus los roles Scrum y el por qué los primeros deberían implementar los segundos.

Rol empresa	Rol Scrum que implementa	Fundamento
Desarrollador de sistemas	Development Team Member	• Desarrolla sistemas para optimizar procesos dentro de la empresa, administrando y desarrollando proyectos de tecnología con el fin de facilitar el trabajo de los colaboradores de la empresa. • Desarrolla mejoras a los sistemas informáticos. • Realiza construcción de software. • Identifica requerimientos. • Tiene la capacidad de auto organizarse. • Son los únicos miembros del equipo que participan en la creación de un incremento.

6.4 Implementación

Para lograr un rápido y exitoso proceso de implementación de la nueva metodología, han de planificarse una serie de actividades orientadas a preparar la estructura de la organización, así como las actividades propias de la ejecución de la metodología Scrum.

6.5 Organización

La adopción de la nueva metodología de trabajo es un cambio que es realizado de forma gradual, se requiere que todos los miembros de la empresa directores de área y usuarios finales comprendan las ventajas, beneficios y consideraciones que se han de obtener con la nueva metodología.

La primera actividad, antes de iniciar este proceso de cambio, es realizar un plan de comunicación donde se involucre a todas las partes interesadas, que comunique todas las bondades que comprende la metodología. Dicho plan de comunicación no ha de implicar una simple difusión de las características de la metodología, esta debe de incluir sesiones de

discusión, a partir de las cuales puedan recopilarse y analizarse las opiniones de todos los involucrados.

Para lograr este objetivo Agile (2013) dice que la empresa debe ser entonces concebida como un grupo global de trabajo, incluyendo a desarrolladores, gerentes, interesados en el producto, etc.; cada uno de los componentes deben ser integrados a lo que se llama la "gran charla". Por un lado entonces, esta conversación servirá para informar e incluir todos los puntos de vista, pero por el otro, ayudará a obtener el apoyo y los aliados necesarios a la hora de integrar la metodología.

A través del convencimiento de las diferentes estructuras de la organización, se esperan obtener los resultados:

Tabla 11

Tabla de resultados esperados

Estructura	Resultado Esperado
Directores	Son los propulsores del cambio, contribuyen y apoyan en la eliminación de los obstáculos que puedan afectar en la ejecución de la metodología.
Coordinadores de desarrollo	Promueven las ventajas de la metodología, contribuyendo a logar una mejor interacción entre los ejecutivos de la empresa y los equipos de desarrollo. Son garantes que el equipo de desarrollo cumpla la metodología.
Equipos de desarrollo	Viven y cumplen las reglas de la metodología, motivándose entre sus miembros a no fallar en dicho propósito.

Todas estas actividades no podrán lograr su objetivo, a menos que los miembros de la organización tengan claros los conceptos y principios que deberán de tomarse en cuenta, al momento de ejecutar las fases y etapas de la metodología Scrum.

6.6 Aplicación de la metodología

En un principio y para que Scrum produzca los efecto esperados, es necesario empezarlo a utilizar en el desarrollo de proyectos pequeños o medianos en lugar de proyectos grandes. Esto mantendrá las cosas simples mientras el Equipo Scrum se acostumbra a las bases.

Una vez que la organización y equipo Scrum está preparada sobre la metodología, es necesario planificar cada una de las actividades a realizar durante la aplicación de la metodología. Para ello han de aplicarse la siguiente serie de etapas a realizar al momento de ejecutar un proyecto.

1. Preparar el Proyecto.
   1.1. Ordenar el Backlog
   1.2. Estimar el Product Backlog
2. Planificar el Sprint
   2.1. Aclarar los requerimientos
   2.2. Estimar las tareas
3. Trabajar el Sprint
   3.1. Crear un espacio común de trabajo
   3.2. Llevar a Cabo el Sprint
   3.3. Reuniones de Seguimiento diario.
4. Terminar a tiempo
   4.1. Medir el Avance
   4.2. Asegurar que el Sprint esta hecho
5. Revisar, mejorar y repetir…

6.7 Preparar el proyecto

6.7.1 Ordenar el Backlog

En esta etapa del proyecto el Product Owner deber de conocer los requerimientos del proyecto, deber ser responsable de priorizar el trabajo a realizar. Deber de tener la capacidad de ser un buen comunicador, que esté comprometido con el éxito del proyecto y que sea capaz y esté dispuesto a dedicar tiempo en desarrollo del mismo.

De no encontrar una persona que reúna estas condiciones es mejor no seguir, ya que se recabaran una serie de requisitos pocos claros y no podrán lograrse los objetivos del proyecto.

El siguiente paso es necesario elaborar la pila del producto (Product Backlog). Esta es una lista de tareas que se tendrán que realizar para completar el proyecto.

Cada uno de estos Ítems deberá ser expresado en términos del negocio, no con definiciones técnicas, ya que deben ser fácilmente comprensibles por los usuarios y clientes.

El Product Backlog. Puede estar conformado por cualquier necesidad relacionado al sistema, no solo nuevos desarrollos, por lo tanto pueden incluirse, mejoras, errores, problemas identificados y riesgos.

