



**Fortalecimiento del proceso en la construcción de software para Banco
Internacional, utilizando Scrum y Cobit**

Franco Benalcázar, Vinicio Javier

Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Maestría en Gerencia de Sistemas

Trabajo de titulación, previo a la obtención del título de Magíster en Gerencia de
Sistemas

Msc. Solis Acosta, Edgar Fernando

5 de agosto del 2020



Document Information

Analyzed document Tesis BI_Vinicio Franco - Capl... CapIV tur.docx (D77264678)
Submitted 7/28/2020 9:47:00 PM
Submitted by SOLIS ACOSTA EDGAR FERNANDO
Submitter email efsolis@espe.edu.ec
Similarity 2%
Analysis address efsolis.espe@analysis.arkund.com

Sources included in the report

SA	JORGE WASHINGTON CORDOVA RAMOS.docx Document JORGE WASHINGTON CORDOVA RAMOS.docx (D53932176)	 1
W	URL: https://e-archivo.uc3m.es/bitstream/handle/10016/26132/TFG_Ana_Retamosa_Santos.pdf Fetched: 10/6/2019 6:46:13 AM	 1
W	URL: https://core.ac.uk/download/pdf/223231984.pdf Fetched: 5/26/2020 1:51:46 PM	 7
W	URL: https://repository.ucc.edu.co/bitstream/20.500.12494/6184/4/2018-RodriguezGuzman... Fetched: 6/5/2020 7:59:47 AM	 1
SA	Tesis.v4.pdf Document Tesis.v4.pdf (D27599736)	 2
W	URL: https://es.slideshare.net/yeltsintorres18/metodologias-para-el-desarrollo-del-software Fetched: 11/22/2019 6:19:37 PM	 1
SA	Universidad de las Fuerzas Armadas ESPE / PAOLA TESIS (V14)(v2).docx Document PAOLA TESIS (V14)(v2).docx (D21514221) Submitted by: caruiz@espe.edu.ec Receiver: caruiz.espe@analysis.arkund.com	 2

Firma:

Solis Acosta, Edgar Fernando

DIRECTOR



VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA
TECNOLÓGICA

CENTRO DE POSGRADOS

CERTIFICACIÓN

Certifico que el trabajo de titulación, "**Fortalecimiento del proceso en la construcción de software para Banco Internacional, utilizando Scrum y Cobit**" fue realizado por el señor **Franco Benalcázar, Vinicio Javier** el mismo que ha sido revisado y analizado en su totalidad, por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolqui, 05 de agosto del 2020

Firma:

Solís Acosta, Edgar Fernando

Director

C.C.: 1803005071



VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE
TECNOLOGÍA

CENTRO DE POSGRADOS

RESPONSABILIDAD DE AUTORÍA

Yo **Franco Benalcázar, Vinicio Javier**, con cédula de ciudadanía n° 1002587432, declaro que el contenido, ideas y criterios del trabajo de titulación: **Fortalecimiento del proceso en la construcción de software para Banco Internacional, utilizando Scrum y Cobit** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 05 de agosto de 2020

Firma:

A handwritten signature in blue ink, which appears to be 'Franco Benalcázar', is written over a horizontal dotted line. The signature is stylized and cursive.

Franco Benalcázar, Vinicio Javier

C.C.: 1002587432



VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE
TECNOLOGÍA

CENTRO DE POSGRADOS

AUTORIZACIÓN DE PUBLICACIÓN

Yo **Franco Benalcázar, Vinicio Javier** autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Fortalecimiento del proceso en la construcción de software para Banco Internacional, utilizando Scrum y Cobit** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangoquí, 05 de agosto de 2020

Firma:

A handwritten signature in blue ink is written over a horizontal dashed line. The signature is stylized and appears to be 'Franco Benalcázar, Vinicio Javier'.

Franco Benalcázar, Vinicio Javier

C.C.: 1002587432

DEDICATORIA

El presente trabajo esta dedico a todas las persona que confiaron y mí de alguna manera contribuyeron con mis ganas y mi esfuerzo por cada día ser mejor y así alcanzar mis objetivos.

Vinicio Franco

AGRADECIMIENTO

A Dios

Por haberme permitido llegar a cumplir mis objetivos.

Vinicio Franco

Tabla de contenidos

Índice de tablas	10
Índice de figuras	11
Acrónimos y abreviaturas.....	12
Resumen	13
Abstract.....	14
Capítulo I: Introducción	15
Antecedentes.....	15
Planteamiento del Problema	15
Objetivos.....	17
Objetivo General	17
Objetivos Específicos	17
Justificación, Importancia y Alcance.....	17
Capítulo II: Fundamentación Teórica	19
Marco Teórico.....	19
Metodología de Desarrollo Tradicional	19
Metodología de Desarrollo MSF	21
Metodología de Desarrollo Ágil.....	23
Comparativo Metodologías Tradicionales versus Agiles.....	25
Marco de Referencia Cobit 2019.....	27
Dominio: Construir, Adquirir e Implementar (BAI)	31
Scrum.....	35
Owasp.....	38
Capítulo III: Metodología Aplicada al Caso de Estudio	42
Fase I: Situación Actual	42
Fase II: Identificar Oportunidades de Mejora.....	65

Fase III: Fortalecimiento del Modelo Actual.....	75
Alcance General.....	75
Definición de Objetivos de Gestión basados en COBIT 2019.....	75
Despliegue de Procesos basados en SCRUM.....	84
Presentación del Modelo Integral Propuesto	93
Capítulo IV: Conclusiones y Recomendaciones	124
Conclusiones	124
Recomendaciones	125
Bibliografía.....	127

Índice de tablas

Tabla 1 <i>Comparativo metodologías tradicionales y ágiles en el desarrollo de software.</i>	25
Tabla 2 <i>Comparación de metodologías.</i>	26
Tabla 3 <i>Fases y procesos fundamentales de Scrum.</i>	36
Tabla 4 <i>Ventajas y desventajas metodología Owasp.</i>	39
Tabla 5 <i>Roles y funciones de los equipos de trabajo por Línea de Servicio.</i>	44
Tabla 6 <i>Iteraciones para todo desarrollo de software.</i>	45
Tabla 7 <i>Relación entre manejo de proyectos, procesos y CVD.</i>	46
Tabla 8 <i>Roles y funciones ciclo de vida del desarrollo de software.</i>	48
Tabla 9 <i>Resumen fases del CVD con objetivos Cobit.</i>	83
Tabla 10 <i>Resumen fases del CVD con procesos fundamentales Scrum.</i>	92
Tabla 11 <i>Roles y funciones ciclo de vida de desarrollo de software.</i>	99
Tabla 12 <i>Resumen del modelo integral propuesto.</i>	121

Índice de figuras

Figura 1 <i>Desarrollo de software en cascada</i>	21
Figura 2 <i>Modelo de proceso de desarrollo de aplicaciones MSF</i>	23
Figura 3 <i>Desarrollo de software ágil</i>	24
Figura 4 <i>Triangulo de la gestión de proyectos tradicional y ágil</i>	27
Figura 5 <i>Objetivos de gobierno y gestión dentro de Cobit 2019</i>	31
Figura 6 <i>Objetivos de gestión del dominio BAI</i>	32
Figura 7 <i>Flujo de Scrum para un sprint</i>	37
Figura 8 <i>Ataque genérico</i>	40
Figura 9 <i>Diez factores de riesgo</i>	41
Figura 10 <i>Estructura de área de desarrollo distribuida por Líneas de Servicio</i>	43
Figura 11 <i>Fases ciclo de vida desarrollo de software</i>	50
Figura 12 <i>Fases propuestas para ciclo de vida desarrollo de software</i>	103
Figura 13 <i>Diagrama de flujo fase de Entendimiento</i>	105
Figura 14 <i>Diagrama de flujo fase Análisis funcional</i>	108
Figura 15 <i>Diagrama de flujo fase Análisis diseño técnico</i>	110
Figura 16 <i>Diagrama de flujo fase Construcción</i>	113
Figura 17 <i>Diagrama de flujo fase Certificación</i>	115
Figura 18 <i>Diagrama de flujo fase Liberación</i>	117
Figura 19 <i>Diagrama de flujo fase Postproducción</i>	120

Acrónimos y abreviaturas

AC: Aseguramiento de la Calidad.

BIA: Construir, Adquirir e Implementar.

CVD: Ciclo de Vida de Desarrollo.

COBIT: Control Objectives for Information and related Technology.

DER: Documento de evaluación de requerimiento.

ISACA: Information Systems Audit and Control Association

I&T: Información y Tecnología.

MSF: Microsoft Solution Framework.

OWASP: Open Web Application Security Project.

SGB: Scrum Guidance Body.

SSII: Sociétés de Services en Ingénierie informatique

TI: Tecnologías de Información.

XP: Extreme Programming.

YAGNI: You Aren't Gonna Need It

Resumen

El proceso de construcción software en Banco Internacional utiliza una metodología de trabajo basada principalmente en un modelo en cascada, la dependencia existente entre sus etapas genera tiempos altos en la entrega de producto al negocio, uno de sus principales problemas es no tener claramente definidas las responsabilidades de los participantes en cada una de las etapas, lo que demora la toma de decisiones funcionales generando reprocesos por entendimientos ambiguos de las necesidades del negocio, impactando en los tiempos de construcción del software.

El presente trabajo busca evaluar y fortalecer el proceso en la construcción del software para Banco Internacional obteniendo productos que generen valor al negocio de manera temprana mediante liberaciones a producción en espacios cortos de tiempo, para este propósito se utilizará COBIT como marco de referencia integrador de procesos del negocio con las tecnologías de la información y SCRUM como marco de trabajo ágil de gestión en la construcción del software, lo que permitirá incrementar la agilidad, eficiencia al entregar producto al negocio.

Palabras clave:

- **METODOLOGÍA DE DESARROLLO**
- **COBIT**
- **SCRUM**

Abstract

The software construction process at Banco Internacional uses a work methodology based mainly on a cascade model, the dependence between its stages generates high times in the delivery of product to the business, one of its main problems is not having clearly defined responsibilities of the participants in each of the stages, which delays the taking of functional decisions generating reprocesses for ambiguous understandings of the business needs, impacting on the times of software construction.

The present work seeks to evaluate and strengthen the process in the construction of the software for the International Bank, obtaining products that generate value to the business in an early way by means of releases to production in short spaces of time, for this purpose COBIT will be used as a framework for integrating processes of the business with information technologies and SCRUM as an agile management framework in software construction, which will increase agility, efficiency when delivering product to the business.

Key words:

- **DEVELOPMENT METHODOLOGY**
- **COBIT**
- **SCRUM**

Capítulo I: Introducción

Antecedentes

En la actualidad existen varios marcos de referencia (frameworks) “utilizados para estructurar, planear y controlar el proceso de construcción de software”. (Gomez, 2017), en tal sentido el considerar uno u otro frameworks es trascendental para la ejecución y éxito de los proyectos de software, sin embargo estos frameworks no necesariamente se adoptan y adaptan a las necesidades de las organizaciones

El proceso de construcción software en Banco Internacional utiliza una metodología de trabajo basada principalmente en las mejores prácticas del marco referencial MSF (Microsoft Solution Framework) el cual es gobernado por un modelo cascada sin responsabilidades claras, la dependencia existente entre su etapas genera tiempos altos de construcción, demorando la entrega de productos al negocio y en algunos casos con funcionalidades no acorde a las necesidades.

Planteamiento del Problema

En la actualidad Banco Internacional cuenta con un procesos de construcción de software poco ágil el cual demora la entrega de producto al negocio, basado principalmente en el marco referencial MSF (Microsoft Solution Framework) y con pequeñas variaciones en la fase de desarrollo que les permite realizar liberaciones parciales de funcionalidad definidas en orden de prioridad dada por el negocio.

Dentro de esta metodología de trabajo establecida en Banco Internacional las responsabilidades de cada etapa no están claramente definida, por lo que el área de Desarrollo termina siendo responsable directo en la ejecución y toma de definiciones para la implementación de proyectos lo que genera insatisfacción por parte del negocio en las soluciones implementadas en producción.

Causas:

- No existe un Marco de Gobierno que garantice una adecuada gestión de Proyecto de Software.
- Metodología de trabajo para desarrollo de software con modelo principal basado en cascada, lo cual genera dependencia entre fases.
- No existe una socialización adecuada en la organización de la metodología de trabajo.

Efectos:

- Las áreas usuarias y áreas de TI trabajan por silos en la implementación de Proyectos.
- No existen definiciones claras de las responsabilidades entre el Negocio y las áreas de TI para la ejecución de Proyectos.
- El área de Desarrollo es un actor determinante en la ejecución del proyecto, el cual en algunos casos asume responsabilidades que no son de su competencia.
- Impacto en los tiempos de construcción de software y demora en la entrega de producto al negocio, por modelo de trabajo.
- Reprocesos en la construcción de software por entendimiento ambiguo del área de Desarrollo de las necesidades del negocio.
- Poca participación activa de los usuarios funcionales responsables de los proyectos en el proceso de construcción de software.
- Descontento por parte del negocio sobre el alcance de los productos entregados.

Objetivos

Objetivo General

- Fortalecer el proceso de construcción de software para Banco Internacional utilizando COBIT como marco de referencia (Frameworks) integrador de procesos del Negocio con las Tecnologías de la Información y SCRUM como marco de trabajo ágil para la implementación de Proyectos/Mantenimientos, lo que nos permitirá incrementar la agilidad, y eficiencia al entregar los productos al negocio, mejorando de esta manera su productividad.

Objetivos Específicos

- Realizar un diagnóstico y análisis de la situación actual del proceso de construcción de software para Banco Internacional, mediante reuniones de trabajo que permitan entender de manera detallada las actividades roles y funciones de los responsables del proceso con la finalidad identificar oportunidades de mejoras.
- Fortalecer la metodología para la construcción de software en Banco Internacional, mediante la utilización de marcos de Referencia (COBIT y SCRUM).

Justificación, Importancia y Alcance

Actualmente el proceso de construcción de software tiene una alta demanda debido a la competitividad de los Negocios, los mismos que requieren productos y servicios tecnológicos que se adapten a las necesidades de sus clientes por lo tanto exigen rapidez, calidad y reducción de costos en su implementación, para asumir estos retos es necesario tener agilidad y flexibilidad durante el proceso de construcción de software, sin embargo el modelo casada que utiliza Banco Internacional para este

efecto limita la creación y entrega de los productos desarrollados de manera temprana al negocio.

Por lo anteriormente mencionado y sus efectos es necesario fortalecer y acelerar el proceso en la construcción del software para Banco Internacional, definiendo claramente roles, funciones y responsabilidades de los actores que intervienen en el mismo, para lo cual se utilizará COBIT como marco de Referencia (frameworks) integrador de procesos del Negocio con las Tecnologías de la Información y SCRUM como marco de trabajo ágil en el proceso de gestión de la construcción de software, lo que permitirá incrementar la agilidad, eficiencia al entregar productos al negocio y de esta manera mejorar la productividad.

Capítulo II: Fundamentación Teórica

Marco Teórico

Metodología de Desarrollo Tradicional

Desarrollo en cascada, es una metodología tradicional que organiza y ordena las etapas del proceso de desarrollo de software, de esta manera cada inicio de fase depende de la finalización de la anterior, al final de cada etapa en el proceso existen hitos o entregables que permiten determinar si el proyecto está listo para avanzar a la siguiente fase (Lopez Provencio, 2015).

El desarrollo en cascada, en relación a la seguridad de software, recomienda realizar un análisis de riesgos antes de iniciar la fase de diseño, donde podrían empezar a visualizar cuales pueden ser los puntos vulnerables del proyecto, no es hasta la fase de implementación que se establecen de manera clara las vulnerables, sin embargo en la fase de verificación es donde se comprueba la seguridad, ya en este punto el desarrollo de software se ha dado por finalizado y cualquier problema de seguridad será solventado con modificaciones sobre el software finalizado (Lopez Provencio, 2015).

A continuación las Fases que tiene esta metodología (Lopez Provencio, 2015):

Requisitos de Software: Fase en la cual se levanta y se analiza las necesidades que debe cumplir el software, como parte de los entregables está el documento de especificaciones funcionales, el mismo que es consensuado con los usuarios finales y define el alcance del desarrollo del software, con lo cual no se pudiera requerir nuevos resultados a mitad del proceso de construcción del software (Toro & Peláez, 2018).

Diseño de Software: Fase en la cual se entiende las necesidades de los usuarios finales del software, desde el punto de vista funcional y técnico, como parte de

los entregables están los documentos de diseño funcional, técnico del software, que contiene la descripción de la estructura relacional global del sistema, la especificación de lo que debe hacer cada una de sus partes, algoritmos necesarios y arquitectura para el cumplimiento de los requerimientos del usuario, insumos necesaria para la fase de construcción (Toro & Peláez, 2018).

Implementación de Software: Fase en la cual se construye y se prueba las necesidades de los usuarios finales del software, mediante la implementación de código fuente, ensamblando componentes del software, construyendo y realizando prueba para garantizar que cumpla con los requisitos desde el punto de vista funcional y técnico (Toro & Peláez, 2018).

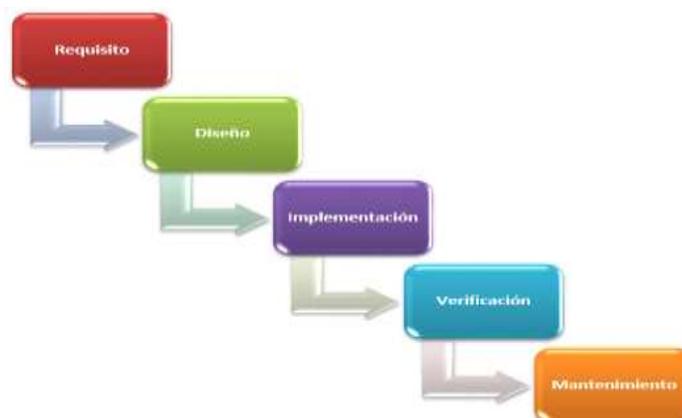
Verificación del Software: Fase en la cual el usuario final certifica en el software las necesidades levantadas en la fase de requerimientos, en esta instancia del proceso ya se realizaron exhaustivas pruebas para comprobar la calidad del software y que este no falle, lo que permite ratificar que las necesidades solicitadas en su momento por la organización fueron cumplidas de manera satisfactoria en su totalidad (Toro & Peláez, 2018).

Mantenimiento del Software: Fase en la cual se realiza la transferencia del software desarrollado para inicia la operatividad del software por parte de los usuarios, se destina un alto porcentaje de recursos en su mantenimiento ya que puede ser que no cumpla con todas las expectativas (Toro & Peláez, 2018).

La figura 1, a continuación indica las fases de la metodología tradicional (modelo cascada) que es utilizada en la actualidad por algunas organizaciones públicas y privadas en el proceso de construcción de Software.

Figura 1

Desarrollo de software en cascada



Nota. Fuente (Lopez Provencio, 2015)

Metodología de Desarrollo MSF

Microsoft® Solution Framework (MSF) es un marco de trabajo flexible y abierto que puede ser adaptado para ajustarse a los requerimientos y necesidades particulares de una organización (Cacuango Díaz, 2017).

El Modelo de Proceso de Desarrollo de MSF describe un ciclo de vida que puede ser usado para desarrollar software de manera exitosa en menor tiempo y de calidad, minimizando riesgos y optimizando recurso humano, estableciendo el orden en el cual se deben realizar las actividades (Cacuango Díaz, 2017).

Microsoft® Solution Framework (MSF) contempla tres modelos dentro de su gestión (Gualotuña Lema , 2016):

Modelo de Equipo: Se caracteriza por organizar el equipo de trabajo de un proyecto y asegurar que los objetivos del mismo se cumplan mediante la vinculación de roles (Gualotuña Lema , 2016).

Modelo de Proceso: Se caracteriza por estructurar y orientar al proyecto a través de un ciclo de vida basado en hitos, los cuales se generan de manera iterativa y flexible. Describiendo las fases, actividades y entregables del proyecto y su relación con el modelo del equipo. Ayudando a mejorar el control del proyecto, minimizar riesgo, mejorar la calidad y reducir el tiempo de entrega (Gualotuña Lema , 2016).

Modelo Gestión de Riesgos: Se caracteriza por suministrar de manera estructurada y dinámica la gestión de riesgos en el proyecto, mediante la toma de decisiones y acciones proactivas para evaluar de forma continua los riesgos y determinar qué riesgos son importantes para implementar estrategias de mitigación. El uso de este modelo y los principios y prácticas que le sirven de base ayudarán a los equipos en centrarse en lo que es más importante, tomar las decisiones correctas y estar mejor preparados para futuras situaciones desconocidas (Gualotuña Lema , 2016).

Este modelo consiste en cinco fases distintas, cuyos nombres dependen del tipo del proyecto en el que se aplica. Cada fase del proceso de desarrollo culmina con un hito visible, tal como se describe a continuación (Gualotuña Lema , 2016):

Fase 1: Visión. En esta fase el equipo y el cliente definen los requerimientos del negocio y los objetivos generales del proyecto. La fase culmina con el hito Visión y Alcance aprobados (Gualotuña Lema , 2016).

Fase 2: Planeación. Durante la fase de planeación el equipo crea un borrador del plan maestro del proyecto, además de un cronograma del proyecto y de la especificación funcional del proyecto. Esta fase culmina con el hito Plan del proyecto aprobado (Gualotuña Lema , 2016).

Fase 3: Desarrollo. Esta fase involucra una serie de releases internos del producto, desarrollados por partes para medir su progreso y para asegurarse que todos sus módulos o partes están sincronizados y pueden integrarse. La fase culmina con el hito Alcance completo (Gualotuña Lema , 2016).

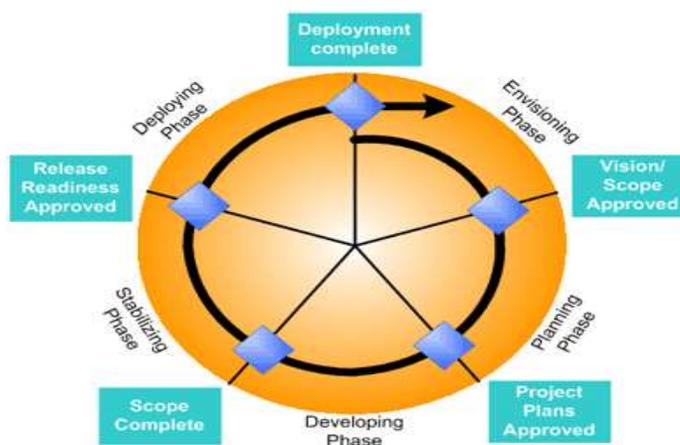
Fase 4: Estabilización. Esta fase se centra en probar el producto. El proceso de prueba hace énfasis en el uso y el funcionamiento del producto en las condiciones del ambiente real. La fase culmina con el hito Release Readiness aprobado (Gualotuña Lema , 2016).

Fase 5: Implantación: En esta fase el equipo implanta la tecnología y los componentes utilizados por la solución, estabiliza la implantación, apoya el funcionamiento y la transición del proyecto, y obtiene la aprobación final del cliente. La fase termina con el hito Implantación completa (Gualotuña Lema , 2016).

La Figura 2 indica, Modelo de Proceso de Desarrollo de Aplicaciones MSF.

