

TEC | Tecnológico de Costa Rica

Área Académica de Administración de Tecnologías de Información

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de Includtec

Trabajo final de graduación para optar al grado de Licenciatura en Administración de Tecnologías de Información

Dionisio Palacio Amador

Cartago Agosto, 2020



Hoja de aceptación

Área Académica de Administración de Tecnologías de Información

Grado académico: Licenciatura

Los miembros del Tribunal Examinador del Área de Administración de Tecnologías de Información recomendamos que el presente Informe Final de Proyecto de Graduación del estudiante Dionisio Palacio Amador sea aceptado como requisito parcial para obtener el grado académico de Licenciatura en Administración de Tecnología de Información.

MARIA JOSE
ARTAVIA
JIMENEZ (FIRMA)

Firmado digitalmente
por MARIA JOSE
ARTAVIA JIMENEZ
(FIRMA)
Fecha: 2020.08.03
09:40:08 -06'00'

Ing. María José Artavia Jiménez, MBA

Profesor Tutor

LUIS CARLOS
NARANJO
ZELEDON
(FIRMA)

Firmado digitalmente
por LUIS CARLOS
NARANJO ZELEDON
(FIRMA)
Fecha: 2020.08.03
14:53:22 -06'00'

M. Phil. Luis Naranjo Zeledón

Lector



Firmado digitalmente por
PEDRO IGNACIO LEIVA
CHINCHILLA (FIRMA)
Fecha: 2020.08.03
16:36:37 -06'00'

Lic. Pedro I. Leiva Chinchilla

Lector

YARIMA TATIANA
SANDOVAL
SANCHEZ
(FIRMA)

Firmado digitalmente
por YARIMA TATIANA
SANDOVAL SANCHEZ
(FIRMA)
Fecha: 2020.08.04
11:01:54 -06'00'

Ing. Yarima Sandoval Sánchez, MBA

Coordinación Trabajo Final de

Graduación

Resumen

El presente proyecto es una investigación enfocada en el tema de aseguramiento de la calidad del software. La investigación se realiza en una pequeña organización dedicada al desarrollo de software que presenta una problemática en medir la calidad del software al no contar con resultados que lo respalden, esto corresponde a que durante el ciclo de vida del desarrollo los procesos de aseguramiento de calidad son empíricos y no estandarizados.

A partir de la problemática, se presenta una propuesta de metodología de aseguramiento de la calidad de software, que permita a la organización cumplir con los objetivos de calidad de cada proyecto que este en desarrollo y los proyectos futuros. Esto, por medio de un conjunto de procesos, actividades y artefactos enfocado en la gestión de requerimientos y verificación y validación de software.

Para abordar la problemática existente, se parte de un análisis de la situación actual para consolidar en una serie de procesos las prácticas actuales de la organización. Luego, se hace una revisión de las normas y buenas prácticas de la industria del software, referente a aseguramiento de la calidad. Posteriormente, con la situación actual y las recomendaciones de las prácticas de la industria se hace un análisis de la brecha, que permite identificar las necesidades de la organización. Por último, ese análisis da paso para la formulación de la propuesta para la organización.

De todo el proceso de desarrollo de la investigación se obtiene que el aseguramiento de la calidad es un proceso que se debe abordar desde el inicio del proyecto y no en solo en las etapas de pruebas del software. Por lo tanto, la propuesta se hace abarcando la gestión de requerimientos y la gestión de verificación y validación de software. Esto involucra, en cuanto a requerimientos, el levantamiento, análisis, especificación, validación y control de cambios, y en cuanto a verificación y validación, la planeación, ejecución, documentación y reporte de pruebas.

Con esta investigación se determina que las prácticas de calidad de la industria son benignas para cualquier organización, para ello se debe realizar un análisis para identificar las aplicaciones de esas prácticas, con el fin de aportar valor al proceso de desarrollo de software, por medio de procesos ágiles y objetivos.

Palabras clave: Calidad, Software, Aseguramiento de calidad (QA), Modelo de calidad, Includtec

Abstract

This project is a research focused on software quality assurance. The research is carried out in a small organization dedicated to software development that presents a problem in measuring software quality as it does not have results that support it, this corresponds to empirical and non-standardized processes of quality assurance during the software life cycle

Based on the problem, a proposal for a software quality assurance method is presented, which allows the organization to meet the quality goals of each current project and futures. Through a set of processes, activities and artifacts focused on requirements management and software verification and validation.

At the beginning, it starts from an analysis of the current situation to consolidate the current practices of the organization in a series of processes. Then, a review of the standards and best practices of the software industry is done, referring to quality assurance. Subsequently, with the current situation and the recommendations of industry practices, an analysis of the gap is made, which allows finding the needs of the organization. Finally, this analysis gives way to the formulation of the proposal for the organization.

From all the research development process it is obtained that the quality assurance is a process that must be addressed from the beginning of the project and not only in the software testing stages. Therefore, the proposal is made encompassing requirements management and software verification and validation management. This involves, in terms of requirements, the elicitation, analysis, specification, validation and control of changes, and in terms of verification and validation, planning, execution, documentation and test reporting.

With this research it is determined that the quality practices of the industry are benign for any organization, for this an analysis must be carried out to identify the applications of those practices, in order to add value to the software development process, through agile and objective processes.

Keywords: Quality, Software, Quality assurance (QA), Quality model, Inlutec.

Tabla de contenido

1. Introducción	1
1.1. Descripción general.....	1
1.2. Antecedentes.....	3
1.2.1. Descripción de la organización	3
1.2.2. Misión.....	3
1.2.3. Visión	4
1.2.4. Sobre la organización	4
1.2.5. Equipo de trabajo.....	6
1.2.6. Proyectos similares realizados dentro o fuera de la organización.	8
1.3. Planteamiento del problema.....	10
1.3.1. Situación problemática.....	11
1.3.2. Beneficios esperados del proyecto	13
1.4. Objetivos.....	14
1.4.1. Objetivo general.....	14
1.4.2. Objetivos específicos	14
1.5. Justificación del proyecto	15
1.6. Alcance del proyecto	17
1.6.1. Aseguramiento de calidad.....	17
1.6.2. Control de calidad	18
1.6.3. Exclusiones del alcance.....	19
1.7. Limitaciones del proyecto	19
1.8. Supuestos del proyecto	20
1.9. Entregables del proyecto.....	20

1.9.1.	Entregables de producto.....	20
1.9.2.	Entregables académicos.....	21
1.9.3.	Gestión del proyecto.....	21
2.	Marco conceptual.....	22
2.1.	Proceso.....	22
2.1.1.	Documentación de procesos.....	22
2.2.	Proyecto y administración de proyectos.....	23
2.3.	Software.....	24
2.4.	Requerimientos de software.....	25
2.4.1.	Requerimientos de software según SWEBOK.....	25
2.4.2.	Requerimientos según ISO/IEC/IEEE 29148:2018(E).....	26
2.5.	Requerimientos de accesibilidad de software.....	28
2.6.	Metodología.....	29
2.6.1.	Metodología de desarrollo de software.....	30
2.6.2.	Metodologías de desarrollo ágil.....	30
2.7.	Modelo de madurez y capacidad.....	33
2.8.	Métricas.....	36
2.9.	Calidad.....	37
2.9.1.	Calidad de software.....	39
2.9.2.	Aseguramiento de la calidad (QA) de software.....	40
2.9.3.	Control de la calidad de software (verificación y validación).....	44
2.10.	Modelo de calidad de software.....	55
2.11.	Prácticas de calidad de software.....	56
2.11.1.	Mejores prácticas.....	56
2.11.2.	Normas y estándares.....	57

3.	Marco metodológico.....	60
3.1.	Tipo de investigación.....	60
3.1.1.	Diseño de la investigación	61
3.2.	Sujetos y fuentes de información.....	64
3.2.1.	Sujetos de información.....	64
3.2.2.	Fuentes de información.....	65
3.3.	Metodología de investigación	67
3.3.1.	Analizar la situación actual del proceso de calidad en Inclutec.....	68
3.3.2.	Analizar la brecha entre el proceso de calidad actual y las recomendaciones de buenas prácticas y estándares.	69
3.3.3.	Elaborar la metodología.....	70
3.3.4.	Identificar herramientas de soporte de QA.....	70
3.4.	Instrumentos de investigación	71
3.4.1.	Observación.....	71
3.4.2.	Entrevistas	72
3.4.3.	Cuestionarios.....	73
3.4.4.	Revisión documental.....	76
3.4.5.	Análisis de los datos	77
3.5.	Operacionalización de las categorías de análisis	78
4.	Análisis de resultados	82
4.1.	Revisión documental	82
4.1.1.	Base teórica para el proyecto	82
4.1.2.	Revisión, descripción y análisis de la documentación de los proyectos de Inclutec	
	86	
4.1.3.	Estado de las pruebas de software.....	95
4.2.	Descripción y análisis de la problemática actual	98