No habrá resistencia si alguna persona quiere proponer nuevas tareas para ser incluidas en el Backlog, sin embargo ninguna podrá ser incluida y priorizada sin la autorización del Product Owner.

Una vez elaborado el Backlog es necesaria la priorización de los mismos. Esta actividad consiste en colocar un orden a las tareas de la lista siguiendo un orden de prioridad. Dicho orden es establecido por el Product Owner.

Se recomienda que las tareas que se encuentren al final sean aquellas que son más difíciles de realizar e inclusive podría no llegar hacerse por ser en algunos casos confusos o encontrarse mal definidos.

Es importante que el Equipo de Desarrollo tenga visión de todas las tareas que incluyen Backlog ya que el no tenerla puede ocasionarle incertidumbre durante el proceso de desarrollo.

6.7.2 Estimar el Product Backlog

Para la estimación de tiempos de desarrollo de las tareas del Backlog se recomienda la estimación mediante la puntuación del sistema de números Fibonacci, los cuales son un conjunto de números científicamente significativos, donde cada número es la suma de los dos anteriores. Por ejemplo:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987...

Por simplicidad, cuando se esté usando estos números para indicar tamaño, se recomienda utilizar el rango entre 1 y 21 ya que nos provee un rango más corto de variación.

Esta consiste en asignar, a cada una de las tareas, una estimación de tiempo, de acuerdo al rango de complejidad que esta representa. Para realizar esta actividad se comienza desde el inicio al final del Backlog.

Para la aplicación de la metodología se recomienda implementar la planeación por Póker. Esta es una técnica segura a través de la cual se evita que las personas se influencien entre sí.

Para realizarla cada miembro del equipo anota su estimado en una tarjeta, y todos muestran sus respuestas a la vez. Con esto se asegura que los miembros con menos experiencia en el equipo estén igualmente comprometidos y no sobre-influenciados por miembros más experimentados. Ayudando a su vez a estimadores menos experimentados a aprender de otros. A partir de este ejercicio, se negocia y se establece el tamaño de cada ítem de la pila en conjunto.

Una vez se ha realizado la estimación de las tareas es necesario que el Product Owner realice una nueva revisión de las prioridades a fin que pueda establecer un nuevo orden en las tareas.

Si alguna prioridad ha cambiado, simplemente se mueve la posición del ítem en el orden de la pila.

6.7.3 Planificar el Sprint

6.7.3.1 Aclarar los requerimientos

La siguiente actividad es realizar una reunión de planeación del Sprint, en la que se cuente la participación de todos los miembros del equipo, en esta además de mostrar el Backlog y prioridades deberá mostrar la duración del Sprint. Esta duración debería ser tomada como un equipo.

Decidir la duración del Sprint es una decisión importante. Scrum sugiere treinta días. Sin embargo este valor parece ser consensuado por los miembros del equipo, ya que puede variar de acuerdo a la naturaleza del proyecto, grado de madurez, tipo de implementación y nivel de conocimiento de los miembros del equipo.

Una vez se ha consensuado la duración del sprint es necesario mantener de forma consistente la duración del mismo; esto ayuda medir de mejor forma el avance del proyecto y del equipo de trabajo. Esto ayuda a poder establecer mejores indicadores que ayuden contar con una mejor estimación en la planificación de los subsecuentes Sprint.

Una vez que se ha establecido la duración del Sprint, es necesario determinar la meta del mismo, para ello han de seleccionarse las primeras tareas del Backlog. A esta etapa el Product Owner no debe esperar que todas las tareas seleccionadas sean terminadas al finalizar el Sprint.

Posterior a la definición del Sprint, su duración y las tareas a incluir, el Product Owner deberá exponer una explicación de los resultados que se espera con la ejecución de los mismos. Esta exposición deberá de generar una iteración entre este y los miembros del equipo de desarrollo.

Cada uno de los puntos acordados de esta discusión, deberán ser colocados en una pizarra o rota folio o cualquier medio que se tenga disponible para la recolección de los requisitos describiendo las diferentes características de estos ates de iniciar el desarrollo.

El resultado final será describir historias de usuario, las cuales son descripciones de los requisitos de forma comprensible para que puedan ser comprendidas por el equipo de desarrollo y posteriormente ser probadas por los usuarios. Para facilitar su comprensión, estas historias pueden ser respaldadas por casos de uso, modelos del dominio y esquemas de interfaces de usuarios que ayuden a describir la funcionalidad esperada de los mismos.

6.7.4 Estimar las tareas

Una vez establecido y clarificado los requisitos del Backlog, se debe pasar los requisitos en tareas y estimar las horas requeridas para completarlas.

En esta etapa es necesaria la participación de todos los miembros del Equipo de Desarrollo, no así la del Product Owner u otro interesado, esto es debido a que esta es una reunión más técnica sobre el trabajo que ha de desarrollar el equipo. Sin embargo, la presencia de ellos ha de ser de mucha ayuda a la comprensión por parte de ellos sobre los aspectos técnicos a desarrollar.

