



CIMAT

Centro de Investigación en Matemáticas, A.C.

DESARROLLO DE UNA GUÍA PARA REFORZAR EL PROCESO DE DESARROLLO DEL PRODUCTO DE SOFTWARE DE STARTUPS EN INCUBACIÓN IMPLEMENTANDO EL ESTÁNDAR ISO/IEC 29110 MEDIANTE PRÁCTICAS ÁGILES

T E S I S

Que para obtener el grado de
Maestro en Ingeniería de Software
con Orientación en
Gestión de Procesos

Presenta
Arturo Lagunas Inocencio

Director de Tesis:
Dra. Mirna Ariadna Muñoz Mata

Co-Director de Tesis:

Autorización de la versión final

Zacatecas, Zac. 19 de Diciembre de 2018

Agradecimientos

A mi madre por el gran amor que me ha brindado en todos los momentos. A mi padre por los sabios consejos que me da. A mis hermanos por todo el cariño que me han brindado desde pequeño.

A Sandy, Ricardo, Mario, Jo, Rodrigo y demás amigos por haber permanecido conmigo en los momentos agradables y difíciles.

A CONACYT y CIMAT por el apoyo financiero que me otorgó para mi estancia en la maestría y el desarrollo del presente trabajo de tesis.

A los doctores, maestros y personal administrativo del Centro de Investigación en matemáticas (CIMAT) Unidad Zacatecas por el aprendizaje académico y apoyo personal que me brindaron a lo largo de mi estancia en la maestría.

A la Dra. Mirna Ariadna Muñoz Mata por el tiempo y paciencia que me brindó para la realización de este trabajo de tesis, el aprendizaje que obtuve en mi estancia en la maestría, los consejos para llegar a ser mejor persona y el inmenso apoyo en los momentos más difíciles de mi vida.

Abstract

Nowadays, the number of Startups in training process keeps growing day to day through incubators of high impact enterprises. Many of these Startups are focused on software development, so it is important that the products and/or services they develop have an appropriate quality to satisfy the customer's needs.

Most of the time, small or very small enterprises prefer an agile software development so that they use Scrum or eXtremme Programming (XP) methodologies, because of the fast response to changes and more participation of the customers within the project they promote. However, the inaccurate implementation of these methodologies affects the quality of the software products and services.

In this context, the ISO/IEC 29110 is a quality standard created to improve the Very Small Entities (VSE) software development implementing better practices of software engineering no matter the model or methodology used.

This research proposes a guide to adopt best practices for the project management and software development, so that it enables strengthening the correct implementation of agile methodologies in Startups in training process.

The validation results of this guide demonstrate that Startups and incubators have a good perception of the guide as support in the implementation of best practices, based on the ISO/IEC 29110 standard using agile practices. This was proved with a case study in which 5 startups and 1 incubator were involved.

Keywords: ISO/IEC 291100, agile methodology, Scrum, eXtremme programming, Startup.

Resumen

Hoy en día, el número de Startups en proceso de formación sigue incrementando día a día a través de las incubadoras de empresas de alto impacto. Muchas de estas Startups se enfocan en el desarrollo de software, por lo que es importante que los productos y/o servicios que desarrollen tengan la calidad adecuada para satisfacer las necesidades de sus clientes.

Este tipo de empresas pequeñas o muy pequeñas prefieren el desarrollo ágil, utilizando metodologías como Scrum o eXtreme Programming (XP), debido a la rápida respuesta a cambios y una mayor participación por parte del cliente en el proyecto que estas promueven. Sin embargo la incorrecta implementación de estas metodologías afecta la calidad del producto/servicio de software.

En este contexto, el ISO/IEC 29110 es un estándar de calidad creado para apoyar Muy Pequeñas Entidades (MPE) a llevar a cabo el desarrollo de software implementando buenas prácticas de ingeniería de software, no importando el modelo o metodología de desarrollo utilizado.

Esta investigación propone una guía para la adopción de buenas prácticas para la gestión y desarrollo de software, reforzando así la correcta implementación de metodologías ágiles en Startups durante su proceso de formación.

Los resultados de la validación de esta guía demuestran que las startups e incubadoras tienen una buena percepción de la guía como apoyo en la implementación de buenas prácticas basándose en el estándar ISO/IEC 29110 mediante el uso de prácticas ágiles. Esto se demostró mediante un estudio de caso llevado a cabo con 5 Startups y 1 incubadora.

Palabras clave: ISO/IEC 291100, agile methodology, Scrum, eXtreme programming, Startup.

Índice

Contenido

Abstract	4
Resumen	5
Índice de Figuras	9
Índice de Tablas	10
Introducción	11
Capítulo 1 Antecedentes.....	12
1.1 Marco teórico	12
1.1.1 ISO/IEC 29110-5-1-1 perfil de entrada.....	12
1.1.2 Metodología de desarrollo de software	20
1.1.3 Desarrollo Ágil.....	20
1.1.4 SCRUM	20
1.1.5 XP	22
1.1.6 Kanban	22
1.1.7 Startup	23
1.1.8 Lean Startup.....	24
1.2 Planteamiento del problema	24
1.3 Objetivos.....	25
1.3.1 Objetivo General.....	25
1.3.2. Objetivos específicos	25
1.4 Hipótesis	26
1.5 Justificación.....	26
Capítulo 2 Estado del arte	27
2.1 Revisión sistemática.....	27
2.1.1 Planificación de la revisión sistemática	27
2.1.1.1 Identificación de la necesidad de la revisión	27
2.1.1.2 Preguntas de investigación	28
2.1.1.3 Cadenas de búsqueda	28

2.1.1.4 Fuentes de datos.....	28
2.1.2 Ejecución de la revisión sistemática	28
2.1.2.1 Criterios de inclusión y exclusión.....	28
2.1.2.2 Selección de estudios primarios	29
2.1.2.3 Aseguramiento de la calidad de los estudios.....	30
2.1.3 Reporte de resultados.....	31
2.1.3.1 ¿Cuáles son las prácticas de metodologías ágiles y prácticas complementarias que usan las empresas para ajustarse al ISO/IEC 29110?.....	31
2.1.3.2 ¿Qué herramientas usan las empresas para la implementación de metodologías ágiles que se adapten al ISO/IEC 29110?	34
2.1.3.3 ¿Qué opinión tienen las empresas que han llevado a cabo la implementación del ISO/IEC 29110?	35
2.2 Trabajos relacionados	36
Capítulo 3 Metodología para el desarrollo de la tesis	38
3.1 Ejecución de la metodología.....	38
Capítulo 4 Propuesta de tesis	39
4.1 Introducción de la propuesta.....	40
4.2 Identificar las definiciones de prácticas/técnicas ágiles.	40
4.3 Realizar un mapeo de prácticas/técnicas ágiles y tareas ISO/IEC 29110 perfil de entrada.....	42
4.4 Definir un proceso de incubación genérico	48
4.5 Realizar mapeo al proceso de incubación genérico.....	50
4.6 Diseñar la guía de implementación	52
4.7 Desarrollar la guía de implementación.....	54
Capítulo 5 Resultados.....	64
5.1 Validación de la guía	64
5.2 Estudio de caso	68
5.2.1 Diseño y planificación del estudio de caso	68
5.2.1.1 Objetivo.....	68
5.2.1.2 Objeto de estudio	69
5.2.1.3 Marco de referencia	69
5.2.1.4 Preguntas de investigación	69

5.2.1.5 Método	70
5.2.2 Preparación de la recogida de datos	70
5.2.3 Recogida de datos.....	71
5.2.4 Análisis de resultados	71
5.2.5 Reporte de resultados.....	73
Capítulo 6 Conclusiones.....	75
6.1 Conclusiones	75
6.2 Trabajos futuros.....	76
6.2 Logros.....	76
Referencias	78
Anexos.....	81
ANEXO A.....	81

Índice de Figuras

Figura 1. Proceso de Gestión del Proyecto	13
Figura 2. Proceso de Implementación de software	14
Figura 3. Tablero Kanban con las columnas marcadas.	23
Figura 4. Flujo de trabajo que se siguió para el desarrollo de la guía.	38
Figura 5. Definiciones de prácticas ágiles.	41
Figura 6. Mapeo entre las prácticas ágiles y las actividades / tareas del ISO/IEC 29110.	43
Figura 7. Ejemplo de justificaciones dadas a cada uno de los puntos del mapeo.	44
Figura 8. Ejemplo de asignación de puntuación y de cobertura para cada uno de los mapeos y justificaciones establecidas.....	46
Figura 9. Porcentaje de cobertura de las prácticas ágiles respecto a las actividades de ISO/IEC 29110.	47
Figura 10. Ciclo de ejecución de la guía.	54
Figura 11. Inicio de la página de la guía.	64
Figura 12. Glosario de términos en inicio de página de la guía.	65
Figura 13. Proceso de incubación genérico en inicio de la página de la guía.	65
Figura 14. Apartados de Inicio de proyecto parte 1	66
Figura 15. Apartados de Inicio de proyecto parte 2.	67
Figura 16. Respuestas de la pregunta 1.	71
Figura 17. Respuestas de la pregunta 2.	72
Figura 18. Respuestas de la pregunta 3.	72
Figura 19. Respuestas de la pregunta 4.	73
Figura 20. Plantilla de Tablero de Tareas.....	81
Figura 21. Plantilla de Historia de Usuario.....	81
Figura 22. Ejemplo de cuestionario de retroalimentación.	82
Figura 23. Ejemplo de Burndownchart.	82

Índice de Tablas

Tabla 1. Roles dentro del ISO 29110. Tomado de (ISO 29110, 2012).....	15
Tabla 2. Productos de Salida.....	16
Tabla 3. Actividades y tareas que se siguen en el estándar. Tomado de (ISO 29110, 2012).....	17
Tabla 4. Actividades y tareas que se siguen en el estándar.....	18
Tabla 5. Resultados obtenidos al aplicar el proceso de selección de estudios primarios.	29
Tabla 6. Lista de estudios primarios.	30
Tabla 7. Resultados de las preguntas de investigación encontradas en los artículos.....	31
Tabla 8. Resultados obtenidos de las prácticas que usan las empresas.....	31
Tabla 9. Resultados obtenidos de las actividades que llevaron a cabo las empresas.	32
Tabla 10. Herramientas usadas para la implementación del ISO/IEC 29110 de manera ágil.....	34
Tabla 11. Resultados obtenidos de las barreras.	35
Tabla 12. Resultados obtenidos de los beneficios.	36
Tabla 13. Modelos de incubación analizados.	48
Tabla 14. Proceso de incubación genérico.	49
Tabla 15. Mapeo del proceso de Gestión del Proyecto.....	50
Tabla 16. Mapeo del proceso de la Implementación del Software.	51
Tabla 17. Glosario de términos.....	52
Tabla 18. Inicio del proyecto.....	55
Tabla 19. Realizar el plan del proyecto.	56
Tabla 20. Análisis de requisitos.....	58
Tabla 21. Ejecución y evaluación del proyecto.....	59
Tabla 22. Identificación de componentes de software.	60
Tabla 23. Desarrollo del software.....	61
Tabla 24. Integración de software.	62
Tabla 25. Cierre del proyecto.....	63
Tabla 26. Preguntas de la encuesta.	70
Tabla 27. Ejemplo de tarjeta CRC.....	82

Introducción

Las Startups son empresas de reciente creación que surgen debido a la necesidad de crear empresas con grandes posibilidades de crecimiento financiero, además de llegar a ser escalables.

Las Startup son formadas en incubadoras de alto impacto, y utilizan como metodología de incubación Lean Startup. Esta metodología ayuda a las Startups a encontrar su Producto Mínimo Viable, es decir el producto que los clientes comprarían, sin centrarse en buenas de desarrollo de software que incrementan la calidad del mismo. Además, las que usan una metodología para el desarrollo de software, prefieren una metodología ágil como SCRUM o eXtremme Programming (XP) debido a que consideran que el uso de estas metodologías reduce el tiempo del desarrollo de software, en costos y esfuerzo, produciendo mejores resultados. Sin embargo, la mayoría de las veces no cuenta con el conocimiento para su adecuada implementación.

Debido a la importancia de hoy en día que tienen las Startups a nivel mundial, esta tesis se enfoca en facilitar la implementación de buenas prácticas de desarrollo de software para fortalecer su proceso de desarrollo de software durante la formación de las mismas.

Para lograrlo esta tesis realiza una propuesta de la implementación del perfil de entrada del estándar ISO/IEC 29110 mediante el uso de prácticas ágiles como una solución para reforzar los procesos de gestión y de desarrollo de software que apoyen en incrementar la calidad de sus productos.

La estructura de esta tesis se compone de 5 capítulos los cuales se detallan a continuación:

Capítulo 1 Antecedentes. Este capítulo explica los conceptos básicos que se usaron para la investigación, así como la problemática, la justificación, los objetivos generales y específicos.

Capítulo 2 Estado del arte. En este capítulo se presenta la situación actual en relación al estándar ISO/IEC 29110 y los entornos ágiles. Para llevar a cabo este capítulo, se realizó una revisión sistemática obteniendo los estudios relacionados.

Capítulo 3 Metodología de desarrollo de la tesis. En este capítulo se describe cómo se realizó la tesis y se describen las actividades llevadas a cabo.

Capítulo 4 Propuesta de tesis. En este capítulo se explica el desarrollo de la guía, y las actividades necesarias que se llevaron a cabo para su desarrollo.

Capítulo 5 Resultados. En este capítulo se describen los resultados obtenidos de las validaciones obtenidas de incubadoras y Startups.

Capítulo 1 Antecedentes

En este capítulo se presentan los antecedentes, comenzando con el marco teórico que contiene la definición de los conceptos fundamentales a partir de los cuales se desarrolla la presente investigación. Posteriormente se presenta: el planteamiento del problema, los objetivos generales y específicos y la justificación.

1.1 Marco teórico

El objetivo del marco teórico es proporcionar un marco de referencia conceptual necesario para comprender la investigación, se integra a partir de enfoques teóricos, estudios y antecedentes en general, que se consideren válidos para la adecuada fundamentación del estudio (Sampieri, Fernández & Baptista, 2010). A continuación se presentan los conceptos fundamentales directamente involucrados con el desarrollo de la presente investigación.

1.1.1 ISO/IEC 29110-5-1-1 perfil de entrada

El ISO/IEC 29110, es un estándar desarrollado por la ISO (Organización Internacional de Normalización) y la IEC (Comisión Electrotécnica Internacional), cuyo propósito es que las MPE's (Muy Pequeñas Entidades) mejoren la calidad de sus productos y servicios, así como el rendimiento de sus procesos. Una MPE es una empresa pequeña que va desde 2 hasta 25 trabajadores. Este modelo no discrimina el uso de diferentes ciclos de vida que existen: cascada, iterativo, incremental, ágil, evolutivo (ISO 29110, 2012).

Algunos de los beneficios que pueden obtener las MPE's son:

1. Entregar al cliente un conjunto acordado de requisitos de proyecto y productos esperados.
2. Dar seguimiento a un proceso sistemático de desarrollo de software, que satisfaga las necesidades del cliente y asegure la calidad del producto.

Este estándar comprende dos procesos: Gestión del Proyecto (GP) (ver Figura 1) e Implementación del Software (IS) (ver Figura 2). Por una parte, el objetivo del proceso de Gestión del proyecto es implementar las tareas que se llevan a cabo en el proceso de Implementación del Software, de manera que cumplan con la calidad, tiempo y costos establecidos. Por otra parte, el proceso de Implementación de Software tiene como objetivo llevar a cabo el análisis, diseño, codificación y las pruebas del software tomando como referencia los requisitos que se especifican y el plan del proyecto actual (ISO 29110, 2012).

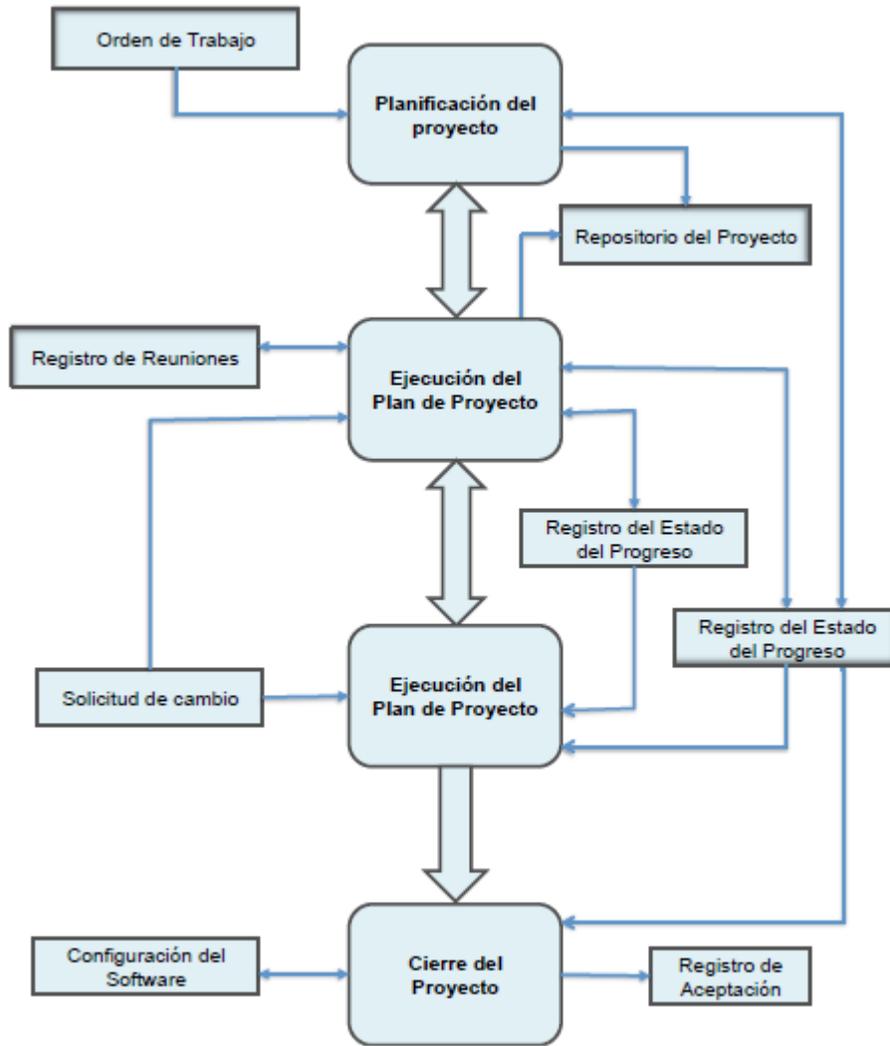


Figura 1. Proceso de Gestión del Proyecto tomado de (ISO 29110, 2012).