4.2.1. Observación de los procesos de gestión de requerimientos y verificación y validación.....	98
4.2.2. Análisis cualitativo de la gestión de requerimientos.....	99
4.2.3. Análisis cualitativo de la gestión de pruebas.....	100
4.2.4. Análisis cualitativo de aseguramiento de calidad.....	100
4.3. Documentación del proceso actual.....	106
4.3.1. Proceso de gestión de requerimientos.....	106
4.3.2. Proceso de pruebas.....	109
4.4. Análisis de brecha.....	122
4.4.1. Comparación de la situación actual y las recomendaciones de la teoría.....	123
4.4.2. Matriz FODA del contexto actual.....	127
4.4.3. Estado actual y estado esperado.....	127
4.4.4. Estrategias para alcanzar el nivel esperado.....	129
5. Propuesta de solución.....	130
5.1. Aspectos generales.....	130
5.1.1. Procedimientos.....	130
5.1.2. Técnicas y artefactos.....	131
5.1.3. Roles.....	131
5.1.4. Conceptualización de la propuesta de aseguramiento de calidad.....	131
5.2. Definición de política de calidad de software.....	134
5.3. Propuesta de la metodología de aseguramiento de calidad.....	136
5.3.1. Proceso de gestión de requerimientos de software.....	136
5.3.2. Proceso de validación y verificación de software.....	160
5.3.3. Costos estimados de la implementación de la propuesta.....	183
5.3.4. Validación de la propuesta.....	185

5.4.	Análisis de herramientas de QA	186
5.4.1.	Criterios de análisis.....	186
5.4.2.	Herramientas seleccionadas.....	189
6.	Conclusiones	195
7.	Recomendaciones	198
8.	Bibliografía.....	200
9.	Anexos.....	205
	Anexo I: Plantilla de minutas.....	205
	Anexo II: Plantilla de gestión de cambios.....	206
	Anexo III: Plantillas de requerimientos actuales.....	207
	Plantilla 1. Requerimientos funcionales del EAW.	207
	Plantilla 2. Requerimientos funcionales de Euler.	209
	Plantilla 3. Requerimientos no funcionales del EAW.	210
	Anexo IV. Plantilla para los casos de pruebas manuales e integración.	211
10.	Apéndices.....	212
	Apéndice A. Instrumentos para la investigación.....	212
	Instrumentos1. Plantilla para observación.	212
	Instrumentos 2. Guía entrevista de requerimientos.....	212
	Instrumentos 3. Guía entrevista de calidad.....	213
	Instrumentos 4. Encuesta sobre aseguramiento de calidad en Inclutec.	214
	Instrumentos 5. Encuesta sobre la gestión de requerimientos en Inclutec.	215
	Instrumentos 6. Plantilla de revisión documentar interna.....	216
	Instrumentos 7. Planilla de revisión documental de normas y buenas prácticas.	216
	Instrumentos 8. Planilla para el análisis de brecha.	217
	Apéndice B. Respuestas de instrumentos.....	218

Respuesta 1: Encuesta sobre requerimientos en Inclutec.	218
Respuesta 2: Encuesta sobre aseguramiento de calidad en Inclutec.	223
Apéndice C. Validación de la propuesta.	240
Validación 1. Requerimientos	240
Validación 2. Evaluación del proceso de verificación y validación.	243
Apéndice D: Minutas con el profesor.	247
Seguimiento con el profesor tutor.	247
Apéndice E: Minutas con la organización.	256
Apéndice F: Visitas del profesor tutor a la organización.	265
Apéndice G: Resumen.	268
Minutas contraparte	268
Minutas profesor tutor	269
Minutas con otros colaboradores de la organización	270
Apéndice F: Guía de implementación.	271

Índice de Figuras

Figura 1. Organigrama de Inlutec. Fuente: Elaboración propia.	6
Figura 2. Modelo ágil basado en Scrum para el desarrollo de proyectos en Inlutec. Fuente (Romero et al., 2019).....	10
Figura 3. Ciclo de desarrollo con Scrum tradicional. (Fuente: Tomado de Requena (2018). ..	32
Figura 4. Incorporación de actividades de QA al ciclo de desarrollo de proyectos. Fuente: Tomado de IEEE (2014).....	41
Figura 5. Estrategia de documentación de pruebas según metodología de Scrum. Fuente: Eira et al. (2018).	54
Figura 6. Principales acciones para llevar a cabo la investigación-acción. Fuente: Tomado de (Hernández et al, 2014).....	63
Figura 7. Esquema de la metodología de desarrollo del proyecto. Fuente: Elaboración propia	68
Figura 8. Temas para la revisión documental. Fuente: Elaboración propia.	83
Figura 9. Proceso actual de gestión de requerimientos. Fuente: Elaboración propia.	107
Figura 10. Proceso actual de gestión de cambios de requerimientos. Fuente: Elaboración propia.....	108
Figura 11. Proceso de pruebas unitarias manuales del proyecto Euler. Fuente: Elaboración propia.....	110
Figura 12. Proceso de pruebas unitarias de los proyectos web. Fuente: Elaboración propia.	111
Figura 13. Proceso actual de pruebas de integración de software.	113
Figura 14. Subproceso actual de ejecución de pruebas. Fuente: Elaboración de propio.	114
Figura 15. Proceso actual de pruebas de accesibilidad en el proyecto Euler. Fuente: Elaboración propia.....	115
Figura 16. Subproceso de ejecución de pruebas de accesibilidad en el proyecto Euler. Fuente: Elaboración propia.....	117
Figura 17. Proceso actual de pruebas de accesibilidad en los proyectos web. Fuente: Elaboración propia.....	119
Figura 18. FODA de la situación actual. Fuente: Elaboración propia.	127
Figura 19. Metodología de desarrollo de software de Inlutec. Fuente: Adaptación de (Romero et al., 2019).....	132

Figura 20. Incorporación de los procesos de aseguramiento al proceso de desarrollo. Fuente: Elaboración propia.....	133
Figura 21. Proceso de recolección y especificación de requerimientos. Fuente: Elaboración propia.....	137
Figura 22. Proceso de gestión de requerimientos durante del desarrollo. Fuente: Elaboración propia.....	140
Figura 23. Proceso para la gestión de cambios de requerimientos. Fuente: Elaboración propia.	144
Figura 24. Organización de las pruebas y documentación en el proyecto. Fuente: Elaboración propia.....	161
Figura 25. Proceso de planificación y ejecución de pruebas de verificación. Fuente: Elaboración propia.....	164
Figura 26. Proceso de la planificación y ejecución de pruebas unitarias. Fuente: Elaboración propia.....	168
Figura 27. Proceso de pruebas de aceptación. Fuente: Elaboración propia.	180

Índice de Tablas

Tabla 1. Equipo de trabajo. Fuente: Elaboración propia.....	8
Tabla 2. Trabajos relacionados al tema de calidad. Fuente: Elaboración propia.....	9
Tabla 3. Elementos básicos de la notación BPMN. Fuente: OMG (2011).	23
Tabla 4. Comparación de los niveles de capacidad y de madurez. Fuente: (CMMI Institute, 2010).	34
Tabla 5. Resumen de los niveles de madurez y capacidad según el CMMI. Fuente: Adaptado de (CMMI Institute, 2010).	36
Tabla 6. Mejores prácticas consideradas en el marco conceptual. Fuente: Elaboración propia.	57
Tabla 7. Normas consideradas en el marco conceptual. Fuente: Elaboración propia.	59
Tabla 8. Sujetos de información en Inclutec. Fuente: Elaboración propia.	65
Tabla 9. Fuentes primarias de información. Fuente: Elaboración propia.....	66
Tabla 10. Instrumentos de entrevistas y su propósito para el proyecto. Fuente: Elaboración propia.....	73
Tabla 11. Lista de cuestionarios aplicados durante el desarrollo del proyecto. Fuente: Elaboración propia.....	76
Tabla 12. Tabla de variables. Fuente: Elaboración propia.....	81
Tabla 13. Normas y buenas prácticas consideradas para el proyecto. Fuente: Elaboración propia.....	85
Tabla 14. Contenido de la documentación de requerimientos del proyecto EAW. Fuente: Elaboración propia.....	87
Tabla 15. Contenido de la documentación de requerimientos del proyecto SICID. Fuente: Elaboración propia.....	89
Tabla 16. Contenido de la documentación de calidad del proyecto SICID. Fuente: Elaboración propia.....	90
Tabla 17. Contenido de la documentación de requerimientos del proyecto Euler. Fuente: Elaboración propia.....	92
Tabla 18. Contenido de la documentación de calidad del proyecto Euler. Fuente: Elaboración propia.....	92