Como resultado de esta actividad se debe obtener:

El número de horas de trabajo que el Equipo de Desarrollo tiene disponible para la realización del Sprint. Este se obtiene calculando el número de horas que ha de trabajar cada uno de los miembros del equipo, deduciéndoles las vacaciones e imprevistos de tiempo.

Dividir los requisitos en tareas. Considera incluir las actividades de diseño, pruebas, etc. Estas no son excluyentes del empleo de la metodología Scrum ya que las metodologías ágiles solo abogan por hacer estos pasos característica por característica, justo a tiempo, en lugar de grandes fases.

Fijar las tareas como resultados. Los resultados son más medibles que las tareas. En lugar de describir que hacer se ha de describir lo que se ha de entregar como resultado.

Estimar el tiempo de las tareas en horas. Se recomienda estimaciones de tareas de un día, porque si una estimación es mucho más grande que esto, los requisitos deben ser divididos para que las tareas sean más pequeñas.

Mantener el compromiso con las tareas del Sprint. Suma todas las estimaciones de las tareas para la Pila seleccionada. Si superan significativamente el tiempo se deberá de reducir el número de Items seleccionados para el Sprint, sin olvidar perder de vista los órdenes de prioridad, previamente establecidos.

6.7.5 Trabajar en el Sprint

6.7.5.1 Crear un espacio común de trabajo

Previo a iniciar el desarrollo del Sprint, es necesario establecer un espacio en el cual podrán colocarse las pizarras que han de servir de herramienta para dar seguimiento al proyecto. Inicialmente no se recomienda utilizar un software para esta etapa, dado que con ello se busca colocar de forma visible las tareas del Sprint y su avance de un solo vistazo. En este lugar se han de realizar las sesiones de pie de todos y donde han de llevarse a cabo las reuniones de avance del proyecto.

La Fig. 13 presenta un ejemplo de cómo organizar las pizarras deberán ser divididas en cinco columnas como mínimo y ser etiquetadas con los siguientes encabezados:

- Pila de Tareas*:* Escribir en un post o tarjeta el número de referencia y una breve descripción de cada ítem de la pila de tareas del Sprint.
- Tareas No Iniciadas*:* Acá se colocan las tareas relevantes de la pila del producto.
- Tareas en progreso*:* Cuando se comienza a trabajar en una tarea, moverla a esta columna.

- Listas para ser verificado: Se colocan tareas que se han finalizado y se encuentren listas para ser revisadas por los usuarios.
- Terminadas: Son aquellas tareas, que ya se completaron y pasaron las pruebas de los usuarios.

Pila de tareas	No iniciadas	En progreso	Lista para ser verificadas	Terminadas
Reporte conozca a su cliente Sincronización de clientes a ERP SAP	Mantenimiento de listas de precios para clientes Mantenimiento de ubicación geográfica del cliente	Monitor de búsqueda de clientes	Validación de documentos de identificación del cliente	Reporte de clientes nuevos

Fig. 13 Pila de tareas del Backlog del Spring.

6.7.6 Llevar a cabo el Sprint

En esta etapa el Equipo de Desarrollo realiza las actividades necesarias para completar las tareas del Scrum. En esta fase el Equipo de Desarrollo está empoderado para tomar las decisiones necesarias para llevar a cabo esta tarea. Cualquier injerencia de la gerencia de la organización elimina responsabilidad sobre el trabajo que realizan.

En esta etapa el trabajo el gerente deberá ser de apoyo y transmisión de opiniones y experiencias, no instrucciones en sí, con esto se busca la auto-organización del equipo para lograr sus metas.

El equipo no deberá de perder de vista el cumplimiento de la duración del Sprint, por lo que no deben de agregarse nuevas tareas, de hacerlo deberán de eliminarse alguna de las

agregadas al Sprint, sin embargo esto deberá ser evaluado durante las reuniones diarias de pie para seguimiento.

Deberán de evitarse los cambios periódicos de prioridades ya que se debe dejar que el Equipo de Desarrollo cumpla los Sprint que se comprometió en el tiempo estimado, haciendo equipos de trabajo productivos.

No deberán de dejarse tareas a medio desarrollo; cuando una tarea se mueve a la columna Listo para ser revisado y terminado, es porque se han cumplido estas dos actividades en un cien por ciento.

En situaciones excepcionales un Sprint puede ser abortado, esto puede darse de forma excepcional cuando la meta del Sprint no va en dirección de cumplir con los objetivos del proyecto, siendo necesario realizar una reevaluación de las prioridades y tareas por realizar.

6.7.7 Reuniones de Seguimiento diario.

A medida se va ejecutando el Sprint, se debe de realizar todos los días reuniones a pie. Estas deben tener a todo el equipo presente e interesados. No solo se requiere la participación del equipo de desarrollo sino que también del Product Owner.

Dada la importancia de la participación de todos los miembros del equipo, es necesario el mantenimiento del orden y puntualidad de la misma.

Para llevarla a cabo el equipo se mantiene en un semi-círculo alrededor de la pizarra del Sprint. En esta etapa cada miembro del equipo presenta un informe, en este debe de responder tres preguntas claves, que deben de ser respondidas en el momento:

¿Qué se ha hecho desde la última reunión? (ayer).