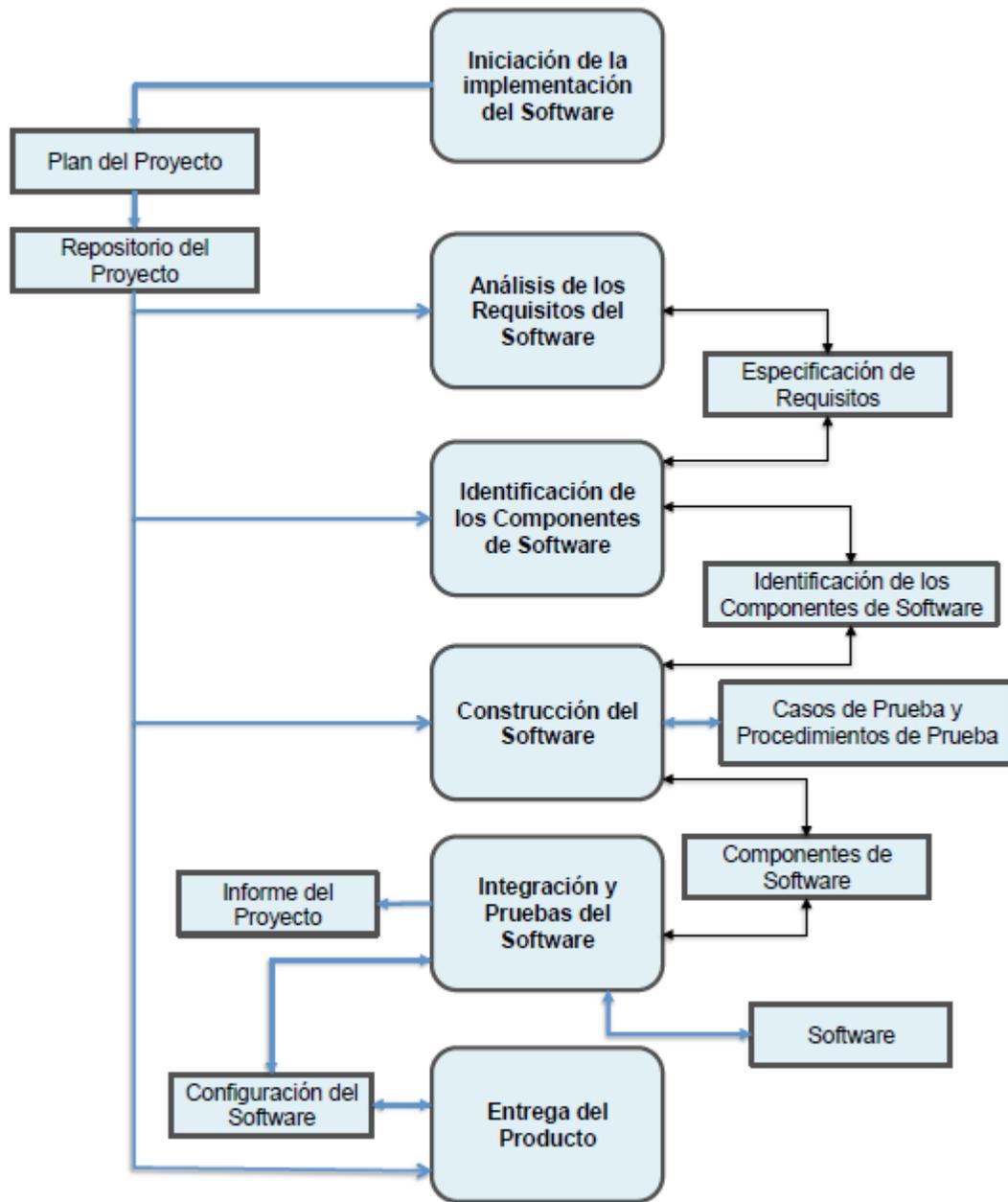


Figura 2. Proceso de Implementación de software tomado de (ISO 29110, 2012).

Como parte de los procesos de Gestión de Proyecto e Implementación del Software el ISO/IEC 29110 menciona los roles mostrados en la Tabla 1:

Tabla 1. Roles dentro del ISO 29110. Tomado de (ISO 29110, 2012).

Rol	Descripción	Abreviatura
Cliente	Organización o persona que recibe un producto o servicio [ISO 12207:2008].	CL
Gerente de Proyecto	Responsable de coordinar la realización del producto o servicio.	GP
Equipo de desarrollo	Responsable de la creación del producto o servicio.	ET

Además, como observa en la Tabla 2, este estándar provee un conjunto de productos de salida (ISO 29110, 2012), que son los elementos resultado de la realización de las actividades que se llevan a cabo en el estándar.

Tabla 2. Productos de Salida. Tomado de (ISO 29110, 2012).

Producto	Descripción	Fuente
Registro de Aceptación	Documento en donde el cliente está de acuerdo con los entregables del proyecto.	Gestión del Proyecto
Solicitud de Cambio	Documento donde se identifican cambios o modificaciones que el cliente solicite.	Implementación de Software, Cliente, Gestión del Proyecto
Registro de Reuniones	Registro de los acuerdos establecidos con el Cliente y/o el Equipo de desarrollo.	Gestión del Proyecto
Registro de Estado de Progreso	Registro del estado del proyecto contra el Plan del Proyecto.	Gestión del Proyecto
Plan de Proyecto	Presenta cómo serán ejecutados los procesos y actividades del proyecto para asegurar su conclusión exitosa, así como la calidad de los productos entregables.	Gestión del Proyecto
Repositorio del Proyecto	Contenedor electrónico para almacenar los productos de trabajo y entregables del proyecto.	Gestión del Proyecto
Especificación de Requisitos	Identificación de los requisitos de software.	Implementación del Software
Software	Elemento de software (código fuente del software y código ejecutable) para el cliente, constituido por una colección de Componentes de Software integrados.	Implementación del Software
Componentes de Software	Un conjunto de productos de software únicos y consistentes identificados incluyendo: - Especificación de los Requisitos - Software	Implementación del Software
Configuración del Software	Un conjunto de productos de software identificados de forma única y consistente.	Implementación del Software
Identificación de Componentes de Software	Información textual y gráfica sobre la estructura del software.	Implementación del Software
Declaración del Trabajo	Descripción del trabajo a ser realizado en relación al desarrollo de Software.	Cliente
Casos de Prueba y Procedimientos de Prueba	Elementos necesarios para probar el código.	Implementación del Software
Informe de pruebas	Documentación de la ejecución de las pruebas.	Implementación del Software

La Tabla 3 muestra el proceso de Gestión del Proyecto con actividades, tareas y productos de salida que sigue el estándar. (ISO 29110, 2012).

Tabla 3. Actividades y tareas que se siguen en el estándar. Tomado de (ISO 29110, 2012).

ISO/IEC 29110 Actividades del proceso Gestión del Proyecto			
Actividad	Rol	Lista de Tareas	Producto de Salida
Planificación del Proyecto	GP ET	Revisar de la Orden de Trabajo.	Orden de Trabajo
	GP ET	Identificar las Tareas específicas a realizar para producir las entregas y sus componentes de software identificados en la Orden de Trabajo.	Plan de Proyecto
	GP ET	Establecer la Duración Estimada para realizar cada tarea.	Plan de Proyecto
	GP ET	Identificar y documentar los recursos.	Plan de Proyecto
	GP ET	Establecer la Composición del Equipo de desarrollo.	Plan de Proyecto
	GP ET	Asignar las fechas estimadas de inicio y finalización para cada una de las tareas.	Plan de Proyecto
	GP	Calcular y documentar el Esfuerzo y Costo Estimados del proyecto.	Plan de Proyecto
	GP ET	Identificar y documentar los riesgos que puedan afectar al proyecto.	Plan de Proyecto
	GP	Generar el Plan de Proyecto	Plan de Proyecto
	GP CLI	Revisar y aceptar las partes del plan del proyecto de manera apropiada. El cliente revisa y acepta el plan del proyecto.	Plan de Proyecto
	GP ET	Establecer el repositorio del proyecto.	Repositorio del Proyecto
Ejecución del Plan de Proyecto	GP ET	Monitorear y registrar el estado de la ejecución del Plan de Proyecto.	Registro de Estado de Proceso
	GP CLI ET	Llevar a cabo reuniones con el cliente.	Registro de Reuniones Solicitud de Cambio
Evaluación y Control del Proyecto	GP ET	Evaluar el progreso del proyecto con respecto al plan del proyecto, comparando.	Registros del estado del progreso [evaluado]
	GP ET	Evaluar y monitorear las solicitudes de cambio realizadas por el cliente.	Solicitud de Cambio [Seguimiento realizado]
	GP ET	Establecer acciones para corregir desviaciones o problemas.	
Cierre del Proyecto	GP ET	Formalizar la finalización del proyecto,	Registro de Aceptación
	GP	Actualizar el Repositorio del Proyecto.	Repositorio del Proyecto

La Tabla 4 muestra el proceso de Implementación de Software con actividades, tareas y productos de salida que sigue el estándar. (ISO 29110, 2012).

Tabla 4. Actividades y tareas que se siguen en el estándar. Tomado de (ISO 29110, 2012).

ISO/IEC 29110 Actividades del proceso Implementación de Software			
Actividad	Rol	Lista de Tareas	Producto de Salida
Iniciación de la Implementación de Software	GP ET	Revisar del Plan de Proyecto actual con los miembros del Equipo de desarrollo.	Plan de Proyecto
	ET	Establecer o actualizar el ambiente de implementación.	
Análisis de Requisitos de Software	GP ET	Asignar tareas a los miembros del Equipo de desarrollo.	
	ET CLI	Documentar o actualizar la Especificación de Requisitos.	Especificación de Requisitos [verificada]
	CLI	Validar y obtener la aprobación de la Especificación de Requisitos.	Especificación de Requisitos [validada]
Identificación de Componentes de Software	GP ET	Asignar tareas a los miembros del Equipo de desarrollo.	
	ET	Entender la Especificación de Requisitos.	
	ET	Documentar o actualizar la Identificación de Componentes de Software.	
Construcción del Software	GP ET	Asignar tareas a los miembros del Equipo de desarrollo.	
	ET	Entender la Identificación de Componentes de Software.	
	ET	Construir o actualizar los Componentes de Software.	Componentes de Software
	ET	Establecer o actualizar los Casos de Prueba y los Procedimientos de Prueba.	Casos de Prueba y Procedimientos de Prueba
	ET	Probar los Componentes de Software.	Componentes de Software [unidades probadas]
Integración y Pruebas del Software	GP ET	Asignar Tareas a los miembros del Equipo de desarrollo.	
	ET	Entender los Casos de Prueba y Procedimientos de Prueba.	

	ET	Integrar el Software usando los Componentes de Software y actualizar los Casos de Prueba y Procedimientos de Prueba.	Software Casos de Prueba y Procedimientos de Prueba [actualizados]
	ET	Realizar las pruebas de Software usando los Casos de Prueba y Procedimientos de Prueba.	Software [probado] Informe de pruebas
	ET	Corregir los defectos encontrados hasta lograr una prueba exitosa.	Software [corregido] Informe de Pruebas [defectos eliminados]
	ET	Incorporar la Especificación de Requisitos y el Software en la Configuración de Software.	Configuración de Software Especificación de Requisitos Software
Entrega de Producto	GP ET	Asignar tareas a los miembros del Equipo de desarrollo.	
	ET	Revisar la comprensibilidad de la Configuración del Software.	
	GP ET	Llevar a cabo la entrega al Gestor del Proyecto y dar soporte a la entrega de acuerdo con el Plan de Proyecto.	Configuración del Software [lista para ser entregada]

1.1.2 Metodología de desarrollo de software

Una metodología de desarrollo de software es un marco de trabajo en donde se estructura la manera en que se lleva a cabo la planeación, ejecución y control del desarrollo de software, proporcionando un proceso sistemático que se sigue para idear, implementar y mantener un producto de software (Centers for Medicare & Medicaid Services, 2008). La metodología describe tres principales etapas:

1. Planeación: Es la etapa inicial, en la que se desarrolla la idea junto con la creación de las actividades que se van a desarrollar.
2. Ejecución: Es la etapa donde se ejecutan las actividades que se realizaron en la Planeación y que conducen a la creación del producto.
3. Control: Es la etapa donde se lleva a cabo la administración y seguimiento de las actividades que se van realizando incluso hasta la posterior entrega del producto.

1.1.3 Desarrollo Ágil

El desarrollo ágil es una forma de desarrollo iterativo incremental en donde se da una solución cada cierto tiempo y donde esa solución añade un valor al producto final que se vaya a desarrollar. El tiempo que toma producir un incremento funcional del producto, potencialmente entregable se denomina Sprint y es aquí donde se lleva a cabo el desarrollo de software, donde este tiempo es a corto plazo. (Dingsøyr, Dybå, & Moe, 2010)

El desarrollo de software de manera ágil se centra mucho en las interacciones con el cliente y con el equipo de desarrollo. También, es importante mencionar que el desarrollo ágil debe estar adaptado de manera que los cambios que se produzcan en el software se aprueben de una manera más rápida. (Highsmith & Cockburn, 2001)

Entre las principales metodologías ágiles se encuentran Scrum y XP, mismas que se describen a continuación.

1.1.4 SCRUM

Scrum es una metodología de desarrollo de software, basada en el modelo Iterativo Incremental, el cual consiste en entregas constantes hacia el cliente y con un valor agregado en cada entrega (Satpathy, 2016).

Scrum cuenta con 3 roles principales que son:

- **Product Owner:** es el miembro que tiene una comunicación directa con el cliente y el encargado de que el trabajo vaya cumpliendo con objetivo del negocio. También, ayuda a escribir las Historias de Usuario.
- **Scrum Master:** como miembro del equipo del proyecto, el Scrum Master es la persona que ayuda al equipo de desarrollo a llevar de manera correcta la metodología y verificar que las Historias de Usuario se estén llevando a cabo en el Sprint correspondiente.
- **Equipo de desarrollo:** el equipo de desarrollo se refiere a las personas profesionales que realizan la codificación, pruebas y diseño del software. El mismo equipo de desarrollo estructura las tareas que se van a realizar y organizan el trabajo.

En Scrum se manejan eventos, los cuales consisten en las actividades que se realizan durante la ejecución de un proyecto de software (Blankenship, Bussa, & Millett, 2011). Estos eventos son:

1. **Sprint Planning:** el Sprint planning se realiza el primer día de cada sprint. Se reúne el Scrum master, product owner y el equipo de desarrollo. El product owner presenta las características que se quieren llevar a cabo y el Equipo de desarrollo determina las tareas necesarias para cumplir con dichas características. Además, se revisan las estimaciones de trabajo para ver si el equipo puede llegar a completar las características requeridas en el sprint, sino es así, las características de prioridad más baja se ponen al final del product backlog, hasta que la carga de trabajo de ese sprint es lo suficientemente buena para comenzar.
2. **Daily Stand-Up:** estas, son reuniones cortas que se realizan entre el Equipo de desarrollo y el scrum master diariamente, para ver la situación del proyecto, qué ha hecho cada miembro, además de ayudar a otros miembros del equipo si lo necesitan. Estas iteraciones no deberían durar más de 15 minutos y el Scrum Master es el que se encarga de moderar estas reuniones
3. **Sprint Review:** la revisión del Sprint se lleva a cabo al final del sprint. Su propósito es que el equipo presente las historias de usuario completadas durante el sprint.
4. **Sprint Retrospective:** en el Sprint Retrospective se revisa lo que el equipo hizo bien, lo que se hizo mal y lo que se podría mejorar

1.1.5 XP

EXtremme programming es una metodología de desarrollo ágil enfocada en producir software a través de iteraciones, mejorar la comunicación con el cliente y el trabajo en equipo (Ron Jeffries, Ann Anderson, 2000). EXtremme programming cuenta con 5 fases en su metodología (Ron Jeffries, Ann Anderson, 2000) que son:

- **Planeación:** esta parte se enfoca en crear las historias de usuario, planificar el número de iteraciones y saber qué historias de usuario deben ser acabadas para cada iteración.
- **Gestión:** la gestión se refiere a la manera en que se va controlando el trabajo en el proyecto, realizando reuniones y midiendo la velocidad con la que el equipo trabaja.
- **Diseño:** en este apartado, se define el diseño del software de una manera simple y usando ciertas técnicas o prácticas que hacen que el diseño no sea tan complejo.
- **Codificación:** EXtremme Programming propone prácticas ágiles que hacen que el Equipo de desarrollo mejore la velocidad con la que producen software, por ejemplo trabajando en parejas o integrando el software de manera frecuente.
- **Pruebas:** la metodología propone la realización de pruebas de aceptación y pruebas unitarias que deben ser realizadas.

1.1.6 Kanban

Kanban es una herramienta que en el ámbito del software fue implementada por primera vez por David Anderson para llevar el control de proyectos de cualquier tipo, que van desde la manufactura hasta el desarrollo de software (Hefley, 2008).

El principal objetivo de Kanban es minimizar el trabajo que se encuentra en proceso de todos los procesos de la organización (Hefley, 2008).

Kanban se basa en la utilización de columnas (Hefley, 2008) como a continuación se describe y se muestra gráficamente en la Figura 3.

- **Columna “Backlog”,** en esta columna se encuentran las tareas pendientes a realizar.
- **Columna “Para Hacer”,** en esta columna se encuentran las tareas seleccionadas que se van a realizar en un tiempo determinado.