Tabla 19. Comparación entre la documentación de requerimientos de los proyectos involucrados. Fuente: Elaboración propia.....	93
Tabla 20. Comparación de la estructura de los requerimientos funcionales de los proyectos involucrados. Fuente: Elaboración propia.....	94
Tabla 21. Contenido del instrumento de documentación Plan de Pruebas actual. Fuente: Elaboración propia.....	97
Tabla 22. Resultado de las observaciones realizadas por el investigador. Fuente: Elaboración propia.....	99
Tabla 23. descripción de las tareas del proceso de requerimientos iniciales. Fuente: Elaboración propia.....	108
Tabla 24. Descripción de las tareas del proceso de requerimientos durante el desarrollo del proyecto. Fuente: Elaboración propia.....	109
Tabla 25. Descripción de las tareas del proceso de pruebas unitarias de Euler. Fuente: Elaboración propia.....	111
Tabla 26. Descripción de las tareas del proceso de pruebas unitarias de los proyectos web. Fuente: Elaboración propia.....	112
Tabla 27. Descripción de las tareas del proceso pruebas de integración. Fuente: Elaboración propia.....	114
Tabla 28. Descripción de las tareas del proceso pruebas de accesibilidad en el proyecto Euler. Fuente: Elaboración propia.....	116
Tabla 29. Descripción de las tareas del subproceso de ejecutar los casos de prueba de accesibilidad. Fuente: Elaboración propia.....	119
Tabla 30. Descripción de las tareas del proceso pruebas de accesibilidad en los proyectos web. Fuente: Elaboración propia.....	121
Tabla 31. Comparación de la aplicación de pruebas del proyecto Euler y los proyectos Web. Fuente: Elaboración propia.....	122
Tabla 32. Comparación del estado actual con las recomendaciones de las normas y buenas prácticas. Fuente: Elaboración propia.....	126
Tabla 33. Estado actual de los procesos de la organización y estado esperado. Fuente: Elaboración propia.....	128

Tabla 34. Descripción de las tareas del proceso de recolección y especificación de requerimientos. Fuente: Elaboración propia.....	139
Tabla 35. Descripción de las tareas del proceso de requerimientos durante el desarrollo. Fuente: Elaboración propia.....	143
Tabla 36. Matriz de clasificación de cambios de requerimientos. Fuente: Elaboración propia.	144
Tabla 37. Descripción de las tareas del proceso de cambios. Fuente: Elaboración propia.	146
Tabla 38. Plantilla para la documentación de escenarios de requerimientos. Fuente: Elaboración propia.....	151
Tabla 39. Ejemplo #1 del uso de la plantilla para la especificación de requerimientos. Fuente: Elaboración propia.....	153
Tabla 40. Ejemplo #2 del uso de la plantilla de especificación de requerimientos. Fuente: Elaboración propia.....	154
Tabla 41. Plantilla para la gestión de cambios. Fuente: Adaptación del Anexo II.....	157
Tabla 42. Métricas para la gestión de requerimientos. Fuente: Elaboración propia.	159
Tabla 43. Descripción de las tareas del proceso de planificación y ejecución de pruebas de verificación. Fuente: Elaboración propia.....	167
Tabla 44. Descripción de las tareas del proceso de planificación y ejecución de pruebas unitarias. Fuente: Elaboración propia.	170
Tabla 45. Plantilla para la documentación de casos de prueba unitarias. Fuente: Elaboración propia.....	172
Tabla 46. Métricas del proceso de verificación. Fuente: Elaboración propia.	178
Tabla 47. Descripción de las tareas del proceso de pruebas de aceptación. Fuente: Elaboración propia.....	182
Tabla 48. Métricas del proceso de validación (aceptación). Fuente: Elaboración propia.	183
Tabla 49. Costos de salario aproximado de las personas en caso de contratación. Fuente: Elaboración propia.....	184
Tabla 50. Costos por horas de capacitación del personal contratado. Fuente: Elaboración propia.....	184
Tabla 51. Costos de salarios aproximados del personal actual. Fuente: Elaboración propia.	185

Tabla 52. Costos por horas de capacitación del personal actual. Fuente: Elaboración propia.	185
Tabla 53. Pesos de los criterios de funcionalidad para el benchmarking. Fuente: Elaboración propia.....	187
Tabla 54. Peso de los criterios técnicos y de precio para el benchmarking. Fuente: Elaboración propia.....	188
Tabla 55. Herramientas consultadas. Fuente: Elaboración propia.	189
Tabla 56. Comparación bajo los criterios de funcionalidad. Fuente: Elaboración propia.....	190
Tabla 57. Comparación bajo precio y criterios técnicos. Fuente: Elaboración propia.....	193
Tabla 58. Puntaje asignado a las herramientas según los criterios técnicos y de precio. Fuente: Elaboración propia.....	194

1. Introducción

1.1. Descripción general.

En este capítulo, se describen los elementos introductorios del trabajo final para optar por el título de Licenciatura en Administración de Tecnologías de Información, que consiste en la propuesta de una metodología enfocada en aseguramiento y control de calidad de software para Inclutec. Este proyecto se desarrolla bajo el perfil profesional de gestor de soluciones, que es uno de los perfiles de la carrera de Administración de Tecnologías de Información (ATI) (Tecnológico de Costa Rica, 2020). La propuesta presentada se hace en el área de gestión de proyectos y enfocado en el área de conocimiento de calidad en proyectos de software. Con el fin de aportar valor al proceso de desarrollo de software en Inclutec, por medio de alternativas para la gestión de un proceso de aseguramiento de calidad.

La propuesta se hace basada en normas y buenas prácticas de calidad que se utilizan actualmente en la industria del software para adaptarlo a las necesidades de un grupo de interés, como es el caso de Inclutec.

Una de las actividades principales de Inclutec es el desarrollo de software, por esta razón el proyecto se enfoca en esa área. Se requiere realizar un análisis que permita identificar requerimientos de aseguramiento y control de calidad en los procesos actuales y tener alternativas para la gestión efectiva del cumplimiento de los requerimientos.

La naturaleza del proyecto enfocado en aseguramiento de la calidad se debe a oportunidades de mejora identificadas en el proceso actual, que pueden representar beneficios para Inclutec. Esto se realiza tomando en cuenta las capacidades y recursos que posee el grupo de interés.

Para llevar a cabo el proyecto se plantea realizar un análisis de la situación actual y una revisión de las recomendaciones de normas y buenas prácticas de calidad, para comparar y determinar la brecha y posibles propuestas de solución. En general, este documento presenta la descripción del proyecto que se desarrolla, se detallan los antecedentes de la organización donde es ejecutado y los aspectos importantes de trabajos relacionados que aportan al presente proyecto.

Del mismo modo, se expone la problemática que presenta el Grupo de Interés Inclutec, específicamente en el área de desarrollo de software y los beneficios que se esperan del proyecto. Además, se indica el objetivo general y los específicos que engloban el alcance.

También, se establecen las restricciones y supuestos con los que cuenta el desarrollo del proyecto y se describen los entregables que se esperan obtener.

En los capítulos posteriores, se presenta la teoría de calidad que se tomará como base para el desarrollo de la propuesta, la metodología utilizada para el desarrollo del proyecto, los resultados obtenidos al aplicar las diferentes fases de la metodología, la propuesta dada a Inlutec y las conclusiones y recomendaciones.

1.2. Antecedentes

En esta sección se describen los detalles de la organización como la visión, misión y valores. El propósito es contextualizar la naturaleza de la organización.

1.2.1. Descripción de la organización

El TEC “[...] es una institución nacional autónoma de educación superior universitaria, dedicada a la docencia, la investigación y la extensión de la tecnología y las ciencias conexas para el desarrollo de Costa Rica. Fue creado mediante ley No. 4.777 del 10 de junio de 1971.” (Tecnológico de Costa Rica, 2019).