Figura 2

Modelo de proceso de desarrollo de aplicaciones MSF.



Nota. Fuente (Gualotuña Lema , 2016)

Metodología de Desarrollo Ágil

Es una metodología con una colección de métodos de ingeniería de software basados en el desarrollo interactivo e incremental, donde la colaboración y participación de grupos auto organizados y multidisciplinarios es fundamental para obtener los requisitos y soluciones. Los métodos ágiles enfatizan las comunicaciones cara a cara en

vez de la documentación, entre las más utilizadas tenemos: SCRUM, XP o Extreme Programming (Lopez Provencio, 2015).

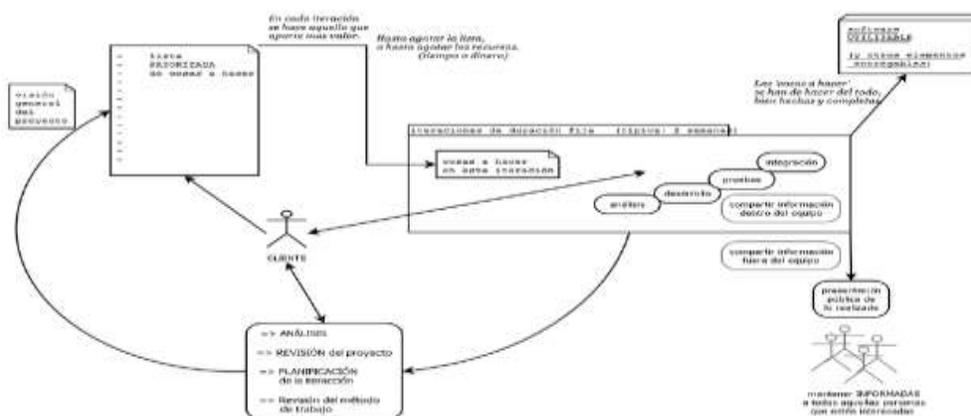
La mayoría de estas metodologías minimiza riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar entre 1 a 4 semanas, en el ciclo de vida del desarrollo cada iteración pasa por las siguientes etapas: Planificación (Iteraciones y fechas de liberación), Análisis de requisitos (Agrupa necesidades para liberaciones cortas), Diseño (Prototipos), Construcción (Desarrollo necesidades), Revisión y Documentación (Entrega liberaciones de software al usuario final) (Lopez Provencio, 2015).

En esta metodología el cliente es quien asigna la prioridad de las tareas, por tal razón no existe mayores controles de seguridad debido a que la seguridad no se suele considerar importante para un cliente, hasta que falla, si bien es cierto la metodologías ágiles permiten integrar con facilidad la seguridad en el desarrollo, es necesario que el equipo de trabajo haga consciente al cliente de los problemas que acarrea dejar de lado la seguridad (Lopez Provencio, 2015).

La Figura 3 indica, el desarrollo de software ágil.

Figura 3

Desarrollo de software ágil.



Nota. Fuente (Lopez Provencio, 2015)

Comparativo Metodologías Tradicionales versus Agiles.

Las metodologías tradicionales trabajan con un enfoque riguroso que se fundamenta en la documentación sobre el proceso de desarrollo de software, realizando énfasis en la planificación global, considerando la totalidad del trabajo a realizar, y una vez detallado se inicia con el ciclo de desarrollo de software. En cambio las metodologías ágiles en muchas ocasiones evaden la documentación y se enfocan en buscar el equilibrio entre proceso/esfuerzo (Molina Montero, Vite Cevallos, & Dávila Cuesta, 2018).

En resumen, estas dos metodologías sostienen dos filosofías opuestas de cómo gestionar y desarrollar de proyectos de software a pesar que no se encuentren claramente delimitadas (Garcia Rodriguez, 2015).

La tabla 1, indica un comparativo de las metodologías tradicionales y agiles en el desarrollo de software.

Tabla 1

Comparativo metodologías tradicionales y agiles en el desarrollo de software.

Aspectos	Tradicional	Ágil
Ciclo de vida	Secuencial: Cascada, Espiral.	Iterativo, modelos evolutivos.
Estilo de desarrollo	Anticipativo.	Adaptativo.
Requisitos	Conocidos, estables, claramente definidos y documentados.	Desconocidos a priori, definidos durante el proyecto.
Arquitectura	Pesada y sobredimensionada para los requisitos actuales y futuros.	Filosofía You Aren't Gonna Need It (YAGNI).
Gestión	Centrada en procesos: mando y control.	Centrada en la gente: liderazgo y colaboración.
Documentación	Detallada, conocimiento explícito.	Ligera, conocimiento táctico.
Metas	Previsibilidad y optimización.	Exploración y adaptación.
Cambio	Aversión al cambio.	Acepta cambios.

Aspectos	Tradicional	Ágil
Organización del equipo	Equipo preestructurados.	Equipo autoorganizados.
Involucramiento del cliente	Pasivo, poca involucrada.	Activo, un miembro más del equipo.
Cultura organizativa	Jerárquica con mando y control establecido.	Menos jerárquica con liderazgo y colaboraciones.
Desarrollo de software	Acercamiento universal con solución predecible y altamente segura.	Acercamiento flexible adaptado a las necesidades particulares del proyecto.
Medida de éxito	Conforme a un plan.	Evaluar globalmente el negocio.

Nota. Fuente (García Rodríguez, 2015)

La tabla 2, indica otros comparativos relevantes para estas dos metodologías para desarrollo de software.

Tabla 2

Comparación de metodologías.

Tradicional	Ágil
Basada en normas procedentes de estándares seguidos por el entorno de Desarrollo.	Basada en heurísticas procedentes de Prácticas de producción de código.
Proceso muy controlado, numerosas Normas.	Proceso menos controlado, con pocos Principios.
Grupos grandes.	Grupos pequeños (<10).
Más artefactos.	Pocos artefactos.
La arquitectura del software es esencial	Menor énfasis en la arquitectura del Software.
Se concibe como un proyecto.	Un proyecto es subdividido en varios Proyectos más pequeños.
Entrega de software al finalizar el desarrollo.	Entregas constantes de software.

Nota. Fuente (Molina Montero, Vite Cevallos, & Dávila Cuesta, 2018)

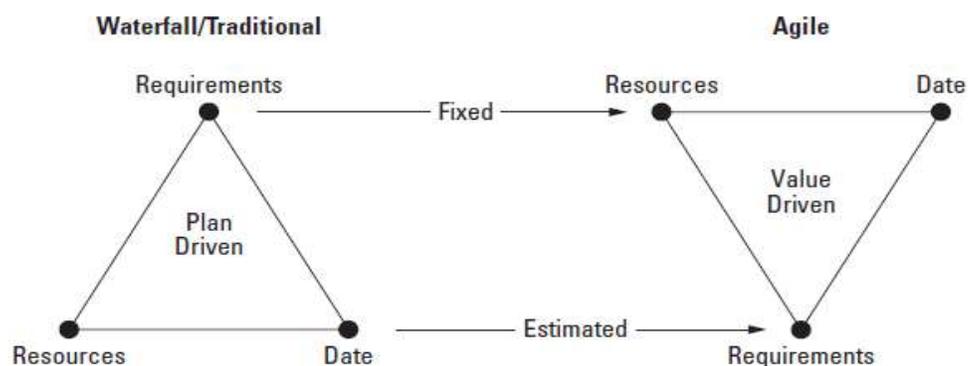
Por otro lado la gestión de proyectos tiene una visión diferente para ambas metodologías. En la tradicional los requisitos se fijan siguiendo a un plan, donde los plazos y recursos son estimados para cumplir el plan. En cambio en la visión ágil los plazos y recursos están fijados y siguiendo un plan de generación de valor, se

desarrollar el proyecto, en función de requisitos cambiantes acordados entre todas las partes periódicamente (Garcia Rodriguez, 2015).

La figura 4, indica un comparativo del triangulado de gestión de proyectos para las 2 metodologías.

Figura 4

Triangulo de la gestión de proyectos tradicional y ágil



Nota. Fuente (Garcia Rodriguez, 2015)

Marco de Referencia Cobit 2019

Durante el paso de los años, se han desarrollado y promocionado marcos de referencia de mejores prácticas para contribuir a entender, diseñar e implementar el GEIT (Gobierno empresarial de información y tecnología), COBIT 2019 se integra y se basa en más de 25 años de desarrollo en este campo (ISACA, 2018).

Desde su nacimiento en la comunidad de auditoría de TI, COBIT ha pasado a ser un marco de referencia de gestión y gobierno de I&T más amplio y exhaustivo, estableciéndose como un marco generalmente aceptado para el gobierno de I&T (ISACA, 2018).

COBIT 2019 se centra en el gobierno de la información y la tecnología (I&T), como marco de referencia reconoce que la información y la tecnología podría encontrarse fuera del departamento de TI tradicional y abarcar toda la información y la tecnología que la empresa genera, procesa y utiliza para lograr sus objetivos, así como la tecnología para respaldarla en toda la empresa (ISACA, 2018).

COBIT 2019 se desarrolló con base en dos grupos de principios (ISACA, 2018):

- Principios que describen los requisitos fundamentales de un **Sistema de Gobierno** para la información y la tecnología de la empresa (ISACA, 2018).
- Principios para un **Marco de Referencia de Gobierno** que puede usarse para crear un sistema de gobierno para la empresa (ISACA, 2018).

Principios del Sistema de Gobierno.- Son seis y se trata de los requisitos fundamentales de un sistema de gobierno para la información y la tecnología de la empresa (ISACA, 2018):

- Proporcionar valor a las partes interesadas.- Cada empresa necesita un sistema de gobierno para satisfacer las necesidades de las partes interesadas y generar valor del uso de la I&T. El valor refleja un equilibrio entre beneficios, riesgos y recursos, y las empresas necesitan una estrategia y un sistema de gobierno práctico para materializar este valor (ISACA, 2018).
- Enfoque holístico.- Un sistema de gobierno para la I&T de la empresa se crea a partir de una serie de componentes que pueden ser de distinto tipo y que funcionan conjuntamente de forma holística (ISACA, 2018).
- Sistema de gobierno dinámico.- Esto significa que cada vez que se cambian uno o más factores del diseño (p. ej. un cambio de estrategia o

tecnología), debe considerarse el impacto de estos cambios en el sistema de GEIT. Una visión dinámica de GEIT llevará a un sistema de GEIT preparado para el futuro (ISACA, 2018).

- Diferenciar gobierno de gestión.- Un sistema de gobierno debe distinguir claramente entre actividades de gobierno y actividades de gestión, y estructuras (ISACA, 2018).
- Adaptado a las necesidades de la empresa.- Un sistema de gobierno debe personalizarse de acuerdo con las necesidades de la empresa, con una conjunto de factores de diseño como parámetros para personalizar y priorizar los componentes del sistema de gobierno (ISACA, 2018).
- Sistema de gobierno íntegro.- Un sistema de gobierno debería cubrir la empresa de principio a fin y centrarse no solo en la función de TI, sino en todo el procesamiento de tecnología e información que la empresa pone en funcionamiento para lograr sus objetivos, independientemente de su ubicación en la empresa (ISACA, 2018).

Principios para el Marco de Referencia de Gobierno.- Son tres y se trata de identificar los principios subyacentes para un marco de referencia de gobierno que pueda usarse para crear un sistema de gobierno para la empresa (ISACA, 2018):

- Basado en un modelo conceptual.- Un marco de referencia de gobierno se debería basar en un modelo conceptual que identifique los componentes principales y las relaciones entre componentes para maximizar la consistencia y permitir la automatización (ISACA, 2018).
- Abierto y flexible.- Un marco de referencia de gobierno debería ser abierto y flexible. Debería permitir la incorporación de nuevo contenido y

la capacidad para abordar nuevos asuntos de la forma más flexible, mientras mantiene la integridad y la consistencia (ISACA, 2018).

- Alineado con los principales estándares.- Un marco de referencia de gobierno debería alinearse con los principales estándares, marcos de referencia y regulaciones relacionados (ISACA, 2018).

COBIT define el sistema de gobierno para cubrir de inicio a fin la empresa, enfocándose no solo en la función de TI, sino en todo el procesamiento de tecnología e información para lo cual utiliza los 7 componentes para crear y sostener un sistema de gobierno: procesos, estructuras organizativas, políticas y procedimientos, flujos de información, cultura y comportamientos, habilidades e infraestructura (ISACA, 2018).

COBIT es un marco de referencia para el gobierno y la gestión de la información y la tecnología que está dirigido a toda la empresa, para lo cual se apoya en 40 objetivos de gobierno y gestión, los mismos que se agrupan en cinco dominios (ISACA, 2018):

Gobierno:

1. Evaluar, Dirigir y Monitorizar (**EDM**):- Su función es evaluar las opciones estratégicas, Orientar y monitorear los logros.

Gestión:

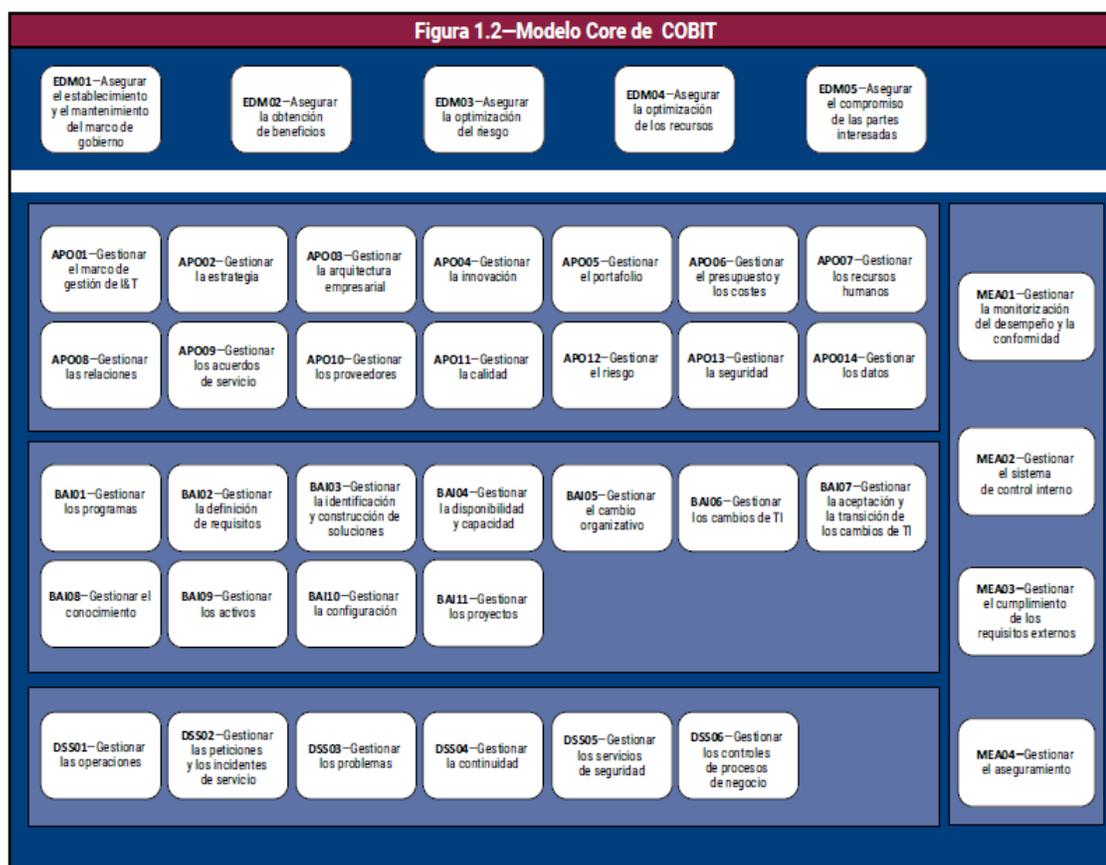
2. Alinear, Planificar y Organizar (**APO**).-Su función es la organización en general, Estrategia y Planificación.
3. Construir, Adquirir e Implementar (**BAI**)- Su función es definir, implementar y velar por la integración en los procesos de negocio.
4. Entregar, Dar Servicio y Soporte (**DSS**).- Su función es garantizar la operativa y el soporte.

5. Monitorizar, Evaluar y Valorar (**MEA**).- Su función es el monitoreo y rendimiento de la operativa.

La figura 5, indica el conjunto completo de los objetivos de gobierno y gestión dentro de COBIT 2019.

Figura 5

Objetivos de gobierno y gestión dentro de Cobit 2019.



Nota. Fuente (ISACA, 2018)

Dominio: Construir, Adquirir e Implementar (BAI)

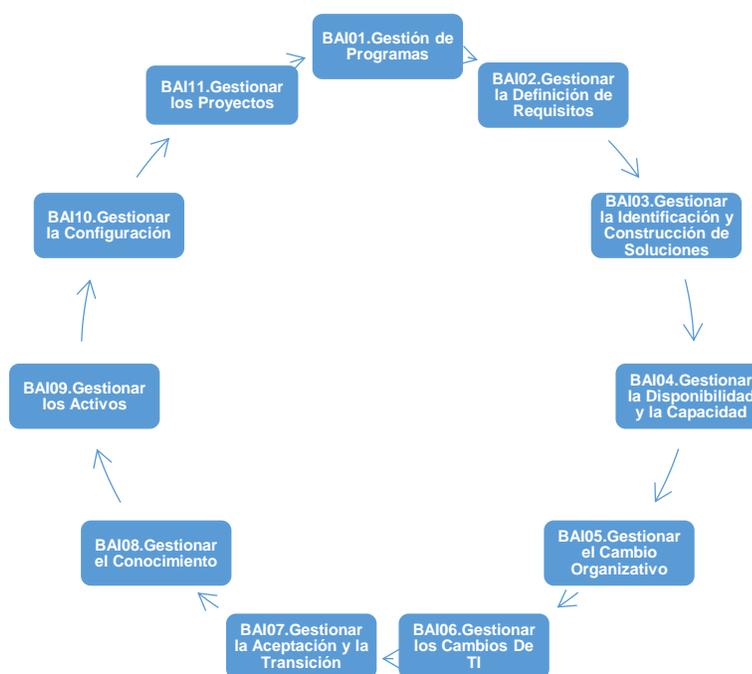
El dominio Construir, Adquirir e Implementar busca satisfacer las necesidades del negocio mediante soluciones o nuevos proyectos que sean entregados dentro del tiempo y presupuesto establecido, por otro lado una vez implementadas e integradas las

soluciones en los procesos del negocio estas operen de manera adecuada y que cualquier cambio realizado no afecten su operatividad (ISACA, 2018) .

La figura 6, indica los 11 objetivos de gestión relacionados al Dominio BAI.

Figura 6

Objetivos de gestión del dominio BAI.



Gestionar Los Programas (BAI01): Este objetivo se encarga de coordinar alineado con la estrategia de la organización el Inicio, planificación, control y ejecución de los programas estándar, así como también su cierre con una revisión post- implementación monitorizando el valor esperado del programa (ISACA, 2018).

Gestionar la Definición de Requisitos (BAI02): Este objetivo se encarga de identificar soluciones y analizar los requerimientos antes de la adquisición o creación asegurándose que estén alineados con los requerimientos estratégicos de la organización, de tal manera que cubran los procesos de negocios, aplicaciones, información/datos, infraestructura y servicios (ISACA, 2018).

Gestionar la Identificación y Construcción de Soluciones (BAI03): Este objetivo se encarga de establecer y mantener soluciones alineadas con los requerimientos de la organización, contemplando el diseño desarrollo, compras/contratación y asociación con proveedores/fabricantes. Adicional a esto gestiona la configuración, preparación y ejecución de pruebas, así como también la gestión de requerimientos y mantenimiento de procesos de negocio, aplicaciones, datos/información, infraestructura y servicios (ISACA, 2018).

Gestionar la Disponibilidad y la Capacidad (BAI04): Este objetivo se encarga de equilibrar necesidades actuales y futuras de disponibilidad, rendimiento y capacidad con una provisión de servicio efectiva en costes, así como también evaluar capacidades actuales, prever necesidades futuras basadas en los requerimientos de la organización y la evolución de riesgos para planificar e implementar acciones que permitan lograr los requerimientos (ISACA, 2018).

Gestionar el Cambio Organizativo (BAI05): Este objetivo se encarga de maximizar la probabilidad exitosa de implementación en toda la organización del cambio de forma rápida y con riesgo reducido, abarcado todo el ciclo de vida del cambio y las partes interesadas del negocio y de TI (ISACA, 2018).

Gestionar los Cambios de TI (BAI06): Este objetivo se encarga de forma controlada de gestionar todos los cambios que se ejecutan en los procesos del negocio, aplicaciones e infraestructura, incluyendo cambios estándar y mantenimientos emergentes. Considerando también normas y procedimientos de cambio, análisis de impacto, priorización y autorización, cambios de emergencia, seguimiento, reporte, cierre y documentación (ISACA, 2018).

Gestionar la Aceptación y la Transición (BAI07): Este objetivo se encarga de hacer operativas y aceptar formalmente las nuevas soluciones, incluyendo la planificación de la implementación, la conversión de los datos y los sistemas, las

pruebas de aceptación, la comunicación, la preparación del lanzamiento, el paso a producción de procesos de negocio o servicios TI nuevos o modificados, el soporte temprano en producción y una revisión post-implementación (ISACA, 2018).

Gestionar el Conocimiento (BAI08): Este objetivo se encarga de mantener disponible de conocimiento relevante, actual, validado y fiable para dar soporte a todas las actividades de los procesos y facilitar la toma de decisiones. Así como también planificar la identificación, recopilación, organización, mantenimiento, uso y retirada de conocimiento (ISACA, 2018).

Gestionar los Activos (BAI09): Este objetivo se encarga de gestionar los activos de TI a través de su ciclo de vida para garantizar que su uso aporta valor a un coste óptimo, que se mantendrán en funcionamiento, que están justificados y protegidos físicamente, y que los activos que son fundamentales para apoyar la capacidad del servicio son fiables y están disponibles. Así como también de administrar las licencias de software para asegurar que se adquiere el número óptimo requerido, se mantienen y despliegan en relación con el uso necesario para el negocio y que el software instalado en la organización cumple con los acuerdos de licencias (ISACA, 2018).

Gestionar la Configuración (BAI10): Este objetivo se encarga de definir y mantener las definiciones y relaciones entre los principales recursos y capacidades necesarios para la prestación de los servicios proporcionados por TI, incluyendo la recopilación de información de configuración, así como también el establecimiento de líneas de referencia, la verificación y auditoría de la información de configuración y la actualización del repositorio de configuración (ISACA, 2018).

Gestionar los Proyectos (BAI11): Este objetivo se encarga de gestionar de manera coordinada y alineada con la estrategia de la organización, el Inicio, planificación, control y ejecución de los proyectos estándar, así como también su cierre con una revisión post-implementación (ISACA, 2018).

Scrum

Scrum consiste en un esfuerzo de colaboración para crear un nuevo producto, servicio u otro resultado tal como se define en la declaración de la visión del proyecto (Project Visión Statement). Los proyectos se ven afectados por limitaciones de tiempo, costos, alcance, calidad, recursos, capacidades organizacionales y demás limitaciones que dificultan su planificación, ejecución, administración y por último, su éxito. Sin embargo, la implementación exitosa de los resultados de un proyecto terminado le proporciona ventajas económicas considerables a una organización. Por lo tanto, es importante que las organizaciones seleccionen e implementen un método adecuado de gestión de proyectos (SCRUMstudy™, 2017).