- Columna “Haciendo”, en esta columna se encuentran las tareas que se están realizando en ese momento.
- Columna “Probando”, en esta columna se encuentran las tareas que se están probando o validando.
- Columna “Finalizado”, en esta columna se encuentran las tareas que se han cubierto por completo y fueron realizadas correctamente.

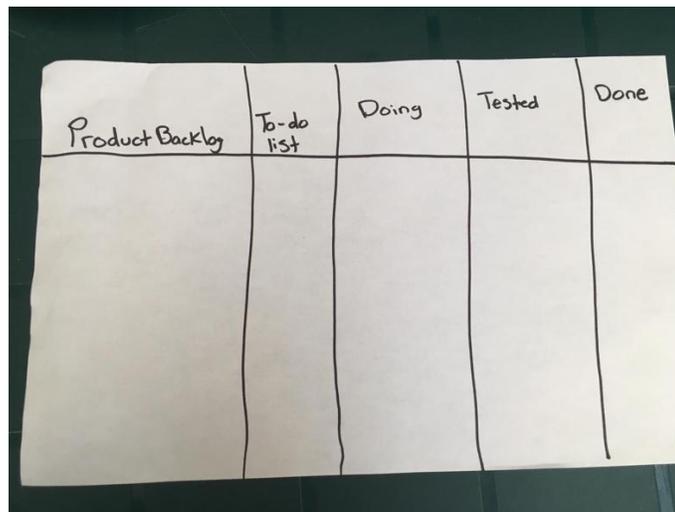


Figura 3. Tablero Kanban con las columnas marcadas.

Es importante recalcar que existen en Kanban políticas que establecen criterios para pasar de un estado de proceso a otro, con el objetivo de controlar que solo aquellas actividades hayan cumplido su función puedan cambiarse de columna (Hefley, 2008).

1.1.7 Startup

De acuerdo a Eric Ries, fundador de Lean Startup: “Una Startup es una institución humana diseñada para entregar un nuevo producto o servicio bajo condiciones extremadamente inciertas” (Ries, 2011).

Una Startup es una empresa de nueva creación con altas posibilidades de crecimiento y escalable. Generalmente, las Startups son empresas emergentes estrechamente relacionadas al uso de tecnologías digitales y siempre buscando una rápida consolidación hacia dos vías, crecer como negocio o ser absorbida por otra empresa (Ries, 2011).

Las principales características de las Startups son:

- Generar productos y servicios innovadores.
- Obtener mayor tolerancia al fracaso.
- Tener equipos de trabajo comprometidos.
- Dar una mayor calidad en el producto/servicio que ofrecen.
- Trabajar con bajos costos de operación.
- Guiarse a través de metodologías y herramientas (Como Lean Startup).

1.1.8 Lean Startup

Lean Startup es una metodología que se sigue para desarrollar un negocio. La característica principal de esta metodología es realizar una hipótesis donde se lanza un producto/servicio al mercado con un valor mínimo para el cliente y con ello poder recibir retroalimentación que sirve para saber si el negocio va por buen camino o si se necesita pivotar la hipótesis (Ries, 2011).

Algunas características de Lean Startup son:

- Crear un producto mínimo viable.
- Realizar experimentos Split-Test.
- Obtener indicadores accionables.
- Decidir si realizar pivoteo.
- Uso del circuito crear-medir-aprender.

1.2 Planteamiento del problema

La industria de software creció aproximadamente más del 10% en el sector de Tecnologías de la Información y Comunicación del 2006 al 2012. Por ello, México busca fomentar la creación de empresas mediante Startups que ayuden a fortalecer la industria de software. Por lo tanto, es necesario que desde su creación las Startups desarrollen productos o servicios con calidad (Economía, 2012).

Aunado a lo anterior, se identifica que este tipo de empresas utilizan metodologías de desarrollo ágiles, debido a que consideran que usando estas metodologías mejoran el tiempo en que desarrollan software, representa menores gastos y menor esfuerzo con mejores resultados. Sin embargo, la mayoría de las veces estas metodologías no son implementadas adecuadamente que se refleja en la falta de implementación de buenas prácticas de ingeniería de software para la realización de sus productos o servicios.

La carencia del uso de buenas prácticas, en específico para la gestión y desarrollo de software, ya que afectan el rendimiento de la empresa y en algunos casos su estabilidad para mantenerse en el mercado.

Enfocado en esta problemática, este trabajo propone reforzar el proceso de formación de las Startups mediante la implementación de buenas prácticas de desarrollo de software.

1.3 Objetivos

El objetivo general y los objetivos específicos para el desarrollo del presente trabajo de investigación se muestran a continuación:

1.3.1 Objetivo General

El objetivo general de este trabajo de tesis es crear una guía que apoye a las Startups en proceso de incubación a reforzar la adopción de buenas prácticas de desarrollo de software mediante el uso de prácticas ágiles.

1.3.2. Objetivos específicos

- Identificar el estado del arte del estándar ISO/IEC 29110 en entornos ágiles, enfocado a: prácticas, implementación, herramientas y experiencias.
- Hacer un mapeo de las prácticas ágiles más comunes con las tareas que se llevan a cabo en el ISO/IEC 29110.
- Realizar un proceso genérico del proceso de incubación de las Startups.
- Diseñar y desarrollar una guía de implementación del ISO/IEC 29110 mediante el uso de prácticas ágiles para ser implementada durante el proceso de incubación de las Startups.
- Validar la guía realizando una encuesta a las startups e incubadoras de empresas.

1.4 Hipótesis

La guía de implementación en las Startups, promoverá el uso de buenas prácticas para reforzar la gestión del proyecto y el desarrollo de software durante su formación.

1.5 Justificación

Las Startups asociadas a la creación de software son actualmente una fuente de crecimiento para el país, por lo que, es importante que durante su formación adopten buenas prácticas para el desarrollo de software que les permita el logro de sus objetivos entre los que resaltan la satisfacción por parte de los clientes (García Rodríguez, Martínez Pérez, & Hernández Vasconcelos, 2018).

Los estándares de calidad ayudan a las empresas a implementar buenas prácticas para mejorar el proceso de desarrollo y así obtener mayor calidad en sus productos o servicios que ofrecen. En este contexto, el ISO/IEC 29110 está enfocado en entidades muy pequeñas (EMP) de desarrollo de software, teniendo el perfil de entrada para empresas como las Startups.

Aun cuando este estándar ayuda en la gestión e implementación de los proyectos de software ofreciendo guías para su ejecución, éstas no se siguen porque las Startups piensan que existe una documentación excesiva en ellas y no reflejan su forma de trabajo.

Por lo tanto, para facilitar la adopción de buenas prácticas propuestas en el ISO/IEC 29110, este trabajo propone la adopción de las buenas prácticas mediante el uso de prácticas ágiles, con lo cual por un lado se disminuye la resistencia para la adopción de buenas prácticas y al mismo tiempo se robustece la metodología ágil utilizada.

Capítulo 2 Estado del arte

En este capítulo se recopilan los resultados obtenidos a través de la realización de una revisión sistemática enfocada en prácticas ágiles y el estándar ISO/IEC 29110 dentro de entornos de desarrollo ágiles, además de incluir un análisis de los trabajos relacionados.

2.1 Revisión sistemática

La Revisión Sistemática de Literatura es una forma en que se pueden identificar, interpretar y evaluar investigaciones recientes, relacionadas al tema del que se quiere obtener información. (Kitchenham et al., 2009).

La revisión sistemática de literatura ha demostrado su viabilidad en otras investigaciones (Hernandez, Munoz, Mejia, & Pena, 2016)(Martinez, Mejia, & Munoz, 2016)(Mejía, Íñiguez, & Muñoz, 2017). Consta de 3 pasos como se describe a continuación (Kitchenham et al., 2009):

- Planificación de la revisión sistemática: establece por qué se necesita realizar la investigación, las preguntas de investigación, cómo y dónde se buscará la información, así como los criterios que se aplicarán al realizar la investigación.
- Ejecución de la revisión: detalla cómo se realizó la búsqueda y selección de estudios primarios, así como la extracción de la información.
- Resultados de la revisión sistemática: se muestra la información para responder las preguntas de investigación.

A continuación se describe la revisión sistemática realizada para esta investigación.

2.1.1 Planificación de la revisión sistemática

La planificación de la revisión sistemática establece por qué se necesita realizar la investigación, las preguntas de investigación, cómo y dónde se buscará la información, así como los criterios que se aplicarán al realizar la investigación

2.1.1.1 Identificación de la necesidad de la revisión

Para desarrollar una guía de implementación del ISO/IEC 29110 en entorno ágiles es necesario conocer cómo se está implementando el estándar en este tipo de organizaciones, los

obstáculos/beneficios que enfrentan, las herramientas y tecnología que usan; así como la percepción que estos tienen del estándar.

2.1.1.2 Preguntas de investigación

- ¿Cuáles son las prácticas de metodologías ágiles que usan las empresas para implementar el ISO/IEC 29110?
- ¿Qué herramientas usan las empresas para la implementación de metodologías ágiles que se adapten al ISO/IEC 29110?
- ¿Qué opinión tienen las empresas que han llevado a cabo la implementación del ISO/IEC 29110?

2.1.1.3 Cadenas de búsqueda

Para responder las preguntas de investigación se tomaron en cuenta sólo dos palabras clave: “ISO/IEC 29110” y “agile”. Posteriormente, se formó la cadena de búsqueda con el operador lógico AND obteniendo como cadena de búsqueda: ISO/IEC 29110 AND agile.

2.1.1.4 Fuentes de datos

Se tomaron aquellas bases de datos donde se publican artículos relacionados a la Ingeniería de Software, como son IEEE, Springer, Scopus, ACM y la Web of Science. Sin embargo, se encontraron pocos resultados, por lo que se optó buscar en otras fuentes como Google y ResearchGate, donde los resultados fueron más favorables.

2.1.2 Ejecución de la revisión sistemática

En la ejecución de la revisión sistemática se detalla cómo se realizó la búsqueda y selección de estudios primarios, así como la extracción de la información.

2.1.2.1 Criterios de inclusión y exclusión

Los criterios de inclusión y exclusión permiten acortar los resultados y sólo enfocar en los estudios que aporten información de interés. Los criterios de inclusión que se definieron son:

1. Artículos dentro del periodo del año 2012 al 2017.
2. Los artículos escritos en inglés o español

3. Aquellos artículos en los que aparezca la palabra ISO/IEC 29110, en el título, abstract y/o palabras clave.
4. El artículo debe mencionar el uso de prácticas ágiles para la implementación del estándar.
5. El artículo debe mencionar el uso de una herramienta en la de soporte para la implementación del estándar.
6. El artículo debe contener un estudio de caso donde se hable de alguna metodología ágil.
7. El artículo debe explicar la manera en que se llevó a cabo la implementación del estándar.
8. El artículo debe incluir opiniones del personal que implementó el estándar.

Los criterios de exclusión que se definieron son:

1. Artículos repetidos en más de una fuente de datos.
2. Artículos no enfocados en la implementación del ISO/IEC 29110 en entornos ágiles.
3. Artículos que sean extensiones de otros artículos.

2.1.2.2 Selección de estudios primarios

Para la selección de estudios primarios, se definió la siguiente estrategia:

1. Introducir la cadena de búsqueda aplicando el criterio de inclusión 1.
2. Aplicar el criterio de inclusión 2.
3. Aplicar el criterio de exclusión 1
4. Aplicar el criterio de exclusión 2.
5. Aplicar los criterios de inclusión 3, 4, 5, 6, 7 y 8.
6. Aplicar el criterio de exclusión 3.

A partir de este proceso de selección, se encontraron los resultados que se detallan en la Tabla 4.

Tabla 5. Resultados obtenidos al aplicar el proceso de selección de estudios primarios.

Paso	Scopus	Web of Science	IEEE	Springer	ResearchGate	Google	Total
1	3	3	3	8	7	2	26

2	3	3	3	8	5	2	24
3	2	0	3	8	2	2	17
4	1	0	3	7	1	0	12
5	0	0	3	4	1	0	8
6	0	0	3	3	1	0	7

2.1.2.3 Aseguramiento de la calidad de los estudios

Se crearon las siguientes preguntas con el objetivo de Asegurar la Calidad (AC) y calificar la fidelidad en los estudios seleccionados, para esta investigación: AC1) ¿El estudio se centra principalmente en el estándar ISO/IEC 29110?; AC2) ¿El estudio se centra casos reales donde se haya implementado el estándar?

Estas preguntas fueron aplicadas a todos los estudios primarios, siendo 7 artículos que cumplieron con los criterios de calidad descritos. La lista de estudios primarios se muestra en la Tabla 6.

Tabla 6. Lista de estudios primarios.

ID del artículo	Nombre del artículo
EP1	Early Stage Adoption of ISO/IEC 29110 Software Project Management Practices A Case Study (O'Connor, 2014).
EP2	Evaluating VSE Viewpoint and Sentiment Towards the ISO/IEC 29110 Standard: A Two Country Grounded Theory Study (Sanchez-Gordon, Rory V. O'Connor, & Ricardo Colomo-Palacios, 2015).
EP3	Factors driving the adoption of ISO IEC 29110 a case study of small software enterprise (Dávila & Pessoa, 2015).
EP4	Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru (Paucar, Laporte, Arteaga, & Bruggmann, 2015).
EP5	Implementing the New ISO/IEC 29110 Systems Engineering Process Standard in a Small Public Transportation Company (Laporte, Tremblay, Menaceur, & Poliquin, 2016).
EP6	Systems engineering and management processes for small organizations with ISO/IEC 29110 (Laporte, Tremblay, Menaceur, Poliquin, & Houde, 2017).
EP7	The Implementation of ISO/IEC 29110 Software Engineering Standards and Guides in Very Small Entities (Laporte & Connor, 2016).

2.1.3 Reporte de resultados

La Tabla 7 muestra la cobertura de los estudios primarios y las preguntas de investigación que se formularon (Ver sección 2.1.1.2, Preguntas de investigación). Posteriormente, se analizan los resultados obtenidos por pregunta de investigación.

Tabla 7. Resultados de las preguntas de investigación encontradas en los artículos.

ID del artículo	Pregunta de investigación		
	1	2	3
EP1	x	x	x
EP2			x
EP3	x		
EP4	x	x	
EP5	x		
EP6	x	x	
EP7	x	x	

2.1.3.1 ¿Cuáles son las prácticas de metodologías ágiles y prácticas complementarias que usan las empresas para ajustarse al ISO/IEC 29110?

En la Tabla 8, se listan las prácticas ágiles que las empresas adoptaron al implementar el estándar. Además, se incluyen prácticas complementarias a las ágiles que las empresas llevaron a cabo para la implementación del estándar.

Tabla 8. Resultados obtenidos de las prácticas que usan las empresas.

ID del artículo	Prácticas ágiles	Prácticas complementarias
-----------------	------------------	---------------------------

EP6 y EP5	<ul style="list-style-type: none"> - Uso de pruebas Unitarias. - Uso de pruebas de Integración. - Uso de pruebas de sistemas. - Uso de Historias de Usuario. - Pruebas en el sitio del cliente. 	<ul style="list-style-type: none"> - Procesar la documentación gráfica y textualmente. - Poner políticas en un sólo documento. - Usar tablas como línea base. - Controlar la terminología de definiciones en el proyecto. - Dividir los proyectos dependiendo de naturaleza y tamaño. - Crear plantillas: solicitud de propuesta, matriz de selección de proveedores, orden de compra y contrato de compra. - Usar una representación gráfica para mostrar la trazabilidad generada en varios productos de trabajo. - Ejecutar de un análisis de cobertura del proceso que se llevó a cabo.
EP4	<ul style="list-style-type: none"> - Uso de Historias de Usuario. - Uso de Product Backlog. - Uso de pruebas unitarias. - Uso de pruebas funcionales. 	<ul style="list-style-type: none"> - Realizar acuerdos vía e-mail. - Usar diseños de sistemas responsivos Web 2.0 - Usar sistemas de aplicaciones basadas en la nube. - Usar tecnologías y plataformas emergentes.
EP1	<ul style="list-style-type: none"> - Uso de tablero kanban. - Uso de Historias de usuario. 	<ul style="list-style-type: none"> - Seguir un enfoque en cascada modificado. - Uso de Excel para planeación del proyecto.
EP7	<ul style="list-style-type: none"> - Verificación en pares. - Revisión de los requisitos y la arquitectura en pares. 	<ul style="list-style-type: none"> - Entender los casos de pruebas procedurales y el ambiente de integración. - Integrar los componentes de software. - Corregir defectos. - Documentar los resultados. - Realizar trazabilidad de requisitos y diseño del producto. - Realizar y verificar documentación operacional y uso de software. - Verificar la línea base de software.

Además, se analizaron las actividades y lecciones aprendidas que se llevaron a cabo para la implementación del ISO/IEC 29110 como se observa en la Tabla 9.

Tabla 9. Resultados obtenidos de las actividades que llevaron a cabo las empresas.

Actividad	Lecciones aprendidas
-----------	----------------------

ID del artículo		Factores Positivos	Factores Negativos
EP1	<ul style="list-style-type: none"> - Desarrollar una guía de implementación propia basada en el Paquete de Puesta en Operación. 		<ul style="list-style-type: none"> - No existen asesorías de cómo aplicar las prácticas día a día. - Se necesitan ejemplos de guías y documentación detallada de cómo implementar el estándar.
EP3	<ul style="list-style-type: none"> - Definición y uso de formatos que se adecuen al estándar. - Capacitación solamente al gerente técnico. - Definición y uso nuevos formatos para cumplir las exigencias del financiamiento. - Adopción de nuevas prácticas que no se realizaban antes. 	<ul style="list-style-type: none"> -Mayor experiencia en el uso de las TI. -La gerencia tuvo mayor permanencia. 	<ul style="list-style-type: none"> -El estándar provocó mayor presión competitiva. -No se vio la utilidad percibida. -No hubo una facilidad de uso percibida. -Se necesita mayor entrenamiento con el usuario.
EP4	<ul style="list-style-type: none"> -Creación de plantillas para cada producto de trabajo. -Concretar pasos necesarios para implementar el estándar. -Desarrollo de listas de control para verificar el cumplimiento de las actividades de la Gestión del Proyecto y la Implementación del Sistema. -Desarrollo de guías de instalación, manuales de usuarios y demos. -Análisis del proceso de desarrollo de la empresa contra los requisitos del estándar. -Uso de listas de control para verificar que la empresa cumple con las condiciones de entrada del estándar. -Creación de una matriz de trazabilidad entre los productos de trabajo del ISO/IEC 29110 y los documentos o herramientas equivalentes de la empresa. 	<ul style="list-style-type: none"> - Se tuvo un nivel de calidad equivalente al CMMI nivel 3. -No se requiere una documentación excesiva. -No existe mucha inversión en recursos y tiempo para implementarse. -Atrae a nuevos clientes y proyectos más grandes. 	