¿Qué se realizara antes de la siguiente reunión? (mañana).

¿Hay algo deteniendo su progreso? (impedimentos).

Si alguna cuestión surge como parte posterior al informe o de ellas requieren una mayor discusión, se debería hacer pero se abstendrán de discutirla en detalle hasta después de la reunión. Solo aquellos necesarios en la discusión pueden permanecer después de que la reunión haya terminado. Todos los demás vuelven al trabajo.

El Scrum Master es la persona responsable de coordinar esta reunión y mantener que se cumpla la duración de la misma; se estima una duración máxima es de 15 minutos. Es su responsabilidad buscar la solución de cualquier impedimento que se haya discutido durante la reunión, todo con el propósito de mantener el equipo enfocado.

6.7.8 Terminar a Tiempo

6.7.8.1 Medir el Avance

Un factor importante en desarrollo de un proyecto es identificar los problemas de manera temprana y con el tiempo necesario para su identificación y corrección. Una de las características que tiene la metodología Scrum es que permite ir realizando las pruebas y midiendo el avance del proyecto durante todas las fases del ciclo de vida del mismo, así como el involucramiento de los usuarios en las etapas del mismo.

Adicionalmente a la medición del avance en la calidad del proyecto, es necesario medir el progreso diario de las tareas del proyecto.

Para conseguir esto la metodología promueve la elaboración del gráfico Daily Burndown ver Fig. 14. Este gráfico es una herramienta muy poderosa que consta de dos, en el eje Y se enumeran cada una de las tareas que se entregaran en un sprint y en el eje X, se anota el tiempo estimado para completarlo, haciendo con ello una gráfica de estimación de desarrollo del sprint.

A medida se va avanzando en el desarrollo de este, los miembros han de ir actualizando su estimación de tareas sobre una base diaria antes de la reunión de pie, y a partir de ahí tener

una visión grafica del avance del proyecto y poder a partir de esto identificar posibles problemas u retrasos en el mismo.

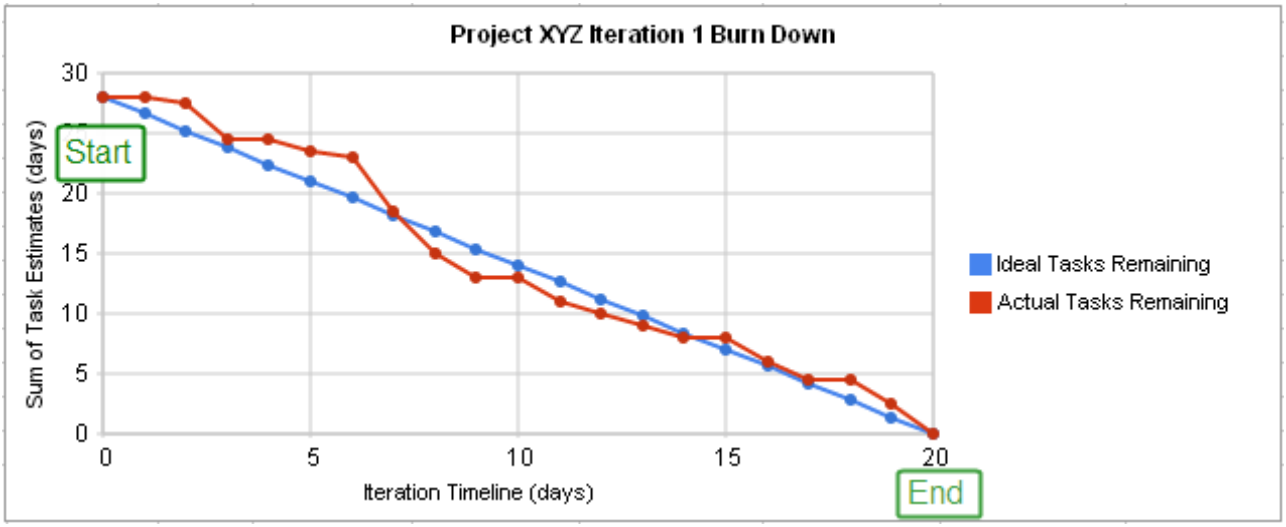

Fig. 14 Diagrama del Daily Burndown.

La ventaja de este enfoque es presentar desde un punto de vista numérico el progreso del Sprint, midiendo las estimaciones originales versus las estimaciones reales, buscando con esto lograr que el equipo pueda completar todas las tareas con un faltante de cero horas.

Cuando la línea real este por debajo de la línea de estimación original, se está en el camino correcto. Cuando está por encima, es porque se está generando algún atraso, siendo necesario evaluar a diario la causa de estos y la forma de superarlos.

Durante la elaboración de este grafico será de mucha utilidad anotar los eventos claves que se han presentado (Speech Bubbles), los cuales serán de mucha utilidad al momento de hacer la revisión del Sprint y comunicación de estas a las partes interesadas para evitar con ello elaborar un informe por separado.