Scrum es uno de los métodos ágiles más populares. Es un framework adaptable, iterativo, rápido, flexible y eficaz, diseñado para ofrecer un valor considerable en forma rápida a lo largo del proyecto. Scrum garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva y de progreso continuo (SCRUMstudy™, 2017).

Una fortaleza clave de Scrum radica en el uso de equipos interfuncionales (cross-functional), auto organizados y empoderados que dividen su trabajo en ciclos de trabajo cortos y concentrados llamados Sprint (SCRUMstudy™, 2017).

El proceso está formado por cinco fases: Inicio; Planificación y Estimación; Implementación, Revisión y Retrospectiva; y Lanzamiento. Estas fases describen a detalle las entradas y salidas asociadas con cada proceso, así como las diferentes herramientas que pueden utilizarse en cada una. Algunas entradas, herramientas y salidas son obligatorias y estas se indican como tales; otras son opcionales dependiendo del proyecto específico, de los requisitos de la organización y/o

lineamientos establecidos por el Scrum Guidance Body de la organización (SGB) (SCRUMstudy™, 2017).

La tabla 3, La fases y Procesos fundamentales de Scrum.

Tabla 3

Fases y procesos fundamentales de Scrum.

Fase	Procesos Fundamentales de Scrum
Inicio	<ul style="list-style-type: none"> • Crear la visión del proyecto • Identificar al Scrum Master y Stakeholder(s) • Formar Equipos Scrum • Desarrollar épica(s) • Crear el Backlog Priorizado del Producto • Realizar la planificación de lanzamiento
Planificación y estimación	<ul style="list-style-type: none"> • Crear historias de usuario • Estimar historias de usuario • Comprometer historias de usuario • Identificar tareas • Estimar tareas • Crear el Sprint Backlog
Implementación	<ul style="list-style-type: none"> • Crear entregables • Realizar Daily Standup • Refinar el Backlog Priorizado del Producto
Revisión y retrospectiva	<ul style="list-style-type: none"> • Demostrar y validar el sprint • Retrospectiva del sprint
Lanzamiento	<ul style="list-style-type: none"> • Enviar entregables • Retrospectiva del proyecto

Nota. Fuente (SCRUMstudy™, 2017)

Como método ágil (Mariño & Alfonso, 2014):

- El desarrollo adaptable, antes que predictivo.
- Orientado a las personas, más que a los procesos.

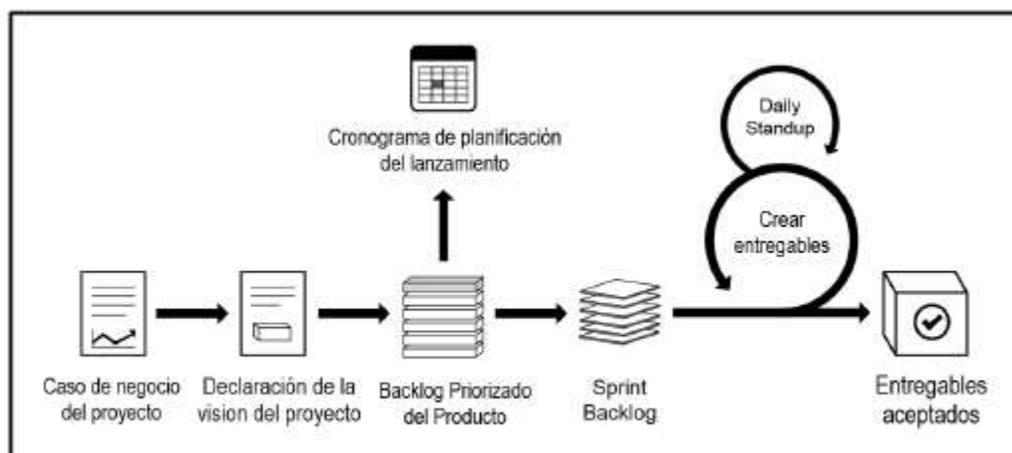
- Utiliza un modelo de construcción incremental basado en iteraciones y revisiones.

Las prácticas empleadas por Scrum para mantener un control ágil en el proyecto son: (1) Revisión de las iteraciones, (2) Desarrollo incremental, (3) Desarrollo evolutivo, (4) Auto organización del equipo y (5) Colaboración (Mariño & Alfonso, 2014).

La figura 7, indica la visión general de flujo de un proyecto Scrum.

Figura 7

Flujo de Scrum para un sprint.



Nota. Fuente (SCRUMstudy™, 2017)

El Flujo de Scrum para un sprint inicia con una reunión de stakeholders, cuyo propósito es crear la visión del proyecto. Después, el Product Owner desarrolla una Backlog Priorizado del Producto (Prioritized Product Backlog) el mismo que contiene un listado de requerimiento del negocio y del proyecto por orden de importancia en forma de una historia de usuario. Cada sprint inicia con una reunión de planificación del sprint (Sprint Planning Meeting) donde las historias de usuario de alta prioridad son consideradas para cada sprint. Un sprint generalmente tiene una duración de una a seis semanas durante las cuales el equipo scrum trabaja en la generación de entregables. Durante la ejecución del sprint, se llevan a cabo Daily Standups rápidos y concretos,

que permite a los miembros del equipo revisar el progreso diario. Al final del sprint, se lleva a cabo una reunión de revisión del sprint (Sprint Review Meeting) en la cual se realiza una demostración de los entregables al Product Owner y a los stakeholders relevantes. El Product Owner acepta los entregables sólo si cumplen con los criterios de aceptación predefinidos. El ciclo del sprint finaliza con una reunión de retrospectiva del Sprint (Retrospect Sprint Meeting), donde el equipo analiza las formas de mejorar los procesos y el rendimiento a medida que avanzan al siguiente sprint (SCRUMstudy™, 2017).

Owasp

Owasp es un proyecto open source de seguridad en aplicaciones web, que fue propuesto desde el año 2003 y ha tenido algunas series de cambios y actualizaciones respecto a los principales riesgos de seguridad en la web durante estos últimos años, el propósito principal de owasp es identificar y mitigar las vulnerabilidades presentadas en un servidor web, con la finalidad de impedir que los usuarios no autorizados accedan a la información privada recopilada en los sistemas de información web, creando conciencia en las organizaciones sobre los riesgos a los que se exponen, el impacto técnico y de negocio que sufrirían sus activos (Briones Pincay & Hernandez Peñaherrera, 2018) .

OWASP muestra una lista de los riesgos críticos sobre el resguardo de datos que se acoplan en los sistemas web y aporta con buenas prácticas para mitigar estas vulnerabilidades (Briones Pincay & Hernandez Peñaherrera, 2018).

OWASP es una organización sin interés de ganancia que maneja diferentes proyectos relacionados con seguridad de la información. Entre los proyectos que gestionan se encuentra la OWASP Testing Guide, la cual contiene tres secciones claramente diferenciados (Lopez Provencio, 2015).

El testing.- Esta sección contiene una rápida introducción sobre testing de seguridad, las principales técnicas utilizadas y los requisitos que se deben cumplir para realizar un testing significativo (Lopez Provencio, 2015).

Framework OWASP.- Esta sección contiene la metodología de trabajo propuesta por OWASP para seguridad de un software. La metodología tiene un listado de consideraciones, recomendaciones y actividades que se debe considerar al momento de realizar el desarrollo de software lo cual permitirá facilitar la detección temprana de problemas de seguridad a nivel del código. Dentro de las recomendaciones de la metodología se propone llevar a cabo una auditoria del software para asegurar que no han quedado vulnerabilidades (Lopez Provencio, 2015).

Métodos de ataque contra una aplicación web.- Esta sección es la más extensa de la guía y se enfoca a explicar en un nivel de detalle las diferentes técnica utilizadas para realizar auditorías de seguridad en aplicaciones web (Lopez Provencio, 2015).

La tabla 4, indica las ventajas y desventajas encontradas en esta metodología.

Tabla 4

Ventajas y desventajas metodología Owasp

Ventajas	Desventajas
Contempla la seguridad de un proyecto.	No contiene mayor detalle sobre algunas fases del ciclo de vida de desarrollo de software solo se comentan por encima.
Metodología que más se acerca a las necesidades de los proyectos de aplicaciones web	Los proyecto cubiertos por esta metodología solo son de aplicaciones web.

Nota. Fuente (Lopez Provencio, 2015).

Para las pruebas de aplicaciones web, OWASP utiliza un procedimiento de que se fundamenta en dos fases principales: pasivo y activo empleando para esto la auditoría de caja negra (Briones Pincay & Hernandez Peñaherrera, 2018).

Caja Negra.- Es un término usado en auditoria para definir el Test a ser utilizado en la ejecución remota sin conocer como está diseñado el sitio web y poder hallar vulnerabilidades para futuros ataques (Briones Pincay & Hernandez Peñaherrera, 2018).

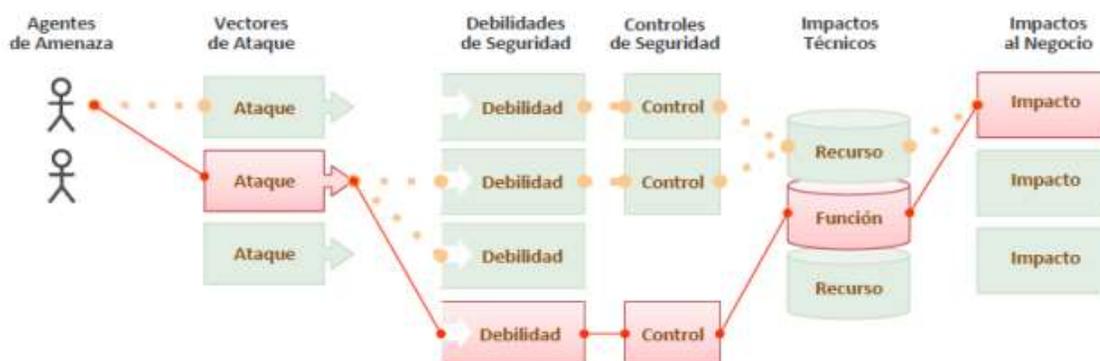
El auditor inicia la ejecución del Test con la visión de un usuario el cual desconoce de la infraestructura del sistema web y se procede al ataque solo con la información pública que brinda el sitio web como el nombre del domino, permitiéndole encontrar lo mismo que podría encontrar el *hacker* ya que el ataque es hecho desde su mismo ambiente, es decir remotamente (Briones Pincay & Hernandez Peñaherrera, 2018).

Riesgos de seguridad en OWASP.- Los hacker pueden hallar vulnerabilidades del sitio web con la finalidad de dañar la integridad de las organizaciones. Cada uno de estos caminos que llegue a tomar puede no tener repercusiones o podría encontrar debilidades, invadiendo los controles de seguridad para acceder a los recursos de la organización y de esa manera causa un impacto técnico y de operatividad (Briones Pincay & Hernandez Peñaherrera, 2018).

La figura 8, indica los caminos en que un hacker puede acceder a la aplicación.

Figura 8

Ataque genérico.



Nota. Fuente: (Briones Pincay & Hernandez Peñaherrera, 2018)

Factores de Riesgo por OWASP.- OWASP tiene un top 10 de los riesgos más utilizados contra los sitios web de las organizaciones y para cada uno de ellos información relevante y genérica sobre la probabilidad y el impacto que pueden llegar a tener (Briones Pincay & Hernandez Peñaherrera, 2018).

La figura 9, indica una sinopsis de los riesgos más importantes de Seguridad de Aplicación 2017 y los elementos de riesgo asociados a cada uno. Estos elementos se establecieron en relación a los inventarios disponibles y a la práctica del equipo Top 10 de OWASP.

Figura 9

Diez factores de riesgo.

RISK	Threat Agents	Attack Vectors			Security Weakness		Impacts	Score
		Exploitability	Prevalence	Detectability	Technical	Business		
A1:2017-Injection	App Specific	EASY: 3	COMMON: 2	EASY: 3	SEVERE: 3	App Specific	8.0	
A2:2017-Authentication	App Specific	EASY: 3	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	7.0	
A3:2017-Sens. Data Exposure	App Specific	AVERAGE: 2	WIDESPREAD: 3	AVERAGE: 2	SEVERE: 3	App Specific	7.0	
A4:2017-XML External Entities (XXE)	App Specific	AVERAGE: 2	COMMON: 2	EASY: 3	SEVERE: 3	App Specific	7.0	
A5:2017-Broken Access Control	App Specific	AVERAGE: 2	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	6.0	
A6:2017-Security Misconfiguration	App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	MODERATE: 2	App Specific	6.0	
A7:2017-Cross-Site Scripting (XSS)	App Specific	EASY: 3	WIDESPREAD: 3	EASY: 3	MODERATE: 2	App Specific	6.0	
A8:2017-Insecure Deserialization	App Specific	DIFFICULT: 1	COMMON: 2	AVERAGE: 2	SEVERE: 3	App Specific	5.0	
A9:2017-Vulnerable Components	App Specific	AVERAGE: 2	WIDESPREAD: 3	AVERAGE: 2	MODERATE: 2	App Specific	4.7	
A10:2017-Insufficient Logging&Monitoring	App Specific	AVERAGE: 2	WIDESPREAD: 3	DIFFICULT: 1	MODERATE: 2	App Specific	4.0	

Nota. Fuente (Briones Pincay & Hernandez Peñaherrera, 2018)

Capítulo III: Metodología Aplicada al Caso de Estudio

Fase I: Situación Actual

Banco Internacional, con la finalidad de asegurar el cumplimiento de objetivos estratégicos de la organización y los compromisos adquiridos para con la entrega de soluciones al negocio tiene una metodología de trabajo para el desarrollo de software, la misma que trata de viabilizar la comunicación y colaboración entre las partes / actores involucrados (áreas usuarias, áreas de control, áreas TI y terceros), mediante instrucciones generales y específicas que norman la administración y ejecución del proceso de ciclo de desarrollo de software sobre las aplicaciones.

Líneas de servicio desarrollo de software: En el área de desarrollo del Banco Internacional tiene las siguientes líneas de servicio de desarrollo de software:

Mantenimiento de Software: Corresponde a todo desarrollo de software relacionado con una corrección de un error/defecto, actualización o mejora en los módulos del Core Bancario y los aplicativos satélites.

Las mejoras y/o actualizaciones según su duración y esfuerzo desde el análisis hasta el soporte postproducción se clasifican en 2 categorías:

- **Mantenimiento Menor.-** Desarrollo hasta 40 horas o hasta 2 semanas incluido el soporte postproducción.
- **Mantenimiento Mayor.-** Desarrollo entre 80 y 160 horas o entre 2 y 4 semanas incluido el soporte postproducción.

Si la solicitud de mantenimiento de software por motivo de “mejora” y/o “actualización” es mayor a las 160 horas de esfuerzo de desarrollo o 4 semanas / 1 mes incluido el soporte postproducción, se considera como Proyecto de Software.

Proyectos de Software: Corresponde a todo desarrollo de software relacionado con la implementación de un nuevo servicio, producto, proceso y aplicativo bancario.

Según su duración y esfuerzo desde el análisis hasta el soporte postproducción se clasifican en 2 categorías:

- **Mini Proyectos.-** Desarrollo entre 161 y 320 horas y más de 4 semanas / 1 mes incluido el soporte postproducción.
- **Proyectos.-** Desarrollo mayor a 321 horas y más de 2 meses incluido el soporte postproducción.

Estructura área de desarrollo: Cada línea de servicio de desarrollo de software cuenta con su respectivo equipo de trabajo conformado por:

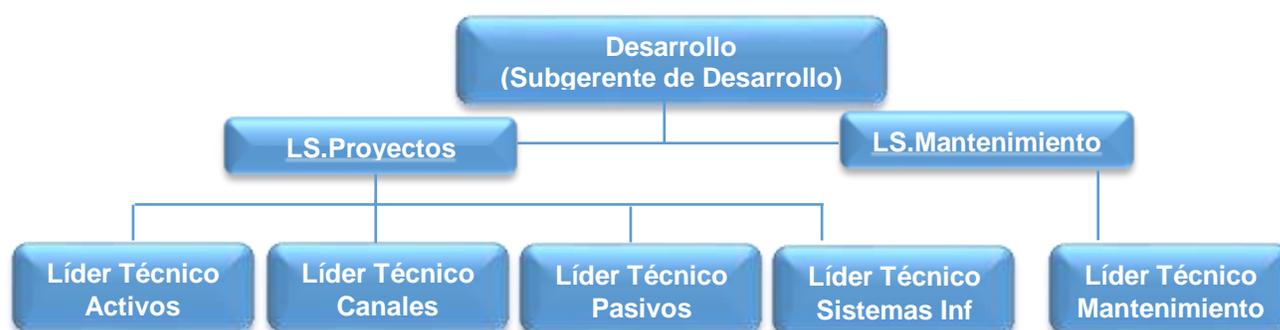
- Líder Técnico.
- Desarrolladores.

Para la línea de servicio de Proyectos de Software se tiene conformado 4 equipos, mientras que en la línea de servicio de Mantenimiento de Software se tiene 1 solo equipo de trabajo.

La figura 10, indica estructura área de desarrollo distribuida por líneas de servicio.

Figura 10

Estructura de área de desarrollo distribuida por Líneas de Servicio



La tabla 5, indica los roles y funciones de los equipos de trabajo para las líneas de servicio.

Tabla 5

Roles y funciones de los equipos de trabajo por Línea de Servicio.

Roles	Funciones
Líder Técnico (Activos-Pasivos- Canales- Sistemas de Información).	<ul style="list-style-type: none"> • Gestion y seguimiento de Mini proyectos y Proyectos durante el ciclo de vida de desarrollo. • Generar documento de especificaciones, Plan de trabajo y Diseño solución. • Responsable del capacity equipo y asignación de trabajo. • Garantizar el cumplimiento de la metodología. • Generación de documentación base del proyecto.
Líder Técnico Mantenimiento	<ul style="list-style-type: none"> • Gestion y seguimiento de los Mantenimientos Menores y Mayores durante el ciclo de vida de desarrollo. • Gestion Incidentes y resolución de problemas detectados en producción. • Generar documento de especificaciones, Plan de trabajo y Diseño solución. • Responsable del capacity equipo y asignación de trabajo. • Garantizar el cumplimiento de la metodología.
Desarrolladores	<ul style="list-style-type: none"> • Participación activa en las etapas de análisis, diseño y planificación. • Ejecutar la construcción del software, cumpliendo estándares de programación y metodología dentro de la planificación establecida. • Atender corrección de incidentes de desarrollo asignados en la etapa de QA, certificación y postproducción • Acompañamiento en la puesta en producción del desarrollo asignado. • Generar documentos técnicos.

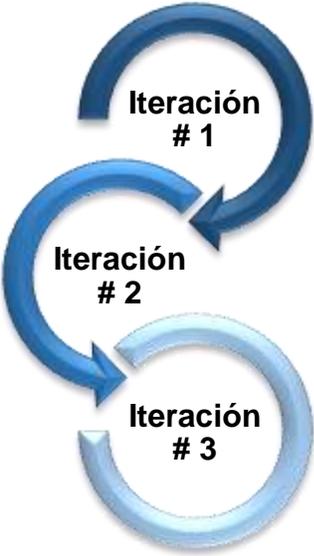
Estrategia y esquema de trabajo: En el proceso de construcción software en Banco Internacional utiliza una metodología para el desarrollo de software basada principalmente en las mejores prácticas del marco referencial MSF (Microsoft Solution Framework) con pequeñas variaciones en la fase de desarrollo que les permite realizar liberaciones parciales de funcionalidad definidas en orden de prioridad dada por el negocio.

Para la ejecución de todo desarrollo de software se establece realizar hasta 3 iteraciones que permitirá entregar producto al negocio, los factores claves que permitirán definir la funcionalidad liberada en cada iteración está en función de las variables alcance, tiempo y disponibilidad de recursos.

La tabla 6, indica las 3 iteraciones para todo desarrollo de software.

Tabla 6

Iteraciones para todo desarrollo de software.

	<p>I#1.-Considera funcionalidad principal del desarrollo para alcanzar los objetivos generales del proyecto. Representa la parte medular del proceso/ servicio/producto a mejor/implementar, (inicio hasta fin). La funcionalidad a implementarse en esta iteración no es factible de ser manejada operativamente dentro del time to market requerido.</p>
	<p>I#2.-Considera funcionalidad complementaria del desarrollo para alcanzar los objetivos específicos del proyecto. Representa reglas del negocio, controles, excepciones del proceso/ servicio/ producto a mejor/ implementar, (inicio hasta fin). La funcionalidad a implementarse en esta iteración puede ser manejada operativamente y no es crítica dentro del time to market requerido.</p>
	<p>I#3.-Considera temas de forma tales como mejoras adicionales de navegabilidad o presentación, consultas y reportes complementarios que general valor a los objetivos específicos del proyecto. La funcionalidad a implementarse en esta iteración puede ser manejada operativamente.</p>

Durante la ejecución de cada interacción se considera las siguientes estrategias:

- **Durante Desarrollo.-** Realizar revisiones de usuario en fase de desarrollo para cada iteración con la finalidad de realizar un solo paso a ambiente de aseguramiento de la calidad.
- **Durante Estabilización.-** Realizar pasó de versión de desarrollo al ambiente de estabilización por cada iteración para realizar pruebas de aseguramiento de la calidad y certificaciones funcionales por iteración con la finalidad de realizar un único pasó a producción.
- **Durante Liberación Producción.-** Realizar paso a producción para cada iteración, realizando pruebas de aseguramiento de calidad y de certificación funcional para cada iteración en ambiente de pruebas.

La tabla 7, indica un resumen a metodología de desarrollo que utilizada Banco Internacional, en donde se mapea/relaciona con el manejo de proyectos, procesos y con el ciclo de vida del desarrollo software.

Tabla 7

Relación entre manejo de proyectos, procesos y CVD.

Manejo de Proyectos	Manejo de procesos	Ciclo de vida desarrollo de Software
Inicio	Visión	Entendimiento Análisis Funcional Análisis y Diseño Técnico
Planificación	Planificación	Planificación
Ejecución	Desarrollo Estabilización	Desarrollo Pruebas Aseguramiento Calidad Certificación Funcional Usuarios Liberación
Cierre	Implementación	Soporte Post producción Cierre

Áreas participantes – roles y funciones: En el proceso de construcción software en Banco Internacional utiliza una metodología para el desarrollo de software basada principalmente en las mejores prácticas del marco referencial MSF (Microsoft Solution Framework) con pequeñas variaciones en la fase de desarrollo que les permite realizar liberaciones parciales de funcionalidad definidas en orden de prioridad dada por el negocio.

Todo nuevo desarrollo tendrá un Gerente de Proyecto quien será el responsable de orquestar y coordinar los requerimientos de la organización de manera integral, así como también la ejecución de Proyecto/Mantenimiento, para esta gestión tendrá a su cargo un equipo de trabajo constituido por:

- Áreas de control.
- Áreas usuarias.
- Tecnología.
- Terceros.