De acuerdo a los resultados obtenidos, se observa que las prácticas ágiles que más usan las empresas son la creación de historias de usuario y su priorización en el product backlog, además del uso de tableros para el seguimiento de las tareas.

Dentro de los factores negativos de las actividades que se llevaron a cabo, es necesario que se proporcione capacitación del estándar al personal, así como la creación de guías con plantillas. A pesar de que el estándar no es tan robusto y no requiere actividades exhaustivas, aún existe un desconocimiento de cómo llevar a cabo dichas actividades.

2.1.3.2 ¿Qué herramientas usan las empresas para la implementación de metodologías ágiles que se adapten al ISO/IEC 29110?

Se analizaron aquellas herramientas utilizadas para la implementación del ISO 29110. Cabe mencionar que muchas de estas herramientas se centran principalmente en la Gestión del Proyecto. La Tabla 10, detalla los resultados.

Tabla 10. Herramientas usadas para la implementación del ISO/IEC 29110 de manera ágil.

ID del artículo	Nombre de la herramienta	Disponible/No disponible	Cobertura
EP1	CVS. SVN. GForge. OpenOffice Calc.	Disponibles.	- Repositorio del proyecto. - Gestión de Proyecto. - Seguimiento de fallos. - Control de versiones. - Mejora de Procesos.
EP6	Microsoft Visio.	Disponible.	- Diagramado de procesos.
EP4	Software privado de una empresa Canadiense.	No Disponible.	- Procesos de Gestión del Proyecto e Implementación del software.
EP4	BPML.	Disponible.	Cubre el proceso de la empresa.
EP7	Bugzilla.	Disponible.	Manejo de defectos.

Como se observa, existen herramientas que cubren ciertas partes de los procesos del estándar, sin embargo sólo existe una herramienta que cubre los dos procesos del estándar, sin embargo, es una herramienta privada.

2.1.3.3 ¿Qué opinión tienen las empresas que han llevado a cabo la implementación del ISO/IEC 29110?

Se analizaron las opiniones que las personas que trabajan en la industria del software tienen acerca de los procesos y la certificación del estándar. Las opiniones se han clasificado en barreras y beneficios como lo muestran las Tablas 11 y 12 respectivamente.

Tabla 11. Resultados obtenidos de las barreras.

ID del artículo	Barrera	Rol
EP1	Es difícil adoptar todo el proceso, puesto que sólo les dan una lista de criterios de actividades y no una guía de implementación detallada.	CEO.
EP1 y EP2	Con la implementación de estándares no necesariamente se percibe un valor añadido.	Miembro del Equipo de desarrollo.
EP1 y EP2	No es necesario seguir un estándar para vender o no sus productos/servicios.	Gerente del Proyecto.
EP1	Los desarrolladores sólo codifican y no tienen tiempo de cumplir las normas del estándar.	Miembro del Equipo de desarrollo.
EP2	Mala interpretación del estándar: estándar debería acoplarse a la compañía y no la compañía al estándar.	Miembro del Equipo de desarrollo.
EP2	No se ve un retorno de inversión monetario, puesto que los estándares requieren mucha documentación y las compañías les resulta difícil implementarlo.	Miembro del Equipo de desarrollo.
EP2	Existe una brecha entre los recursos humanos y las fuentes financieras.	Miembro del Equipo de desarrollo.
EP2	No se puede sobrecargar de documentación en entornos ágiles, a pesar de que se necesita documentación formal, por lo que es difícil adoptar el estándar a la realidad de la compañía.	Miembro del Equipo de desarrollo.

Tabla 12. Resultados obtenidos de los beneficios.

ID del artículo	Beneficio	Rol
EP1	Esfuerzo mínimo en la adopción del estándar.	Gerentes de Proyecto.
EP2	Crecimiento de la empresa.	Miembro del Equipo de desarrollo.
EP2	Disminución de costos de la compañía.	Miembro del Equipo de desarrollo.
EP2	Software más controlado con menos tiempo en el desarrollo.	Miembro del Equipo de desarrollo.

Con la información mostrada en las tablas anteriores se identifica la existencia de más barreras que beneficios en la implementación del estándar. Las barreras son percibidas en su mayoría por los miembros del equipo de desarrollo de software, por ejemplo desarrolladores o arquitectos.

Entre las principales barreras se encuentran: (1) la percepción de que los estándares no cumplen sus necesidades; (2) sólo es un gasto monetario de dinero que no se tiene; (3) la carencia de guías de implementación, y (4) la falta de cultura de procesos.

Algunos de los beneficios que perciben son: (1) el uso del estándar crea nuevas posibilidades de abrirse a nuevos mercados; (2) no lo perciben tan robusto y (3) puede llevarse a cabo un software controlado con un mínimo esfuerzo.

2.2 Trabajos relacionados

En esta sección se presentan los trabajos relacionados con la presente investigación, los cuales se tomarán como base para la realización de la propuesta del presente trabajo de investigación.

En *Scrum+* el autor (Galvan, Mora, O'Connor, Acosta, & Alvarez, 2015) realizó un análisis de la relación del proceso de gestión de proyecto del ISO/IEC 29110 perfil de entrada con prácticas ágiles. Este análisis se realizó a partir de la herramienta denominada IDEF0, la cual sirvió para el diseño y modelado de decisiones, acciones y actividades que se llevan a cabo en el estándar ISO 29110. En este artículo, se analizaron tres metodologías de desarrollo ágil que son XP, Scrum y UPEDU.

En *Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110*, los autores (Pasini, Esponda, Boracchia, & Pesado, 2013) realizan una comparación de Scrum con el estándar ISO 29110 en el perfil básico. Además propone roles y documentos propios, así como modificar las actividades de los dos procesos que son Gestión de Proyecto e Implementación de software que se llevan a cabo en el estándar ISO 29110.

Capítulo 3 Metodología para el desarrollo de la tesis

3.1 Ejecución de la metodología

En el capítulo 3 se llevó a cabo una metodología para el desarrollo de la tesis y las tareas que se siguieron se muestran en la Figura 3.

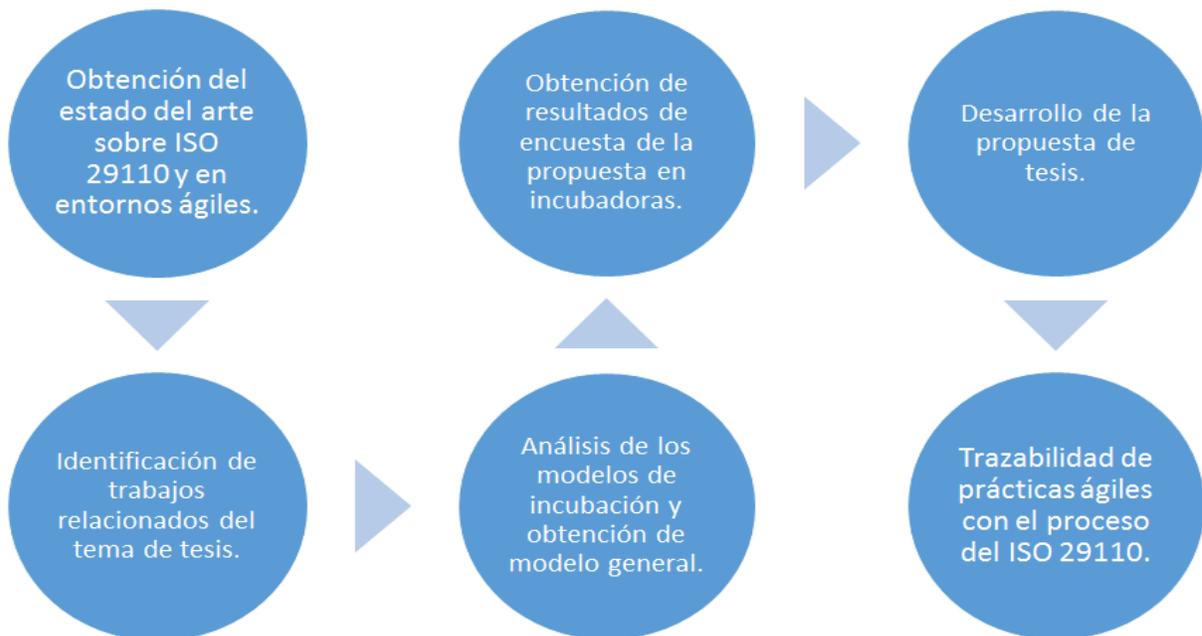


Figura 4. Flujo de trabajo que se siguió para el desarrollo de la guía.

Obtención del estado del arte sobre ISO/IEC 29110 y en entornos ágiles.- El primer paso que se realizó fue la obtención del estado del arte a través de una revisión sistemática enfocada en el ISO/IEC 29110 y prácticas ágiles. Los resultados de esta se muestran en el capítulo 2.

Identificación de trabajos relacionados del tema de tesis: se identificaron trabajos relacionados a la propuesta de tesis como guías, métodos o herramientas. Estos resultados se muestran en el capítulo 2.

Análisis de los procesos de incubación y obtención del proceso genérico: en este paso se analizaron diferentes procesos de incubación obtenidos vía internet y de incubadoras de

Zacatecas a través de entrevistas, para poder aplicar la propuesta de tesis. Los resultados se muestran en el capítulo 4.

Mapeo de prácticas ágiles con el proceso del ISO 29110.- Se desarrolló un mapeo en la que se identificaban las prácticas ágiles de las metodologías ágiles y cómo estas cubrían partes del proceso del ISO 29110. Los resultados se muestran en el capítulo 4.

Desarrollo de la propuesta de tesis.- En este paso se desarrolló la guía analizando las prácticas ágiles con el ISO/IEC 29110 perfil de entrada, dando como resultado la guía para la implementación del ISO/IEC 29110 perfil de entrada. La guía se muestra en el capítulo 4.

Obtención de resultados de encuesta de la propuesta en incubadoras.- Se validó la guía llevando la propuesta a incubadoras y Startups para que analizaran la factibilidad del uso de la guía en empresas en proceso de incubación. Los resultados se muestran en el capítulo 5.

Capítulo 4 Propuesta de tesis

En este capítulo se muestra el desarrollo de la propuesta de la guía de implementación del ISO/IEC 29110 con las prácticas ágiles.

4.1 Introducción de la propuesta

Para el desarrollo de la guía, se definieron 6 actividades que a continuación se describen:

1. **Identificar las prácticas/técnicas ágiles:** en esta actividad se realizó una búsqueda y selección de las prácticas ágiles más utilizadas para la gestión de proyectos y el desarrollo de software.
2. **Realizar un mapeo de prácticas/técnicas ágiles y tareas del ISO/IEC 29110:** en esta actividad se realiza un análisis a nivel de definición entre las prácticas ágiles seleccionadas y las tareas establecidas en el estándar ISO/IEC 29110 en su perfil de entrada, posteriormente se realiza un mapeo entre las prácticas ágiles y las prácticas del estándar.
3. **Definir un proceso de incubación genérico:** en esta actividad se analizan las etapas de los modelos existentes de incubación, obtenidos mediante una búsqueda en internet y mediante entrevistas con incubadoras, para establecer un proceso de incubación genérico.
4. **Realizar mapeo al proceso de incubación genérico:** en esta actividad se realiza un mapeo entre el proceso genérico y los procesos que cubre el estándar ISO/IEC 29110 en su perfil de entrada.
5. **Diseñar de la guía de implementación:** en esta actividad se seleccionan los elementos que conformarán la guía de implementación.
6. **Desarrollar la guía de implementación:** en esta actividad se desarrolla cada una de las fases y elementos que contiene la guía de implementación de acuerdo al diseño establecido.

4.2 Identificar las definiciones de prácticas/técnicas ágiles.

Para comenzar con el desarrollo de la propuesta, se realizó una búsqueda para la identificación de las definiciones de las prácticas ágiles más utilizadas para la gestión de proyectos y desarrollo de software.

Una vez seleccionadas se recopiló información de las prácticas. Posteriormente se escribieron en forma de tabla en Excel donde se registró la siguiente información: nombre de la práctica, descripción y tipo de elemento ágil.

Cabe resaltar que para este trabajo de investigación un elemento ágil puede hacer referencia a una técnica o a una práctica ágil.

En la Figura 5 se muestra un fragmento de la tabla de este análisis. El resto de la tabla se encuentra en:

https://drive.google.com/file/d/1YcoP25PmL1bU2AIa9_6jFSEVHjOBrcyV/view?usp=sharing.

	A	B	C
1	Scrum		Tipo de elemento
2	Sprint Planning	El Sprint planning se realiza el primer día de cada sprint. Se reúne el Scrum Master, Product Owner y el equipo de desarrollo. El Product Owner presenta las características que se tienen que hacer y el equipo determina las tareas necesarias para implementar esas características. Se revisa las estimaciones de trabajo para ver si el equipo tiene el tiempo necesario para completar las características requeridas en el sprint, sino es así, las características de prioridad más baja van atrás del product backlog, hasta que la carga de trabajo de ese sprint es lo suficientemente buena para comenzar.	Práctica
3	Planning Poker	Planning Poker es un juego que alienta a los miembros del equipo a dar sus calificaciones honestas sobre la complejidad de una historia de usuario en relación a otras historias. Las herramientas requeridas para el juego son simples: Puedes usar tu hacerlas o comprar un set de cartas Planning Poker para llevarlo a cabo.	Técnica
4	Daily Stand-Ups	Estas son reuniones cortas que se realizan entre el equipo de trabajo y el Scrum Master diariamente, para ver la situación del proyecto, qué ha hecho cada miembro, ayudar a otros miembros del equipo si lo necesitan. Estas iteraciones no deberían pasar más de 15 minutos y el Scrum Master es el que se encarga de moderar estas reuniones.	Práctica
5	Sprint Review	La revisión del Sprint se lleva a cabo al final del sprint. Su propósito es que el equipo presente las historias de usuario completadas durante el sprint	Práctica
6	Sprint Retrospectives	Aquí se revisa lo que el equipo hizo bien, lo que se hizo mal y lo que se podría mejorar	Práctica
7			
8			
9			
10	EXTREME PROGRAMMING		
11	Planning Game	La planeación de juego captura las características del sistema como historias de usuario, y define las liberaciones del proyecto. Durante la planeación de la iteración, las historias de usuario seleccionadas para la iteración se descomponen en tareas de desarrollo.	Práctica
12	System Metaphor	Cuando haya comunicación con los negocios, ayuda tener un lenguaje ubicuo para que los sistemas complejos puedan ser explicados fácilmente.	Práctica
13	Create Simple Design	Un alto diseño del sistema ocurre al principio y durante una iteración. Un diseño simple confía en el continuo mejoramiento de una mentalidad que aprovecha la refactorización y el desarrollo incremental. Los analistas necesitan asegurarse que el diseño evoluciona junto con los requerimientos.	Práctica
14	On-Site Customer	Durante una iteración, es ideal para el equipo de trabajo siempre tener presente al cliente para asistir en la ayuda que el equipo necesite. Mientras las reuniones físicas son ideales, si no se puede es preferible también hacerlas virtuales que no hacerlas. El punto importante es tener un cliente disponible que pueda proveer respuestas así como los desarrolladores exploran los detalles de una historia de usuario.	Práctica
15	Team Sitting Together	Es importante para todo el equipo comprometerse al proyecto sin estar a grandes distancias uno del otro. Esto mejora las comunicaciones e imparte un sentimiento de camaradería.	Técnica
16	Pair Programming	Programación en pares es una práctica de desarrollo que empareja desarrolladores para que puedan trabajar en un mismo problema juntos. Un par compartirá la misma máquina de desarrollo, y mientras uno codifica, el otro asistirá con las decisiones de diseño y mirará adelante hacia las siguientes características.	Técnica

Figura 5. Definiciones de prácticas ágiles.

4.3 Realizar un mapeo de prácticas/técnicas ágiles y tareas ISO/IEC 29110 perfil de entrada

Para realizar el mapeo fue necesario realizar los siguientes 4 pasos:

1. Selección de elementos.

Para la realización del mapeo de las prácticas/técnicas ágiles con las tareas que se llevan a cabo en el ISO/IEC 29110 perfil de entrada, primero se definieron los elementos de proceso que se iban a analizar:

- a. Actividad (ISO/IEC 29110): la descripción de manera general de lo que se debe llevar a cabo.
- b. Entradas: son los productos de trabajo intermedios requeridos para la realización de las actividades del ISO/IEC 29110.
- c. Salidas: son los productos de trabajo que resultan de llevar a cabo las actividades del ISO/IEC 29110.
- d. Tarea: es una acción mediante la cual se puede llevar a cabo la actividad. El ISO/IEC 29110 sugiere un conjunto de tareas, sin embargo, éstas pueden ser realizadas de alguna otra forma, pero siempre cumpliendo con el objetivo de la actividad.
- e. Práctica (Ágil): son acciones que se llevan a cabo durante el desarrollo de software.
- f. Técnica (Ágil): es un procedimiento en que la práctica se lleva a cabo. En esta se detallan los pasos que se deben seguir para que la práctica se realice.

2. Análisis de contenido de prácticas ágiles y tareas ISO/IEC 29110.

- a. Se revisó el contenido tanto de las prácticas ágiles como de las tareas del ISO/IEC 29110, para establecer la relación entre cada una de ellas. Para el análisis se utilizó una hoja de Excel en la que las columnas se registraron las tareas y en las filas se registraron las prácticas ágiles. Posteriormente la relación se fue indicando dependiendo de la tarea y la práctica utilizando el ID de cada una de ellas. Ejemplo 1-2.2 significa que se mapea la práctica 1 ágil con tarea 2.2 del ISO/IEC 29110.