6.7.9 Asegurar que el Sprint esta hecho

La ejecución y seguimiento de cada una de las etapas anteriores, ha de permitir lograr el propósito planteado durante la fase de planeación, el cual es terminar las tareas del Sprint en el

tiempo estimado. Sin embargo cuando se refiere a que una tarea ha sido terminada, es porque ha sido finalizada en su cien por ciento y habiendo cumplido con las pruebas y satisfecho las necesidades planteadas por los usuarios y el Product Owner al inicio del Sprint.

6.7.10 Revisar, mejorar y repetir

Esta etapa es realizada en pro de alcanzar los siguientes objetivos:

- Permitir a los miembros del equipo mostrar lo que han logrado y demostrar su contribución al producto.
- Permitir a todos los tomadores de decisión ver lo que se ha hecho y proveer valiosa retroalimentación en una base regular, mientras aún hay tiempo de tomarla en consideración.
- Ayudar al equipo a permanecer enfocado en el tiempo del Sprint. nadie quiere aparecer en la revisión del Sprint con nada útil que demostrar.

Para lograrlo, una vez completadas las etapas es necesario ejecutar el Sprint Review. Esta es una reunión que a través de la cual se busca revisar avance del Sprint. Esta debe de contar con la presencia de todos los miembros del equipo además de los tomadores de decisiones. ¡Entre más partes interesadas, mejor!.

Dentro de la metodología de esta revisión, se busca reflexionar sobre cómo fueron las cosas durante el Sprint. Es una oportunidad para que el equipo discuta el Sprint y consideren como se pueden mejorar las cosas. No debemos de olvidar que esta reunión no es para toma de decisión, sino que esta debe de hacerse inmediatamente después de la revisión del Sprint.

Como parte de esta revisión se deberán de realizar las actividades siguientes:

- Revisar el gráfico de seguimiento.
- Revisar la "velocidad" del equipo.

- Discutir lo que fue bien y asegurar que se repita la siguiente vez.
- Discutir y entender que podría haber ido mejor.
- Decidir lo que el equipo hará diferente en el siguiente Sprint.

Al finalizar de esta etapa el equipo contara con una valiosa fuente de información sobre el producto, su rendimiento y sobre algunos impedimentos en su entorno que le permitirán conocer:

- Cómo se ve el software después del último Sprint.
- Retroalimentación sobre el producto desarrollado hasta entonces.
- Capacidad del equipo entregar lo que se había comprometido en la planificación del Sprint.
- La velocidad del equipo y lo que es alcanzable en una iteración típica.
- Elementos que afectaron positiva o negativamente.
- Elementos que se han de mejorar en el siguiente Sprint.

Una vez hecho esto se deberá repetir el proceso, aplicando las lecciones aprendidas en el Sprint anterior.

7 Conclusiones

- Las metodologías ágiles son un conjunto de valores, principios y prácticas, representan un paradigma diferente para generar a los clientes valor en sus productos o servicios de una forma mucho más rápida.
- Las metodologías de desarrollo de proyectos han evolucionado de un enfoque secuencial a uno más dinámico o ágil adaptándose cada vez a las necesidades de los dominios de complejidad de las empresas, aunque no son una "bala de pata" para la solución a todos los problemas presentan marcos de trabajo que permiten potenciar al recurso humano, permitiendo que éstos sean equipos auto gestionados y dediquen más tiempo a la calidad de sus productos o servicios en lugar de actividades administrativas que no aportan valor a la empresa.
- La metodología Scrum es un marco de trabajo con roles definidos, artefactos y reuniones que permiten potenciar la entrega de productos o servicios mediante la adopción de principios como el enfoque, respeto, compromiso, franqueza y valor. Es un framework enfocado en las personas donde ven expuesto su trabajo y deben manejarlo de forma colaborativa.
- La metodología Scrum puede adaptarse según la complejidad de los proyectos, pero esta se adapta en base a equipos Scrum, conformados por Product Owner, Scrum Master y Equipos de desarrollo.
- Para poder implementar la metodología es necesario conocer el contexto de la empresa, involucrar a la gerencia de informática, capacitar al personal previamente a la implementación, es importante que se empiece por proyectos pequeños para ir adaptando la metodología según las necesidades. Debe existir

comunicación y tener un alto grado de cohesión en el equipo, incluyendo a los clientes o usuarios finales. La metodología es altamente colaborativa y debe existir mucha disciplina por parte de los involucrados para que pueda fluir sin problemas.

- El apoyo al departamento de informática por parte de la gerencia informática es importante, ya que permite empoderar a los equipos con los roles definidos por el marco de trabajo Scrum. Para conseguir el apoyo, las herramientas y marcos de trabajo deben estar acorde y en sintonía los objetivos empresariales.
- La priorización del back log es una tarea conjunta y colaborativa, requiere de la participación de todo el Scrum Team e incluso de los usuarios finales para poder priorizar ítems del Backlog ya sea por valor para la empresa, dependencia de actividades y el impacto a nivel técnico que ésta pueda tener por conceptos de refactorización que puedan aumentar la complejidad del requerimiento.
- Para asignar los roles de Scrum es importante conocer los roles actuales de la empresa, deben ser compatibles a las actividades que cada uno realiza de tal forma que éstos puedan adaptarse con menor dificultad a la metodología. Otro valor agregado es la experiencia sobre la metodología y aptitudes propias que el personal pueda tener, esto ayudará a definir que rol de Scrum puede desempeñar durante la implementación.