Las áreas de Control (Auditoría, Seguridad de la Información, Control Interno) nombran un responsable de contacto, Gestion el mismos que tendrá la responsabilidad de tomar los definiciones requeridas en la ejecución de los Proyecto/Mantenimiento por otro lado las áreas Usuarias (Negocio, Marketing, Operaciones) también nombran su responsable de contacto, por parte de Tecnología se tendrá como responsable al Líder Técnico del área de desarrollo y los Terceros o proveedores tendrán a uno o varios responsables dependiendo del Proyecto/Mantenimiento que se esté ejecutando en los cuales siempre existirá una contraparte del negocio y técnica con la finalidad de garantizar de manera integral la necesidad.

La tabla 8, indica un resumen de los roles y funciones de los participantes en la Metodología de Desarrollo de Software.

Tabla 8*Roles y funciones ciclo de vida del desarrollo de software*

Roles	Funciones
Gerente de Proyecto	<ul style="list-style-type: none"> • Identificar y coordinar con áreas de control y usuarias su participación como responsables en la ejecución del Proyecto/Mantenimiento lo que permitirá establecer el alcance para el desarrollo, así como él también la elaboración de las especificaciones funcionales y no funcionales, alcance de las pruebas de certificación funcional. • Coordinar y gestionar con áreas usuarias y de control involucradas la entrega de Insumos requeridos por el Líder Técnico • Gestionar las convocatorias de certificación funcional a áreas usuarias involucradas una vez confirmado por Líder Técnico la habilitación del ambiente de pruebas para certificación funcional. • Seguimiento, confirmación y aprobación de las certificaciones funcionales de las diferentes áreas usuarias y de control involucradas. • Soporte y facilitador para resolución de novedades y conflictos entre actores del proyecto, así como también de notificar avances a los interesados. • Administrar contratos de proyectos con Terceros (proveedores, clientes).
Áreas de usuarios	<ul style="list-style-type: none"> • Participar en la definición del alcance y especificaciones funcionales y no funcionales y certificación del Proyecto/Mantenimiento. • Asistir, ejecutar y aprobar la certificación funcional de los casos y/o escenarios de pruebas definidos para validar la completitud y calidad del desarrollo implementado.

- Definir en conjunto al Líder Técnico los aplicativos de terceros involucrados para las pruebas de certificación.

Roles	Funciones
Áreas de control	<ul style="list-style-type: none"> • Participar en la definición de especificaciones funcionales y no funcionales y diseño solución del proyecto/mantenimiento. • Definir y certificar cumplimiento de lineamientos generales referente a cambios en los programas de los aplicativos del banco a nivel de documentación, seguridad informática y cumplimiento metodología de trabajo.
Tecnología- Líder Técnico	<ul style="list-style-type: none"> • Participar en la definición del alcance y especificaciones funcionales y no funcionales del Proyecto/Mantenimiento. • Diseñar y desarrollar soluciones tecnológicas satisfaciendo las necesidades del negocio en función de las especificaciones levantadas. • Realizar estimaciones de esfuerzos y tiempos para cada una de las etapas del Proyecto/Mantenimiento. • Gestionar y coordinar con las áreas de TI (Infraestructura, Operación / Monitoreo, Comunicaciones, soporte técnico, Aseguramiento de la calidad) y/o las contrapartes técnicas de terceros la definición diseño solución, pruebas técnicas y correcciones respectivas del desarrollo implementado.

Fases ciclo de vida desarrollo de software:

La figura 11, indica las fases del ciclo de vida de desarrollo establecida en Banco internacional.

Figura 11

Fases ciclo de vida desarrollo de software.



FASE 1 – Entendimiento: En esta fase se realiza generación del requerimiento funcional de alto nivel indicando el objetivo general, justificación, descripción general del alcance funcional y/o proceso, servicios y/o productos involucrados y beneficios del proyecto/mantenimiento. El QUE y PORQUE.

Responsable de la Fase: Gerente Proyecto/Mantenimiento.

Participantes: Gerente Proyecto/Mantenimiento, Líder Técnico, puntos de contacto de áreas usuarias involucradas, Comité de proyectos.

Actividades:

- Identificar la necesidad del negocio (mejora, corrección u oportunidad).
Responsable: Gerente Proyecto/ Mantenimiento.
- Identificar las destrezas requeridas por el equipo. **Responsable:** Gerente Proyecto/ Mantenimiento + Líder Técnico.
- Reunir los requerimientos iniciales. **Responsable:** Gerente Proyecto/ Mantenimiento.+ Líder Técnico.
- Identificar las partes involucradas directa e indirectamente con la solicitud. **Responsable:** Gerente Proyecto/Mantenimiento + Líder Técnico.
- Definir metas, asunciones y limitantes. **Responsable:** Gerente Proyecto/ Mantenimiento + Líder Técnico.
- Determinar la criticidad, impacto de no realizar la solicitud. **Responsable:** Gerente Proyecto/ Mantenimiento.

Entregables:

- Documento de evaluación de requerimiento (DER) – aplica solo en proyectos. **Responsable:** Gerente Proyecto/Mantenimiento + Líder Técnico.
- Documento Inicial de Plan de proyecto. – aplica solo en proyectos.
Responsable: Gerente Proyecto/ Mantenimiento.
- Presentación Kick off de inicio – aplica solo en proyecto (opcional).
Responsable: Gerente Proyecto/ Mantenimiento.
- Número Ticket Mesa de ayuda - aplica solo en Mantenimientos
Responsable: Gerente Proyecto/ Mantenimiento.

- Listado de áreas involucradas con respectivo responsable.
Responsable: Gerente Proyecto/ Mantenimiento + Líder Técnico.
- Documento funcional inicial. **Responsable:** Gerente Proyecto/ Mantenimiento.
- Diagrama alto nivel de procesos involucrados (opcional). **Responsable:** Gerente Proyecto/ Mantenimiento.

FASE 2 – Análisis funcional: En esta fase se realiza la recopilación de información y se obtiene el detalle funcional para el desarrollo requerido. Detalle del QUE.

Responsable de la Fase: Gerente Proyecto/Mantenimiento.

Participantes: Gerente Proyecto/Mantenimiento, Líder Técnico, Desarrolladores, áreas usuarias involucradas directa e indirectamente en el proceso, servicio o producto a modificar /crear, áreas de control –seguridad información y gobierno información, analista de pruebas de AC, Contra parte técnica de terceros involucrados.

Actividades:

- Recopilar la mayor cantidad de información posible. **Responsable:** Gerente Proyecto/ Mantenimiento + Líder Técnico.
- Identificar los procesos, servicios, productos y aplicativos tecnológicos involucrados en el desarrollo. **Responsable:** Gerente Proyecto/ Mantenimiento + Líder Técnico.
- Involucrar a todas las áreas usuarias involucradas directamente o indirectamente en el proceso, servicio o producto a modificar /crear.
Responsable: Gerente Proyecto/Mantenimiento.

- Generar detalle de requisitos funcionales del desarrollo a realizar.

Responsable: Gerente Proyecto/Mantenimiento.

Entregables:

- Plan de proyecto actualizado. **Responsable:** Gerente Proyecto/Mantenimiento.
- Documento especificaciones funcionales y no funcionales aprobado (firmado o confirmado vía email). **Responsable:** Gerente Proyecto/Mantenimiento + Áreas usuarias y de control involucradas.

FASE 3 – Análisis y diseño técnico: En esta fase se determinan cambios e impacto técnico para generar requisitos funcionales definidos. EL COMO.

Responsable de la Fase: Líder Técnico.

Participantes: Líder Técnico, Desarrolladores, Analista de pruebas AC, áreas de producción de tecnología (Infraestructura, Redes), Seguridad de la Información, Contra parte técnica terceros.

Actividades:

- Determinar las especificaciones técnicas (actividades/tareas) para implementar los requisitos funcionales definidos en los documentos especificaciones funcionales y no funcionales. **Responsable:** Líder Técnico.
- Definir arquitectura diseño de la solución. **Responsable:** Líder Técnico.

Entregables:

- Documento Especificaciones técnicas. **Responsable:** Líder Técnico.

- Diseño de Solución/diagrama arquitectura (opcional). **Responsable:** Líder Técnico.
- Listado de componentes/programas que se requerirá modificar.
Responsable: Líder Técnico + Desarrolladores.
- Listado de actividades/ tareas de implementación por requisito funcional (opcional). **Responsable:** Líder Técnico.

FASE 4 – Planificación: En esta fase se generará el cronograma de trabajo integral del desarrollo de software.

Responsable de la Fase: Líder Técnico.

Participantes: Líder Técnico, Desarrolladores, Analista de pruebas AC, Gerente Proyecto/Mantenimiento, Áreas Usuarias, Contra parte técnica terceros.

Actividades:

- Generar estructuras actividades/tareas de fase de desarrollo con respectivos tiempos de implementación. **Responsable:** Líder Técnico.
- Realizar reunión de definición alcance pruebas funcionales en base a documentos especificaciones funcionales y no funcionales.
Responsable: Gerente Proyecto/Mantenimiento + Analista Pruebas AC + áreas usuarias y de control involucradas.
- Generar listado de escenarios/casos de prueba funcional y tiempos de ejecución certificación funcional con respectivas áreas usuarias y de control involucradas. **Responsable:** Gerente Proyecto/Mantenimiento.
- Generar y consolidar listado de escenarios/casos de prueba funcional con posibles combinaciones pruebas y técnicas (carga, stress, inspección

código, otros) con respectivos responsables y tiempos de ejecución pruebas AC. **Responsable:** Analista pruebas AC.

- Identificar requisitos de ambiente de pruebas para ejecución de pruebas funcionales y técnicas **Responsable:** Líder Técnico + Analista pruebas AC + Gerente Proyecto/ Mantenimiento + áreas usuarias + áreas control.
- Confirmar pre-requisitos de ambiente de desarrollo. **Responsable:** Líder Técnico + Desarrolladores.

Entregables:

- Cronograma de trabajo integral de desarrollo tecnológico (desarrollo + pruebas AC + certificación + liberación + tiempo post producción).
Responsable: Líder Técnico.
- Actualizar Documento de especificaciones funcionales y no funcionales.
Responsable: Gerente Proyecto/Mantenimiento.
- Documento de especificaciones técnicas final. **Responsable:** Líder Técnico.
- Diagrama arquitectura / diseño solución final aprobado por equipo TI (firmado o confirmado vía email) –opcional. **Responsable:** Líder Técnico.
- Matriz consolidada de Listado de escenarios /casos de prueba para fase desarrollo (pruebas técnicas/unitarias), de QA y de certificación.
Responsable: Líder Técnico + Analista Pruebas AC + Gerente Proyecto/Mantenimiento y Usuarios.
- Listado consolidado de requisitos de ambiente desarrollo y ambiente de pruebas requerido para iniciar fase de desarrollo y para pruebas funcionales respectivamente. **Responsable:** Líder Técnico + Desarrolladores + Analista pruebas AC.

FASE 5 – Desarrollo: En esta fase se construye/codificar los requisitos funcionales y no funcionales solicitados en el desarrollo descritos en el documento de especificaciones técnicas.

Responsable de la Fase: Líder Técnico.

Participantes: Líder Técnico, Desarrolladores, Versionamiento de código de AC, Contra parte técnica terceros.

Actividades:

- Confirmación equipo de desarrollo de cada plataforma confirmando responsable por plataforma. **Responsable:** Líder Técnico.
- Preparar ambiente de desarrollo (aplicativos internos, aplicativos externos, data de prueba, instalación fuentes de programas requeridos).
Responsable: Desarrolladores.
- Crear carpeta de requerimiento de desarrollo en directorio compartido del respectivo grupo de trabajo (ej: activos) estructura de directorios de documentación del desarrollo. **Responsable:** Líder Técnico
- Envío de path de directorio del requerimiento. **Responsable:** Líder Técnico.
- Solicitud de fuentes a responsable versionamiento de código de AC de programas a utilizar para el requerimiento asignado. **Responsable:** Desarrolladores (java, .net y/o RPG).
- Entrega de fuentes. **Responsable:** Versionamiento de código de AC

- RPG Colocación de los componentes en el ambiente de desarrollo.
- JAVA/NET asignación de branch en svn y permisos acceso para Desarrolladores que se solicitó.
- Armar ambiente con fuentes entregados para instalar y configurar ambiente de proyecto. **Responsable:** Desarrolladores.
- Codificación a nivel de aplicación y/o base de datos de requisitos funcionales y no funcionales definidos. **Responsable:** Desarrolladores.
- Ejecutar y generar resultados pruebas técnicas/unitarias de funcionalidad implementada. **Responsable:** Desarrolladores.
- Entrega de manual de instalación y configuración que contiene una guía de instalación y la lista de componentes. **Responsable:** Líder Técnico + Desarrolladores.

Entregables:

- Inventario /Listado de componentes modificados y/o creados con su respectiva descripción e información técnica. **Responsable:** Líder Técnico + Desarrolladores.
- Notificar Listado de escenarios/casos de prueba adicionales considerados en pruebas técnicas/unitarias. **Responsable:** Líder Técnico + Desarrolladores.
- Listado consolidado escenarios/casos de prueba considerados en pruebas técnicas/unitarias de desarrollo. **Responsable:** Líder Técnico + Desarrolladores.

- Resultados pruebas técnicas/unitarias ejecutadas. **Responsable:** Líder Técnico + Desarrolladores.
- Diccionario Datos y diagrama entidad-relación base de datos actualizado (opcional). **Responsable:** Líder Técnico + Desarrolladores.
- Manual técnico de componentes de aplicación y base de datos (opcional). **Responsable:** Líder Técnico + Desarrolladores.
- Manual de instalación de solución para ambiente de QA. **Responsable:** Líder Técnico + Desarrolladores.

FASE 6 – Pruebas AC: En esta fase se asegurar la calidad y estabilizar el desarrollo implementado en el ambiente de pruebas previo a la certificación funcional de usuarios.

Responsable de la Fase: Analista Pruebas AC.

Participantes: Gerente Proyecto/Mantenimiento, Analista pruebas AC, Desarrolladores, Líder Técnico, Contra parte técnica terceros, Infraestructura (opcional), Redes (opcional), Usuarios de áreas involucradas (opcional), usuarios de áreas de control (opcional), Versionamiento de código de AC.

Actividades:

- Versionamiento fuentes de programas para QA y producción.
Responsable: Versionamiento de código de AC.
- Ejecutar pruebas funcionales que aseguren integridad de información y combinación de escenarios. **Responsable:** Analista Pruebas AC.
- Ejecutar pruebas funcionales definidas en inventario/matriz escenarios / casos de prueba. **Responsable:** Analista Pruebas AC.

- Validar, registrar y reportar novedades (errores/defectos). **Responsable:** Analista Pruebas AC.
- Estabilizar ambiente de pruebas. **Responsable:** Analista Pruebas AC.
- Brindar soporte y Preparar data de prueba. **Responsable:** Analista Pruebas AC.
- Capacitar y apoyar a usuarios en reporte de novedades y metodología de trabajo de ejecución pruebas funcionales. **Responsable:** Analista Pruebas AC.

Entregables:

- Ambiente de pruebas estable con requisitos de data, aplicaciones, software y hardware solicitado. **Responsable:** Analista Pruebas AC.
- Matriz consolidada actualizada de Listado de escenarios/casos de prueba. **Responsable:** Analista Pruebas AC.
- Guion de pruebas (detalle paso a paso) de cada escenario/caso de prueba. (opcional). **Responsable:** Analista Pruebas AC.
- Informe resultados de pruebas técnicas y funcionales realizadas. **Responsable:** Analista Pruebas AC.
- Confirmación/aprobación fase pruebas de calidad. **Responsable:** Analista Pruebas AC.

FASE 7 – Certificación: En esta fase se realiza la certificación funcional del desarrollo implementado en el ambiente de pruebas.

Responsable de la Fase: Gerente Proyecto/Mantenimiento.

Participantes: Gerente Proyecto/Mantenimiento, Analista pruebas AC, Desarrolladores, Líder Técnico, Áreas de control, Áreas usuarias, Contra parte técnica terceros, Infraestructura (opcional), Redes (opcional).

Actividades:

- Certificar funcionalmente la implementación correctamente todos los requisitos funcionales definidos en el documento de especificaciones.
Responsable: Áreas usuarias.
- Certificar funcionalmente el correcto comportamiento de los módulo y aplicativos relacionados con el desarrollo. **Responsable:** Áreas usuarias.
- Reportar y generar documentación de novedades (errores/defectos) que se detecten durante la certificación funcional. **Responsable:** Áreas usuarias y de control.
- Revisar y validar documentación de novedades (errores/defectos) con el analista de pruebas AC. **Responsable:** Áreas usuarias y de control + Gerente Proyecto/Mantenimiento.
- Validar solución de novedades reportadas por áreas usuarias.
Responsable: Áreas usuarias y control + Gerente Proyecto/Mantenimiento.
- Solucionar novedades técnicas reportadas por usuarios (errores/defectos). **Responsable:** Desarrolladores + Líder Técnico.
- Seguimiento de avances de usuarios en ejecución de certificación funcional. **Responsable:** Gerente Proyecto/Mantenimiento.
- Seguimiento de avances de atención de novedades reportadas al equipo de TI. **Responsable:** Gerente Proyecto/Mantenimiento.

- Seguimiento y validación de corrección novedades reportadas por usuarios. **Responsable:** Líder de Técnico.

Entregables:

- Matriz consolidada actualizada de Listado de escenarios/casos de prueba. **Responsable:** Analista de pruebas AC.
- Confirmación/aprobación certificación funcional por parte de cada área usuaria involucrada vía email. **Responsable:** Gerente Proyecto/Mantenimiento + Área usuarias + Áreas control.
- Entrega de documento de especificaciones funcionales y no funcionales final. **Responsable:** Gerente Proyecto/Mantenimiento.
- Entrega de documento de especificaciones técnicas final. **Responsable:** Líder Técnico.
- Entrega de los controles de cambios aprobados implementados. **Responsable:** Gerente Proyecto/Mantenimiento.
- Entrega consolidada de aprobaciones certificación funcional de usuarios. **Responsable:** Gerente Proyecto/Mantenimiento.
- Matriz de novedades reportada con respectivo responsable, resultado y estatus. (opcional). **Responsable:** Analista Pruebas AC + Líder Técnico + Gerente Proyecto/ Mantenimiento.

FASE 8 – Liberación: En esta fase se preparar liberación de desarrollo implementado a ambiente de producción.

Responsable de la Fase: Líder Técnico.

Participantes: Gerente Proyecto/Mantenimiento, Desarrolladores, Líder Técnico, Contra parte técnica terceros, Infraestructura (opcional), Redes (opcional), áreas usuarias, seguridad información, Versionamiento de código de AC.

Actividades:

- Coordinar y validar con partes involucrados actividades de liberación a producción. **Responsable:** Líder Técnico + Gerente Proyecto/Mantenimiento.
- Versionamiento de componentes/programas modificados y/o creados. **Responsable:** Versionamiento código AC.
- Entrega de documentación/habilitantes para paso a producción. **Responsable:** Líder Técnico.
- Confirmar fecha y hora de liberación a producción de desarrollo implementado. **Responsable:** Líder Técnico + Responsable Producción.
- Ejecutar liberación de programas/componentes modificados/creados a producción. **Responsable:** Líder Técnico + Desarrolladores + Responsable Producción.

Entregables:

- Certificar funcionalmente la implementación correctamente todos los requisitos funcionales definidos en el documento de especificaciones. **Responsable:** Áreas usuarias.
- Código fuente de programas modificados y/o creados. **Responsable:** Líder Técnico + Desarrolladores

- Scripts de bases de datos de actualización parámetros. **Responsable:** Líder Técnico + Desarrolladores.
- Versionamiento código de componentes aplicativos y base de datos. **Responsable:** Versionamiento de código de AC.
- Documento Especificaciones funcionales definitivas. **Responsable:** Gerente Proyecto/Mantenimiento.
- Documento Especificaciones técnicas definitivas. **Responsable:** Líder Técnico.
- Diseño solución técnico definitivo. **Responsable:** Líder Técnico + Desarrolladores.
- Inventario de componentes/programas modificados y/o creados. **Responsable:** Líder Técnico + Desarrolladores.
- Manual de instalación y operación definitivo. **Responsable:** Líder Técnico + Desarrolladores.
- Aprobaciones certificación funcional por parte áreas usuarias y áreas de control. **Responsable:** Líder Técnico
- Plan de implementación y Rollback. **Responsable:** Líder Técnico + Desarrolladores.

FASE 9 – Soporte postproducción: En esta fase se atiende novedades que se presenten en periodo de monitoreo una vez liberado el desarrollo en producción.

Responsable de la Fase: Líder Técnico.

Participantes: Gerente Proyecto/Mantenimiento, Desarrolladores, Líder Técnico, Contra parte técnica terceros, Infraestructura (opcional), Redes (opcional).

Actividades:

- Registro y seguimiento solución de novedades (errores/defectos) reportados en producción. **Responsable:** Líder Técnico + Gerente Proyecto/Mantenimiento.
- Corrección de novedades reportadas en producción. **Responsable:** Gerente Proyecto/Mantenimiento.
- Formalizar cierre de proyecto una vez finalizado periodo de monitoreo (pasar una quincena, fin de mes y/o fecha de ejecución funcionalidad). **Responsable:** Líder Técnico + Gerente Proyecto/Mantenimiento.

Entregables:

- Código de programas modificados y/o creados para corrección de novedades reportadas. **Responsable:** Líder Técnico + Desarrolladores.
- Acta o mail confirmación de cierre de proyecto. **Responsable:** Gerente Proyecto/Mantenimiento.

Gestion control de cambios: Durante el Ciclo de vida de desarrollo, se puede presentar cambios funcionales al alcance definido en el documento de especificaciones funcionales generado en la fase de análisis, incluso desde la fase de planificación.

Los controles de cambios pueden ser solicitados por alguno de los actores/partes involucradas en el proyecto sin embargo se debe tener las siguientes consideraciones:

Actividades:

- Generar documento de especificación funcional de control de cambios. **Responsable:** Gerente Proyecto/Mantenimiento.
- Aprobar control de cambios. **Responsable:** Gerente Proyecto/Mantenimiento.

- Estimar impacto técnico y tiempos implementación/desarrollo control de cambios **Responsable:** Líder Técnico + Desarrolladores.
- Confirmar impacto tiempos pruebas de aseguramiento calidad
Responsable: Analista de pruebas AC.
- Confirmar impacto tiempos pruebas de certificación funcional.
Responsable: Gerente Proyecto/Mantenimiento + Áreas Usuarías+ Áreas de Control.
- Actualizar cronograma de trabajo integral de desarrollo tecnológico.
Responsable: Líder Técnico + Gerente Proyecto/Mantenimiento.

Fase II: Identificar Oportunidades de Mejora

Dentro de las fases del ciclo de vida de desarrollo del software para Banco Internacional se han identificado las siguientes problemáticas y oportunidades de mejora:

FASE 1 – Entendimiento:

Problemática:

- Los Gerentes de Proyecto/Mantenimiento se definen en función del área de donde salen las iniciativas del negocio, en algunos casos pueden ser los Gerentes, Subgerentes o Jefes.
- Los Gerentes de Proyecto/Mantenimiento no tienen una visión integral de las necesidades del negocio y carecen de habilidades (blandas, duras), conocimiento funcional y bases técnicas requeridas para gestionar las responsabilidades actuales establecidas en el ciclo de vida de desarrollo de software.