Para completar este análisis, se realizaron dos plantillas por cada proceso. A manera de ejemplo, en la Figura 6 muestra un fragmento de la tabla en Excel donde las filas representan las prácticas ágiles y las columnas representan las tareas que se llevan a cabo en el estándar ISO/IEC 29110. El resto de la tabla se encuentra

en:

[https://drive.google.com/file/d/1YcoP25PmL1bU2AIa9_6jFSEVHjOBrcyV/vi
ew?usp=sharing](https://drive.google.com/file/d/1YcoP25PmL1bU2AIa9_6jFSEVHjOBrcyV/vi
ew?usp=sharing)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1																								
2																								
3	1.2																							
4	1.6																							
5	1.7																							
6																								
7																								
8																								
9																								
10	1.2																							
11	1.3																							
12	1.6																							
13	1.1																							
14	1.10 x																							

Figura 6. Mapeo entre las prácticas ágiles y las tareas del ISO/IEC 29110.

3. Escribir justificación del mapeo de la práctica ágil y las tareas del ISO/IEC 29110 perfil de entrada.

Para cada una de estas relaciones establecida a través del mapeo, se realizó una justificación basada en el razonamiento llevado a cabo para establecer la relación. La Figura 7 muestra un fragmento de la tabla donde se realizaron las justificaciones establecidas para las relaciones. El resto de la tabla se encuentra en: https://drive.google.com/file/d/1YcoP25PmL1bU2AIa9_6jFSEVHjOBrcyV/view?usp=sharing.

Implementación del Sistema	
Punto	Justificación
1-2.2	Durante la reunión de la planeación del Sprint, se lleva a cabo la selección de Historias de Usuario realizadas por el Product Owner para esa iteración, además de que el equipo también realiza el Product Backlog en esta parte del desarrollo, lo cual equivaldría a la lista de Requisitos de la ISO 29110 siendo ágil.
1-2.3	El Product Owner realiza las Historias de Usuario junto con el Cliente para validar de que son las HU que se van a realizar en el proyecto. Es necesario que al finalizar las HU el cliente firme algún documento que valide y apruebe las HU.
4-2.2	Al final de cada iteración se le presenta al cliente lo que se obtuvo y el cliente puede o no replanificar el Proyecto, asignando o quitando HU.
4-2.3	Al final de cada iteración se lleva a cabo el Sprint Review en el que el cliente revisa junto con el equipo los requisitos que se van a realizar y ver si existen nuevos requisitos. Sin embargo, no se obtiene evidencia alguna de que el cliente está de acuerdo con los nuevos requisitos.
6-2.2	Durante el Planning Game, se realizan y se seleccionan las HU que se realizarán en la iteración, además de que se descomponen estas HU en tareas específicas que los desarrolladores tendrán que realizar.
6-2.3	El cliente al estar en la reunión, valida y aprueba las HU pero al igual que en el Sprint Planning se debe crear un documento donde el cliente firme que está de acuerdo.
8-3.2	El equipo de trabajo entiende los requisitos en esta parte para poder realizar el diseño simple del software.
8-3.3	El equipo realiza un diseño simple (siguiente cualquier táctica que conozca), analizando las HU dadas. El principio de la práctica es hacer sólo el diseño de las HU de la iteración que se llevará a cabo.
9-2.2	Los requisitos en ágil se representan como HU. Estas HU son pedidas al cliente, el cual siempre está disponible para atender dudas o peticiones que el equipo requiera. Sin embargo, la práctica sólo cumple con identificar la fuente y los requisitos, pero no se realiza un documento como ER, puesto que las historias de usuario son representadas en un Product Backlog.
	El cliente al estar siempre disponible para cualquier cosa que el equipo necesite, está de acuerdo con las decisiones acordadas, sin

Figura 7. Ejemplo de justificaciones dadas a cada uno de los puntos del mapeo.

4. Asignar puntuación y grado de cobertura para cada relación establecida con base en su justificación.

Tomando como base la justificación y siguiendo las puntuaciones por valor numéricos establecidos en (Morales-Trujillo, Oktaba, Ventura, & Torres, 2013), se asignó una puntuación a cada mapeo establecido, y con base en estos valores establecido la cobertura, como a continuación se describe:

- T representa una puntuación de 1, que significa que la práctica cumple totalmente la relación.
- M representa una puntuación de 0.7, que significa que la práctica cumple mayormente la relación.

- P representa una puntuación de 0.3, que significa que la práctica cumple parcialmente la relación.
- N representa una puntuación de 0, que significa que la práctica no tiene ninguna relación.

La Figura 8 muestra la asignación de puntuación y de cobertura, para cada relación establecida de acuerdo a su contenido. Además, en la en la Figura 9, se muestra el promedio de cobertura de las prácticas ágiles en relación a las tareas del ISO/IEC 29110. El resto de las tablas se encuentra en: https://drive.google.com/open?id=12bCvslwD7Yx1AlmlH8SbFFi-D2xVGY_y

Implementación del Sistema			
Punto	Justificación	Nivel de cobertura	Puntuación
1-2.2	Durante la reunión de la planeación del Sprint, se lleva a cabo la selección de Historias de Usuario realizadas por el Product Owner para esa iteración, además de que el equipo también realiza el Product Backlog en esta parte del desarrollo, lo cual equivaldría a la lista de Requisitos de la ISO 29110 siendo ágil.	M	0.7
1-2.3	El Product Owner realiza las Historias de Usuario junto con el Cliente para validar de que son las HU que se van a realizar en el proyecto. Es necesario que al finalizar las HU el cliente firme algún documento que valide y apruebe las HU.	M	0.7
4-2.2	Al final de cada iteración se le presenta al cliente lo que se obtuvo y el cliente puede o no replanificar el Proyecto, asignando o quitando HU.	M	0.7
4-2.3	Al final de cada iteración se lleva a cabo el Sprint Review en el que el cliente revisa junto con el equipo los requisitos que se van a realizar y ver si existen nuevos requisitos. Sin embargo, no se obtiene evidencia alguna de que el cliente está de acuerdo con los nuevos requisitos.	M	0.7
6-2.2	Durante el Planning Game, se realizan y se seleccionan las HU que se realizarán en la iteración, además de que se descomponen estas HU en tareas específicas que los desarrolladores tendrán que realizar.	M	0.7
6-2.3	El cliente al estar en la reunión, valida y aprueba las HU pero al igual que en el Sprint Planning se debe crear un documento donde el cliente firme que está de acuerdo.	M	0.7
8-3.2	El equipo de trabajo entiende los requisitos en esta parte para poder realizar el diseño simple del software.	T	1
8-3.3	El equipo realiza un diseño simple (siguiente cualquier táctica que conozca), analizando las HU dadas. El principio de la práctica es hacer sólo el diseño de las HU de la iteración que se llevará a cabo.	M	0.7
9-2.2	Los requisitos en ágil se representan como HU. Estas HU son pedidas al cliente, el cual siempre está disponible para atender dudas o peticiones que el equipo requiera. Sin embargo, la práctica sólo cumple con identificar la fuente y los requisitos, pero no se realiza un documento como ER, puesto que las historias de usuario son representadas en un Product Backlog.	P	0.4
9-2.3	El cliente al estar siempre disponible para cualquier cosa que el equipo necesite, está de acuerdo con las decisiones acordadas, sin embargo, no existe un documento con su firma donde apruebe estas decisiones.	P	0.4
11-4.1	Al realizar Pair Programming, una persona realiza la codificación y la otra ayuda en cuanto a ver las decisiones de diseño que se aplicarán.	T	1
11-4.2	Los desarrolladores en esta parte entienden las características de los componentes de software que van a desarrollar.	T	1

Figura 8. Ejemplo de asignación de puntuación y de cobertura para cada uno de los mapeos y justificaciones establecidas.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		1. Planificación del Proyecto																	
23	19 Project Velocity	1.3			0.7														
24	20 Move people																		
25	21 Dedicated Integration Computer																		
26	22 Class-responsibility-collaboration card																		
27	23 Create a Spike Solution																		
28	24 Don't do extra functionality																		
29	25 Code refactoring																		
30	26 Create Acceptance Tests																		
31	Puntuación																		
32	45.79%	5.1	0	1	1	0	1	0.7	1	0	0	0.4	0	2	0.7	0.7	0.7	1	
33																			

Figura 9. Porcentaje de cobertura de las prácticas ágiles respecto a las tareas del ISO/IEC 29110.

4.4 Definir un proceso de incubación genérico

Para la realización del mapeo de los modelos de incubación, por un lado, se realizó una búsqueda en internet de modelos de incubación que se usan para la creación de Startups de alto rendimiento. Por otro lado, se entrevistaron un conjunto de incubadoras de la ciudad de Zacatecas.

Después de analizar la información de cada modelo de incubación, se identificaron las fases y actividades que se mencionaban en la mayoría de los modelos. Una vez identificadas las actividades que más veces se repetían, se realizó la definición del proceso de incubación genérico. Las fases y actividades de los modelos de incubación propuestos se muestran en la Tabla 13:

Tabla 13. Modelos de incubación analizados.

Modelo	Fases del proceso de incubación			
	Selección	Pre-incubación	Incubación	Post-incubación
Modelo UTZAC	<ul style="list-style-type: none"> -Centro ideológico del proyecto (Validar la idea del negocio). -Modelaje financiero pre-operativo (Modelo de negocios). 	<ul style="list-style-type: none"> - Realizar estrategia experimental (Prototipo validado). - Realizar simulación comercial (Cuasi empresa) 	<ul style="list-style-type: none"> - Formalización de la empresa (Plan de negocios) - Puesta en marcha - Realizar planeación estratégica - Generar utilidades 	<ul style="list-style-type: none"> - Estructura administrativa - Procesos de producción - Expansión comercial
Modelo Bolivia	<ul style="list-style-type: none"> - Identificación de segmentos - Eventos - Screening de emprendedores - Banco de proyectos 	<ul style="list-style-type: none"> - Del Flow - Registro de emprendimientos - Diagnóstico de emprendimientos - Generación de propuesta de valor 	<ul style="list-style-type: none"> - Portafolio de proyectos invertibles - Acuerdos de incubación - Plan de incubación - Validación modelo de negocio (Uso de lean Startup) - Formulación plan de negocio - Demo day 	<ul style="list-style-type: none"> - Dirección Estratégica - Acuerdos de inversión - Gestión de Riesgos - Negociación de términos de salida
Modelo BCS	<ul style="list-style-type: none"> - Difusión - Selección 	<ul style="list-style-type: none"> - Realizar el modelo de negocio - Crear el producto mínimo viable 	<ul style="list-style-type: none"> - Poner en marcha el producto mínimo viable - Crear el plan de negocios 	<ul style="list-style-type: none"> - Petición de recursos mediante capital semilla. - Capacitación constante.

Como se observa en la Tabla 13, los modelos de incubación se componen de un proceso que contiene 4 fases. Por lo que, se decidió que estas fases se tomaran como base para la realización del proceso de incubación genérico, puesto que en cada fase se realizan ciertos productos de trabajo que ayudan a cumplir algunos de los artefactos que son necesarios para cumplir con el estándar ISO/IEC 29110.

Se observa además que la metodología de desarrollo de negocios Lean Startup es mencionada por la mayoría de los modelos de incubación, por lo que se identificó qué las fases de Lean Startup deben estar incluidas como parte de alguna fase del proceso de incubación genérico. El proceso genérico resultante se muestra en la Tabla 14.

Tabla 14. Proceso de incubación genérico.

Proceso de incubación genérico					
Selección	Pre-Incubación	Incubación			Post-Incubación
	Modelo de Negocios	Crear	Medir	Aprender	Empresa formalizada
Selección a ideas para incubar.	Presentar la idea ante la incubadora	Desarrollo de la idea del negocio			Formalización de la empresa, pasando de ser una Startup a una empresa consolidada
Revisión de las ideas para su aprobación.	Crear modelo de negocios	Validar idea de negocio realizando prototipos			Acercar el crecimiento mediante apoyos financieros
	Evaluar el modelo de negocios	Presentar producto a los clientes target			
	Orientar a emprendedores dando talleres o cursos	Se desarrolla el Plan de Negocios			

4.5 Realizar mapeo al proceso de incubación genérico

Se realizó un mapeo del proceso genérico de incubación de las Startups con los dos procesos del estándar ISO/IEC 29110 que son Gestión de Proyecto e Implementación de Software, los resultados obtenidos se muestran en las Tablas 15 y 16.

En primer lugar, la Tabla 15 muestra el mapeo de las actividades relacionadas con la Gestión del Proyecto: planificación, evaluación y control y cierre del proyecto. En segundo lugar, la Tabla 16 muestra el mapeo de las actividades relacionadas con la Implementación de Software: inicio, análisis, identificación de componentes, desarrollo y pruebas.

Tabla 15. Mapeo del proceso de Gestión del Proyecto

ISO/IEC 29110	Proceso de incubación genérico de las Startups				
	Pre-incubación	Incubación			Post-incubación
Proceso de Gestión de Proyectos	Modelo de negocios	Crear	Medir	Aprender	Empresa formalizada
Planificación del Proyecto	Actividades clave, Recursos clave (Canvas Business Model). – Identificar tareas, Identificar y documentar los recursos (Plan de proyecto).				
Ejecución del Plan del Proyecto				Validación de hipótesis, Retroalimentación del prototipo (Lean Startup). – Reuniones con el cliente, Solicitudes de cambios.	
Evaluación y Control del Proyecto		Elegir si pivotar. - Evaluar solicitudes de cambio.			
Cierre del Proyecto					Cuando se finaliza el proceso de incubación de la Startup, se constituye como empresa formal.

Tabla 16. Mapeo del proceso de la Implementación del Software.

ISO/IEC 29110	Proceso de incubación genérico de las Startups				
	Pre-incubación	Incubación			Post-incubación
Proceso de Implementación de Software	Modelo de negocios	Crear	Medir	Aprender	Empresa formalizada
Iniciación de la implementación del software	El equipo conformado sabe lo que se va a desarrollar. – Lograr entendimiento, Establecer entorno de implementación				
Análisis de Requisitos del cliente	Actividades a realizar debe haber una de obtención de requisitos. – Obtener requisitos del cliente (Historias de usuario)			Pivotar. – Cambios en requisitos o nuevos requisitos	
Identificación de componentes de software		Identificación de componentes			
Construcción del software		Construcción de software			
Integración y pruebas del sistema.		Ejecutar pruebas del prototipo y corregir defectos			
Entrega del producto.					Prototipo lanzado al mercado (Lean Startup). – Acta de aceptación

4.6 Diseñar la guía de implementación

Como primer paso de la guía de implementación se desarrolló un glosario de término. Este glosario toma en cuenta elementos de metodologías ágiles como Scrum o XP, así como también la metodología de desarrollo de negocios denominada Lean Startup. La Tabla 17 muestra los términos y definiciones de este glosario.

Tabla 17. Glosario de términos.

Término	Definición
HU	Historia de usuario que se llevará a cabo dentro de un sprint.
PMV	Producto Mínimo Viable
Release Plan	El reléase plan especifica cuáles historias van a ser implementadas para cada liberación del sistema y la fecha estimada para esas liberaciones.
Sprint	La traducción de sprint es iteración. Dentro de esta guía se maneja como iteración al lapso de tiempo que se tiene para completar ciertas historias de usuario.
Sprint Review	La revisión del Sprint se lleva a cabo al final del sprint. Su propósito es que el equipo presente las historias de usuario completadas durante el sprint.
Sprint Retrospective	Aquí se revisa lo que el equipo hizo bien, lo que se hizo mal y lo que se podría mejorar. Además dentro del proceso de incubación se toman acuerdos para hacer con respecto al PMV.
Plan de proyecto	Presenta cómo serán ejecutados los procesos y actividades del proyecto para asegurar su conclusión exitosa, así como la calidad de los productos entregables.
Tarjetas CRC	Es una herramienta para un diseño mínimo de cómo debe estar estructurado el software, usando Clases, Responsabilidades y Colaboradores.
Sprint Backlog	Se refieren a todas las HU que deben ser completadas en el Sprint.

La guía se constituye de fases, las cuales a su vez se dividen en apartados, cada uno con un propósito específico. Los apartados y su propósito se listan a continuación:

1. Proceso de incubación: en este apartado se describe dónde se aplica cada fase en el proceso de incubación genérico. Este apartado sirve como guía para saber cuándo empezar a hacer cada fase de la guía en el momento en que se esté incubando la Startup.
2. Objetivo: el objetivo describe qué se espera realizar en cada fase de la guía.
3. Productos de entrada: se refieren a los productos de trabajo requeridos para trabajar en cada fase.

4. Productos de salida: producto(s) obtenidos al realizar cada fase.
5. Roles: se refieren a los integrantes que llevan a cabo cada fase. Estos varían dependiendo si se sigue la metodología de Scrum o la metodología de XP.
6. ¿Cómo llevarlo a cabo?: en este apartado se describen los pasos que se deben realizar en cada fase mediante el uso de prácticas ágiles, mediante los cuales se generan los productos de salida.
7. Recomendaciones: las recomendaciones se refieren al uso de técnicas/prácticas no ágiles adicionales, que los usuarios de la guía pueden implementar para cumplir con ciertos aspectos que algunas prácticas ágiles no cumplen.

Las fases que componen a la guía son:

1. Inicio del proyecto.
2. Realizar el plan de proyecto.
3. Análisis de requisitos.
4. Identificar componentes de software.
5. Ejecución y evaluación del proyecto.
6. Desarrollo del software.
7. Integración del software.
8. Cierre del proyecto.

La forma para ejecutar la guía de implementación se muestra en la Figura 10. Como se observa en la figura, el ciclo inicia con la fase de Inicio del Proyecto para pasar después a la Realización del Plan de Proyecto. Una vez realizadas estas dos fases, se inicia un ciclo que comprende las fases de Análisis de Requisitos, la Identificación de Componentes de Software, el Desarrollo de Software, y finalizando con la Integración de Software. Este ciclo es controlado mediante la fase de Ejecución y Evaluación del Proyecto. Una vez terminado el producto a desarrollar, se inicia la fase de Cierre del Proyecto.