Dentro del TEC se encuentra el Centro de Investigación en Computación (CIC), el cual es “[...] una entidad adscrita a la Escuela de Ingeniería en Computación del Tecnológico de Costa Rica. Fue creado en 1992 y se apoya en el sólido prestigio logrado por la escuela en sus 35 años de existencia.” (Tecnológico de Costa Rica, 2019).

El CIC (Centro de Investigación en Computación) busca la incorporación de tecnología para solución de problemas en la región de Centroamérica y el Caribe, haciendo énfasis en la generación, adaptación, incorporación y difusión de conocimientos informáticos (Tecnológico de Costa Rica, 2019).

Inlutec no posee misión y visión, sin embargo, al estar adjunto al CIC se utiliza la visión y misión del CIC y del TEC.

1.2.2. Misión

A continuación, la misión del TEC:

Contribuir al desarrollo integral del país, mediante formación del recurso humano, la investigación y la extensión; manteniendo el liderazgo científico, tecnológico y técnico, la excelencia académica y el estricto apego a las normas éticas, humanísticas y ambientales, desde una perspectiva universitaria estatal de calidad y competitividad a nivel nacional e internacional.

(Tecnológico de Costa Rica, 2019).

La misión del CIC es: “Incorporar, adecuar y generar tecnologías computacionales acordes con las necesidades de los sectores productivos y de servicios” (Tecnológico de Costa Rica, 2019).

1.2.3. Visión

A continuación, la visión del TEC:

El Instituto Tecnológico de Costa Rica seguirá contribuyendo mediante la sólida formación del talento humano, el desarrollo de la investigación, la extensión, la acción social y la innovación científico-tecnológica pertinente, la iniciativa emprendedora y la estrecha vinculación con los diferentes actores sociales a la edificación de una sociedad más solidaria e inclusiva; comprometida con la búsqueda de la justicia social, el respeto de los derechos humanos y del ambiente.

(Tecnológico de Costa Rica, 2019).

La visión del CIC es: “Ser líderes en el campo de tecnologías computacionales en la región de Centroamérica y el Caribe, mediante la realización de actividades orientadas a la generación, adaptación, incorporación y difusión de conocimientos informáticos” (Tecnológico de Costa Rica, 2019).

1.2.4. Sobre la organización

Inlutec es un grupo de interés adjunto al CIC, que realiza investigación aplicada, desarrolla e implementa tecnologías con accesibilidad para toda la población nacional. Este grupo tiene como objetivo fomentar la inclusión social en el ámbito tecnológico.

Las principales actividades que caracterizan a Inlutec son:

- Analizar el contexto de las poblaciones más vulnerables, mediante la investigación, con el fin de tener bases para el desarrollo de innovaciones sociales y tecnológicas.
- Desarrollar proyectos tecnológicos y accesibles para apoyar equiparación e igualdad de oportunidades para todas las personas.
- Promover a nivel nacional la accesibilidad, con el fin de contribuir en el mejoramiento de la realidad de las poblaciones más vulnerables.
- Tecnologías Inclusivas para todas las personas.

Además, Inlutec actualmente está trabajando en los siguientes proyectos:

- a. Plataforma Internacional de Edición de Lengua de Señas (PIELS):** Proyecto en desarrollo que busca apoyar en la reducción de las brechas existentes en el proceso de comunicación entre la población oyente y la población sorda, por medio de una

herramienta (avatar) que brinde la traducción entre texto en español y su equivalente en Lengua de Señas Costarricense (LESCO).

- b. Sistema Costarricense de Información sobre Discapacidad (SICID):** Este es un proyecto en desarrollo y consiste en una plataforma web para el CONAPDIS (Consejo Nacional de la Persona con Discapacidad), que busca fortalecer las capacidades de Costa Rica para generar información adecuada y de calidad sobre discapacidad, que permita la toma de decisiones como país, para que este segmento pueda ejercer plenamente sus derechos.
- c. Editor de Recursos de Aprendizaje STEM¹ (EULER):** Consiste en un editor científico - matemático accesible para personas con discapacidad visual. Apoya en la lectura, edición, exploración, importación y exportación de fórmulas a formatos estándar; facilitando los procesos de enseñanza – aprendizaje de las matemáticas en escenarios donde al menos una persona tiene discapacidad visual.

Dentro de los proyectos terminados está la traducción de información textual a lenguas indígenas costarricenses. Con la colaboración de estudiantes indígenas del TEC y de sus comunidades, se traducen textos en cinco de las lenguas indígenas autóctonas costarricenses, a saber: Bribri, Brunca, Cabécar, Maleku, y Ngäbe. Esto contribuye con el rescate de las lenguas indígenas, evidenciando la existencia de la riqueza cultural costarricense, además que fortalece el propósito de accesibilidad e inclusión social hacia las tecnologías.

1.2.4.1. Propuesta de valor

Según Naranjo (2019), la organización se basa de los siguientes valores:

- Honestidad
- Trabajo en equipo.
- Inclusividad.
- Orientación al cliente.
- Honestidad.
- Respeto.

1 STEM: Acrónimo de Ciencia, Tecnología, Ingeniería y Matemática.

1.2.5. Equipo de trabajo

A continuación, se muestra el organigrama de Inlutec y se detalla la información de los interesados del proyecto.

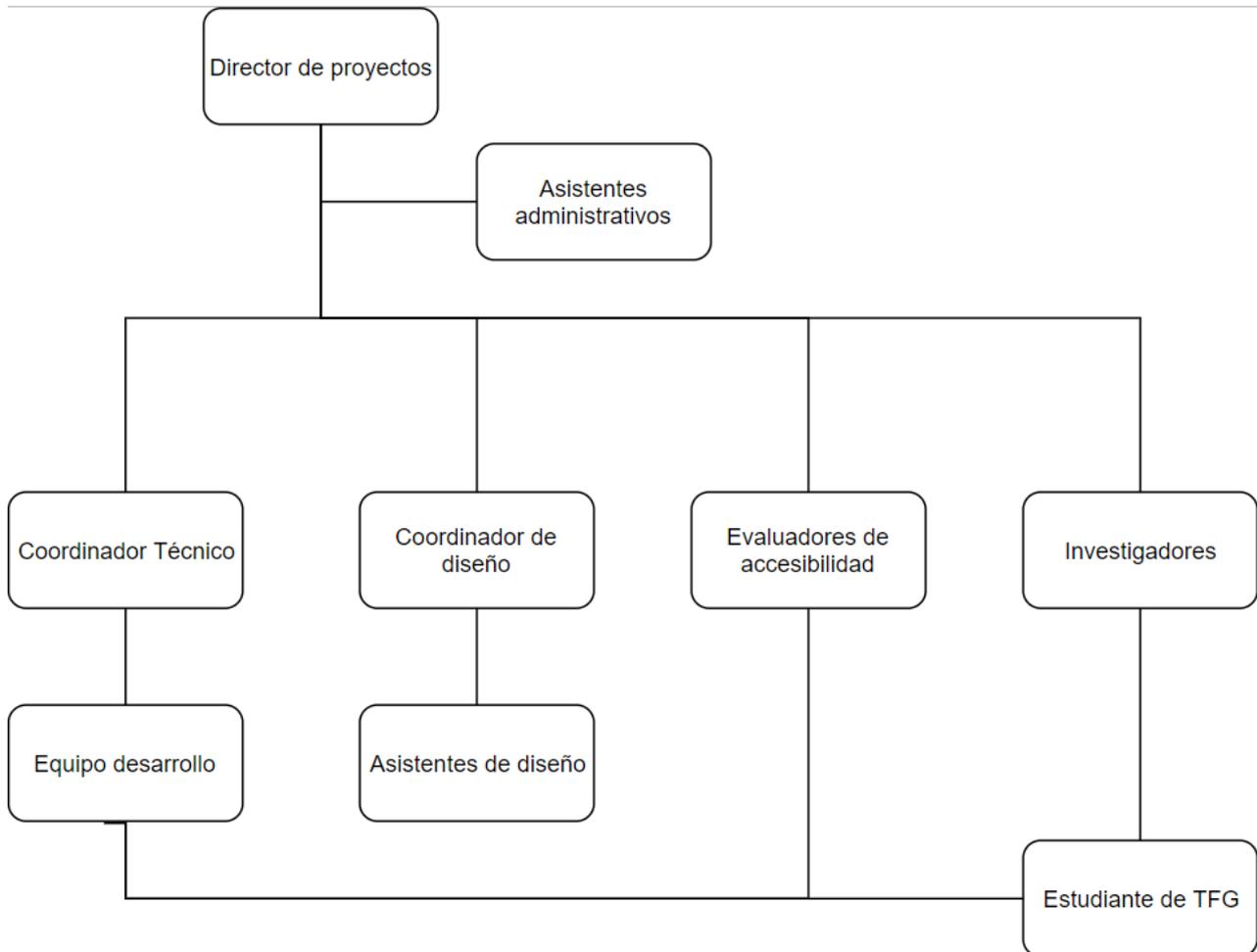


Figura 1. Organigrama de Inlutec. Fuente: Elaboración propia.