- A pesar de existir responsabilidades claramente definidas dentro del ciclo de vida de desarrollo de software, en la práctica estas no son asumidas en su totalidad por el Gerentes de Proyecto/Mantenimiento (negocio) y Líder Técnico (Tecnología), generando una brecha alto de riesgo en la gestión del proyecto.
- No existe un equipo de trabajo multidisciplinario asignados al proyecto que pueda solventar de manera temprana las necesidades técnicas del Proyecto/Mantenimiento.
- No se cuenta con una herramienta para gestión de requerimientos que permita llevar la trazabilidad de los Proyectos/ Mantenimientos durante el ciclo de vida de desarrollo de Software.

Oportunidades de Mejora:

- Crear un nuevo acto dentro del ciclo de vida de desarrollo de software llamado “Integrador de las necesidades del Negocio-TI”, con habilidades bandas, técnicas, funcionales y de gestión cuya responsabilidad principal será generar sinergia, empatía e integración con el Gerente de Proyecto/Mantenimiento para garantizar la implementación eficiente de las necesidades del negocio.
- Establecer un equipos de trabajos multidisciplinarios asignados a los Proyectos/Mantenimientos conformado por Líder Técnico + Desarrolladores + Analistas de AC + Infraestructura (bajo demanda) y Redes (bajo demanda).
- Adquirir una herramienta para llevar la trazabilidad de la gestión de los requerimientos, con la finalidad de que el Gerente de

Proyecto/Mantenimiento, Integrador de las necesidades del Negocio -TI y el equipo multidisciplinario se encuentren informados de los avances.

- Crear el Backlog Priorizado del Producto.

FASE 2 – Análisis funcional:

Problemática:

- Existe poca participación e involucramiento de las áreas que serán las responsables y ejecutoras de las soluciones implementadas en el Proyectos/Mantenimientos.
- En muchas ocasiones el negocio no tiene claro lo que pide, lo que genera funcionalidades solicitadas ambiguas.
- No existe un método adecuado para realizar un levantamiento de las necesidades del negocio (requerimientos).
- Las áreas usuarias y áreas de TI trabajan por silos en la ejecución de Proyectos/Mantenimientos.
- No existe la participación de súper usuarios o usuarios con buenos conocimientos de los procesos del negocio para apoyar en esta fase.
- No existe el asesoramiento de TI al negocio para realizar el levantamiento de necesidades a un nivel detallado.
- En muchas ocasiones en esta fase no se cierre el alcance de los Proyectos/Mantenimientos, incluso se dejan abiertos detalles cruciales o transversales, que luego generan reproceso.

Oportunidades de Mejora:

- Establecer un método claro integrar para el levantamientos de necesidades del negocio utilizando *talleres de historias de usuario*, lo

cual permitirá tener un mayor entendimiento de las necesidades del negocio y poder definir criterios de aceptación, el entregable formal de estos talleres serán las historias de usuarios que describan la funcionalidad desde el punto de vista de los usuarios, fáciles de entender y procurando no dejar abiertos detalles cruciales o transversales de los Proyectos/Mantenimientos, que luego generan reproceso.

FASE 3 – Análisis y diseño técnico:

Problemática:

- En muchos casos se realiza análisis y diseños técnicos incompletos, ya que la fase de análisis funcional no están completamente cerrada o el levantamiento de necesidades funcionales es ambiguo.
- No se realiza análisis y diseño integrales, ya que Líderes Técnicos tiene especializaciones solo en algunos procesos del Banco y no se apoyan en los desarrolladores con experiencia en esta actividad.
- Los Diseños técnicos se plasman en función de la experiencia de los Líderes Técnicos, sin utilizar lineamientos de construcción de software y sin consulta/acordar con los desarrolladores experimentados asignados al Proyecto/Mantenimiento.
- No existe una participación formal y comprometida de las áreas de TI involucradas en el ciclo de vida de desarrollo de software.

Oportunidades de Mejora:

- Con base en las historias de usuario creadas se debe realizar reuniones de trabajo entre el Gente de Proyecto/Mantenimiento+ Integrador del Negocio-TI + equipo multidisciplinario con la finalidad de aclarar las

necesidades del negocio y estimar el esfuerzo requerido para implementar la funcionalidad descrita en cada historia de usuario.

- Con el equipo multidisciplinario realizar un análisis técnico detallado de los componentes necesarios para realizar la implementación integrar las historias de usuario.
- Identificar a un nivel detallado las tareas necesarias para implementar las historias de usuario, generando un Backlog para ser priorizado y gestionado.
- Realizar una planificación para los entregables, en ciclos cortos con la finalidad de entregar producto visible de manera temprana al negocio, con un plazo entre 4 y 6 semanas, sin embargo esta planificación debe ser fija en todo el Proyecto/Mantenimiento entre 4 a 6 semanas.

FASE 4 – Planificación:

Problemática:

- No se cumple los hitos planificados en las fechas comprometidas, en algunos casos los Líderes Técnicos justifican que las necesidades del negocio fueron ambiguas.
- Mala estimación de esfuerzos y de recursos requeridos, debido a que no existe un detalle claro de las necesidades del negocio a ser implementadas.

Oportunidades de Mejora:

- Las actividades ejecutadas en esta Fase se las puede realizar y mejorar en la Fase de **Análisis y Diseño Técnico**, de esta manera la **Planificación** se convierte en un entregable de la misma, por tal razón

La **Fase de Planificación** ya no sería parte del ciclo de vida de desarrollo de Software.

FASE 5 – Desarrollo:

Problemática:

- Analistas programadores no aplican buenas prácticas de programación en la construcción de código para el desarrollo de Software.
- No existen estándares de programación actualizados para los diferentes lenguajes de programación utilizados en Banco Internacional (RPG, JAVA, .Net).
- No existe lineamientos de Arquitectura y los de Seguridad se encuentran desactualizados.
- El código fuente de los programas son confuso y dificultan el mantenimiento de los mismos.
- No existe un procedimiento establecido para realizar pruebas unitarias o de programador de manera adecuada.
- No existe procesos establecidos para garantizar la calidad de los desarrollos desde el punto de vista técnico.
- No todos los analistas programadores cuentan con el conocimiento de los procesos que tiene el Banco que se requieren Implementar o realizar un mantenimiento.

Oportunidades de Mejora:

- Se debe aprovechar la experiencia y el conocimiento que tienen algunos Analistas Programadores claramente identificados dentro del equipo de Desarrollo.

- Mantener actualizado los estándares de programación para los diferentes lenguajes de programación existentes en Banco Internacional.
- Crear y mantener actualizado los lineamientos de Arquitectura y Seguridad (OWASP) para construcción de Software dentro del ciclo de vida de desarrollo.
- Realizar el aseguramiento de calidad (QA) al desarrollo de Software.
- Utilizar herramientas, que permitan garantizar las buenas prácticas de desarrollo como:
 - Eliminación de código repetitivo y redundante.
 - Separar los métodos y las funciones en rutinas más pequeñas.
 - Definir claramente las variables y los nombres de los métodos.
 - Simplificar el diseño del código.
 - Hacer que el código sea más fácil de entender y de modificar.
- La ejecución de pruebas unitarias de los desarrolladores con base en los criterios de aceptación de las historias de usuario.
- Preparar y ejecutar un plan de pruebas que permita validar los componentes de manera individual e integrada de la solución.
- Realizar un control y monitoreo diario de las asignaciones del equipo multidisciplinario.
- Refinamiento del Backlog Priorizado del Producto generado en fases anteriores.

FASE 6 – Pruebas AC:

Problemática:

- Existe personal nuevo y proveedores que no tienen conocimiento de los procesos del Banco Internacional, en la sentido la curva de aprendizaje es lenta.
- Interactúan muy tarde dentro del ciclo de vida de desarrollo “Pruebas de AC”, lo que genera reprocesos.
- No existen herramientas que permitan automatizar pruebas.

Oportunidades de Mejora:

- La actividades ejecutadas en esta Fase se las puede realizar en la Fase de **Desarrollo**, donde se aplicaría todo el aseguramiento de calidad (QA) en conjunto con el equipo de desarrollo y aprovechando las oportunidades de mejora de esta fase, por tal razón La **Fase de Pruebas AC** ya no sería parte del ciclo de vida de desarrollo de Software.

FASE 7 – Certificación:

Problemática:

- Los usuarios asignados como responsables de realizar la certificación de los Proyectos/Mantenimientos no tiene claro su alcance.
- Los usuarios asignados en esta etapa para la mayoría de Proyectos/Mantenimientos, no son usuarios expertos o dueños del producto y tampoco fueron parte de la fase de Análisis Funcional.

Oportunidades de Mejora:

- Esta Fase debe estar bajo la responsabilidad del Integrador de las necesidades del Negocio-TI, el será el responsable de coordinar y gestionar las actividades, con base en los criterios de aceptación

acordados para las historias de usuarios y la participación activa de los usuarios expertos y áreas de control.

- Demostrar y validar el entregable con el propósito de asegurar que se obtenga la aprobación y aceptación del Producto desarrollado.

FASE 8 – Liberación:

Problemática:

- No existe una área travesar autónoma, que congele, custodie y despliegue los componentes desarrollados en los diferentes ambientes del ciclo de vida desarrollo.
- En algunos casos los componentes entregados por QA, para la Fase de liberación en producción no son las últimas versiones probadas en las **Fases de Pruebas de QA y Certificación.**
- En esta Fase existe mucha dependencia del equipo de desarrollo para realizar la liberación en producción.
- No se tiene establecidos horarios fijos para ejecutar las liberaciones en producción, por tal motivo en ocasiones se afecta la disponibilidad de los servicios de TI.
- Se realiza una reunión de cambio cada 15 días que trata de organizar estas liberación en producción asignado una fecha de pase, sin embargo no se tiene una visión de manera integral los posibles impactos, colaterales y acciones de mitigación que se debe tener al moneo de liberar los Proyectos/Mantenimientos en producción.
- No existe un checklist de necesidades y actividades que se debe seguir para liberar los Proyectos/Mantenimientos en producción

Oportunidades de Mejora:

- Crear un nuevo acto dentro del ciclo de vida de desarrollo de software “Versionador de Componentes”, con la finalidad de que este sea quien congele, custodie y despliegue los componentes desarrollados en los diferentes ambientes del ciclo de vida desarrollo.
- Debe existir estrategia de despliegue de Proyectos/Mantenimientos en producción.
- Basados en la criticidad y disponibilidad de los servicios de TI se debe establecer un horario adecuado para realizar la implementación de Proyectos/Mantenimientos en producción.
- Se debe realizar una retrospectiva de la implementación de Proyectos/Mantenimientos en producción, con enfoque en lecciones aprendidas.

FASE 9 – Soporte postproducción:**Problemática:**

- Con la finalidad de solventar los incidentes que se pueda presentar en producción dependiendo de la criticidad, las correcciones realizadas en esta fase son parche, no definitivas y se queda así en producción.
- En algunas ocasiones al brindar el soporte se realizan ajustes a los componentes que corrigen el incidente pero se dañan otra funcionalidad.

Oportunidades de Mejora:

- Para cada Proyectos/Mantenimientos se debe establecer un plan de acción que solucione los incidentes, errores identificados en esta Fase, el mismo que tiene que garantizar una correcta implementación en

producción con soluciones definitivas que no afecten a otras funcionalidades, para lo cual se tiene que hacer una verificación del entregable antes de su liberación en producción.

Fase III: Fortalecimiento del Modelo Actual

Alcance General

El alcance general de la propuesta se base en dos marcos de referencia (frameworks) complementarios, COBIT que nos permitirá mediante su Dominio: **Construir, Adquirir e Implementar (BAI)** y la utilización de sus objetivos de gestión integrar de manera eficiente las fases del ciclo de vida de desarrollo actual y SCRUM como método adaptable, iterativo y colaborativo garantiza en cada una de las fases agilidad para entregar de manera temprana productos al negocio.

Lo que permitirá Fortalecer el Proceso de construcción de Software en Banco Internacional y mejorar la productividad.

Definición de Objetivos de Gestión basados en COBIT 2019.

Luego de identificar la problemática existen y oportunidades de mejora dentro del ciclo de vida de desarrollo de software para Banco Internacional, Utilizando el Dominio de Gestión de COBIT: **Construir, Adquirir e Implementar (BAI)**, que se encarga de la definición, adquisición e implementación de soluciones y su integración en los procesos de negocio, se ha definido 3 Objetivos de gestión que nos permitirán fortalecer la construcción de software en Banco Internacional.

- **BAI02** — Gestionar la definición de requisitos.
- **BAI03** — Gestionar la identificación y construcción de soluciones.
- **BAI07** — Gestionar la aceptación y la transición de los cambios de TI.

BAI02 — Gestionar la Definición de Requisitos: Este objetivo se encarga de identificar soluciones y analizar los requerimientos antes de la adquisición o creación asegurándose que estén alineados con los requerimientos estratégicos de la organización, de tal manera que cubran los procesos de negocios, aplicaciones, información/datos, infraestructura y servicios (ISACA, 2018).

BAI03 — Gestionar la Identificación y Construcción de Soluciones: Este objetivo se encarga de establecer y mantener soluciones alineadas con los requerimientos de la organización, contemplando el diseño desarrollo, compras/contratación y asociación con proveedores/fabricantes. Adicional a esto gestiona la configuración, preparación y ejecución de pruebas, así como también la gestión de requerimientos y mantenimiento de procesos de negocio, aplicaciones, datos/información, infraestructura y servicios (ISACA, 2018).

BAI07 — Gestionar la Aceptación y la transición de los cambios de TI: Este objetivo se encarga de hacer operativas y aceptar formalmente las nuevas soluciones, incluyendo la planificación de la implementación, la conversión de los datos y los sistemas, las pruebas de aceptación, la comunicación, la preparación del lanzamiento, el paso a producción de procesos de negocio o servicios TI nuevos o modificados, el soporte temprano en producción y una revisión post-implementación (ISACA, 2018).

A continuación se procede a definir en cada una de las Fase del ciclo de vida de desarrollo de software los *Objetivos y Practica de la Gestión* que serán utilizados.

FASE 1 – Entendimiento:

Objetivo: BAI02 — Gestionar la Definición de Requisitos.

Practica de Gestión:

BAI02.01 Definir y mantener los requisitos funcionales y técnicos del negocio: Mediante la utilización del caso de negocio se identifica, prioriza, especifica y

acuerda los requisitos de información, funcionales, técnicos y de control del negocio que permiten cubrir el alcance de todas las iniciativas necesarias para lograr los resultados esperados de la solución (ISACA, 2018).

FASE 2 – Análisis funcional:

Objetivo: BAI02 — Gestionar la Definición de Requisitos

Practica de Gestión:

BAI02.01 Definir y mantener los requisitos funcionales y técnicos del

negocio: Mediante la utilización del caso de negocio se identifica, prioriza, especifica y acuerda los requisitos de información, funcionales, técnicos y de control del negocio que permiten cubrir el alcance de todas las iniciativas necesarias para lograr los resultados esperados de la solución (ISACA, 2018).

BAI02.02 Realizar un estudio de factibilidad y formular soluciones

alternativas: Mediante un estudio de factibilidad identificar las posibles soluciones alternativas, evaluando su viabilidad y seleccionar la opción apropiada. Si es necesario realizar un piloto para determinar posibles mejoras (ISACA, 2018).

BAI02.04 Obtener la aprobación de requisitos y soluciones: Coordinar la retroalimentación con las partes interesadas en etapas claves predeterminadas, con la finalidad de obtener la aprobación y autorización del dueño del producto para los requisitos funcionales y técnicos, estudios de factibilidad (ISACA, 2018).

FASE 3 – Análisis y diseño técnico:

Objetivo: BAI03 — Gestionar la Identificación y Construcción de Soluciones.

Practica de Gestión:

BAI03.01 Diseño de soluciones de alto nivel: Desarrollar y documentar desde el punto de vista técnico diseños de alto nivel para la solución, procesos de negocio y flujos de trabajo, asegurándose que las partes interesadas participen y aprueben el diseño. También se utiliza técnicas de desarrollo por fases o Agiles acordadas y apropiadas para la organización. Asegurar la alineación con la estrategia de Tecnología y la arquitectura empresarial. Volver a evaluar y actualizar los diseños cuando se presenten problemas significativos durante las fases de diseño detallado o construcción (ISACA, 2018).

BAI03.02 Diseñar componentes detallados para la solución: Desarrollar, documentar y elaborar diseños detallados de forma progresiva. Utilizar técnicas de desarrollo Agile por fases o acordadas y apropiadas, abordando todos los componentes de la solución desde el punto de vista técnico y procesos. Asegurarse de que el diseño detallado incluya acuerdos de nivel de servicio (SLA) internos y externos, así como acuerdos de nivel operativo (OLA) (ISACA, 2018).

FASE 4 – Planificación:

Objetivo: BAI03 —Gestionar la Identificación y Construcción de Soluciones.

Practica de Gestión:

BAI03.04 Adquirir los componentes de la solución: Adquirir los componentes de la solución basados en el plan de adquisiciones, de acuerdo con los requisitos y diseños detallados, los principios y estándares de arquitectura, y los procedimientos generales de adquisición y contratos de la compañía, requisitos de QA y estándares de aprobación. Asegurarse de que el proveedor identifica y aborda todos los requisitos legales y contractuales (ISACA, 2018).

FASE 5 – Desarrollo:**Objetivo: BAI03 – Gestionar la Identificación y Construcción de Soluciones.****Practica de Gestión:**

BAI03.03 Desarrollar los componentes de la solución: Desarrollar de manera progresiva los componentes de la solución en un entorno independiente, de acuerdo con los diseños, estándares, requisitos de desarrollo, documentación y de aseguramiento de la calidad (QA). Asegurarse de que se abordan todos los requisitos de control, infraestructura en los procesos de negocio (ISACA, 2018).

BAI03.05 Construir soluciones: Instalar y configurar soluciones e integrarlas con las actividades del proceso del negocio. Durante esta ejecución se debe implementar medidas de control, seguridad, privacidad y auditabilidad para proteger los recursos y asegurar la disponibilidad y la integridad de los datos. Actualizar el catálogo de productos o servicios para reflejar las nuevas soluciones (ISACA, 2018).

BAI03.09 Gestionar los cambios a los requisitos: Realizar un seguimiento al estado de requisitos individuales (incluidos todos los requisitos rechazados) durante el ciclo de vida del desarrollo de Software. Gestionar la aprobación de cambios a los requisitos (ISACA, 2018).

FASE 6 – Pruebas AC:**Objetivo: BAI03 —Gestionar la Identificación y Construcción de Soluciones.****Practica de Gestión:**

BAI03.06 Realizar el aseguramiento de calidad (QA): Desarrollar, aprovisionar y ejecutar un plan de aseguramiento de la calidad (QA) alineado con el sistema de

gestión de la calidad (QMS), lo que permitirá obtener la calidad especificada en la definición de los requisitos, políticas y procedimientos de calidad de la organización (ISACA, 2018).

BAI03.07 Preparar las pruebas de la solución: Establecer un plan de pruebas y los ambientes necesarios para probarlos componentes de la solución de manera individuales e integrados. Incluir los procesos de negocio y los servicios de soporte, aplicaciones e infraestructura (ISACA, 2018).

BAI03.08 Ejecutar las pruebas de la solución: Durante el desarrollo, ejecutar pruebas de manera continua (incluidas pruebas de control), de acuerdo con el plan de pruebas definido y las prácticas de desarrollo en el ambiente apropiado. Incluir también a los dueños de los procesos de negocio y a los usuarios finales en el equipo de pruebas. Identificar, registrar y priorizar los errores y problemas que se identificaron durante las pruebas.

FASE 7 – Certificación:

Objetivo: BAI07 — Gestionar la aceptación y la transición de los cambios de TI.

Practica de Gestión:

BAI07.03 Plan de pruebas de aceptación: Establecer un plan de pruebas de aceptación aprobado por las partes interesadas, basado en estándares de toda la organización, el cual definirá roles, responsabilidades y criterios de entrada y salida (ISACA, 2018).

BAI07.04 Establecer un entorno de pruebas: Establecer un ambiente de pruebas seguro y representativo del proceso de negocio planificado y del ambiente de operaciones de TI, en cuanto a rendimiento, capacidad, seguridad, controles internos,

prácticas operativas, calidad de los datos, requisitos de privacidad y cargas de trabajo (ISACA, 2018).

BAI07.05 Realizar pruebas de aceptación: Realizar las pruebas de los cambios de forma independiente de acuerdo con el plan de pruebas de aceptación definido antes de la migración al ambiente operativo en producción (ISACA, 2018).

FASE 8 – Liberación:

Objetivo: BAI07 — Gestionar la aceptación y la transición de los cambios de TI.

Practica de Gestión:

BAI07.01 Establecer un plan de implementación: Establecer un plan de implementación aprobado por las partes interesadas, que cubra la conversión de sistemas, datos, criterios de pruebas de aceptación, comunicación, formación, preparación de puestas en producción, paso a producción, soporte temprano en producción, plan de fallback/backup y revisión posterior a la implementación (ISACA, 2018).

BAI07.02 Planificar la conversión de procesos de negocio, sistemas y datos: Prepararse para la migración de los procesos de negocio, datos de servicios e infraestructura de tecnología como parte de los métodos de desarrollo de la organización, incluyendo pistas de auditoría y un plan de recuperación si falla la migración (ISACA, 2018).

BAI07.06 Promover a producción y gestionar las liberaciones (releases): Promover la solución aceptada al negocio y a las operaciones, gestionar las liberaciones de los componentes. Cuando sea apropiado, ejecutar la solución como una implementación piloto o en paralelo con la solución antigua durante un período definido

y comparar el comportamiento y los resultados. Si se producen problemas significativos, volver al entorno original usando el plan de fallback/backup (ISACA, 2018).

FASE 9 – Soporte postproducción:

Objetivo: BAI07 — Gestionar la aceptación y la transición de los cambios de TI.

Practica de Gestión:

BAI07.07 Proporcionar soporte oportuno en producción: Durante un periodo de tiempo acordado, proporcionar un soporte oportuno a los usuarios y a las operaciones de tecnología para resolver problemas y ayudar a estabilizar la nueva solución en producción (ISACA, 2018).

BAI07.08 Realizar una revisión post-implementación: Realizar una revisión post-implementación para confirmar los resultados previstos por el usuario o cliente, adicional identificar las lecciones aprendidas y desarrollar un plan de acción. Evaluar el rendimiento y los resultados reales del servicio nuevo o modificado, en comparación con el rendimiento (ISACA, 2018).

La tabla 9, indica un resumen de las Fases del ciclo de vida desarrollo y un mapeo con los Objetivos COBIT que serán utilizados para fortalecen la construcción de Software en Banco Internacional.