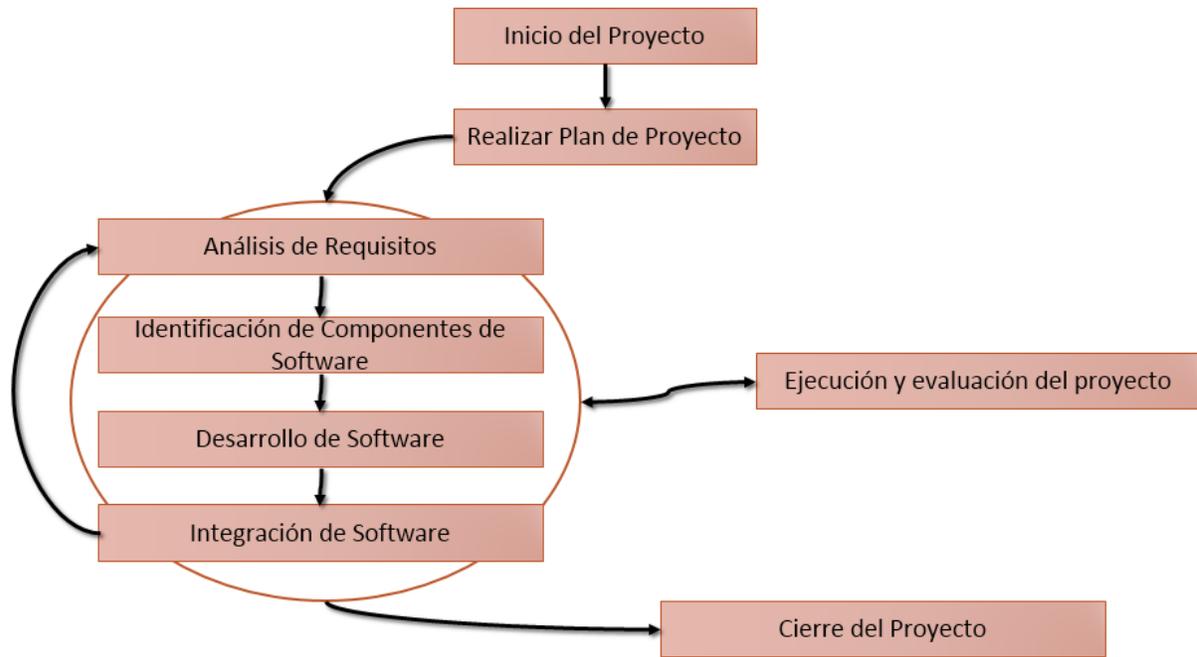


Figura 10. Ciclo de ejecución de la guía.

4.7 Desarrollar la guía de implementación

En esta sección se describe la guía desarrollada como a continuación se indica: la Tabla 18 muestra la parte de la guía desarrollada para el inicio del proyecto; la Tabla 19 muestra la parte de la guía desarrollada para realizar el plan del proyecto; la Tabla 20 muestra la parte de la guía desarrollada para el análisis de requisitos; la Tabla 21 muestra la parte de la guía desarrollada para llevar a cabo la ejecución y evaluación del proyecto; la Tabla 22 muestra la parte de la guía desarrollada para realizar el diseño de software; la Tabla 23 muestra la parte de la guía desarrollada para llevar a cabo la codificación del software, la Tabla 24 muestra la parte de la guía desarrollada para integrar el software; la Tabla 25 muestra la parte de la guía desarrollada para concluir el proyecto.

Tabla 18. Inicio del proyecto.

Inicio del proyecto		
Objetivo: Obtener el entorno donde se va a realizar el proyecto.		Fase del proceso de incubación: Esta fase se lleva a cabo en el proceso de pre-incubación.
Productos de entrada	Productos de salida	Roles
1. Nombre del proyecto.	1. Repositorio del proyecto.	1. Equipo de desarrollo. 2. Líder de equipo (XP) o Scrum master (Scrum).
¿Cómo llevarlo a cabo?		
<ol style="list-style-type: none"> 1. Para iniciar con la implementación del software, primero se debe hacer una reunión entre el equipo de desarrollo y el líder del equipo. 2. Se discute el nombre que se le dará al repositorio y las herramientas que se van a usar. 3. Un miembro del equipo realiza el repositorio del proyecto. Este repositorio puede ser en la nube, a nivel local o de ambas formas. 		
Recomendaciones		
<p>Es recomendable usar un repositorio colaborativo en la nube donde todos los miembros del equipo de desarrollo hagan sus contribuciones y se tenga control de acceso.</p> <p>Algunos ejemplos de repositorios colaborativos en la nube son:</p> <ul style="list-style-type: none"> • GitHub • GitLab 		

Tabla 19. Realizar el plan del proyecto.

Realizar el plan del proyecto		
<p>Objetivo: Desarrollar el plan de proyecto para guiar el desarrollo del PMV a lo largo de la formación de la Startup. En el plan de proyecto se realiza lo siguiente:</p> <ol style="list-style-type: none"> 1. Identificar y estimar duración de las tareas que se van a realizar. 2. Identificar los recursos y material con el que se cuenta. 3. Identificar quiénes van a participar dentro del proyecto. 	<p>Dentro del proceso de incubación: Dentro del proceso de pre-incubación se realiza el modelo de negocios. A partir de ese modelo, se obtiene cierta información para el plan de proyecto.</p>	
Productos de entrada	Productos de salida	Roles
<ol style="list-style-type: none"> 1. Modelo de negocios. 	<ol style="list-style-type: none"> 1. Plan de proyecto. 2. Tablero de tareas. 	<ol style="list-style-type: none"> 1. Líder del proyecto (XP) o Scrum master. 2. Product Owner (Scrum). 3. Equipo de desarrollo. 4. Cliente potencial.
¿Cómo llevarlo a cabo?		
<p>Antes de iniciar con el plan de proyecto, se tuvo que haber validado y aceptado la idea del modelo de negocios. Posteriormente, se tienen que realizar los siguientes pasos:</p> <ol style="list-style-type: none"> 1. Primer Sprint: <ol style="list-style-type: none"> a. Utilizar el Product Backlog (con sus respectivas Historias de Usuario (HU)), resultado de la realización del primer Análisis de Requisitos. b. Realizar un Release Plan con el equipo de desarrollo y el líder del proyecto de las tareas que se van a realizar. Para llevar a cabo el Release Plan se realizan los siguientes pasos: <ol style="list-style-type: none"> i. Establecer el número de Sprints del proyecto. ii. Identificar la duración del Sprint, se puede utilizar la práctica ágil Velocity. iii. Estimar la fecha de terminación cada una de las HU. iv. Seleccionar y priorizar las HU que se van a realizar en cada Sprint. v. Las HU seleccionadas para el Sprint se colocan en el Sprint Backlog. vi. Describir los recursos con los que se cuentan como herramientas de desarrollo, financieros, humanos. 2. Realizar el Tablero de Tareas. El tablero debe estar visible para todos los integrantes del proyecto, puesto se observa el progreso de las HU. Un ejemplo de tablero se muestra en el ANEXO A Figura 20. 		
Recomendaciones		
<ol style="list-style-type: none"> 1. Es importante que dentro del plan de proyecto también se identifiquen los riesgos. Por ejemplo que no haya suficiente recurso o el producto no le sea agradable al usuario. 2. Se recomienda hacer un cronograma de las HU a realizar, asignando al responsable de la HU, el tiempo estimado y fecha de entrega. 3. Para la realización de HU, se pueden tomar ideas a partir de las actividades o tareas clave que se identifican en el modelo de negocios. 4. Si se está haciendo un producto/servicio a la medida, se recomienda realizar un contrato con el cliente al que se le va a desarrollar el software como acción preventiva para minimizar que el cliente no quiera adquirir el producto/servicio. 		

Tabla 20. Análisis de requisitos.

Análisis de requisitos		
Objetivo: Escribir las Historias de Usuario (HU) que se van a desarrollar.	Dentro del proceso de incubación: Esto se lleva a cabo dentro de la subfase crear de la fase Incubación del Proceso de Incubación propuesto.	
Productos de entrada	Productos de salida	Roles
1. Modelo de negocios.	1. Historias de Usuario (HU). 2. Product Backlog.	1. Equipo de desarrollo. 2. Líder de equipo (XP) o Scrum master (Scrum). 3. Product Owner (Scrum).
¿Cómo llevarlo a cabo?		
<ol style="list-style-type: none"> 1. Dentro del proceso de desarrollo es necesario que alguien asuma el rol de Product Owner, ya que será el encargado de saber lo que el cliente busca. Su labor principal es crear las HU que se llevarán a cabo en colaboración con el equipo de desarrollo. 2. Se lleva a cabo el Release Planning en donde se crean las HU con las siguientes características: <ol style="list-style-type: none"> 1. Número de la HU 2. Título de la HU. 3. Estimación realizada por el equipo de desarrollo (Puntos función o Planning Poker). 4. Prioridad dada por el Product Owner. 5. Iteración en que se va a realizar. 6. Estado de la HU (Ejemplo: Producto Backlog, Doing, Done). 3. La descripción de la HU se puede redactar de la siguiente manera: <ol style="list-style-type: none"> 1. Rol de quien ejecuta la acción 2. ¿Qué va a ejecutar? 3. ¿Para qué? 4. En el ANEXO A Figura 21 se encuentra una plantilla de una Historia de Usuario. Además, se tienen que identificar las tareas específicas a realizar dentro de esas HU, para ello se debe: <ol style="list-style-type: none"> 5. Establecer el Sprint en que se va a realizar la HU. 6. Listar las tareas específicas necesarias para llevar a cabo dicha HU. 7. Marcar las tareas específicas que ya se llevaron a cabo. 		
Recomendaciones		
Si existe un cliente en particular, el Product Owner debe tener una excelente comunicación con él, ya que el Product Owner será el encargado de traducir lo que el Cliente necesita.		

Tabla 21. Ejecución y evaluación del proyecto.

Ejecución y Evaluación del proyecto		
Objetivo: El objetivo es mantener mayor control del proyecto, registrando el avance del proyecto y analizando cambios que se llevan a cabo cuando se pivotea el PMV.	Dentro del proceso de incubación: Esta fase se lleva a cabo cuando se esté realizando el PMV, dentro de la fase de Incubación en la subfase aprender.	
Productos de entrada	Productos de salida	Roles
1. Sprint Backlog. 2. Tablero de HU.	1. Burndownchart.	1. Equipo de desarrollo. 2. Líder de equipo (XP) o Scrum master (Scrum). 3. Cliente potencial.
¿Cómo llevarlo a cabo?		
<ol style="list-style-type: none"> 1. Con el Tablero de HU se sigue el progreso del proyecto. A continuación se explica el funcionamiento del Tablero de Historias de Usuario: <ol style="list-style-type: none"> 1. En la sección “Product Backlog” se ponen las HU que faltan por realizar. 2. En la sección “To-do list” se ponen las HU que ese están realizando durante el Sprint. 3. En la sección “Doing” se ponen las HU que se están realizando. 4. En la sección “Tested” se ponen las HU que están siendo probadas e integradas al PMV. 5. En la sección “Done” se ponen las HU ya integradas en el PMV. 2. Durante el transcurso del Sprint, se realizan los Daily meetings que consisten en: <ol style="list-style-type: none"> 1. Realizar una reunión de 15 minutos al principio del día con el equipo de desarrollo. 2. Identificar con el equipo de desarrollo las tareas que faltan realizar. 3. Sólo se deben discutir asuntos relacionado con el proyecto. 3. Al finalizar un ciclo de desarrollo, se realiza la validación del PMV por los clientes potenciales dentro del Sprint Review. 4. A través de los Sprint Review se llevan a cabo reuniones con el cliente. En estas reuniones se realiza lo siguiente: <ol style="list-style-type: none"> 1. Dar a probar al cliente el PMV. 2. Proporcionar un cuestionario para obtener retroalimentación, si se desea, se puede sustituir el cuestionario por una entrevista para recibir retroalimentación. 3. A partir de la retroalimentación, analizar los cambios realizados por el cliente al PMV. Estos cambios deben reflejarse en nuevas HU. 4. De haber nuevas HU se debe ejecutar nuevamente la fase de Análisis de Requisitos. 5. El Product Backlog se vuelve a reorganizar y se vuelven a colocar las HU más importantes para entregar dentro del Sprint. 5. Cuando se muestre el PMV se debe obtener retroalimentación del cliente. Un ejemplo de este cuestionario de retroalimentación se encuentra en el ANEXO A Figura 22. 6. Con esta retroalimentación se toma la decisión de pivotar el PMV. Esto significaría un cambio. 7. Una vez se llevó a cabo el Sprint Review se realiza un nuevo Sprint. 8. Realizar el Burndownchart de la siguiente manera: <ol style="list-style-type: none"> 1. En una hoja de cálculo poner la relación del número de Historias de Usuario completadas y las Historias de Usuario planificadas en los Sprint planificados. 2. Graficar la relación. 9. Un ejemplo de plantilla de Burndownchart se encuentra en el ANEXO A Figura 23. 		
Recomendaciones		
<ol style="list-style-type: none"> 1. Se debe tener evidencia de la fecha en que se terminaron las HU. Esto se puede llevar a cabo tomando una foto al tablero con la fecha del día en que se tomó. 2. Cuando se pruebe el PMV con el cliente, es recomendable que se tenga evidencia de las reuniones. 		

3. Si se realiza un cambio cuando se llevó un contrato, es recomendable tener evidencia de la aceptación del cliente de los cambios a realizar. Esta evidencia se obtiene a través de la firma del cliente.

Tabla 22. Identificación de componentes de software.

Identificación de componentes de software		
<p>Objetivo: El objetivo es realizar un diseño mínimo del software de cómo deberían estar conectados los componentes que se van a realizar del PMV. Este diseño no tiene que ser el diseño completo del PMV. Para llevar a cabo un pequeño diseño se proponen las Tarjetas Clase-Responsabilidades-Colaboradores (Tarjetas CRC) y diagramas C4.</p>	<p>Dentro del proceso de incubación: El diseño de software se lleva a cabo cuando se esté realizando el PMV dentro del proceso de incubación en la subfase crear.</p>	
Productos de entrada	Productos de salida	Roles
1. Historias de Usuario.	1. Tarjetas CRC. 2. Diagrama C4 (Opcional).	1. Equipo de desarrollo.
¿Cómo llevarlo a cabo?		
<ol style="list-style-type: none"> 1. Con las HU identificadas y asignadas, cada miembro del equipo de desarrollo realiza las correspondientes tarjetas CRC. 2. Estas tarjetas CRC son útiles en un desarrollo Orientado a Objetos. A continuación se describe que va en cada sección de una tarjeta CRC: <ol style="list-style-type: none"> 1. En el apartado Clase, se escribe el nombre de la Clase que se va a desarrollar. 2. En el apartado Responsabilidad, se escribe la función que realiza la clase. 3. En el apartado Colaboración se escriben las clases con las que se conecta dicha clase. 3. Un ejemplo de la tarjeta CRC se encuentra en el ANEXO A Tabla 27. 		
Recomendaciones		
<ol style="list-style-type: none"> 1. El diseño tiene que ser mínimo ya que hacer un diseño de todo el sistema provocará mucho esfuerzo y al tener que realizar un PMV este cambia parcial o completamente. 2. También, se pueden usar los diagramas C4 para poder comunicar a toda la empresa la idea del PMV. Al usar estos diagramas se puede llegar al nivel que necesites. 		

Tabla 23. Desarrollo del software.

Desarrollo del software		
Objetivo: Programar el PMV a partir de las HU y el diseño realizado de las tarjetas CRC.	Dentro del proceso de incubación: El PMV se codifica en la fase de crear, cuando se esté realizando el PMV.	
Productos de entrada	Productos de salida	Roles
1. Tarjetas CRC. 2. Historias de Usuario.	1. Componente del software programado.	1. Equipo de desarrollo.
¿Cómo llevarlo a cabo?		
<ol style="list-style-type: none"> 1. El desarrollo de software se lleva a cabo a partir de las tarjetas CRC realizadas en la “Identificación de componentes de software”. También, el desarrollo se puede llevar a cabo por pares, de tal manera que mientras una persona codifica la otra revisa el código y verifica que esté cumpliendo con la HU y las tarjetas CRC. 2. Una vez que se realicen los componentes de software, es necesario realizar las pruebas y corregir los errores hasta que pasen las pruebas. 3. El código que se va creando se va subiendo al repositorio previamente creado al “Inicio del Proyecto”. 		
Recomendaciones		
<p>Una manera de desarrollar el software es realizando primero las pruebas y después desarrollando los componentes. Se puede realizar usando la práctica TDD, que es una forma de desarrollar software. Para realizar esta práctica se codifican primero las pruebas que el software debe pasar y el resultado que se debe de obtener, para después codificar el componente.</p> <p>Si no se tiene mucha experiencia en programación, se puede realizar de la manera tradicional creando primero los componentes y si existe algún defecto corregirlo hasta que la prueba sea exitosa.</p>		

Tabla 24. Integración de software.

Integración de software		
Objetivo: El objetivo es desarrollar la integración del software a partir de la Integración continua.	Dentro del proceso de incubación: De la misma manera que el diseño y el desarrollo, las pruebas se realizan en el proceso de incubación, cuando se esté realizando el PMV en la subfase crear de la fase Incubación.	
Productos de entrada	Productos de salida	Roles
1. Componentes de software programados.	1. Componentes de software integrados.	1. Equipo de desarrollo. 2. Líder de equipo (XP) o Scrum master (Scrum).
¿Cómo llevarlo a cabo?		
<ol style="list-style-type: none"> 1. Para llevar a cabo la integración de una manera ágil, se puede realizar la integración continua a partir de los componentes de software que el equipo de desarrollo realice. Para ello, se pueden utilizar distintas herramientas como Travis CI o Jenkins. 2. Para llevarlo a cabo, se pueden seguir las instrucciones en estos links de las herramientas propuestas: Travis CI: https://travis-ci.org/. Documentación: https://docs.travis-ci.com/ Jenkins: https://jenkins.io/. Documentación: https://jenkins.io/doc/ 		
Recomendaciones		
Si se integra mediante alguna herramienta como Travis CI en la nube, se obtiene un registro automático de las pruebas que fallaron al momento de integrar el PMV. Es importante ir guardando estos reportes de prueba para tener evidencia de las pruebas hechas al PMV.		