El proyecto se realiza en colaboración con los coordinadores técnicos y equipo de desarrollo del proyecto Inlutec. En la Tabla 1, se muestra las responsabilidades de los interesados en el proyecto.

Posición Laboral	Rol en la empresa	Rol en el proyecto	Contacto
Director de proyectos	<ul style="list-style-type: none"> • Encargado de la toma de decisiones estratégicas respecto a los proyectos. • Contacto directo con los clientes. • Asignación de tareas a los equipos de trabajo. • Coordinador de financiación. 	Patrocinador del proyecto	machacon@itcr.ac.cr
Coordinador del proyecto Euler y web	<ul style="list-style-type: none"> • Distribución de las tareas en el equipo de desarrollo. • Dar seguimiento a los avances de las tareas distribuidas. • Velar por el cumplimiento de la metodología de desarrollo utilizada. 	Responsable en el acompañamiento del desarrollo del proyecto.	vimora@itcr.ac.cr
Equipo de desarrollo	<ul style="list-style-type: none"> • Desarrollo de los productos de software. • Velar por el cumplimiento de las funcionalidades con los requerimientos. 	Involucrados directos en las necesidades, intereses y expectativas.	

Posición Laboral	Rol en la empresa	Rol en el proyecto	Contacto
Estudiante de TFG	<ul style="list-style-type: none"> Asistente de desarrollo de software. Encargado de pruebas del proyecto EULER. 	Encargado en el desarrollo del TFG en el tema de aseguramiento de calidad.	dag1411@gmail.com

Tabla 1. Equipo de trabajo. Fuente: Elaboración propia.

1.2.6. Proyectos similares realizados dentro o fuera de la organización.

Inlutec no ha desarrollado algún trabajo relacionado con calidad, sin embargo, a nivel del TEC se ha identificado un trabajo enfocado en este tema y los demás han sido desarrollados fuera de la organización. Estos trabajos serán insumos para plantear el proyecto y como referencia para el desarrollo de este. En Tabla 2, se muestran.

Nombre	Descripción	Autor
Diseño de sistema para el aseguramiento y el control de la calidad en los proyectos de software del TEC Digital	Se realiza la propuesta de integración de dos elementos para la ejecución del propio sistema: 1) herramientas de software que soporten las actividades planteadas y 2) un modelo de calidad de software para la verificación, validación y evaluación de esta a través de todo el ciclo de vida de desarrollo de los proyectos de software del TEC Digital. (Sánchez, 2017)	Michael Sánchez Soto

Nombre	Descripción	Autor
Propuesta de una metodología de control de calidad para los proyectos de automatización, basado en las mejores prácticas de ISTQB, caso SOIN S.A.	La metodología propuesta se fundamenta en tres pilares que son: principios, métodos y técnicas. Dentro de los cuales se desarrollan los componentes requeridos para fundamentar el conocimiento, definir los procesos, y desarrollar las herramientas necesarias para garantizar la calidad. (Ocampo, 2017)	María Fernández Ocampo
Establecimiento de la metodología para aseguramiento de la calidad en ArcSight QA Costa Rica.	La investigación deriva una propuesta para la creación de una metodología enfocada en el aseguramiento de la calidad, siguiendo las mejores prácticas de la industria con respecto al aseguramiento de calidad del software y usando un modelo de desarrollo ágil. (Aguilar, 2018)	Alejandro Aguilar Mora

Tabla 2. Trabajos relacionados al tema de calidad. Fuente: Elaboración propia.

1.3. Planteamiento del problema

En esta sección se describe la situación problemática encontrada en Inlutec, la cual motiva el desarrollo del proyecto, así como la mención de los beneficios esperados del producto.

En Inlutec se desarrollan diferentes proyectos basados en tecnologías web, escritorio y animación, como lo son SICID, EULER y PIELS. El equipo de desarrollo se encarga de realizar el planeamiento para el diseño, codificación y pruebas de los componentes de software que necesitan los productos. Para el desarrollo se utiliza un modelo ágil basado en Scrum que permite tener un control de los requerimientos y su cumplimiento durante el desarrollo, como se muestra en la Figura 2.

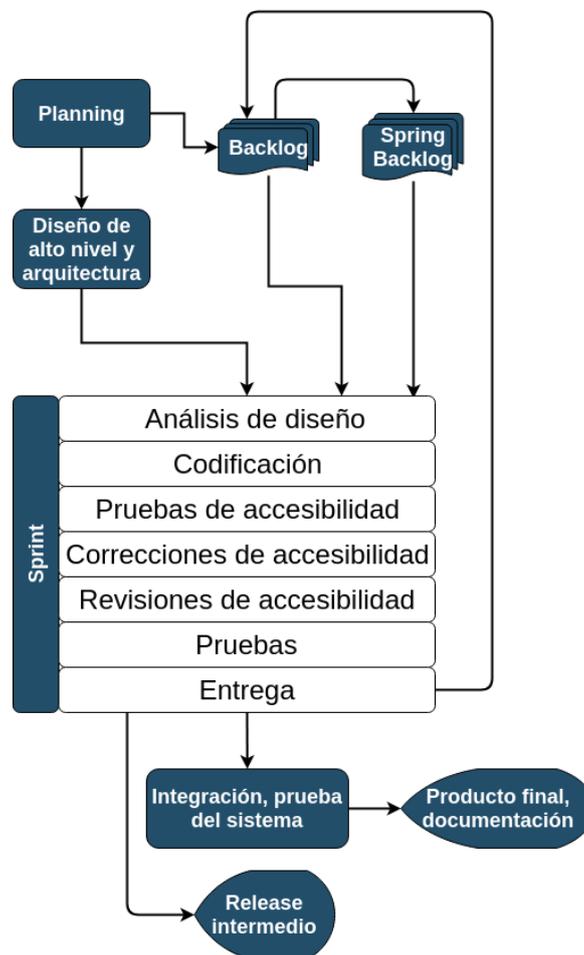


Figura 2. Modelo ágil basado en Scrum para el desarrollo de proyectos en Inlutec. Fuente (Romero et al., 2019)

El modelo de desarrollo inicia posterior a la definición de requerimientos, cuenta con dos partes importantes de pruebas, las de accesibilidad y las pruebas generales del funcionamiento del software y finaliza con entregas de software intermedios o la entrega final.

De las pruebas, que son importantes para verificar el estado del software se describen las siguientes.

- **Pruebas de accesibilidad:** Para estas pruebas se presta una atención cuidadosa pues es el motivo de existencia del grupo de interés. Estas pruebas se realizan por los evaluadores de accesibilidad de Inlutec.
- **Pruebas:** Son las pruebas generales de funcionamiento, las cuales se hacen por *sprint*, sin embargo, en ocasiones son los mismos desarrolladores quienes evalúan el software al final.

Además de la metodología de desarrollo, para los proyectos web se cuenta con tres servidores, el servidor de desarrollo, pruebas y producción (servidor en la nube AWS - *Amazon Web Services*). Las funcionalidades que se desarrollan para las soluciones tienen que pasar por esos tres servidores. La transición del software a los diferentes servidores se hace por *sprint* terminado o por funcionalidad según lo consideren los desarrolladores.

1.3.1. Situación problemática

A continuación, se hace un enfoque específico en la situación problemática y sus diferentes partes.

1.3.1.1. Estado de los requerimientos

La solución que se desarrolle inicia desde los requerimientos que se definan, sin embargo, actualmente el proceso de gestión no vela por la actualización y control de los requerimientos, lo que causa situaciones de retrabajo por inconsistencia en los requerimientos, cambios constantes durante el desarrollo y resultados de pruebas rechazados.

1.3.1.2. Estándares o buenas prácticas de calidad

Como se evidencia en la Minuta 1 del Apéndice E, los equipos de desarrollo han hecho el esfuerzo de implementar pruebas de las funcionalidades generales y accesibilidad al final del *sprint*, sin embargo, aseguramiento de calidad es un proceso que incluye actividades como planeación, análisis, implementación, reporte y resultados de pruebas (Roshdy, 2018).