Tabla 9

Resumen fases del CVD con objetivos Cobit

Ciclo de vida de desarrollo	Cobit	
Fase	Objetivos	Practica de Gestión
Entendimiento	BAI02 — Gestionar la definición de requisitos	BAI02.01 Definir y mantener los requisitos funcionales y técnicos del negocio.
Análisis Funcional		BAI02.01 Definir y mantener los requisitos funcionales y técnicos del negocio. BAI02.02 Realizar un estudio de factibilidad y formular soluciones alternativas. BAI02.03 Gestionar el riesgo de los requisitos. BAI02.04 Obtener la aprobación de requisitos y soluciones.
Análisis y Diseño Técnico	BAI03 — Gestionar la identificación y construcción de soluciones	BAI03.01 Diseño de soluciones de alto nivel.
Planificación		BAI03.02 Diseñar componentes detallados para la solución.
Desarrollo		BAI03.04 Adquirir los componentes de la solución BAI03.03: Desarrollar los componentes de la solución. BAI03.05 Construir soluciones. BAI03.09 Gestionar los cambios a los requisitos.
Pruebas AC	BAI03 — Gestionar la identificación y construcción de soluciones	BAI03.06 Realizar el aseguramiento de calidad (QA)
		BAI03.07 Preparar las pruebas de la solución. BAI03.08 Ejecutar las pruebas de la solución.
Certificación	BAI07 — Gestionar la aceptación y la transición de los cambios de TI	BAI07.03 Plan de pruebas de aceptación.
		BAI07.04 Establecer un entorno de pruebas. BAI07.05 Realizar pruebas de aceptación.
Liberación		BAI07.01 Establecer un plan de implementación. BAI07.02 Planificar la conversión de procesos de negocio, sistemas y datos. BAI07.06 Promover a producción y gestionar las liberaciones (releases).
Soporte Postproducción		BAI07.07 Proporcionar soporte oportuno en producción. BAI07.08 Realizar una revisión post-implementación.

Despliegue de Procesos basados en SCRUM

Una vez definidos los objetivos de gestión de COBIT, Utilizando SCRUM como método adaptable, iterativo, rápido, flexible y eficaz, que garantiza transparencia en la comunicación y crea un ambiente de responsabilidad colectiva para crear un nuevo producto, se ha definido 16 Procesos a nivel de ejecución con sus respectivas entradas, salida y herramientas aplicables a la realidad de Banco Internacional que nos permitirán fortalecer el proceso de construcción de software.

A continuación se procede a definir en cada una de las Fase del ciclo de vida de desarrollo de software los Procesos que serán utilizados.

FASE 1 – Entendimiento:

Proceso: Crear la Visión del Proyecto.- En este proceso se crea la visión del proyecto (Project Vision Statement), la misma que proporcionará un enfoque de todo el proyecto a partir del caso de negocio existente (SCRUMstudy™, 2017).

Entrada:

- Caso de negocio del proyecto.

Salida:

- Declaración de la visión del proyecto.

Herramienta:

- Reunión de visión del proyecto.

Proceso: Formar Equipos Scrum.- En este proceso el Product Owner seleccionan a los miembros del Equipo Scrum con la colaboración del Scrum Master (SCRUMstudy™, 2017).

Entrada:

- Product Owner.
- Scrum Master.

Salida:

- Equipo Scrum identificado.

Herramienta:

- Selección del equipo Scrum.

Proceso: Crear el Backlog Priorizado del Producto.- En este proceso la épica(s) es refinada, detallada y luego priorizada para armar el Backlog Priorizado del Producto del proyecto, estableciendo también los criterios de terminado (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Épica(s).

Salida:

- Backlog Priorizado del Producto.

Herramienta:

- Métodos de priorización de historias del usuario.

FASE 2 – Análisis funcional:

Proceso: Crear historias de usuario.- En este proceso se crean las historias de usuario y los criterios de aceptación. Las historias de usuario se diseñan para garantizar que los requerimientos del cliente estén claramente representados y que todos los interesados las puedan entender completamente, adicional se incorporan al Backlog Priorizado del Producto (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Backlog Priorizado del Producto.
- Declaración de la visión del proyecto.

Salida:

- Historias de usuarios.
- Criterio de aceptación de historias del usuario.

Herramienta:

- Experiencia en la redacción de historias de usuario.
- Talleres de historias del usuario.

FASE 3 – Análisis y diseño técnico:

Proceso: Estimar historias de usuario.- En este proceso el Product Owner aclara las historias de usuario con la finalidad de que el Scrum Master y el Equipo Scrum hagan una estimación sobre el esfuerzo necesario para desarrollar la funcionalidad requerida (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Historias de usuarios.

Salida:

- Historias de usuarios estimadas.

Herramienta:

- Métodos de estimación.

Proceso: Comprometer historias de usuario.- En este proceso el Equipo Scrum se compromete a entregar al Product Owner las historias de usuario comprometidas para un sprint (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Historias del usuario estimadas.
- Duración del sprint.

Salida:

- Historias de usuarios comprometidas.

Herramienta:

- Reuniones de planificación del sprint.

FASE 4 – Planificación:

Proceso: Identificar tareas.- En este proceso las historias de usuario comprometidas se dividen en tareas específicas y se almacenan en una lista de tareas (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Historias de usuarios comprometidas.

Salida:

- Historias de usuarios comprometidas actualizadas.

Herramienta:

- Reuniones de planificación del sprint.

Proceso: Estimar tareas.- En este proceso el equipo principal de Scrum estima el esfuerzo necesario para completar cada tarea en la lista de tareas (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Lista de tareas.

Salida:

- Effort Estimated Task List

Herramienta:

- Reuniones de planificación del sprint.

- Métodos de estimación.

Proceso: Crear el Sprint Backlog.- En este proceso el equipo principal de Scrum lleva a cabo reuniones de planificación del sprint donde el grupo elabora un Sprint Backlog que contiene todas las tareas a ejecutar en el sprint (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Effort Estimated Task List.
- Duración del sprint.

Salida:

- Sprint Backlog.

Herramienta:

- Reuniones de planificación del sprint.

FASE 5 – Desarrollo:

Proceso: Crear entregables.- En este proceso el Equipo Scrum trabaja en las tareas del Sprint Backlog para crear los entregables del sprint. Los problemas que enfrenta el Equipo Scrum pueden actualizarse en el Impediment Log (o registro de impedimentos) (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Sprint Backlog.
- Scrumboard.

Salida:

- Entregables del sprint.
- Scrumboard actualizado.

Herramienta:

- Experiencia del equipo.
- Otras herramientas de desarrollo.

Proceso: Realizar Daily Standup.- En este proceso se lleva a cabo diariamente una reunión altamente focalizada con un time-box asignado. Es un foro para que el Equipo Scrum se ponga al día sobre sus progresos y sobre cualquier impedimento que pudieran estar enfrentando en la ejecución del Sprint (SCRUMstudy™, 2017).

Entrada:

- Equipo Scrum.
- Scrum Master.
- Sprint Burndown Chart.

Salida:

- Sprint Burndown Chart actualizada

Herramienta:

- Daily Standup.
- Tres preguntas diarias.

Proceso: Refinamiento del Backlog Priorizado del Producto.-En este proceso constantemente se actualiza y refina el Backlog Priorizado del Producto. En función de los cambios y priorizaciones existentes en el momento (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum.
- Backlog Priorizado del Producto.

Salida:

- Backlog Priorizado del Producto actualizado

Herramienta:

- Reunión de revisión del Backlog Priorizado del Producto.

FASE 6 – Pruebas AC y FASE 7 – Certificación.

Proceso: Demostrar y validar el sprint.- En este proceso el Equipo Scrum demuestra los entregables del sprint al Product Owner y a los interesados. El objetivo de esta actividad es lograr la aprobación y aceptación del Product Owner respecto al producto o servicio (SCRUMstudy™, 2017).

Entrada:

- Equipo principal de Scrum
- Entregables del sprint.
- Sprint Backlog.
- Criterios de aceptación.

Salida:

- Entregables aceptados.

Herramienta:

- Reuniones de revisión del Sprint.

Proceso: Retrospectiva del sprint.-En este proceso el Scrum Master y el Equipo Scrum se reúnen para discutir las lecciones aprendidas durante el sprint. Dicha información se documenta con la finalidad de poder utilizarse en futuros sprints (SCRUMstudy™, 2017).

Entrada:

- Equipo Scrum.
- Scrum Master.
- Salidas de Demostrar y validar el sprint.

Salida:

- Agreed Actionable Improvements.

Herramienta:

- Reunión de retrospectiva del Sprint.

FASE 8 – Liberación.

Proceso: Enviar entregables.- En este proceso se hace la entrega o la transición de los entregables aceptados a los interesados. La conclusión satisfactoria del sprint se documenta en un acuerdo de entregables funcionales (SCRUMstudy™, 2017).

Entrada:

- Product Owner
- Interesados.
- Entregables aceptados.

Salida:

- Acuerdo de entregables funcionales

Herramienta:

- Métodos de desplazamiento organizacional.

Proceso: Retrospectiva del proyecto.-En este proceso se concluye el proyecto, los interesados de la organización y los miembros del equipo principal de Scrum se reúnen para hacer una retrospectiva del proyecto e identificar, documentar e internalizar las lecciones aprendidas, las mismas que serán consideradas para los futuros proyectos.

Entrada:

- Equipo principal de Scrum
- Interesados.

Salida:

- Mejoras Aceptadas (agreed actionable improvements).

Herramienta:

- Reunión de la retrospectiva del proyecto.

La tabla 10, indica un resumen de las Fases del ciclo de vida desarrollo y un mapeo con los Procesos Fundamentales de SRCUM que serán utilizados para fortalecer la construcción de Software en Banco Internacional.

Tabla 10

Resumen fases del CVD con procesos fundamentales Scrum.

Ciclo de vida de desarrollo	Srcum	
<u>Fase</u>	<u>Fase</u>	<u>Procesos fundamentales</u>
Entendimiento	Inicio	<ul style="list-style-type: none"> • Crear la visión del proyecto. • Formar Equipos Scrum. • Crear el Backlog Priorizado del Producto.
Análisis Funcional Análisis y Diseño Técnico Planificación	Planificación y Estimación	<ul style="list-style-type: none"> • Crear historias de usuario. • Estimar historias de usuario. • Comprometer historias de usuario. • Identificar tareas. • Estimar tareas.
Desarrollo	Implementación	<ul style="list-style-type: none"> • Crear el Sprint Backlog. • Crear entregables. • Realizar Daily Standup. • Refinamiento del Backlog Priorizado del Producto.
Pruebas AC Certificación Liberación	Revisión y Retrospectiva Lanzamiento	<ul style="list-style-type: none"> • Demostrar y validar el sprint. • Retrospectiva del sprint. • Enviar entregables. • Retrospectiva del sprint.
Postproducción		<ul style="list-style-type: none"> • Retrospectiva del proyecto.

Presentación del Modelo Integral Propuesto

El presente modelo integral propuesto se enfoca en fortalecer el proceso de construcción de software en Banco Internacional, el mismo que actualmente tiene una metodología para el desarrollo de software basada en las mejores prácticas del marco referencial MSF (Microsoft Solution Framework), para lo cual mediante la utilización de COBIT y apalancado en los objetivos de gestión del dominio **(BIA)** nos permitirá reestructurar las fase del ciclo de vida de desarrollo actual, buscando una optimización a nivel de proceso. Por otro lado SCRUM como método adaptable a las necesidades de la organización, iterativo y colaborativo aportará en cada una de las fases con entradas, salidas y herramientas que permitirá agilizar la entrega de producto al negocio.

Estrategia y esquema de trabajo a utilizar: En el proceso de construcción software propuesto para Banco Internacional, se establecer la creación de 2 nuevos roles relevantes dentro del ciclo de vida de desarrollo:

- Integrador de las necesidades del Negocio-TI.
- Versionador de componentes.

Los mismos que contribuirán con la integración y eficiencia del ciclo de vida de desarrollo, que mediante la utilización de una estrategia de desarrollo iterativo generaran entregables graduales con valor al negocio.

Con la finalidad de fortalecer la metodología de trabajo actual se propone los siguientes objetivos:

- Establecer sinergia, empatía entre el Negocio y Tecnología para la gestión de eficiente de Proyectos/Mantenimientos, trabajando para alcanzar objetivos comunes.

- Mediante un esquema de priorización claramente definido con el negocio, asegurando que los requerimientos que general mayor valor sean los primeros en liberarse.
- Establecer un proceso iterativo de entregas continuas de valor al negocio tan frecuente como este lo requiera.
- Dividir el trabajo en ciclos de trabajos cortos y necesarios para la creación de entregables, incluyendo desarrollo, pruebas y garantizando la calidad del producto.

Para la ejecución de todo desarrollo de software se establecen las siguientes consideraciones:

- Se asigna un Integrador de las necesidades del Negocio-TI, quien será el encargado de generar sinergia, empatía con el Gerente de Proyecto/Mantenimiento y velara por la implementación del producto desde el punto de vista técnico de manera integral.
- Definir el equipo de trabajo multidisciplinario asignado a la ejecución del Proyecto/Mantenimiento en función de la experiencia y especialización que tengan los colaboradores de Tecnología, este equipo estará conformado por Líder Técnico + Desarrolladores + Analistas de pruebas AC + Infraestructura (bajo demanda) y Redes (bajo demanda).
- Basado en las necesidades del negocio “Doc de Especificaciones funcionales”, se creara un Backlog Priorizado del Producto, utilizando el siguiente método de priorización en el orden indicado:
 - Considera funcionalidad principal del desarrollo que represente la parte medular del proceso/servicio/producto a mejorar/implementar, la misma que no es factible ser manejada operativamente.

- Considera funcionalidad complementaria del desarrollo que representa reglas del negocio, controles, excepciones del proceso/ servicio/ producto a mejor/ implementar, la funcionalidad a implementarse puede ser manejada operativamente.
- Considera temas de forma tales como mejoras adicionales de navegabilidad o presentación, consultas y reportes complementarios, la funcionalidad a implementarse puede ser manejada operativamente.
- Se procede a crear de historias de usuario, con la finalidad de garantizar que las necesidades del negocio estén claramente identificadas y entendibles, adicional a esto se deben generar los criterios de aceptación que son las condiciones que satisfacer las historias de usuario y permiten aprobar o rechaza la funcionalidad, todo esto mediante la ejecución de reuniones o talleres de creación de historias, posteriormente las historias de usuario y criterios de aceptación serán incluidas en el Backlog Priorizado del Producto.
- Se realiza la reunión de planificación del sprint, donde el Gerente de Proyecto/Mantenimiento en conjunto con el Integrador de las necesidades del Negocio-TI, son los encargados de aclarar y explicar las historias de usuarios del Backlog Priorizado del Producto en el orden de priorización al equipo multidisciplinario con la finalidad que estos puedan realizar las siguientes actividades:
 - Estimar historias de usuario, utilizando métodos de estimación como, juicio de experto, planning poker (hasta llegar a un acuerdo).

- Identificar y estimar tareas en función del esfuerzo necesario desde el punto de vista personas y recursos para implementar los entregables asociados a las historias de usuarios comprometidas, lo que nos permitirá crear el Sprint Backlog que posteriormente será implementado en producción.
- La duración de un Sprint se establecer entre 2 o 6 semanas esto dependerá del alcance del Proyecto/Mantenimiento, importante considerar que una vez determinada la duración del sprint esta se mantiene igual para todos los sprint definidos, lo que permitirá generar la planificación del Proyecto/Mantenimiento.

Durante la ejecución de cada sprint se considera las siguientes estrategias:

- **Construcción.-** El equipo multidisciplinario al crear los entregables deberá tener las siguientes consideraciones:

Desarrolladores

- Utilizar estándares de programación para los diferentes lenguajes de programación existentes en Banco Internacional.
- Considerar y aplicar los estándares de Arquitectura y Seguridad de la Información para desarrollos de Software.
- Realizar pruebas unitarias a los componentes con base a los Criterios de aceptación de las historias de usuario.
- Realizar pruebas de vulnerabilidad SQL injection, Denegación de servicios entre otros basado en la Guía de Pruebas OWASP.

Analistas de AC

- Realizar el aseguramiento de calidad (QA) a los entregables.

- Utilizar herramientas que permitan verificar y garantizar las buenas prácticas a nivel de desarrollo:(Eliminación de código repetitivo y redundante, Separar los métodos y las funciones en rutinas más pequeñas, Definir claramente las variables y los nombres de los métodos, Simplificar el diseño del código, Hacer que el código sea más fácil de entender y de modificar).
- Preparar ambiente con los prerrequisitos necesarios para ejecutar las pruebas de la solución
- Reportar incidentes encontrados para su resolución.

El equipo multidisciplinario realizara reuniones diarias de máximo 15 minutos con la finalidad de hacer un seguimiento a las actividades realizadas y asignar nuevas para todos los miembros del equipo, esto permitirá actualiza y refina el *Backlog Priorizado del Producto* para futuras planificaciones en los siguientes sprints, el Integrador de las necesidades del Negocio-TI será el encargado de mantener continuamente informado al Gerente de Proyecto/Mantenimiento sobre cualquier cambio que se pueda presentar durante la ejecución del sprints y en el *Backlog Priorizado del Producto*.

Por otro lado en la construcción se debe garantizar la detección y corrección temprana de errores a nivel de componentes así como también los criterios de aceptación para las historias de usuarios con la finalidad de realizar entregas validadas al ambiente de Certificación.

- **Certificación.-** El Integrador de las necesidades del Negocio-TI en conjunto con el Gerente de Proyecto/Mantenimiento, serán los responsable de coordinar y gestionar esta actividad, con base en los criterios de aceptación acordados para las historias de usuarios y con

la participación activa de los usuarios expertos y áreas de control, con la finalidad de demostrar y validar funcionalmente el sprint para lograr la aprobación y aceptación del respecto al producto que será entregando al negocio, posterior a esto realizar el paso de las solución a Liberación.

- **Liberación.-** Realizar el paso a producción o la transición de los entregables aceptados para cada sprint del Proyecto/Mantenimiento, luego de cada liberación el equipo multidisciplinario se reunirá para discutir las lecciones aprendidas durante la ejecución del sprint, las mismas servirán para aplicar en las siguientes liberaciones.
Una vez liberado el último sprint en producción, el Gerente de Proyecto/Mantenimiento reúne a los interesados y a todo el equipo de trabajo para hacer una retrospectiva del Proyecto/Mantenimiento identificando, documentando e interiorizando las lecciones que se aprendieron, con la finalidad de implementarlas en futuros Proyecto/Mantenimiento.
- **Postproducción.-** Realizar durante los 2 primeros meses de liberación un soporte oportuno a los usuarios y la operación del producto o servicio entregado al negocio, solventado problemas, incidentes y errores que se puedan presentar en este periodo, con la finalidad de garantizar su estabilidad, realizando ajustes o correcciones definitivas.

Roles y funciones: Para fortalecer el proceso de construcción software en Banco Internacional se ha realizado algunos ajustes sobre los Roles y Responsabilidades existente apalancados en la creación de nuevos roles, con la finalidad de logra una integración eficiente entre las necesidades del negocio y

tecnología la misma que durante la ejecución del ciclo de vida de desarrollo permitirá generaran entregables graduales con valor al negocio.

Todo nuevo desarrollo tendrá un Gerente de Proyecto/Mantenimiento un Integrador de las necesidades del Negocio-TI quienes en conjunto serán los responsables de manera integral de garantizar la implementación del proyecto desde el punto de vista técnico, apoyados por el equipo multidisciplinario, usuarios expertos, áreas de control y versionamiento.

La tabla 11, indica un resumen de los roles, funciones modificados y nuevos de los participantes en la Metodología del Ciclo de Vida para el Desarrollo de Software.

Tabla 11

Roles y funciones ciclo de vida de desarrollo de software.

Roles	Funciones
Gerente de Proyecto/Mantenimientos	<ul style="list-style-type: none"> • Identificar y coordinar con áreas de control y usuarias su participación con responsabilidad en la ejecución del Proyecto/Mantenimiento. • Gestionar en conjunto con Integrador de las necesidades del Negocio-TI de manera integral la implementación de las necesidades del negocio durante la ejecución de ciclo de vida de desarrollo de software. • En conjunto con el Integrador de las necesidades del Negocio-TI serán los encargados brindar soporte a las necesidades de los equipos multidisciplinarios y ser los facilitador para solucionar novedades, conflictos entre actores del Proyecto/Mantenimiento. • Con base a las necesidades del negocio definir el <i>Backlog Priorizado del Producto</i> y acordar los criterios de aceptación de las historias de usuarios.

Roles	Funciones
Integrador de las Necesidades del negocio-TI	<ul style="list-style-type: none"> • Aprobar el <i>Sprint Backlog</i>, su durabilidad y la planificación para la implementación del mismo basado en sprints. • Gestionar las convocatorias de certificación funcional con áreas usuarias involucradas, previa coordinación con el Integrador de las necesidades del Negocio-TI. • Seguimiento, confirmación y aprobación de las certificaciones funcionales realizadas por las diferentes áreas usuarias y de control involucradas. • Mantener adecuadamente informados de los avances a los interesados. • Administrar contratos de proyectos con Terceros (proveedores, clientes). • Apoyado de sus habilidades (técnicas, funcionales y de gestión), será el responsable de generar sinceraría y buscar objetivos comunes con el Gerente de Proyecto/Mantenimiento para garantizar la implementación de manera eficiente del proyecto durante todo el ciclo de vida del desarrollo de software. • En función de su experiencia técnica y el alcance del Proyecto/Mantenimiento será el encargado identificar la cantidad de colaboradores y los perfiles necesarios de Tecnología que conformarán el equipo multidisciplinario asignado al proyecto. • Apoyar al Gerente de Proyecto/Mantenimiento en el proceso de <i>Priorizado Backlog Producto</i> y criterios de aceptación historias de usuario. • En conjunto con el Gerente de Proyecto/Mantenimiento será el responsable de aclarar y explicar las historias de usuarios del <i>Backlog Priorizado del Producto</i> al equipo multidisciplinario.

Roles	Funciones
	<ul style="list-style-type: none"> • Coordina la creación del <i>Sprint Backlog</i> lo que permitirá establecer el cronograma de planificación del Proyecto/Mantenimiento. • Realizar reuniones diarias de seguimiento al equipo multidisciplinario con la finalidad de revisar el estado de las asignaciones realizadas y planificadas. • Vocero oficial encargado de mantener continuamente informado al Gerente de Proyecto/Mantenimiento sobre cualquier cambio que se pueda presentar durante la ejecución de los <i>Sprints</i> y en el <i>Backlog Priorizado del Producto</i>. • En conjunto con el Gerente de Proyecto/Mantenimiento, serán los responsable de coordinar y gestionar con los usuarios expertos la certificación funcional.
<p>Equipo Multidisciplinario</p>	<p><u>Todos</u></p> <ul style="list-style-type: none"> • Apoyar al Líder técnico en realizar la estimación de las historias de usuario. • Apoyar al Líder técnico identificando y estimando tareas en función del esfuerzo necesario desde el punto de vista personas y recursos para implementar los entregables. • Asistir a la reuniones de seguimiento diarios con los insumos necesarios e identificando problemáticas que no permitan avanzar en la planificación. • Interiorizar las lecciones aprendidas de cada sprint y Proyecto/ Mantenimiento liberado. <p><u>Desarrolladores</u></p> <ul style="list-style-type: none"> • Responsables de construir las entregas utilizando los estándares de programación, arquitectura y seguridad de la información existentes en Banco Internacional. • Realizar pruebas unitarias a los componentes con base a los criterios de aceptación de las historias de usuario.