8 Recomendaciones

- Todo el equipo deberá tener pensamiento ágil, en particular para hacer coaching y acompañamiento continuo.
- Los miembros del equipo y la organización deben tener presente la cultura de la calidad.
- Antes de comenzar hay que tener claro el proceso y la interacción entre las personas. “La gente tiene que trabajar cara a cara”. Debe haber un alto grado de cohesión en el equipo, incluyendo a los usuarios.
- No pensar que Scrum va a solucionar todos los problemas, incluidos los personales., Scrum por sí solo no es suficiente, debe acompañarse de un conjunto de prácticas y otros métodos preferiblemente ágiles.
- Si no se cuenta con alguien que ejecute algunos de los roles que requiere la metodología, no se debe implementar, es necesario el involucramiento y participación de todos ellos.
- No inventar ni ignorar pasos. Aplicar la metodología tal cual es.
- Mantener retroalimentación continua de todos los participantes.
- Se requiere disciplina. Ser estricto con la duración. Siempre empezar a tiempo y cumplir con los 15 minutos máximos de la reunión de pie.
- Si alguno de los miembros del equipo que no puedan asistir tienen que ser reemplazados por un miembro que esté presente. Esta persona debe informarle al equipo sobre el progreso del miembro ausente.
- Solventar los impedimentos; si alguien del equipo no esté seguro de algo, necesita información, está esperando alguna cosa, el Scrum Master tiene que

resolver cada impedimento en veinticuatro horas. Si no puede hacerlo dentro de las veinticuatro horas, deberá reportarlo cuanto antes a la organización, para poder recibir ayuda.

- Mantener los artefactos de Scrum visibles. Por ejemplo pizarras visibles o que todos los involucrados en un Equipo Scrum tengan acceso a un software de control y seguimiento ágil.
- Facilitar según sea necesario, usualmente será menos de lo esperado. NO gestionar, NO decirle a los demás qué hacer.
- No permitir que otras personas, dueños del producto, líderes, gerentes, etc. interrumpan el Scrum Diario. Sólo pueden hablar los miembros del equipo.
- Mantener a todos los miembros del equipo de los inconvenientes presentados y solventados, así como de los avances aspectos relevantes durante la ejecución del sprint.

9 Bibliografía

Agile, L. q. (12 de 2 de 2013). *ERICHBUHLER.* Recuperado el 09 de 08 de 2016, de Lo que no se dice de Agile: https://agilib.org/2013/02/12/lo-que-no-se-dice-de-agile/

Alaimo, D. M. (2013). *PROYECTOS ÁGILES CON SCRUM: Flexibilidad, aprendizaje, innovación y colaboración en contextos complejos.* Cdad. Autónoma de Buenos Aires, 1049, Argentina: Kleer.

Balaji, S., & Sundarajan Murugaiyan, M. (29 de Junio de 2012). *WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC.* Obtenido de International Journal of Information Technology and Business Management: http://www.jitbm.com/Volume2No1/waterfall.pdf

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., . . . Sutherland, J. (Septiembre de 2016). *Principles behind the Agile Manifesto*. Obtenido de agilemanifesto.org: http://agilemanifesto.org/principles.html

Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham, W., Fowler, M., . . . Thomas, D. (2001). *Manifesto for Agile Software Development*. Obtenido de agilemanifesto.org: http://agilemanifesto.org

Beriguete De Leon, A. C. (31 de Enero de 2012). *Escuela de Organización Industrial*. Obtenido de Blogs EOI Awilda Carolina Beriguete de Leon: http://www.eoi.es/blogs/awildacarolinaberiguete/2012/01/31/las-actividades-de-la-gestion-de-proyectos/

Borbón Ardila, N. I. (2013 de Marzo de 2013). *NORMA DE EVALUACIÓN ISO/IEC 9126*. Obtenido de Evaluación de Software: http://actividadreconocimiento-301569-8.blogspot.com/2013/03/norma-de-evaluacion-isoiec-9126.html

Centers of Medicare and Medicaid Services. (Marzo de 2008). *SELECTING A DEVELOPMENT APPROACH.* Obtenido de CMS.gov: Centers of Medicare and Medicaid Services: https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf

Diagnostico y Soluciones. (2007). *¿Qué es la ISO 9001?* Obtenido de Diagnostico y Soluciones Servicios Integrales: https://www.diagnosticoysoluciones.com/novedades/que-es-la-iso-9001/

Ecured. (s.f.). *ISO 15504*. Obtenido de Ecured: https://www.ecured.cu/ISO_15504

Ecured. (s.f.). *Norma ISO/IEC 14598*. Obtenido de Ecured: https://www.ecured.cu/Norma_ISO/IEC_14598

Gómez Fuentes, M. D., Cervantes Ojeda, J., & González Pérez, P. P. (2012). *Notas del curso: Administración de proyectos.* México, DF: Casa Abierta al Tiempo.