Además, es un proceso que se ejecuta durante todo el ciclo de vida del software, desde los requerimientos hasta el cierre del proyecto. De esta manera, aunque se involucran etapas de pruebas, no cuentan con procesos que estén documentados bajo estándares o buenas prácticas de aseguramiento de calidad.

1.3.1.3. *Estado de la calidad en desarrollo*

El método de pruebas para las funcionalidades generales, que aplican los desarrolladores, consiste en utilizar las funciones del software y revisar que se obtenga el resultado esperado. Los resultados que se documentan no tienen seguimiento y en otros casos no se documenta, sino que el esfuerzo se centra en solucionar lo identificado como defectos o errores.

No se hace un seguimiento de la documentación de resultado de pruebas, tanto de éxitos como errores. Esto, dificulta conocer o medir el grado de alineación o cumplimiento de los requerimientos. A raíz de esto, surge una falta de información sobre del estado de calidad de los componentes de software y sus características a nivel interno. Además, los objetivos y métricas de calidad están ausentes, esto tiene impacto si se quiere hacer cambios para la mejora continua y le resta relevancia.

1.3.1.4. *Controles de pruebas*

El equipo debe pasar el software por los diferentes servidores, desde desarrollo y pruebas hasta que el producto esté listo para ser liberado en el servidor de producción. Cuando el desarrollador considera que se pueden ejecutar pruebas lo llevan al servidor respectivo (servidor de pruebas).

Para hacer la transición del software en desarrollo a pruebas para su evaluación, no se tiene control de esa etapa, en el sentido que, no se documentan la transición, ejecución y los resultados obtenidos. Entonces, se desconoce si esa etapa de pruebas tiene resultados positivos para el ciclo de desarrollo.

El control que existe es cuando se tiene que pasar al servidor de producción, donde hay una persona encargada para hacer la liberación y garantizar que sea exitosa, sin afectar el resto de las aplicaciones que se estén ejecutando.

Una liberación exitosa no significa que el software tenga la calidad esperada. Al colocarse el software en producción se espera que las nuevas características funcionen y no haya una afectación de las actuales, sin embargo, no se tiene la certeza y en ocasiones causa problemas a los usuarios, como se evidencia en la Minuta 1 del Apéndice E.

1.3.1.5. *Reporte de los usuarios y hotfixes*

Cuando los usuarios experimentan problemas a causa de nuevas características o funcionalidades actuales con errores, debido a las situaciones mencionadas anteriormente,

el equipo de desarrollo debe atender *hotfixes* para poder solventar la demanda de los usuarios y en ocasiones se desconoce el origen de los errores, lo que atrasa el proceso.

Ante toda esta problemática surge la pregunta, ¿Qué necesita Inlutec para asegurar que los productos de los proyectos cumplan con los requerimientos de calidad esperados por las partes interesadas?

1.3.2. Beneficios esperados del proyecto

Con la realización de este proyecto se espera que la organización solviente la problemática y obtenga los siguientes beneficios:

- Se contará con un proceso documentado, adaptado a las condiciones de la organización, que permita asegurar la calidad del software que se desarrolle.
- Aportar una guía que permita a los involucrados en desarrollo de software controlar la calidad del producto.
- Contar con métricas para controlar el avance de las actividades de aseguramiento de calidad y tomar acciones preventivas.
- La organización podrá contar con información sobre la calidad de los proyectos y emprender acciones para la mejora continua, basándose en hechos.
- Permitir el cumplimiento de los requerimientos iniciales, y ante la posibilidad de cambios de prioridades, se podrán entregar funcionalidades que respondan a los requerimientos.
- Analizar el riesgo y tomar decisiones adecuadas ante cambios en los proyectos.

Beneficios indirectos del proyecto

- La entrega de productos con la calidad esperada por el cliente aporta a la imagen del grupo de interés, no solo por las propuestas y desarrollo de proyectos innovadores sino en la calidad de estos.
- Generar oportunidades para nuevas contrataciones de proyectos por demanda o financiamiento para los proyectos de investigación en accesibilidad al proporcionar evidencia sobre el proceso de calidad que se lleva a cabo para desarrollar un producto.
- La sostenibilidad del grupo de interés depende del financiamiento o contrataciones que se obtengan, de este modo, la idea de calidad es tener un soporte en la atracción de proyecto y por ende ser sostenibles.

1.4. Objetivos

En esta sección se detalla el objetivo general y los objetivos específicos del proyecto.

1.4.1. Objetivo general

Proponer una metodología de calidad de software basada en estándares y buenas prácticas de aseguramiento y control de calidad, en el periodo del primer semestre 2020, que permita que el proceso de desarrollo de software y sus resultados en Inlutec sean los esperados por las partes interesadas.

1.4.2. Objetivos específicos

- Caracterizar el proceso de calidad actual utilizado en Inlutec por medio de observación, consulta con los desarrolladores y revisión de documentación para entender su funcionamiento.
- Analizar la brecha entre el proceso de calidad actual y las recomendaciones de buenas prácticas, estándares y marcos de referencia por medio de una evaluación de las partes del proceso actual contra las recomendaciones para determinar las necesidades de calidad.
- Elaborar la metodología de aseguramiento y control de calidad para los proyectos de software basado en la brecha entre las recomendaciones de calidad y el proceso actual para ajustarla a las necesidades de Inlutec.
- Recomendar herramientas de software que se ajusten a la metodología de aseguramiento y control de calidad propuesta, esto por medio de un benchmarking para agilizar el proceso en los equipos de desarrollo de Inlutec.

1.5. Justificación del proyecto

A continuación, se describe la justificación del proyecto del proceso de aseguramiento y control de calidad en Inlutec.

Como se mencionó, Inlutec desarrolla varios proyectos de software donde la característica principal es su enfoque en la accesibilidad. Algunos proyectos son iniciativas del grupo de interés que generan desarrollos subsecuentes y eventualmente serán demandado por clientes potenciales, en el contexto del desarrollo del trabajo final de graduación se les llamará proyectos por iniciativa (por ejemplo, EULER). Los otros son por demanda de alguna institución (por ejemplo, SICID). Es imperativo para Inlutec que los productos de estos proyectos tengan la calidad que esperan los interesados.

El motivo principal del desarrollo de proyectos de software de investigación en accesibilidad es generar un producto estable, que permita la promoción del grupo de interés y facilite la generación de oportunidades de obtención de financiamiento y de este modo, asegurar la constancia del centro de interés.

Los proyectos para las instituciones son el ingreso principal del grupo de interés, esto significa que se deben entregar los productos con una calidad que sea percibida como buena por parte de los interesados, esto en términos de funcionalidad, accesibilidad, estabilidad, entre otros, de manera que haya conformidad de las partes. Lo anterior, es importante porque garantiza la sostenibilidad económica de Inlutec, y la posibilidad de seguir incursionando en temas de accesibilidad.

Inlutec debe proporcionar una imagen que le permita mantener y atraer financiamiento para los proyectos actuales o nuevos que sean financiados por otras entidades. La forma de hacerlo es garantizando que los productos de los proyectos actuales sean de calidad, sin embargo, al no existir un proceso ni un encargado para la ejecución, se tiene una alta incertidumbre de la calidad.

Además, tanto en proyectos de investigación como desarrollo para instituciones, desde sus inicios, se han caracterizado por estar sujetos a cambios constantes, ya sea por redefinición de prioridades o por experimentación de nuevas características.

Ante esta situación, la estructura de desarrollo debe ser adaptada a un alto componente de flexibilidad, a medida que el equipo ha madurado en este tipo de ambiente han podido adaptarse a las condiciones de cambio. Sin embargo, el hecho de la existencia de cambios

en un proyecto, según la teoría de administración de proyectos, puede tener impactos en diferentes direcciones tales como costo, tiempo, alcance y calidad.

Cuando hay cambios de prioridades, el equipo de desarrollo tiene que solventar la necesidad, y centrar los esfuerzos en un componente de software específico para liberarlo con prontitud. Como no existe un proceso de calidad ni una figura que lo lleve a cabo, el riesgo de impacto en la calidad será alto, pues no se somete a pruebas suficientes para determinar si está apto para ser liberado sin afectar las funcionalidades existentes.

De este modo, se justifica un proceso de aseguramiento y control de calidad con su respectivo rol, que le permita a Inclutec cumplir con la calidad esperada y puedan generar beneficios para su sostenibilidad.