Roles	Funciones
	<ul style="list-style-type: none"> • Garantizar los insumos requeridos (Hoja Pase, componentes listos) por versionamiento una vez finalizado el desarrollo para realizar el despegue al ambiente que le corresponda. <p><u>Analista de QA</u></p> <ul style="list-style-type: none"> • Preparar ambiente con los prerrequisitos necesarios para ejecutar las pruebas de funcionalidad a los entregables. • Realizar el aseguramiento de calidad (QA) a los entregables. • Utilizar herramientas que permitan verificar y garantizar las buenas prácticas a nivel de desarrollo • Garantizar la detección temprana de errores a nivel de componentes así como también los criterios de aceptación para las historias de usuarios. • Reportar incidentes encontrados para su corrección. <p><u>Infraestructura</u></p> <ul style="list-style-type: none"> • Responsable de proveer y administrar los componentes de infraestructura necesarios para implementar los Proyectos/Mantenimientos. • Apoyar al Líder Técnico bajo demanda durante el ciclo de vida de desarrollo. <p><u>Redes</u></p> <ul style="list-style-type: none"> • Responsable garantizar los enlaces de comunicación que hay entre los diversos componentes y con terceros para implementar los Proyectos/Mantenimientos. • Apoyar al Líder Técnico bajo demanda durante el ciclo de vida de desarrollo.
Versionamiento	<ul style="list-style-type: none"> • Realizar una adecuada gestión en el control de versiones con la finalidad de custodiar los componentes entregados por construcción que serán desplegados en todos los ambientes del ciclo vida de desarrollo. • Garantizar que todos los ambientes definidos en el ciclo de vida de desarrollo para los Proyectos/Mantenimientos se

Roles	Funciones
	<p>encuentren operativos y unificados con las últimas versiones de los componentes puestos en producción.</p> <ul style="list-style-type: none"> Garantizar que los cambios aprobados por el usuario experto durante la ejecución del Proyecto/Mantenimiento sean los mismos que se liberen en producción.

Fases ciclo de vida desarrollo de software:

La figura 12, indica las fases del Ciclo de vida de desarrollo de Software disminuidas a 7 fases y propuesta para Banco Internacional.

Figura 12

Fases propuestas para ciclo de vida desarrollo de software.



FASE 1 – Entendimiento: En esta fase se realiza la generación del requerimiento funcional de alto nivel basado en el caso de negocio, lo que permitirá crear la visión del Proyecto/Mantenimiento que servirá de inspiración y proporcionará un solo objetivo para todos los interesados.

Responsable de la Fase: Gerente Proyecto/Mantenimiento.

Participantes: Gerente Proyecto/Mantenimiento, Tecnología, Integrador de las necesidades del Negocio-TI, áreas usuarias involucradas.

Actividades:

- Identificar la necesidad del negocio (mejora, corrección u oportunidad), visión de Proyecto/Mantenimiento. **Responsable:** Gerente Proyecto/Mantenimiento.
- Asignación formal del Integrador de las necesidades del Negocio-TI. **Responsable:** Tecnología.
- Definir el equipo de trabajo multidisciplinario asignado a la ejecución del Proyecto/Mantenimiento en función de la experiencia y especialización que tengan los colaboradores de Tecnología. **Responsable:** Gerente Proyecto/ Mantenimiento + Integrador de las necesidades del Negocio-TI.
- Identificar las partes involucradas directa e indirectamente con el Proyecto/Mantenimiento solicitado. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.
- El requerimiento funcional de alto nivel, se refina, detalla y luego se priorizada para el *Backlog Priorizado del Producto*. **Responsable:** Gerente Proyecto/ Mantenimiento.+ Integrador de las necesidades del Negocio-TI.

Entregables:

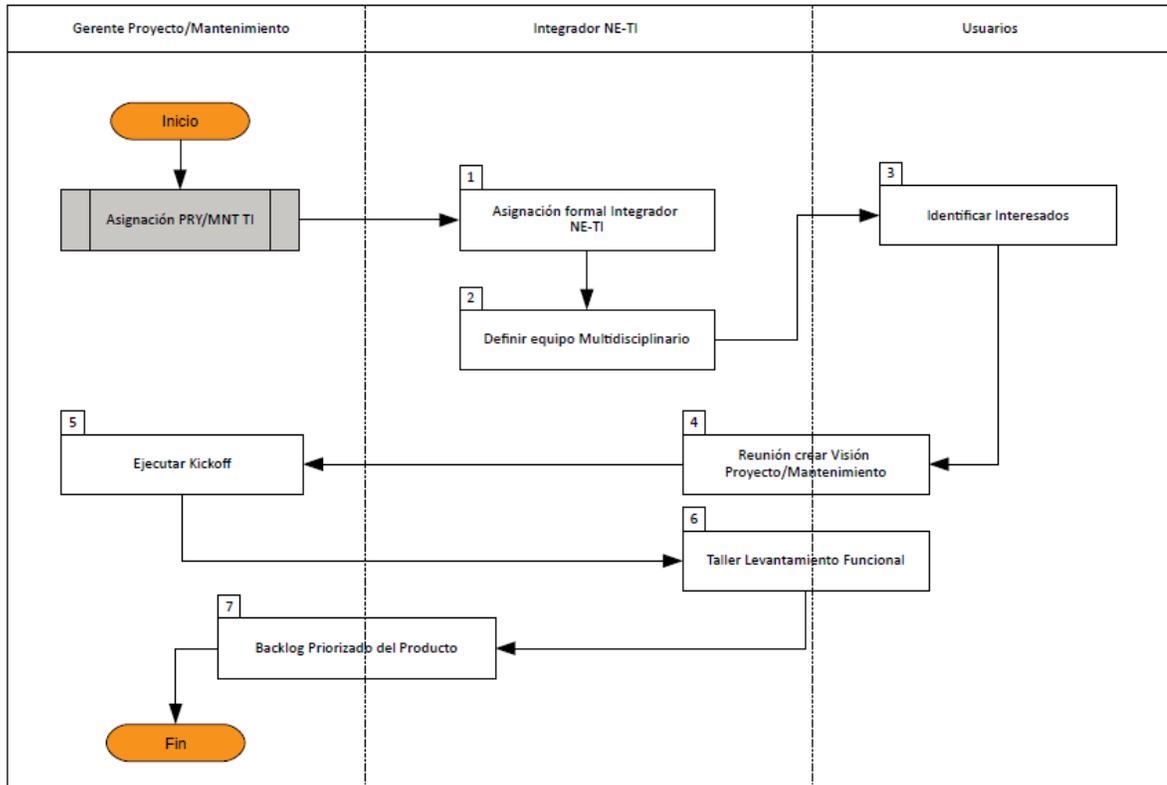
- Documento de evaluación de requerimiento (DER) – aplica solo en proyectos. **Responsable:** Gerente Proyecto/Mantenimiento+ Integrador de las necesidades del Negocio-TI.

- Documento Inicial de Visión del proyecto. – aplica solo en proyectos.
Responsable: Gerente Proyecto/ Mantenimiento.
- Presentación Kick off de inicio – aplica solo en proyecto (opcional).
Responsable: Gerente Proyecto/ Mantenimiento.
- Número Ticket Mesa de ayuda - aplica solo en Mantenimientos
Responsable: Gerente Proyecto/ Mantenimiento.
- Matriz de involucrados roles y responsabilidades – aplica solo en proyecto. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.
- Requerimiento funcional (necesidades del Negocio) inicial.
Responsable: Gerente Proyecto/ Mantenimiento.
- *Backlog Priorizado del Producto.* **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.

Diagrama de Flujo: La figura 13, indica el flujo de manera general Fase de Entendimiento.

Figura 13

Diagrama de flujo fase de Entendimiento.



FASE 2 – Análisis funcional: En esta fase se recopila información adicional y se obtiene el detalle de la funcional requerida para la construcción, con la finalidad de garantizar que las necesidades del negocio estén claramente identificadas y entendibles para todos los interesados del Proyecto/Mantenimiento.

Responsable de la Fase: Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI.

Participantes: Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI, equipo multidisciplinario, áreas usuarias involucradas directa e indirectamente en el proceso, áreas de control, seguridad información y contra parte técnica de terceros involucrados.

Actividades:

- Recopilar la mayor cantidad de información posible. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.
- Crear las historias de usuario basado en el *Backlog Priorizado del Producto*, mediante talleres de historias de usuario que permitirán garantizar que los requerimientos del negocio estén claramente entendibles para todos los interesados del Proyecto/Mantenimiento. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.
- Diseñar los criterios de aceptación, que delimitaran explícitamente las condiciones que deben satisfacer cada una de las historias de usuario, lo que permitirá aceptar y negar la funcionalidad. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.

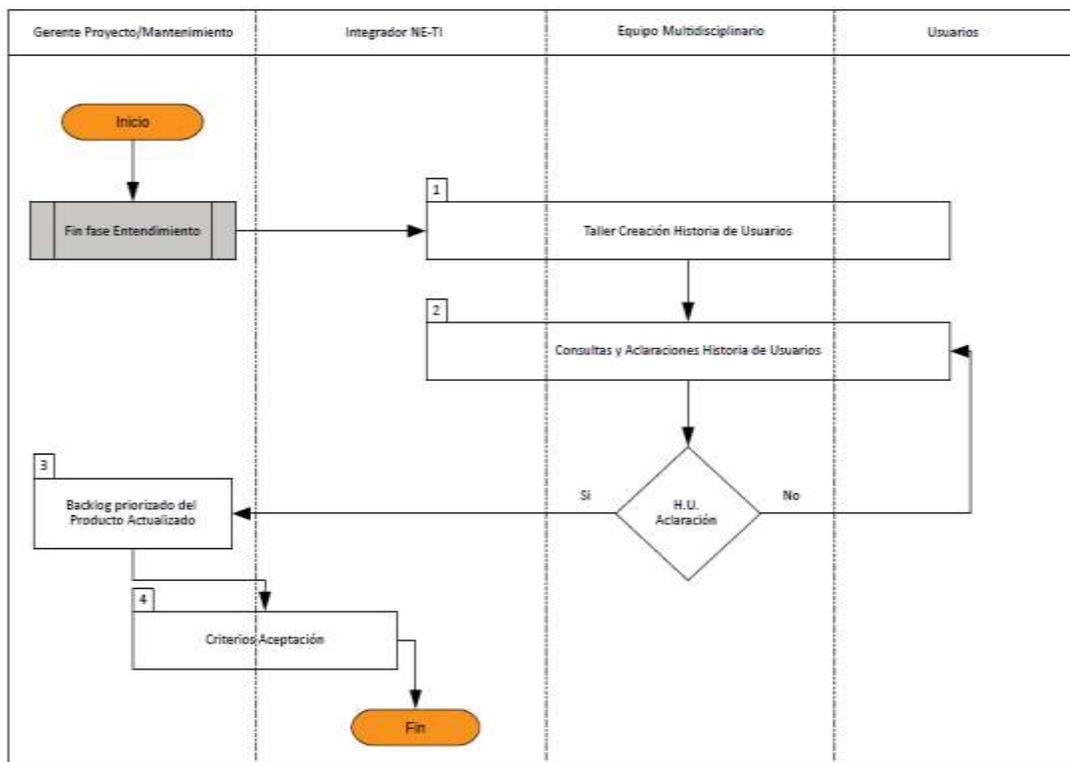
Entregables:

- Historias de usuarios. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.
- Criterio de aceptación por cada historia del usuario. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.
- *Backlog Priorizado del Producto* actualizado. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.

Diagrama de Flujo: La figura 14, indica el flujo de manera general Fase de Análisis Funcional.

Figura 14

Diagrama de flujo fase *Análisis funcional*



FASE 3 – Análisis y diseño técnico: En esta fase se entiende las necesidades del negocio desde el punto de vista técnico, se realiza estimación de tareas en función del esfuerzo necesario desde el punto de vista personas y recursos para implementar los entregables comprometidos.

Responsable de la Fase: Integrador de las necesidades del Negocio-TI.

Participantes Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI, Equipo multidisciplinario y contra parte técnica de terceros involucrados.

Actividades:

- Reuniones de planificación del sprint con todo el equipo multidisciplinario
Responsable: Integrador de las necesidades del Negocio-TI.
- Estimar y comprometer historias de usuario, utilizando métodos de estimación como juicio de experto, planning póker. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- Identificar y estimar tareas, para identificar esfuerzo necesario desde el punto de vista personas y recursos. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- Definir la durabilidad del Sprint para el Proyecto/Mantenimiento.
Responsable: Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- Crear el *Sprint Backlog*, con todas las tareas a completarse en el sprint.
Responsable: Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- Definir arquitectura diseño de la solución. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.

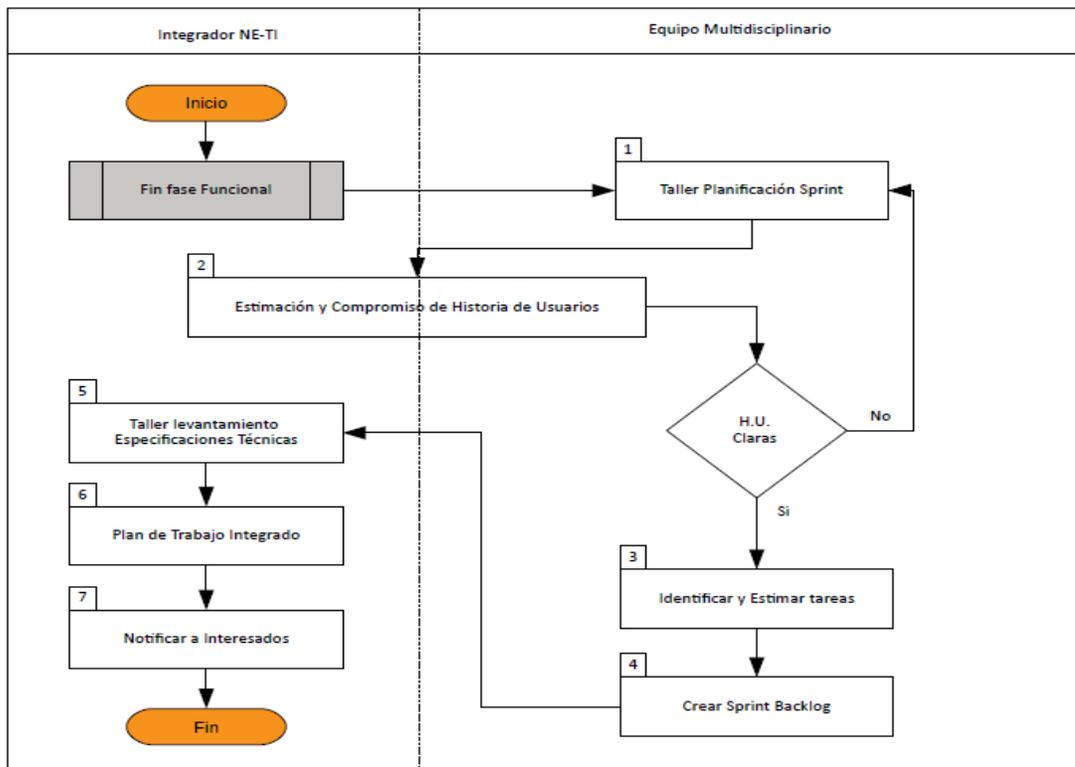
Entregables:

- Documento Especificaciones técnicas. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- Sprint Backlog del Proyecto/Mantenimiento. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- Planificación Integral del Proyecto/Mantenimiento. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.

Diagrama de Flujo: La figura 15, indica el flujo de manera general Fase de Análisis Diseño Técnico.

Figura 15

Diagrama de flujo fase Análisis diseño técnico



FASE 4 – Construcción: En esta fase se crean los entregables del sprint, que satisfacen de manera incremental las necesidades del negocio, establecidas en el Proyecto/Mantenimiento utilizando estándares y realizando pruebas de vulnerabilidad basado en la Guía de Pruebas OWASP.

Responsable de la Fase: Equipo Multidisciplinario.

Participantes: Integrador de las necesidades del Negocio-TI, Equipo multidisciplinario, Contra parte técnica de terceros involucrados y Versionamiento.

Actividades:

- Preparar ambiente de construcción (aplicativos internos, aplicativos externos, data de prueba, instalación fuentes de programas requeridos).
Responsable: Equipo multidisciplinario.
- Construir entregas utilizando estándares de programación, arquitectura y seguridad de la información (Guía de Pruebas OWASP) existentes en Banco Internacional. **Responsable:** Equipo multidisciplinario.
- Ejecutar las pruebas técnicas/unitarias basadas en los casos de aceptación de las historias de usuario a entregar. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- Crear manual de instalación y configuración que contiene una guía de instalación y la lista de componentes a ser versionados. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- Versionar componentes del Sprint en el ambiente indicado **Responsable:** Versionamiento.
- Preparar ambiente con los prerrequisitos necesarios para ejecutar las pruebas de funcionalidad a los entregables. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- Realizar el aseguramiento de calidad (QA) a los entregables.
Responsable: Equipo multidisciplinario.
- Verificar y garantizar las buenas prácticas, estándares y lineamientos a nivel de construcción. **Responsable:** Equipo multidisciplinario.

- Ejecutar pruebas de carga, stress **Responsable:** Equipo multidisciplinario.
- Detección temprana de errores a nivel de componentes así como también los criterios de aceptación para las historias de usuarios. **Responsable:** Equipo multidisciplinario.
- Reportar incidentes encontrados para su corrección. **Responsable:** Equipo multidisciplinario.
- Corrección de Incidentes. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- Realizar reunión de seguimiento diarios para identificar problemáticas, avances y nuevas asignaciones. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.

Entregables:

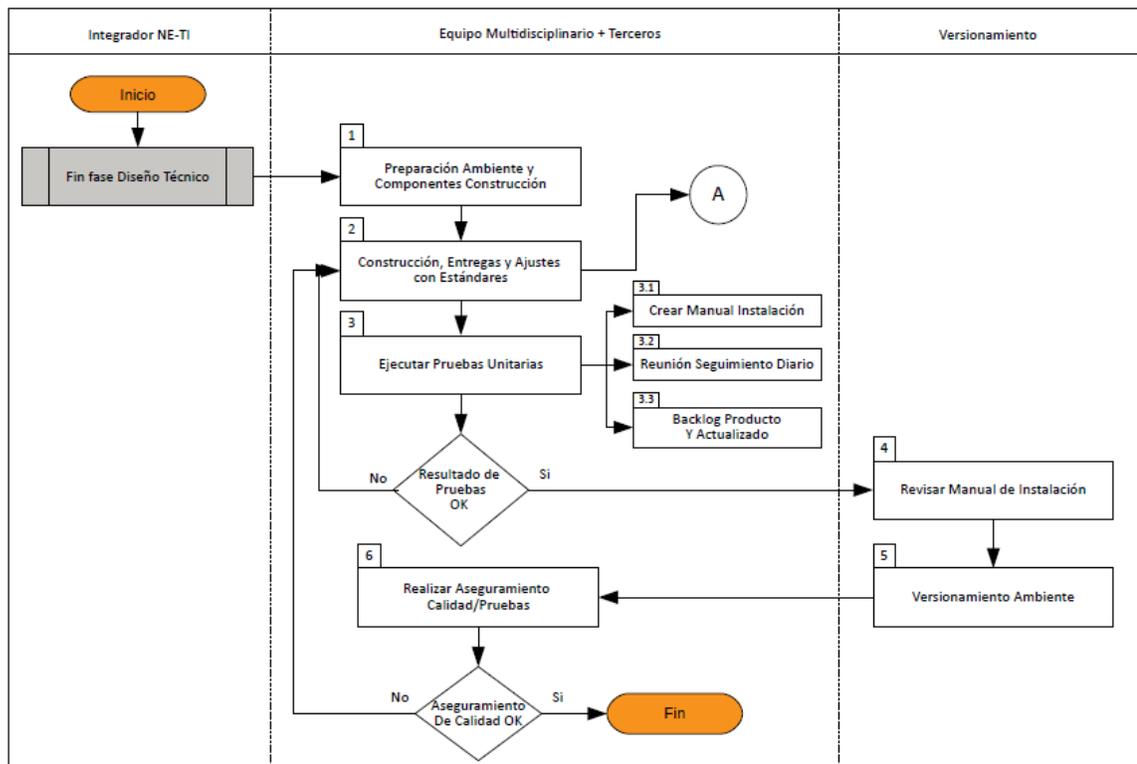
- Entregables planificados, validados del sprint. **Responsable:** Equipo multidisciplinario.
- Manual de instalación y configuración como guía de instalación y la lista de componentes a ser versionados. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- *Backlog Priorizado del Producto* actualizado, en función de los entregables cumplidos y controles de cambios ingresados. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario + Equipo multidisciplinario.

- Planificación Integral del Proyecto/Mantenimiento actualizada.
Responsable: Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- Documento Especificaciones técnicas actualizado. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.

Diagrama de Flujo: La figura 16, indica el flujo de manera general Fase de Construcción.

Figura 16

Diagrama de flujo fase Construcción.



FASE 5 – Certificación: En esta fase se realiza la certificación funcional de los entregables del sprint, con el propósito de lograr la aprobación y aceptación del producto entregado al negocio.

Responsable de la Fase: Integrador de las necesidades del Negocio-TI + Gerente Proyecto/Mantenimiento.

Participantes: Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI, Equipo multidisciplinario, Áreas de control (seguridad información, auditoría, gobierno información), Áreas usuarias (responsable del área y usuarios), Contra parte técnica terceros.

Actividades:

- Reunión de revisión del sprint para aceptar los entregables que cumplan con los criterios de aceptación de las historias de usuario y rechazar los entregables no aceptables **Responsable:** Áreas usuarias y de control.
- Reportar novedades (errores/defectos) encontradas durante la certificación funcional. **Responsable:** Áreas usuarias y de control.
- Corrección de novedades reportadas por usuarios (errores/defectos). **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- Validar, aprobar controles de cambios solicitadas por el negocio. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.