Gosset, N. (s.f.). *Certificación ISO 9003*. Obtenido de eHow en español: http://www.ehowenespanol.com/certificacion-iso-9003-sobre_93599/

Hayes, S., & Andrews, M. (2014). *An Introduction to Agile Methods.* Obtenido de Pace University Seidenberg School of CSIS: http://csis.pace.edu/~marchese/CS616/Agile/IntroToAgileMethods.pdf

INTECO: Instituto Nacional de Tecnologías de la Comunicación. (Marzo de 2009). *INGENIERIA DEL SOFTWARE: METODOLOGIAS Y CICLOS DE VIDA.* Obtenido de Universidad Nacional Aierta y a Distancia: http://datateca.unad.edu.co/contenidos/301569/guia_de_ingenieria_del_software.pdf

ISO 25000. (s.f.). *La familia de normas ISO/IEC 25000*. Obtenido de ISO 25000 calidad del producto software: http://iso25000.com/index.php/normas-iso-25000

James, M. (2012). *Scrum Reference Card.* Obtenido de Scrum Reference Card: http://scrumreferencecard.com/ScrumReferenceCard_v0_9l-es.pdf

Jan, S. A. (25 de Marzo de 2016). *Understanding Agile Using the Cynefin Framework*. Obtenido de Scrum Alliance: https://www.scrumalliance.org/community/articles/2016/march/understanding-agile-using-the-cynefin-framework

Mahanti, A. (2006). *Challenges in Enterprise Adoption of Agile Methods – A Survey.* Obtenido de University of Calgary Department of Computer Science: http://pages.cpsc.ucalgary.ca/~amahanti/Papers/Aniket_JCIT2006.pdf

Nerur, S., Mahapatara, R., & Mangalaraj, G. (Mayo de 2005). *Challenge of Migrating to Agile Methodologies.* Obtenido de Salisbury University: http://faculty.salisbury.edu/~xswang/Research/Papers/SERelated/Agile/p72-nerur.pdf

Palmquist, M. S., Lapham, M. A., Miller, S., Chick, T., & Ozkaya, I. (Octubre de 2013). *Parallel Worlds: Agile and Waterfall Difference and Similarities.* Obtenido de Software Engieering Institute Carnegie Mellon University: https://resources.sei.cmu.edu/asset_files/TechnicalNote/2013_004_001_62918.pdf

Project Managament Institute. (2012). *A guide to the Project MAnagement Body of Knowledge Fifth Edition.* Newtown Square, Pennsylvania: Project Managament Institute, Inc.

Royce, W. W. (Octubre de 2013). *MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS.* Obtenido de USC University of Southern California: http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf

Rubin, K. S. (Septiembre de 2016). *Essential Scrum: A practical guide to the most popular agile process.* Obtenido de Agile for all: http://agileforall.com/wp-content/uploads/2014/07/Essential_Scrum_Chapter_2-.pdf

S. Pressman, R. (2006). *Ingeniería del software un enfoque préctico, septima edición.* Mexico, D.F.

Saenz Arteaga, A. R. (Diciembre de 2012). *El éxito de la gestión de proyectos: un enfoque entre la tradicional y lo dinámico.* Obtenido de Tesis doctorales en Xarxa: http://www.tdx.cat/bitstream/handle/10803/117483/Arturo_Saenz_%20Tesis_2012_Rev_1.pdf

Schwaber, K., & Sutherland, J. (Julio de 2016). *La guía de Scrum.* Obtenido de Scrum Guides: http://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish.pdf

Sierra Caballeo, K. A., Valencia, J. I., & Moralez, J. J. (s.f.). *NORMA ISO/IEC 9004-2*. Obtenido de Normas ISO - Calidad de software: http://tecnicosistemastecnologoadsi.blogspot.com/2016/02/norma-isoiec-9004-2.html

Singh, R. (s.f.). *INTERNATIONAL STANDARD ISO/IEC 12207 SOFTWARE LIFE CYCLE PROCESS.* Obtenido de Abelia: http://www.abelia.com/docs/12207cpt.pdf

Sinha, R. (2010). *Prácticas Ágiles - Desarrollo de software con un enfoque ágil.* Obtenido de Project Management Institute: https://americalatina.pmi.org/latam/KnowledgeCenter/Articles/~/media/2DE8D1887A4B4360ACF630B82716FAA0.ashx

Snowden, D. F., & Boone, M. E. (10 de Julio de 2014). *A leaders Framework for decision making.* Obtenido de Australian Catholic University: https://www.acu.edu.au/__data/assets/pdf_file/0006/659004/David_Snowden_A_Leaders_Framework_for_Decision_Making.pdf

Sutherland, J. (2011). *The Scrum papers: Nut, Bolts, and Origins of an Agile Framework.* Obtenido de HB Agency: http://www.hbagency.com/scrum/tl_files/scrum_inc/documents/ScrumPapers.pdf

10 Anexos.

Anexo A: Instrumento para captura de entrevista a los expertos.