1.6. Alcance del proyecto

En el desarrollo de este proyecto se busca realizar la aplicación de la teoría de calidad basada en estándares, buenas prácticas y marcos de referencia para desarrollar una metodología adaptada a los procesos de desarrollo de software llevados a cabo en Inlutec. El propósito es garantizar la funcionabilidad, confiabilidad, seguridad, usabilidad, rendimiento, eficiencia, integración y accesibilidad de los productos de software desarrollados.

La metodología propuesta será tanto para los proyectos por iniciativa de Inlutec como los proyectos por demanda, de forma que todo el proceso de desarrollo y calidad en Inlutec sea trabajado de la misma manera. La información necesaria se obtiene trabajando directamente con el equipo de desarrollo y los otros interesados.

1.6.1. Aseguramiento de calidad

El trabajo va en línea con aseguramiento de la calidad que involucra los requerimientos y el control de calidad (verificación y validación).

Aseguramiento es un proceso que conlleva acciones planificadas de calidad:

El aseguramiento de la calidad persigue construir confianza en que las salidas futuras o incompletas, también conocidas como trabajo en curso, se completarán de tal manera que se cumplan los requisitos y expectativas establecidos. El aseguramiento de la calidad contribuye al estado de certeza sobre la calidad, mediante la prevención de defectos a través de procesos de planificación o de inspección de defectos durante la etapa de implementación del trabajo en curso.

(PMI, 2013, p.242)

El equipo de desarrollo cuenta con la metodología de desarrollo de software basada en Scrum, en esta propuesta no se contempla hacerle cambios a esa forma de trabajo, sin embargo, se incorporan las prácticas de aseguramiento de calidad como acompañamiento al ciclo de desarrollo.

Como se mencionó en el problema, calidad no solo se limita a ejecución de pruebas, sino que lleva planeación, análisis y diseño, implementación y evaluación desde los requerimientos y pruebas hasta la entrega de productos. Por lo tanto, a partir de la teoría de calidad se propone la metodología para los proyectos de Inlutec.

Como parte del trabajo no se contempla todo el ciclo desarrollo de software, los esfuerzos para aseguramiento de calidad son en las siguientes etapas:

- Requerimientos.
- Pruebas de accesibilidad.
- Correcciones de accesibilidad.
- Revisiones de accesibilidad.
- Pruebas.

No se contempla dentro del proyecto aplicar aseguramiento de calidad en las siguientes etapas:

- *Planning*.
- Diseño de alto nivel y arquitectura.
- Análisis de diseño.
- Codificación.

Como se evidencia en la Minuta 1 del Apéndice E, el motivo de no contemplar los puntos anteriores es porque en términos de metodología de desarrollo de software, tienen ciertos elementos definidos. Se cuenta con estándares internos para la arquitectura y desarrollo y en la parte de *planning*, se rige por la capacidad y urgencia de los requisitos en *backlog*, que son decisiones que toma el líder del proyecto.

Se buscará que el proceso de aseguramiento tenga un enfoque ágil, de forma que se pueda acoplar a la metodología de desarrollo actual.

1.6.2. Control de calidad

El control de calidad es el proceso que “utiliza un conjunto de técnicas operativas y de tareas para verificar que las salidas entregadas cumplirán los requisitos” (PMI, p.248). A partir de la metodología de aseguramiento de calidad, se involucra el proceso de control de calidad (verificación y validación), que en términos generales son las pruebas de software.

Se espera que la metodología de aseguramiento permita que los productos finales cuenten con las características de calidad acordadas con las partes interesadas. Además, que permita documentar y asegurar de manera formal el cumplimiento de los objetivos.

El proceso de control de calidad está enfocado en las etapas de pruebas e Integración y pruebas del sistema de la metodología de desarrollo actual utilizada (Ver Figura 2). Estas

etapas se centran en las pruebas de verificación y validación que se deben realizar sobre el software, son principalmente pruebas de integración que se hacen al final de los *sprints* y previos a las entregas o demostraciones formales.

1.6.3. Exclusiones del alcance

Se debe considerar las siguientes exclusiones:

- Para requerimientos se busca definir y la clasificar los requerimientos funcionales y no funcionales, sin embargo, para la verificación y aceptación se centra en pruebas de integración funcionales y de accesibilidad no funcionales.
- No se hace aseguramiento de calidad del sistema sino de software, por lo tanto, se deja de lado arquitectura, codificación y aspectos de hardware. Se centra principalmente el software, sus características y resultado de pruebas de verificación de funcionalidad y accesibilidad.
- La organización no tiene procesos definidos, por lo tanto, la propuesta que realizada será el primer paso. Debido a esto la propuesta será a alto nivel, no se entrará en detalles por la madurez de la organización.
- La metodología está enfocada completamente para el desarrollo de software, no contempla algún otro tipo de productos que brinde Inlutec.

1.7. Limitaciones del proyecto

En esta sección se muestran las posibles limitaciones que pueden tener un impacto considerable sobre el proyecto, principalmente de forma negativa.

Las limitaciones identificadas son las siguientes:

- Las prioridades en cuanto a desarrollo de software son cambiantes, de existir la necesidad de dedicar esfuerzo a otras actividades limitará el desarrollo del proyecto.
- Uno de los principales insumos para el desarrollo del proyecto se obtendrá de la experiencia del equipo de desarrollo, y se contará con ellos para ir revisando el avance del proyecto. Sin embargo, si se ellos se encuentran limitados de tiempo para entrevistas o responder encuestas, el proceso de desarrollo del proyecto se puede ver afectado.
- Al no existir un proceso de calidad y las actividades de pruebas que se realizan no son estandarizadas, las respuestas de los desarrolladores pueden ser diferentes

según el proyecto al que pertenezca. Esto puede limitar la caracterización del proceso actual.

- Los desarrolladores trabajan bajo una metodología de desarrollo ágil, lo que significa que la propuesta de calidad debe ajustarse a su forma de trabajo, para no afectar el flujo de trabajo que llevan actualmente.
- Existen estándares y marcos de referencia que no están disponibles de forma gratuita, esto puede limitar tener información actualizada.

1.8. Supuestos del proyecto

Para el desarrollo del trabajo final de graduación, se estima que los siguientes elementos serán de apoyo por parte de la organización:

- A pesar de la suspensión del curso lectivo, se cuenta con el aval de la organización para continuar con el desarrollo del proyecto.
- Se aprueba el teletrabajo como alternativa para llevar a cabo las actividades del desarrollo del proyecto.
- La organización tiene un canal de comunicación activo durante el tiempo de trabajo para realizar cualquier consulta con los miembros del equipo.
- Se dispone del tiempo dentro de la organización para el desarrollo del proyecto.
- Si no existen complicaciones que los desarrolladores tengan que resolver urgentemente, se cuenta con su disponibilidad para consultas, entrevistas y encuestas.

1.9. Entregables del proyecto

En esta sección se describen todos los entregables que tiene el proyecto, tomando en cuenta los de gestión, los académicos y los entregables del producto solicitados por la organización.

1.9.1. Entregables de producto

Los entregables del producto serán los que se obtengan como resultado de ejecución del proyecto. A continuación, se describen los productos que se entregan a la organización.

- **Metodología de aseguramiento y control de calidad:** Este entregable es toda la propuesta de aseguramiento de calidad para las etapas de requerimientos, pruebas de accesibilidad y pruebas generales de funcionamiento. Cuenta con el proceso, artefactos y roles necesarios para la aplicación de aseguramiento de calidad en los procesos de gestión de requerimiento y verificación y validación de software.

- **Comparación de herramientas:** Se hace una comparación de herramientas de software enfocadas en aseguramiento de la calidad, donde se indica el valor que puede aportar a la organización.
- **Guía de implementación:** Es una guía con recomendaciones para la implementación de la metodología, enfocada en garantizar la adopción de la propuesta.

1.9.2. Entregables académicos

En esta sección se describen los entregables académicos del proyecto. A continuación, se detallan:

- **Informe del Trabajo Final de Graduación (TFG):** Este entregable se desarrolla de forma paralela respecto a los demás entregables, y cuenta con la descripción de todas las etapas y procesos del desarrollo del proyecto. Es la documentación completa del proyecto, desde la base teórica de inicio hasta los resultados obtenidos.

1.9.3. Gestión del proyecto

A continuación, se muestran los instrumentos que se utilizarán para la gestión del proyecto, estos son para recolectar evidencia del proceso de desarrollo del proyecto.