Entregables:

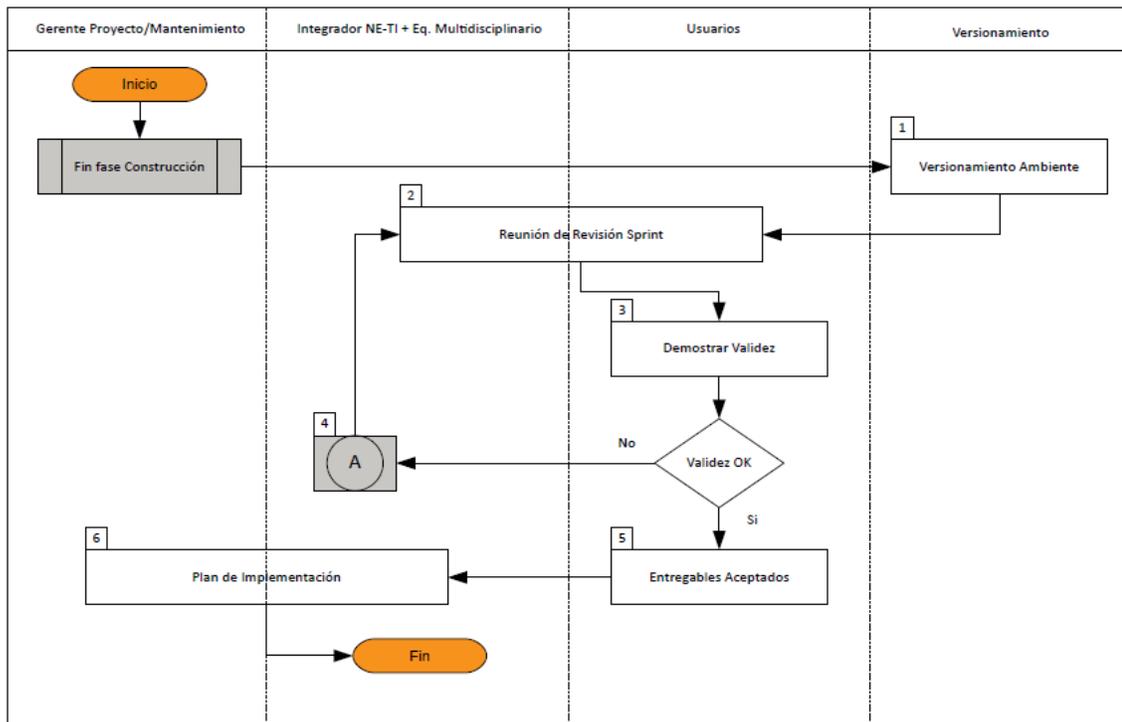
- Entregables del sprint aceptados **Responsable:** Gerente Proyecto/Mantenimiento + Áreas usuarias y de control.

- Entregables del sprint rechazados **Responsable:** Gerente Proyecto/Mantenimiento + Áreas usuarias y de control.
- Manual de instalación y configuración como guía de instalación y la lista de componentes a ser versionados en producción. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- Plan de implementación y Rollback. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- Entrega de los controles de cambios aprobados implementados. **Responsable:** Gerente Proyecto/Mantenimiento.

Diagrama de Flujo: La figura 17, indica el flujo de manera general Fase de Certificación.

Figura 17

Diagrama de flujo fase Certificación



FASE 6 – Liberación: En esta fase se hace la entrega o la transición de los entregables aceptados en la fase de certificación funcional para ser liberados en producción.

Responsable de la Fase: Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI.

Participantes: Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI, Equipo multidisciplinario, Áreas de control (seguridad información, auditoría, gobierno información), Áreas usuarias (responsable del área y usuarios), Contra parte técnica terceros, Versionamiento.

Actividades:

- Coordinar y validar con partes interesadas la liberación a producción.

Responsable: Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.

- Revisión del manual de instalación y configuración como guía de instalación y la lista de componentes a ser versionados en producción.

Responsable: Versionamiento.

- Ejecutar liberación de componentes/programas modificados y/o creados en producción. **Responsable:** Versionamiento + Equipo multidisciplinario + Contra parte técnica terceros
- Realizar reunión de retrospectiva del Sprint, con la finalidad de interiorizar las lecciones aprendidas en la ejecución del sprint. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.

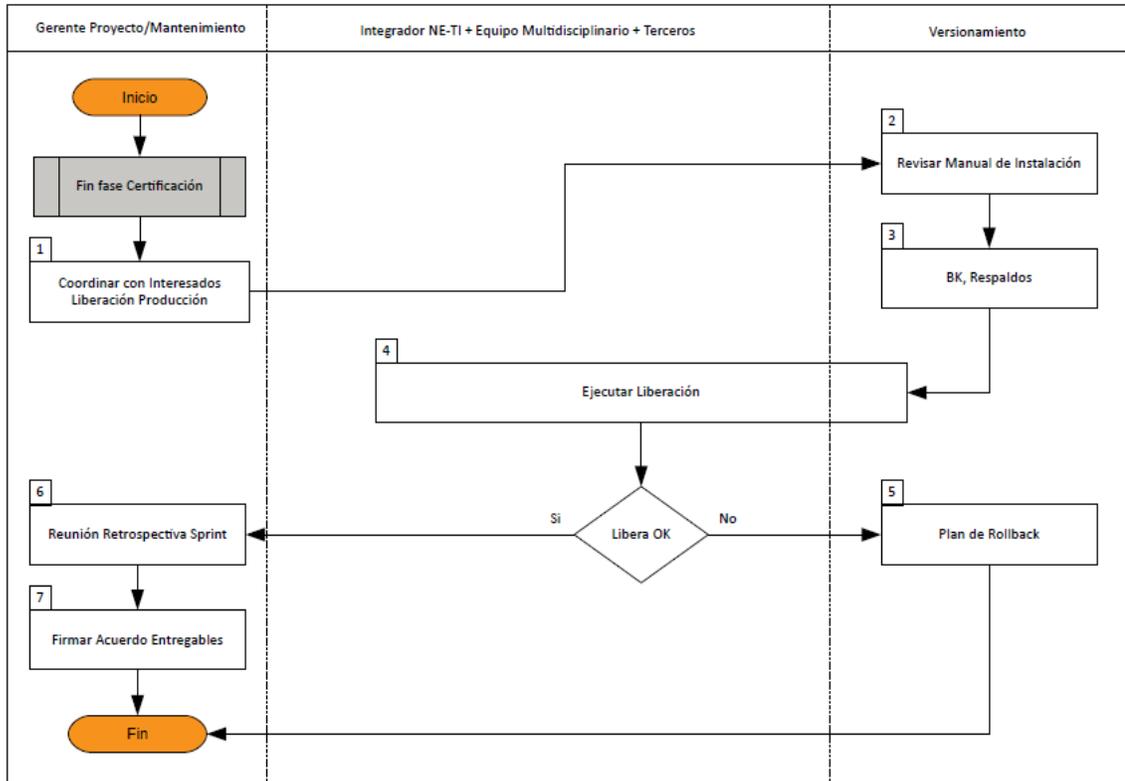
- Firmar en acuerdo de entregables funcionales del Proyecto/Mantenimiento. **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.

Entregables:

- Producto liberado en producción: **Responsable:** Versionamiento + Equipo multidisciplinario + Contra parte técnica terceros.
- Componentes, código fuente de programas modificados y/o creados. **Responsable:** Equipo multidisciplinario + Contra parte técnica terceros.
- Acuerdo de entregables aceptados del Proyecto/Mantenimiento **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.
- Documento funcional y técnica consolidada del Proyecto/Mantenimiento **Responsable:** Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.

Diagrama de Flujo: La figura 18, indica el flujo de manera general Fase de Liberación.

Figura 18 *Diagrama de flujo fase Liberación*



FASE 7 – Soporte postproducción: En esta fase se proporcionar un soporte oportuno en producción, atiende novedades presentadas en el periodo de monitoreo una vez liberado el producto en producción.

- **Responsable de la Fase:** Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI.

Participantes: Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI, Equipo multidisciplinario, Contra parte técnica terceros, versionamiento.

Actividades:

- Reportar novedades (errores/defectos) encontradas en producción.

Responsable: Gerente Proyecto/Mantenimiento, Integrador de las necesidades del Negocio-TI.

- Correcciones definitivas a las novedades reportadas en producción.
Responsable: Equipo multidisciplinario + Contra parte técnica terceros.
- Ejecutar liberación de componentes/programas modificados y/o creados en producción. **Responsable:** Versionamiento + Equipo multidisciplinario + Contra parte técnica terceros.
- Realizar Reunión de la retrospectiva del proyecto con la finalidad de identificar, documentar e interiorizar las lecciones que se aprendieron y a futuro aplicar en otros proyectos. **Responsable:** Gerente Proyecto/Mantenimiento
- Formalizar cierre de Proyecto/Mantenimiento una vez finalizado periodo de monitoreo (un mes de usabilidad en producción). **Responsable:** Gerente Proyecto/Mantenimiento.

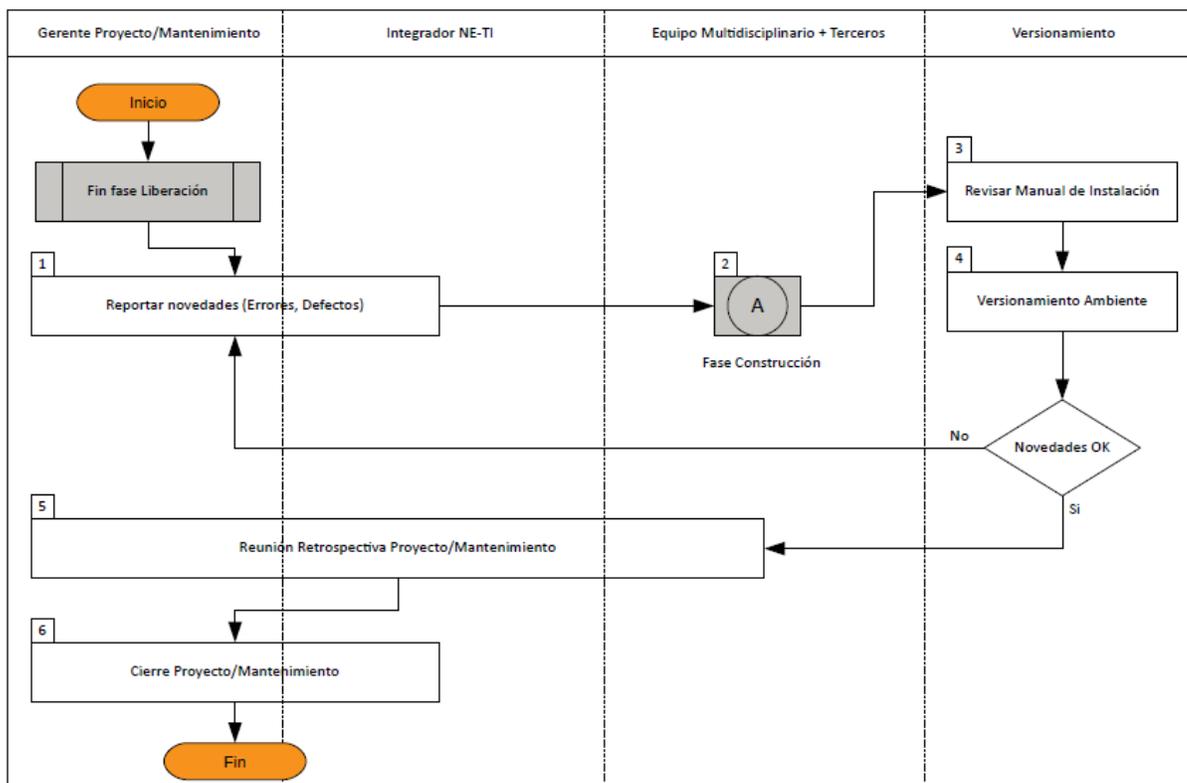
Entregables:

- Componentes, código fuente de programas modificados y/o creados.
Responsable: Equipo multidisciplinario + Contra parte técnica terceros.
- Documento de lecciones Aprendidas del Proyecto/Mantenimiento
Responsable: Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI + Interesados.
- Acta o mail confirmación de cierre de proyecto. **Responsable:** Gerente Proyecto/Mantenimiento.

Diagrama de Flujo: La figura 19, indica el flujo de manera general Fase de Postproducción.

Figura 19

Diagrama de flujo fase Postproducción



Gestion control de cambios: Durante el Ciclo de vida de desarrollo software, se puede presentar cambios funcionales al alcance definido al Proyecto/Mantenimiento a afectando al *Backlog Priorizado del Producto*, este impacto será receptado en las fase de construcción y certificación del sprint que será liberado a producción.

Los controles de cambios pueden ser solicitados por alguno de los actores/partes involucradas en el Proyecto/Mantenimiento sin embargo se debe tener las siguientes consideraciones:

Actividades:

- Generar Solicitud de Control de Cambio **Responsable:** Áreas usuarias y de control.

- Validar, aprobar controles de cambios solicitadas por el negocio.
Responsable: Gerente Proyecto/Mantenimiento + Integrador de las necesidades del Negocio-TI.
- Estimar impacto técnico y tiempos implementación / construcción control de cambios y tiempos de aseguramiento calidad **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- *Backlog Priorizado del Producto* actualizado, en función de los controles de cambios ingresados. **Responsable:** Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.
- Planificación Integral del Proyecto/Mantenimiento actualizada.
Responsable: Integrador de las necesidades del Negocio-TI + Equipo multidisciplinario.

La tabla 12, indica un resumen del Modelo Integral Propuesto para fortalecer el ciclo de vida de desarrollo utilizando COBIT y SCRUM en Banco Internacional.

Tabla 12

Resumen del modelo integral propuesto.

Ciclo de vida de desarrollo		Cobit	Scrum	
<u>Fase</u>	<u>Objetivo</u>	<u>Practica de Gestión</u>	<u>Fase</u>	<u>Proceso</u>
Entendimiento	BAI02 Gestionar definición requisitos	— BAI02.01 Definir la y mantener los requisitos funcionales y técnicos del negocio.	Inicio	<ul style="list-style-type: none"> • Crear la visión del proyecto. • Formar Equipos Multidisciplinario. • Crear el Backlog Priorizado del Producto.

Ciclo de vida de desarrollo		Cobit	Scrum	
<u>Fase</u>	<u>Objetivo</u>	<u>Practica de Gestión</u>	<u>Fase</u>	<u>Proceso</u>
Análisis Funcional		<p>BAI02.01 Definir y mantener los requisitos funcionales y técnicos del negocio.</p> <p>BAI02.02 Realizar un estudio de factibilidad y formular soluciones alternativas.</p> <p>BAI02.04 Obtener la aprobación de requisitos y soluciones.</p>	Planificación y Estimación	<ul style="list-style-type: none"> • Crear historias de usuario.
Análisis y Diseño Técnico	BAI03 Gestionar identificación y construcción de soluciones	<p>— BAI03.01 Diseño de soluciones de alto nivel.</p> <p>BAI03.02 Diseñar componentes detallados para la solución.</p> <p>BAI03.04 Adquirir los componentes de la solución</p>		<ul style="list-style-type: none"> • Estimar historias de usuario. • Comprometer historias de usuario. • Identificar tareas. • Estimar tareas. • Crear el Sprint Backlog.
Construcción		<p>BAI03.03: Desarrollar los componentes de la solución.</p> <p>BAI03.05 Construir soluciones.</p> <p>BAI03.06 Realizar el aseguramiento de calidad (QA)</p> <p>BAI03.07 Preparar las pruebas de la solución.</p> <p>BAI03.08</p>	Implementación	<ul style="list-style-type: none"> • Crear entregables. • Realizar Daily Standup. • Refinamiento del Backlog • Priorizado del Producto.

Ciclo de vida de desarrollo		Cobit	Scrum	
<u>Fase</u>	<u>Objetivo</u>	<u>Practica de Gestión</u>	<u>Fase</u>	<u>Proceso</u>
Certificación	BAI07 Gestionar la aceptación y transición de los cambios de TI	— Ejecutar las pruebas de la solución. BAI03.09 Gestionar los cambios a los requisitos. BAI07.03 Plan de pruebas de la aceptación. BAI07.04 Establecer un entorno de pruebas. BAI07.05 Realizar pruebas de aceptación.	Revisión y Retrospectiva	• Demostrar y validar el sprint
Liberación		BAI07.01 Establecer un plan de implementación. BAI07.02 Planificar la conversión de procesos de negocio, sistemas y datos. BAI07.06 Promover a producción y gestionar las liberaciones (releases).	Lanzamiento	• Enviar entregables • Retrospectiva del sprint
Soporte Postproducción		BAI07.07 Proporcionar soporte oportuno en producción. BAI07.08 Realizar una revisión post-implementación.		• Retrospectiva del proyecto.

Capítulo IV: Conclusiones y Recomendaciones

Conclusiones

- Durante la ejecución del ciclo de vida de desarrollo software en Banco Internacional, se genera tiempos altos en la entrega de producto al negocio afectando a su productividad y competitividad frente a otras instituciones financieras.
- El presente trabajo fortalece el proceso de construcción en Banco Internacional mediante la utilizando 2 marcos de referencia (frameworks) complementarios COBIT y SCRUM, permitiendo optimizar las fases del Ciclo de Vida de Desarrollo de software y generando producto visible al negocio de manea temprana, mejorando su productividad y competitividad.
- Mediante el nuevo rol definido en el ciclo de vida de desarrollo de software “*Integrador de las necesidades del Negocio-TI*” que trabaja muy de la mano con el Gerente del Proyecto/Mantenimiento, se logra establecer la sinergia, empatía y la integración entre el Negocio y Tecnología, la misma que estaba fragmentada
- La creación del rol *Integrador de las necesidades del Negocio-TI* al estar dentro de tecnología permitirá garantizar la creación del producto de manera integral durante la ejecución de las fases del ciclo de vida de desarrollo.
- La unificación de las Fase de Desarrollo y de Pruebas de Aseguramiento en una nueva fase llamada *Construcción*, permite optimizar actividades, entregables, dependencias y tiempos de ejecución dentro del ciclo de vida de desarrollo.
- Crear un equipo multidisciplinario permite de manera temprana e integral solventar todas las necesidades técnicas del Proyecto/Mantenimiento.

Recomendaciones

- Muy importante realizar una socialización efectiva (método y persona) del Fortalecimiento del procesos de construcción software a nivel de Vicepresidentes del Banco Internacional, con la finalidad de lograr el compromiso desde los niveles superiores de la organización.
- Realizar talleres prácticos con los mandos medios de la organización (Gerentes, Subgerentes, Jefes), donde se pueda entender claramente los roles, responsabilidades que tendrán como Gerentes de Proyectos/Mantenimientos durante la ejecución del ciclo de vida de desarrollo de software, así como también el rol y responsabilidades del *Integrador de las necesidades del Negocio-TI*.
- Reforzar los talleres de entendimiento de roles y funciones dentro de la ejecución del ciclo de vida de desarrollo de software con una periodicidad trimestral, dada la rotación de personal que existe en áreas de la Organización.
- Tecnología deberá realizar un monitoreo y control con una frecuencia trimestral sobre la aplicabilidad, cumplimiento y buen uso de la metodología de desarrollo de Software, buscando siempre la mejora continua.
- Actualizar estándares de construcción para los diferentes lenguajes de programación utilizados en Banco Internacional (RPG, JAVA, .Net).
- Crear y mantener actualizados lineamientos de Arquitectura y de Seguridad que rijan la construcción de Software dentro del ciclo de vida de desarrollo.
- Utilizar herramientas como Kiuwan, Micro Focus Fortify entre otros que permitan de manera eficiente garantizar las buenas prácticas a nivel de desarrollo y vulnerabilidad.

- Para los aplicativos WEB poner mayor enfoque en las pruebas de vulnerabilidad basado en la Guía de Pruebas OWASP, lo que permitirá realizar un escaneo para detectar, vulnerabilidades, SQL injection, Denegación de servicios, XSS/Script/HTML, entre otras bondades que ofrece la Guía.
- Utilizar una herramienta de Gestión de Requerimientos que permitan de manera Integral tener una trazabilidad e información del estado del Proyectos/Mantenimientos.

Bibliografía

- Briones Pincay, G. H., & Hernandez Peñaherrera, E. B. (09 de 03 de 2018). *Auditoria de seguridad del servidor web de la empresa Publynex s.a. utilizando mecanismos basados en Owasp*. <http://repositorio.ug.edu.ec: http://repositorio.ug.edu.ec/bitstream/redug/26837/1/B-CINT-PTG-N.249%20Briones%20Pincay%20Gerson%20Hamner.%20Hern%20c3%a1ndez%20Pe%20c3%b1a%20herrera%20Erika%20Bel%20c3%a9n.pdf>
- Cacuango Díaz, L. E. (01 de 01 de 2017). *Propuesta metodológica para gestión de proyectos de desarrollo de software personalizado y marco de trabajo para soporte técnico de la empresa Voicecenter que presta soluciones de sistemas de Call Center*. <http://dspace.udla.edu.ec/handle/33000/6995: http://dspace.udla.edu.ec/bitstream/33000/6995/1/UDLA-EC-TMGSTI-2017-04.pdf>
- Cataldi, Z., Lage, F., Pessacq, R., & García Martínez, R. (01 de 01 de 1996). *Ingeniería de software educativo*. https://es.wikipedia.org/wiki/Desarrollo_en_cascada#cite_note-3: http://www.iidia.com.ar/rgm/comunicaciones/c-icie99-ingenieriasoftwareeducativo.pdf
- Domínguez, P. (30 de 10 de 2017). *Gestiona tu proyecto de desarrollo*. <https://openclassrooms.com: https://openclassrooms.com/en/courses/4309151-gestiona-tu-proyecto-de-desarrollo/4538221-en-que-consiste-el-modelo-en-cascada>
- García Rodríguez, M. J. (01 de 07 de 2015). *Estudio comparativo entre las metodologías ágiles y las metodologías tradicionales para la gestión de proyectos software*. <http://digibuo.uniovi.es/dspace/handle/10651/32457: http://digibuo.uniovi.es/dspace/handle/10651/32457>
- Gomez. (10 de 05 de 2017). *Top 5 Metodologías de Desarrollo de Software*. Megapractical: <https://www.megapractical.com/blog-de-arquitectura-soa-y-desarrollo-de-software/metodologias-de-desarrollo-de-software>.
- Gualotuña Lema, D. A. (01 de 07 de 2016). *Migración de infraestructura de servicios microsoft utilizando la metodología MSF*. <http://repositorio.utc.edu.ec/handle/27000/3689: http://181.112.224.103/bitstream/27000/3689/1/T-UTC-000039.pdf>
- ISACA. (05 de 12 de 2018). *COBIT® 2019 Objetivos de Gobierno y Gestión*. www.isaca.org
- Lopez Provencio, F. (15 de 01 de 2015). *Metodologías para el desarrollo*. Metodologías para el desarrollo: <http://upcommons.upc.edu/bitstream/handle/2099.1/24902/103275.pdf?sequence=1>
- Mariño, S. I., & Alfonzo, P. L. (4 de 12 de 2014). *Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación*.

[Http://www.redalyc.org/revista.oa?id=849:](http://www.redalyc.org/revista.oa?id=849)
<https://www.redalyc.org/pdf/849/84933912009.pdf>

Mendoza, L. E., Grimán, A., & Pérez, M. (09 de 10 de 2008). *Especialización de msf para el desarrollo basado en componentes de sistemas colaborativos*. WillyDEV
Especializacion MSF: <https://es.scribd.com/document/7331780/WillyDEV-Especializacion-MSF>

Molina Montero, B., Vite Cevallos, H., & Dávila Cuesta, J. (01 de 06 de 2018). *Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software*. [Http://revistaespirales.com/index.php/es:](http://revistaespirales.com/index.php/es)
<http://revistaespirales.com/index.php/es/article/view/269>

SCRUMstudy™. (12 de 05 de 2017). *Guía SBOK™ - 3ra Edición*. [Www.scrumstudy.com:](http://www.scrumstudy.com)
www.scrumstudy.com

Toro , A., & Peláez, L. E. (15 de 01 de 2018). *Validación de un modelo para el aseguramiento de la calidad del software en MIPYMES que desarrollan software en el Eje Cafetero*. [Http://www.scielo.org.co/pdf/ecei/v12n23/1909-8367-ecei-12-23-00084.pdf:](http://www.scielo.org.co/pdf/ecei/v12n23/1909-8367-ecei-12-23-00084.pdf) <http://www.scielo.org.co/pdf/ecei/v12n23/1909-8367-ecei-12-23-00084.pdf>