Tabla 25. Cierre del proyecto.

Cierre del proyecto		
Objetivo: El objetivo del cierre del proyecto es dar por concluido el ciclo de formación de la Startup, es decir, cuando se encuentra el nicho de mercado y sabe lo que va a realizar como producto/servicio a vender.		Dentro del proceso de incubación: Esta fase se lleva a cabo al final de la fase de Incubación.
Productos de entrada	Productos de salida	Roles
1. PMV finalizado.	1. PMV identificado. 2. Software finalizado para el cliente (Si se realizó un contrato). 3. Firma de conformidad con el cliente (Si se realizó un contrato).	1. Equipo de desarrollo. 2. Cliente potencial.
¿Cómo llevarlo a cabo?		
<ol style="list-style-type: none"> 1. Después de llevar a cabo la última reunión con el cliente, se descubre el PMV final. Es aquí cuando se cierra el ciclo de formación de la Startup. 2. Al finalizar esta fase la incubadora guía a la Startup y se decide su rumbo (Si va a crecer o va a ser absorbida por otra empresa). 3. Si se llegó a hacer un contrato con algún cliente, se debe entregar el producto/servicio. 		
Recomendaciones		
<ol style="list-style-type: none"> 4. Si se llegó a realizar un contrato o acuerdo con el cliente, es recomendable recabar la firma del cliente donde se tenga evidencia que acepta el producto/servicio que se desarrolló. 		

Capítulo 5 Resultados

En este capítulo se presentan los resultados de la validación de la guía. Para la validación de la guía se desarrolló un estudio de caso. Este capítulo describe en primer lugar el desarrollo de estudio de caso, y en segundo lugar se apoya de gráficas para analizar los resultados obtenidos.

5.1 Validación de la guía

Para la validación de la guía propuesta en esta tesis, se desarrolló la guía digitalmente a través de un sitio Web en Sistema de Gestión de Contenidos llamado Wordpress, de esta forma, se pretende facilitar el uso de la guía por las Startups. A continuación se describen las pantallas principales resultantes de la guía.

Como se observa en las Figuras 11, 12 y 13 se muestra: en la Figura 11, la pantalla principal de la guía donde se describen sus fases y la forma de ejecutar la guía; en la Figura 12, la pantalla que contiene el glosario de términos y en la Figura 13, la pantalla con el proceso de incubación genérico establecido.

Guía de implementación del ISO/IEC 29110 en Startups

En este apartado se describe el contenido que tiene cada una de las fases de la guía de implementación.

- Inicio del proyecto
- Realizar el plan del proyecto
- Ejecución y evaluación del proyecto
- Análisis de requisitos
- Identificación de componentes de software
- Desarrollo del software
- Integración de software
- Cierre del proyecto

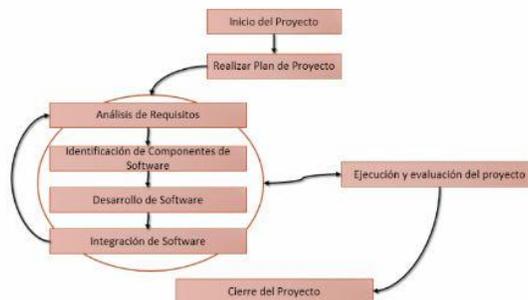


Figura 11. Inicio de la página de la guía.

Glosario de términos

Término	Definición
HU	Historia de usuario que se llevará a cabo dentro de un sprint.
PMV	Producto Mínimo Viable
Release Plan	El reléase plan especifica cuáles historias van a ser implementadas para cada liberación del sistema y la fecha estimada para esas liberaciones.
Sprint	La traducción de sprint es iteración. Dentro de esta guía se maneja como iteración al lapso de tiempo que se tiene para completar ciertas historias de usuario.
Sprint Review	La revisión del Sprint se lleva a cabo al final del sprint. Su propósito es que el equipo presente las historias de usuario completadas durante el sprint.
Sprint Retrospective	Aquí se revisa lo que el equipo hizo bien, lo que se hizo mal y lo que se podría mejorar. Además dentro del proceso de incubación se toman acuerdos para hacer con respecto al PMV.
Plan de proyecto	Presenta cómo serán ejecutados los procesos y actividades del proyecto para asegurar su conclusión exitosa, así como la calidad de los productos entregables.

Figura 12. Glosario de términos en inicio de página de la guía.

El modelo de incubación que se siguió es el siguiente:

Modelo de incubación genérico					
Selección	Pre-Incubación	Incubación			Post-Incubación
	Modelo de Negocios	Crear	Medir	Aprender	Empresa formalizada
Se seleccionan las ideas que se pretenden incubar.	Presentar la idea ante la incubadora.	Desarrollo de la idea del negocio.			Formalización de la empresa, pasando de ser una Startup a una empresa consolidada.
Se ven todas las ideas de negocio dentro del banco de proyectos.	Crear modelo de negocios	Validar idea de negocio realizando prototipos			Acelerar el crecimiento mediante apoyos financieros.
	Evaluar el modelo de negocios	Presentar producto a los clientes target			
	Orientar a emprendedores dando talleres o cursos.	Se desarrolla el Plan de Negocios.			

Figura 13. Proceso de incubación genérico en inicio de la página de la guía.

Cada una de las fases contiene los apartados descritos en el Diseño de la guía (Capítulo 4 de la presente tesis). A manera de ejemplo la Figura 14 muestra de la fase de Inicio, los apartados de proceso de incubación, objetivo, productos de entrada y productos de salida. Finalmente, la Figura 15 muestra los apartados Roles, ¿Cómo llevarlo a cabo? y Recomendaciones. El resto de la guía se encuentra en el siguiente link: agileiso.wordpress.com.

Inicio del proyecto

Proceso de incubación

Esta fase se lleva a cabo en el proceso de pre-incubación.

Objetivo

Obtener el entorno donde se va a realizar el proyecto.

Productos de entrada

- Nombre del proyecto
-

Productos de salida

- Repositorio del proyecto
-

Figura 14. Apartados de Inicio de proyecto parte 1

Roles



Equipo de desarrollo.



Líder de equipo (XP) o Scrum master (Scrum)

¿Cómo llevarlo a cabo?

1. Para iniciar con la implementación del software, primero se debe hacer una reunión entre el equipo de desarrollo y el líder del equipo.
2. Se discute el nombre que se le dará al repositorio y las herramientas que se van a usar.
3. Un miembro del equipo realiza el repositorio del proyecto. Este repositorio puede ser en la nube, a nivel local o de ambas formas.

Recomendaciones

- Es recomendable usar un repositorio colaborativo en la nube donde todos los miembros del equipo de desarrollo hagan sus contribuciones y se tenga control de acceso. Algunos ejemplos de repositorios colaborativos en la nube son:
 - GitLab
 - GitHub

Figura 15. Apartados de Inicio de proyecto parte 2.

5.2 Estudio de caso

Para la validación de la guía se utilizó el método empírico de estudio de caso, dado que se centra en estudiar y comprender cómo se estructura la guía propuesta. Para el estudio de caso se siguieron los siguientes pasos:

1. **Diseño y Planificación del estudio de caso:** Se especifica la forma en que se diseñó y planificó el estudio de caso antes de su ejecución.
2. **Preparación de la recogida de datos:** se refiere a la forma en que se llevó a cabo la recolección de datos y se especifican las preguntas realizadas en el cuestionario.
3. **Recogida de datos:** se describen los pasos llevados a cabo para la ejecución de la recolección de datos.
4. **Análisis de resultados:** se muestran los resultados obtenidos de la recolección de datos.
5. **Reporte de resultados:** se describen las conclusiones del estudio de caso.

5.2.1 Diseño y planificación del estudio de caso

Dentro de la planificación del estudio de caso se describen los siguientes elementos:

- **Objetivo** - ¿Lo que se quiere lograr?
- **Objeto de estudio** - ¿Lo que se estudia?
- **Teoría** - Marco de referencia
- **Preguntas de investigación** - ¿Qué se quiere conocer?
- **Método** – La forma en que se recogen los datos

5.2.1.1 Objetivo

Evaluar la viabilidad de la guía para su uso en las Startups en proceso de incubación.

5.2.1.2 Objeto de estudio

El estudio de caso se centra en 5 Startups que hasta la fecha están en proceso de incubación y 1 incubadora de Startups. Tanto las Startups como la incubadora se encuentran en el estado de Zacatecas, México.

5.2.1.3 Marco de referencia

En los últimos años se han creado cada vez más Startups enfocadas en el desarrollo de software para resolución de problemas de diferentes industrias mediante el desarrollo de soluciones con el uso de tecnología. Estas Startups reciben un ciclo de formación por parte de las incubadoras de empresas, las cuales les brindan herramientas para su crecimiento.

Sin embargo, a pesar de que las incubadoras les brindan gran apoyo para poder crecer, éstas no les brindan prácticas de desarrollo de software que les ayuden a mejorar la calidad de los productos/servicios que lleven a cabo. Esto se puede notar en la carencia de guías que ayuden a las Startups a usar buenas prácticas de desarrollo.

En este contexto se propone como solución crear una guía con buenas prácticas de desarrollo de software enfocadas al estándar ISO/IEC 29110 y prácticas ágiles tomadas de diferentes metodologías de desarrollo de software ágiles. Esta guía brinda a las Startups buenas prácticas tomadas del estándar ISO/IEC 29110 de una manera ágil, además de que se adapta a la metodología para el desarrollo de negocios Lean Startup.

Para ello, se creó un sitio web en Wordpress que brinda prácticas ágiles y recomendaciones que las Startups pueden seguir para obtener mejor control del proyecto. Además de llevar a cabo el ciclo de vida de desarrollo de software siguiendo prácticas ágiles.

5.2.1.4 Preguntas de investigación

Las preguntas de investigación llevadas a cabo son:

- ¿Las Startups usarían una guía que refuerce el proceso de incubación de las Startups a través de buenas prácticas de desarrollo de software?
- ¿La guía digital podría ser usada por las Startups para fortalecer sus buenas prácticas de desarrollo de software?

5.2.1.5 Método

El método para la recolección de datos utilizado es una encuesta web. Esta encuesta fue elaborada en el software de encuestas Google Forms, ya que es una herramienta gratuita y que permite analizar los resultados que se obtienen de manera rápida.

5.2.2 Preparación de la recogida de datos

La recolección de datos se realizó a través de dos fases:

1. Se realizó una presentación del contexto de la tesis, así como la presentación de la guía digital a las Startups y a la incubadora.
2. Se envió el link del formulario en Google Forms a través de sus direcciones de correo y dichas Startups e incubadora respondieron la encuesta.

En la Tabla 26 se muestran las preguntas de la encuesta, de las cuales las primeras cuatro se enfocan en la validación de la guía, y la quinta, se enfoca en cómo podría mejorar la guía. La recolección de respuestas se realizó utilizando la escala Likert con los siguientes niveles de respuesta:

- a) Totalmente en desacuerdo
- b) En desacuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) De acuerdo
- e) Totalmente de acuerdo

Tabla 26. Preguntas de la encuesta.

Número de pregunta	Pregunta
1	¿Considera que la guía puede ayudar a proyectos de Startup enfocados al software?
2	¿Considera que las fases que cubre la guía son suficientes para cubrir un proyecto de Startup?
3	¿Considera que la información es suficiente para alumnos egresados que quieran fundar su propia Startup?
4	¿Considera que si se sigue la guía se obtendrá mayor control del desarrollo del software en el proyecto de Startup?
5	Sugerencias para mejorar la guía

5.2.3 Recogida de datos

Para la recogida de datos se realizaron los siguientes pasos:

1. Se realizaron reuniones con las Startups e incubadora por separado.
2. Se presentó el contexto de la tesis y cómo es que está constituida la guía.
3. Se presentó la guía en formato electrónico a través de Wordpress.
4. Una vez presentada la guía electrónica, los representantes de las Startups e incubadora contestaron la encuesta de la validación de la misma.
5. Se realizó un reporte de los datos obtenidos a partir de las respuestas de la encuesta.

5.2.4 Análisis de resultados

A partir de la presentación de la guía electrónica a las Startups e incubadora se obtuvieron los siguientes resultados.

En la Figura 16 se muestran los datos correspondientes a la primera pregunta relacionada con la validación de la guía, enfocada en conocer si es viable aplicación de la guía en proyectos de desarrollo de software. Como se observa en la figura, la mitad de los encuestados están de acuerdo en que la guía es viable, donde el 33% está muy de acuerdo y el 16.7% tiene una respuesta neutra.

¿Considera que la guía puede ayudar a proyectos de Startup enfocados al software?

6 respuestas

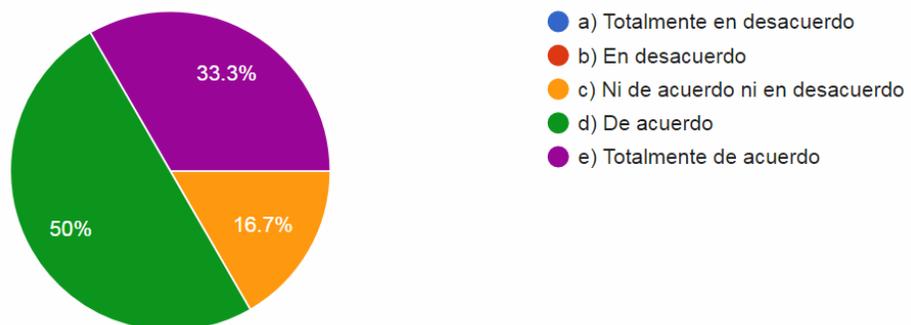


Figura 16. Respuestas de la pregunta 1.

En la Figura 17 se muestran los datos correspondientes a la segunda pregunta relacionada con la validación de la guía, enfocada en conocer si las fases propuestas que cubre la guía son suficientes en proyectos de desarrollo de software. Como se observa en la figura, la mitad está en total acuerdo que son suficientes, mientras que la otra mitad tiene una respuesta neutra.

¿Considera que las fases que cubre la guía son suficientes para cubrir un proyecto de Startup?

6 respuestas

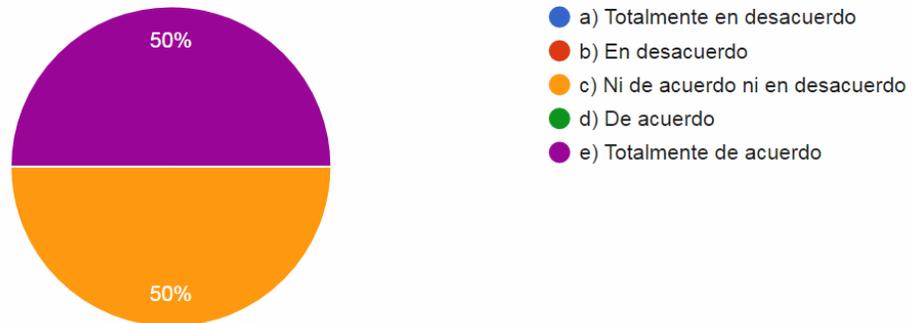


Figura 17. Respuestas de la pregunta 2.

En la Figura 18 se muestran los datos correspondientes a la tercera pregunta relacionada con la validación de la guía, enfocada en conocer si la información de la guía es suficiente para su ejecución en Startups centradas en proyectos de desarrollo de software. Como se observa en la figura, la mayoría de los encuestados tiene una respuesta neutra en relación a que la información es suficiente, sin embargo ninguno de los encuestados estuvo en desacuerdo con dicha pregunta. Además, el 33.4% están de acuerdo o totalmente de acuerdo en que la información es suficiente.

¿Considera que la información es suficiente para alumnos egresados que quieran fundar su propia Startup?

6 respuestas

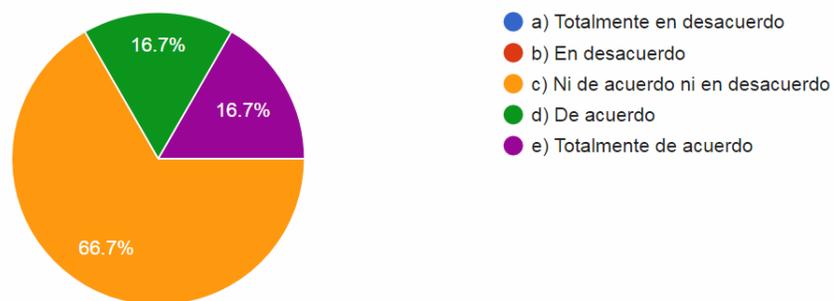


Figura 18. Respuestas de la pregunta 3.

Finalmente, en la Figura 19 se muestran los datos correspondientes a la cuarta pregunta, relacionada con la validación de la guía, enfocada en conocer si el uso de la guía podría mejorar la gestión de los proyectos de software. Como se observa en la figura, la mitad estuvo en acuerdo y la otra mitad en desacuerdo, obteniendo así una respuesta favorable para el uso de la guía.

¿Considera que si se sigue la guía se obtendrá mayor control del desarrollo del software en el proyecto de Startup?

6 respuestas

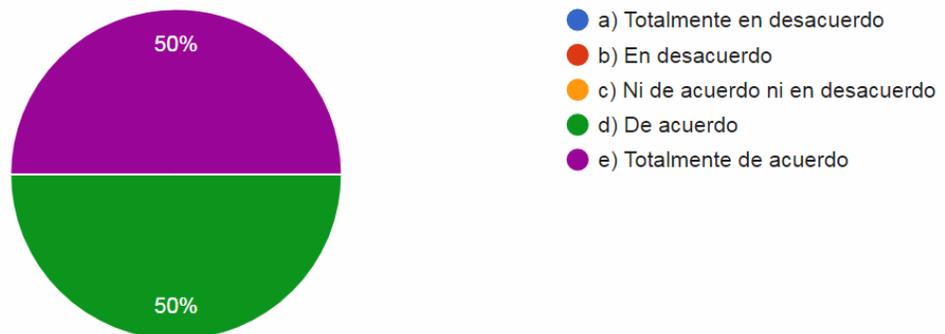


Figura 19. Respuestas de la pregunta 4.