Entrevista dirigida a los expertos Scrum	
Fecha de entrevista:	
Profesión del entrevistado:	
Puesto de trabajo:	

Rubro:	Buenas prácticas para implementar la metodología
1	¿Cuáles considera que son factores de éxito para implementar Scrum?

Rubro:	Buenas prácticas para implementar la metodología
2	¿Qué estrategias ha implementado para abordar el tema de la cultura organizacional y la renuencia al cambio, respecto a la metodología Scrum?

Rubro:	Buenas prácticas para implementar la metodología
3	¿Cuáles considera que son los factores de éxito que pueden influir positivamente en el desarrollo de proyectos con Scrum?

Rubro:	Buenas prácticas para implementar la metodología
4	En base a su experiencia, ¿qué factores negativos han influenciado para que las tareas de desarrollo no puedan finalizar en el tiempo estimado?

Rubro:	Buenas prácticas para implementar la metodología
5	¿Cuál ha sido su estrategia para planificar las reuniones con el equipo de desarrollo y el Product Owner, sin que afecte negativamente el progreso de las actividades?

Rubro:	Buenas prácticas para implementar la metodología
6	¿Qué estrategias de comunicación han sido exitosas con los clientes e interesados durante la ejecución del proyecto?

Rubro:	Validación de la metodología
7	¿Qué tipos de empresa pueden adoptar la metodología Scrum?

Rubro:	Validación de la metodología
8	¿La metodología SCRUM puede ser aplicable a cualquier tipo de proyecto de desarrollo de software, pequeño, mediano o grande?

Rubro:	Validación de la metodología
9	¿Cómo perfilaría al recurso humano, para otorgar los roles: Product Owner, Scrum Master y Desarrollador?

Rubro:	Recomendaciones
10	¿Cuál es la cantidad mínima de recurso humano que debe existir en un departamento de desarrollo de sistemas, para la implementación de Scrum?

Rubro:	Recomendaciones
11	¿Considera importante la implementación de una herramienta que permita la gestión de proyectos ágiles?

Rubro:	Recomendaciones
12	Scrum valora el Software en funcionamiento sobre la documentación, pero durante las fases del ciclo implementadas, ¿qué puntos son importantes a documentar?

Rubro:	Recomendaciones
13	En base a su experiencia, ¿cuáles han sido beneficios al implementar Scrum y cuál ha sido su estrategia para comunicar los resultados a la gerencia de la empresa e interesados?

Rubro:	Recomendaciones
14	Ante múltiples proyectos en la organización, ¿cómo configuraría los equipos Scrum?

Rubro:	Recomendaciones
15	Para describir los requerimientos, ¿considera necesario solo el uso de Historias de Usuario o Historias de Usuarios complementadas con Casos de Uso?

Rubro:	Recomendaciones
16	¿Qué estrategia han utilizado en sus Equipos de Desarrollo para realizar una estimación de tiempos apegada a la realidad?

Rubro:	Recomendaciones
17	Desde su punto de vista, ¿Considera que los roles de Scrum deben ser estáticos y pertenecer a una estructura organizativa, o deben ser dinámicos e independientes de la estructura organizativa de la empresa?

Anexo B: Instrumento para captura de entrevista a miembros de la empresa.

Entrevista dirigida a miembros de la empresa	
Fecha de entrevista:	
Profesión del entrevistado:	
Puesto de trabajo:	

Rubro:	Metodología de desarrollo actual
1	¿Qué tipo de metodología de desarrollo de software utilizan?

Rubro:	Metodología de desarrollo actual
2	Al momento de iniciar un proyecto de desarrollo nuevo, ¿cuentan con una metodología de trabajo establecida, podría describirla?

Rubro:	Metodología de desarrollo actual
3	¿La metodología actual responde a las necesidades de la empresa?

Rubro:	Problema actual
4	¿Considera que la metodología actual involucra a todos los miembros del equipo de forma integral en todas las fases del proyecto?

Rubro:	Problema actual
5	Describa cuales son las ventajas y desventajas de la metodología actual

Rubro:	Problema actual
6	¿Considera acertada la estimación de tiempos al planificar las actividades de desarrollo?

Rubro:	Problema actual
7	¿En qué fases del desarrollo del proyecto se concentra la documentación?

Rubro:	Problema actual
8	¿Cómo gestionan el cambio de requerimientos una vez el proyecto está en marcha?

Rubro:	Nivel de conocimientos de Scrum
9	¿Qué conocimientos posee de la metodología Scrum?

Rubro:	Nivel de conocimientos de Scrum
10	¿Ha participado en proyectos de desarrollo, ejecutados bajo la metodología Scrum?

Rubro:	Nivel de conocimientos de Scrum
11	¿Qué rol desempeñó y cuáles son sus experiencias al respecto?

Rubro:	Nivel de conocimientos de Scrum
12	¿Posee algún diplomado o certificación sobre la metodología Scrum?

Rubro:	Expectativas de la metodología
13	¿Cuáles son sus expectativas al adoptar esta metodología?

Rubro:	Expectativas de la metodología
14	¿Considera que la organización se encuentra preparada para implementar esta metodología? Describa porque si o por qué no