Por último, se obtuvieron las respuestas correspondientes a la quinta pregunta enfocada en pedir sugerencias para mejorar la guía. Como se observa a continuación, sólo hubo 3 comentarios al respecto que son:

- *“La guía debe ser más amigable y digerible aún para un equipo especializado. Lo que pueda ayudar al diálogo con un equipo multidisciplinar.”*
- *“Creo que es importante especificar que es para un Startup de software ya que hay varios tipos de Startup de ahí en más todo excelente.”*
- *“Algunas Startup, no cuentan con el recurso humano suficiente para llevar a cabo todas las tareas necesarias, por lo que podrían incluirse técnicas y/o metodologías para la organización y desarrollo de proyectos con poco recurso humano.”*

5.2.5 Reporte de resultados

Durante la presentación de la guía digital, los participantes se mostraron interesados en conocer como esta guía podría ayudarles a adoptar buenas prácticas de desarrollo de software con un enfoque ágil. Esto refleja la importancia de las metodologías de desarrollo ágiles, las cuales tienen un auge hoy en día debido a que apoyan a las empresas a alcanzar sus metas sin un esfuerzo exhaustivo, pero además el interés por reforzar estas metodologías mediante la adopción de prácticas de formales de ingeniería de software, recolectadas en modelos y estándares de calidad.

La interpretación de los resultados obtenidos es la siguiente:

- La adopción de la guía sería de gran ayuda en las Startups, puesto que, a pesar de que siguen un proceso de formación en las incubadoras de empresas, éstas no se enfocan en proporcionar buenas prácticas de desarrollo de software, lo que conlleva a que las Startups no implementen buenas prácticas.
- Las fases que se tienen en la guía podrían mejorar a pesar de que no hubo una respuesta desfavorable en esta pregunta y se obtuvo un pequeño porcentaje de respuesta favorable. Sin embargo la mayoría de los encuestados no se pronunció por una respuesta favorable o desfavorable lo que podría indicar que es necesario que los encuestados usen la guía en el desarrollo de un producto.
- A pesar de que a la mayoría de los encuestados no se inclinan en alguna respuesta a favor o en contra sobre si la información es suficiente en la guía, no existen respuestas desfavorables en este sentido, lo que indica que la información sí es suficiente para llevar a cabo la implementación de buenas prácticas de desarrollo de software.
- Todas las Startups perciben la guía digital como una herramienta que podría ayudarles a obtener un mayor control en la gestión de la realización de sus productos o servicios. En este sentido se puede deducir que aún existe una carencia de cómo gestionar los proyectos en Startups enfocados en el software siguiendo buenas prácticas.

Capítulo 6 Conclusiones

6.1 Conclusiones

Las Startups hoy en día han logrado tener gran impacto en el país, esto se ve reflejado en un gran apoyo por parte de los estados como Zacatecas. Sin embargo, a pesar de que los gobiernos a nivel ciudad, estado y país ofrecen grandes beneficios a quienes pretendan formar o tengan una Startup enfocada en el desarrollo de software, la mayoría de las veces estas se preocupan más en los productos/servicios que generan, que la formación en el uso de buenas prácticas de desarrollo de software que estas pueden utilizar para incrementar la calidad de sus productos.

En este contexto, los estándares de calidad ayudan a mejorar la calidad de los productos/servicios que las empresas desarrollan, sin embargo aun cuando existen guías de implementación que las empresas como las Startups pueden usar, por lo que, éstas requieren un esfuerzo exhaustivo para la implementación de dichos estándares.

Debido a esta problemática, la propuesta de tesis se centró en desarrollar una guía que ayude a reforzar las buenas prácticas de desarrollo de software de las Startups, siguiendo un enfoque ágil. Con esta guía se pretende que las Startups en proceso de formación implementen o mejoren buenas prácticas de desarrollo de software siguiendo un enfoque ágil y que además estén respaldadas por el estándar ISO/IEC 29110.

En este contexto, la guía propuesta y almacenada en el sitio web en Wordpress, tuvo una buena aceptación para cubrir con la problemática de reforzar el proceso de incubación de las Startups con buenas prácticas de desarrollo de software.

Con respecto a los objetivos específicos alcanzados con el desarrollo de esta tesis se listan a continuación:

- Se obtuvo el estado del arte respecto a la implementación del ISO/IEC 29110 en entornos ágiles.
- Se realizó un mapeo de las prácticas ágiles de Scrum y XP respecto a las tareas establecidas en el estándar ISO/IEC 29110.
- Se identificaron diferentes modelos de incubación encontrados en Web y que existen en la ciudad de Zacatecas, para la realización de un proceso de incubación genérico.
- A partir del proceso de incubación genérico, se realizó un mapeo del proceso respecto al estándar ISO/IEC 29110.

- Se desarrolló una guía con buenas prácticas ágiles enfocadas en el estándar ISO/IEC 29110 y prácticas ágiles que se obtuvieron a partir del mapeo generado.
- Se realizó un sitio web abierto al público para albergar la guía.
- Se validó la guía por parte de 5 Startups y 1 incubadora mediante un estudio de caso, confirmando que la guía podría ayudar a las Startups a implementar buenas prácticas de desarrollo de software.

Con los resultados obtenidos se concluye que la guía puede ayudar a las Startups en proceso de formación, en la implementar buenas prácticas de desarrollo de software enfocadas en el ISO/IEC 29110 y prácticas ágiles, sin que esta guía se vea como un esfuerzo extra que tienen que realizar.

6.2 Trabajos futuros

Los trabajos futuros se describen en los siguientes puntos:

- Mejorar el diseño de la guía en la plataforma Wordpress haciendola más amigable al público.
- Publicación de artículos relacionados con guía y los resultados obtenidos del estudio de caso.

6.2 Logros

Logros académicos :

1. Publicación del artículo del estado del arte en el CISTI 2018:
 Muñoz, M., Mejía, J., & Lagunas, A. (2018). Implementation of the ISO / IEC 29110 standard in agile environments : A systematic literature review. 13th Iberian Conference on Information Systems and Technologies (CISTI), 13, 1–6.
2. Colaboración en el desarrollo de una propuesta de guía para la implementación en desarrollo del estándar ISO/IEC 29110 perfil básico en entornos ágiles.
3. Participación en el proyecto de acompañamiento a las empresas en su proceso de certificación en el estándar ISO/IEC 29110.

4. Colaboración en proyecto de análisis de viabilidad de financiamiento para la obtención de nuevo transporte público en el estado de Zacatecas.

Referencias

- Blankenship, J., Bussa, M., & Millett, S. (2011). *Pro Agile .NET Development with Scrum*.
<https://doi.org/10.1007/978-1-4302-3534-7>
- Centers for Medicare & Medicaid Services. (2008). Selecting a development approach. *Centers for Medicare & Medicaid Services*, 1–10. <https://doi.org/10.1016/j.csi.2016.06.003>
- Dávila, A., & Pessoa, M. (2015). Factors driving the adoption of ISO/IEC 29110: A case study of a small software enterprise. *Proceedings - 2015 41st Latin American Computing Conference, CLEI 2015*.
<https://doi.org/10.1109/CLEI.2015.7360042>
- Dingsøy, T., Dybå, T., & Moe, N. B. (2010). Agile software development: Current research and future directions. *Agile Software Development: Current Research and Future Directions*, (June 2014), 1–238. <https://doi.org/10.1007/978-3-642-12575-1>
- Economía, S. de. (2012). *Tecnologías de la Información (TI)*. Retrieved from <http://www.2006-2012.economia.gob.mx/comunidad-negocios/industria-y-comercio/informacion-sectorial/tecnologias-de-la-informacion-ti>
- Galvan, S., Mora, M., O'Connor, R. V., Acosta, F., & Alvarez, F. (2015). A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process. *Procedia Computer Science*, 64(October 2015), 188–195. <https://doi.org/10.1016/j.procs.2015.08.480>
- García Rodríguez, J. F., Martínez Pérez, L., & Hernández Vasconcelos, M. A. (2018). Las empresas de base tecnológica (start-ups), sinónimo de innovación, competitividad e intangibles. su importancia en el crecimiento y desarrollo económico. el caso de tabasco, méxico. In *DINÁMICA ECONÓMICA Y PROCESOS DE INNOVACIÓN EN EL DESARROLLO REGIONAL* (pp. 967–985).
- Hefley, C. (2008). Kanban Roadmap: How to Get Started in 5 Steps. *Kanban Blog*. Retrieved from <https://leankit.com/learn/kanban/what-is-kanban/%5Cnhttp://leankit.com/learn/kanban/what-is-kanban/%5Cnhttp://kanbanblog.com/explained/>
- Hernandez, L., Munoz, M., Mejia, J., & Pena, A. (2016). Gamification in software engineering teamworks: A systematic literature review. *2016 International Conference on Software Process Improvement (CIMPS)*, 1–8. <https://doi.org/10.1109/CIMPS.2016.7802799>
- Highsmith, J., & Cockburn, A. (2001). Agile Software Development : The Business of Innovation. *Science*, 34(9), 120–123. <https://doi.org/10.1109/2.947100>
- ISO 29110. (2012). *TECHNICAL REPORT ISO / IEC TR Software engineering — Lifecycle profiles for Very Small Entities (VSEs) — Management and engineering guide : Generic profile group : Entry profile* (Vol. 2012).
- Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering - A systematic literature review.

Information and Software Technology, 51(1), 7–15.
<https://doi.org/10.1016/j.infsof.2008.09.009>

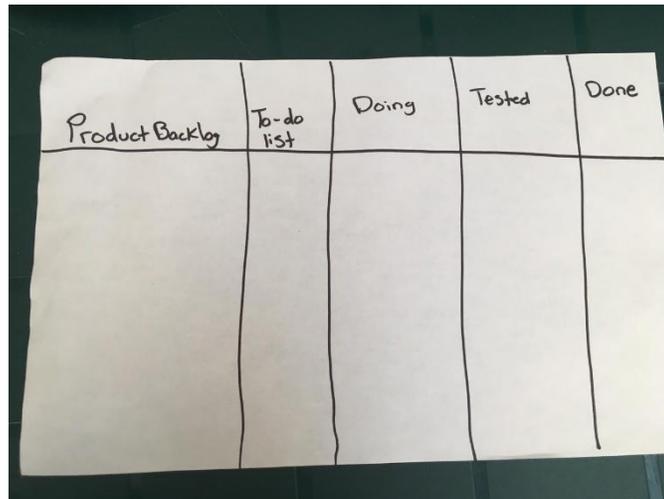
- Laporte, C. Y., & Connor, R. V. O. (2016). The Implementation of ISO/IEC 29110 Software Engineering Standards and Guides in Very Small Entities. *Evaluation of Novel Approaches to Software Engineering*, 703, 162–179. <https://doi.org/10.1007/978-3-319-56390-9>
- Laporte, C. Y., Tremblay, N., Menaceur, J., & Poliquin, D. (2016). Implementing the New ISO/IEC 29110 Systems Engineering Process Standard in a Small Public Transportation Company. *Systems, Software and Services Process Improvement*, 633, 15–29. <https://doi.org/10.1007/978-3-319-44817-6>
- Laporte, C. Y., Tremblay, N., Menaceur, J., Poliquin, D., & Houde, R. (2017). Systems engineering and management processes for small organizations with ISO/IEC 29110: An implementation in a small public transportation company. *11th Annual IEEE International Systems Conference, SysCon 2017 - Proceedings*, 8. <https://doi.org/10.1109/SYSCON.2017.7934718>
- Martinez, J., Mejia, J., & Munoz, M. (2016). Security analysis of the Internet of Things: A systematic literature review. *2016 International Conference on Software Process Improvement (CIMPS)*, 1–6. <https://doi.org/10.1109/CIMPS.2016.7802809>
- Mejía, J., Íñiguez, F., & Muñoz, M. (2017). Data Analysis for Software Process Improvement: A Systematic Literature Review. *Advances in Intelligent Systems and Computing*, 569, 257–266. https://doi.org/10.1007/978-3-319-56535-4_5
- Morales-Trujillo, M. E., Oktaba, H., Ventura, T., & Torres, R. (2013). From MoProSoft Level 2 to ISO/IEC 29110 Basic Profile: Bridging the Gap. *CLEI Electronic Journal*, 16(1), 3. Retrieved from http://www.scielo.edu.uy/scielo.php?script=sci_arttext&pid=S0717-50002013000100003&nrm=iso
- O'Connor, R. (2014). Early Stage Adoption of ISO/IEC 29110 Software Project Management Practices: A Case Study. *Software Process Improvement and Capability Determination SE - 20*, 477, 226–237. https://doi.org/10.1007/978-3-319-13036-1_20
- Pasini, A. C., Esponda, S., Boracchia, M., & Pesado, P. (2013). Q-Scrum: una fusión de Scrum y el estándar ISO/IEC 29110. *XVIII Congreso Argentino de Ciencias de La Computación*, 898–909. Retrieved from <http://hdl.handle.net/10915/32421>
- Paucar, L. G., Laporte, C. Y., Arteaga, J., & Bruggmann, M. (2015). Implementation and Certification of ISO/IEC 29110 in an IT Startup in Peru. Retrieved from https://www.researchgate.net/publication/273455091_Implementation_and_Certification_of_ISOIEC_29110_in_an_IT_Startup_in_Peru
- Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*.
- Ron Jeffries, Ann Anderson, C. H. (2000). *Extreme Programming Installed*. (A.-W. Professional, Ed.) (1st ed.).

Sanchez-Gordon, M.-L., Rory V. O'Connor, & Ricardo Colomo-Palacios. (2015). Evaluating VSEs Viewpoint and Sentiment Towards the ISO/IEC 29110 Standard: A Two Country Grounded Theory Study. *Communications in Computer and Information Science*, 526, 114–127.
<https://doi.org/10.1007/978-3-319-19860-6>

Satpathy, T. (2016). *Cuerpo de conocimiento de SCRUM*.

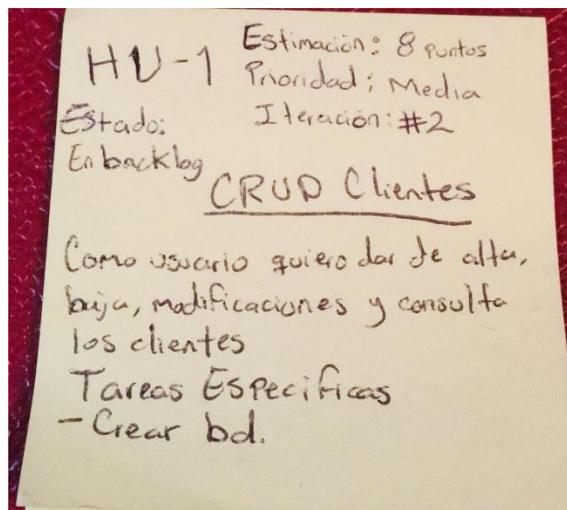
Anexos

ANEXO A



Product Backlog	To-do list	Doing	Tested	Done

Figura 20. Plantilla de Tablero de Tareas.



HU-1 Estimación: 8 puntos
 Prioridad: Media
Estado: Iteración: #2
En backlog

CRUD Clientes

Como usuario quiero dar de alta, baja, modificaciones y consulta los clientes

Tareas Específicas
- Crear bd.

Figura 21. Plantilla de Historia de Usuario.

Retroalimentación de clientes con PMV

Las respuestas se contestan con una escala del 1 al 10 indicando que tan de acuerdo está. La escala para contestar se presenta a continuación.

- 1 – Nada de acuerdo.
- 2 – Poco de acuerdo.
- 3 – Más o menos de acuerdo.
- 4 – Muy de acuerdo.
- 5 – Completamente de acuerdo.

Preguntas

- 1.- ¿El producto tiene algo innovador?
- 2.- ¿El producto es de mi agrado?
- 3.- Usaría el producto
- 4.- Pagaría por el producto
- 5.- ¿Qué me gustaría cambiar del producto/servicio?

Figura 22. Ejemplo de cuestionario de retroalimentación.

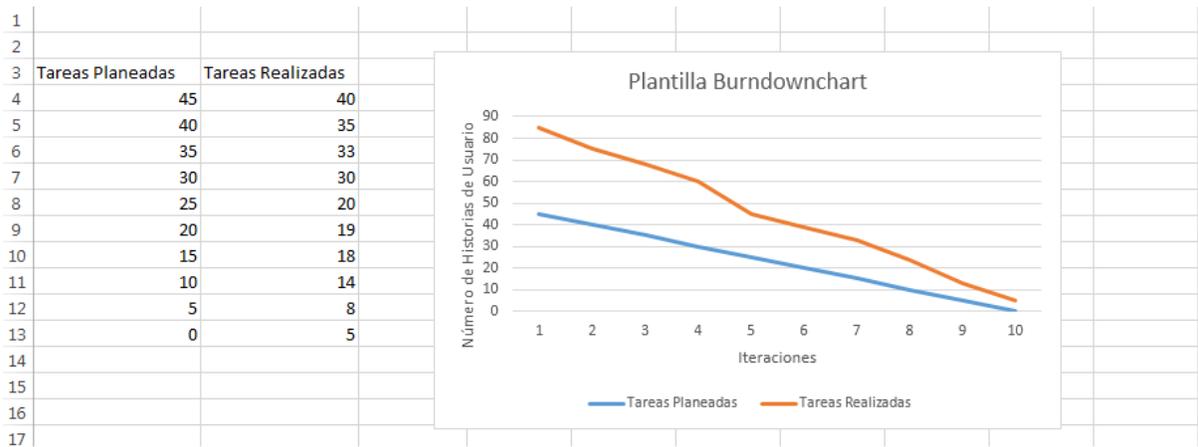


Figura 23. Ejemplo de Burndownchart.

Tabla 27. Ejemplo de tarjeta CRC.

Nombre de la Clase: Imagen	
Responsabilidad:	Colaborador:
<ol style="list-style-type: none"> 1. Cargar una imagen en la plataforma web. 2. Ampliar la imagen 3. Guardar la imagen en la base de datos 	<p>ImageIO</p> <p>EscalImagen</p>