1.9.3.1. *Minutas*

Para el seguimiento del proyecto se realizarán reuniones que involucran requerimientos, revisiones de avances, acuerdos, solicitudes y aprobaciones de cambio. El propósito es tener respaldos de las reuniones realizadas. La plantilla para las minutas se encuentra en el Anexo I.

1.9.3.2. *Gestión de cambios*

La gestión de cambios se realiza para llevar el control de los eventuales cambios que puedan surgir durante el desarrollo del proyecto, la plantilla que se utilizará para la gestión se encuentra en el Anexo II y considera el registro de los cambios propuestos y sus respectivas implicaciones en tiempo, costo y alcance.

2. Marco conceptual

Este apartado consta de los conceptos y definiciones necesarias para el desarrollo del TFG; son la base para su sustentación.

De manera inicial se abordan definiciones como metodología, proyecto, y software. Luego, se introducen las definiciones de calidad, calidad de software, aseguramiento y control de calidad. Posteriormente se describen las normas o mejores prácticas asociadas a desarrollo de software.

2.1. Proceso

Un proceso se define como una de actividades que se ejecutan en un orden para alcanzar un fin. Se pueden encontrar definiciones teóricas como las siguientes:

- ITIL (Information Technology Infrastructure Library) lo define como: "Un conjunto estructurado de actividades diseñadas para alcanzar un objetivo específico" (Axelos, 2011, p. 43).
- El Project Management Institute (PMI) establece que un proceso es una serie sistemática de actividades dirigidas a producir un resultado final de tal forma que se actuará sobre una o más entradas para crear una o más salidas (PMI, 2013).

En el campo del desarrollo de software se llevan a cabo diferentes procesos tales como: requerimientos, diseño, construcción, pruebas y liberación. La ejecución de esos procesos va a depender de las organizaciones.

2.1.1. Documentación de procesos

La notación aceptada para la documentación de procesos es el *Business Process Modeling and Notation* (BPMN) creado por el *Object Management Group* (OMG). Esta se centra en estandarizar procesos en una notación gráfica que sea entendible por todas las personas relacionadas (OMG, 2011)

La OMG (2011) define una serie de elementos básicos que se deben conocer para la documentación de procesos. En la tabla 3, se muestran algunos elementos del BPMN.

Elementos	Descripción	Notación
Eventos	Algo que ocurre durante el curso de un proceso. Afectan el flujo del proceso y generalmente tiene asociado una causa.	

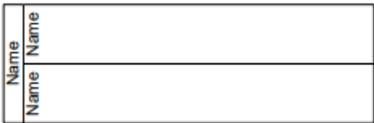
Elementos	Descripción	Notación
Actividad o tarea	Son las actividades que deben ser ejecutadas por algún rol en el transcurso del proceso.	
Compuertas	Es utilizada para unir o separar la secuencia de un proceso, dependiendo de las condiciones que ocurran.	
Flujo de secuencia	Se utiliza para mostrar el orden y secuencia del proceso.	
Pool	Es un contenedor de actividades o conjunto de tareas de uno o varios roles.	
Lane	Es una subsección dentro de un pool que contienen las actividades o conjuntos de actividades de un rol.	

Tabla 3. Elementos básicos de la notación BPMN. Fuente: OMG (2011).

2.2. Proyecto y administración de proyectos

Kerzner (2017), indica que se considera proyecto al conjunto de actividades que cumplan con las siguientes características:

- Tiene un objetivo específico enfocado en generar valor.
- Tiene definido un tiempo de inicio y fin.
- Tiene un presupuesto definido.
- Involucra recursos humanos y no humanos.

Por otra parte, el Project Management Institute (PMI) (PMI, 2013), da una definición de proyecto resumida en “esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único” (p. 3). Generalmente los proyectos son actividades que terminan en un nuevo o mejorado producto, servicio, ambiente o proceso (Kerzner, 2002).

A partir de la definición de proyecto, toda la aplicación de conocimiento, habilidades, herramientas y técnicas que se realice en las actividades de proyectos se conoce como administración o gestión de proyectos (PMI, 2013).

Según PMI (2013), en la administración de proyectos se involucran etapas como: iniciación, planeación, ejecución, monitoreo y control y cierre de los proyectos. Durante esas etapas se tienen que gestionar los elementos que involucra un proyecto, como los son:

- Alcance.
- Calidad.
- Cronograma.
- Presupuesto.
- Recursos y riesgos.

Los elementos anteriores se conocen como restricciones y el objetivo de administrar un proyecto es poder cumplir con las restricciones desde inicio del proyecto hasta el cierre. Con esto se busca que los proyectos concluyan con todos los requerimientos y cumpla con las expectativas de los interesados.

2.3. Software

El software se entiende como un sistema computacional que puede ejecutarse en algún dispositivo electrónico, principalmente móviles o computadoras. El ISO (2010), define el término software de tres maneras:

- Es la totalidad o parte de los programas, procedimientos, reglas y documentación asociada de un sistema de procesamiento de información.
- Programas de computadoras, procedimientos, documentación y datos relacionados con el funcionamiento de un sistema informático.
- Programa o conjunto de programas que se ejecutan en una computadora.

En la actualidad el software está presente en todas partes, ya que permite la delegación de procesos mentales de alta sofisticación en sistemas computacionales (Wands, 2007).

Las formas más comunes de software pueden ser las aplicaciones web, que pueden ser utilizadas con internet y herramientas de navegación, o aplicaciones para móviles y computadoras. Estas se pueden encontrar como sitios web, aplicaciones de comercio electrónico, juegos y muchas otras soluciones de software útiles para el día a día (Malan, 2020).

Las empresas u organizaciones que desarrollan software tienen que enfrentar retos para entregar productos más rápido, con menos riesgo y sin sacrificar la calidad (Meulen, 2017).

En todo ese proceso se tiene que procurar que las expectativas de los clientes, respecto al software, sean aceptables.

2.4. Requerimientos de software

Un requerimiento se puede tomar como una característica o propiedad que se tiene que exponer y explicar para que se pueda resolver un problema del mundo real (Bourque y Fairley, 2014).

Los requerimientos comúnmente responden a las necesidades y las expectativas de un determinado grupo de interesados, para identificarlas se tienen que aplicar procesos de obtención, análisis, especificación y validación, para que los proyectos de software no sean altamente vulnerables (Bourque y Fairley, 2014).

2.4.1. Requerimientos de software según SWEBOK

La guía *Software Engineering Body of Knowledge Guide* (SWEBOK), de manera general, describe conocimiento aceptado acerca de la ingeniería de software, enfocándose en 15 áreas de conocimiento (KA) (Bourque y Fairley, 2014). De las 15 áreas que cubre el SWEBOK, se utilizará principalmente el área de requerimientos de software, pruebas de software y calidad de software.

Bourque y Fairley (2014) indica que existen requerimientos tanto de proceso como de productos, a continuación, se describe cada una:

- **Requerimientos de producto:** Son los requerimientos sobre el software que será desarrollado. Este tipo de requerimiento puede generar requerimientos de proceso de forma implícita. Por ejemplo, en un sistema de matrícula se debe verificar que el estudiante cumpla con los requisitos antes de registrar un curso.
- **Requerimientos del proceso:** Son restricciones en el desarrollo del software. Estos requerimientos pueden ser impuestos por los interesados o por un tercero. Por ejemplo, el software tiene que estar escrito en Java o alguna regulación.

También, Bourque y Fairley (2014) describe que, dentro de los requerimientos del producto, existen requerimientos funcionales y no funcionales, estos se describen a continuación:

- **Requerimientos funcionales:** describe las funciones que el software tiene que realizar.

- **Requerimientos no funcionales:** Son restricciones o requerimientos de calidad, estos se pueden clasificar en desempeño, mantenibilidad, seguridad, fiabilidad y entre otros. La solución de software, independientemente de las funcionalidades se tiene que ajustar a estos requerimientos.

Para garantizar que se cumpla el objetivo de un proyecto, los requerimientos deben ser gestionados. El propósito de la gestión de requerimientos es eliminar toda la ambigüedad posible y garantizar que el resultado refleja las necesidades y expectativas de los interesados. Bourque y Fairley (2014) en el SWEBOOK recomiendan los siguientes procesos:

- **Obtención de requerimientos:** se refiere a la procedencia de los requerimientos de software y el método que se utilizará para recolectarlos. Desde aquí se empieza a comprender la situación y el problema que existe. En esta etapa se identifican las partes interesadas y se establecen relaciones entre el equipo de desarrollo y el cliente.
- **Análisis de requerimientos:** En esta etapa se busca identificar y resolver conflictos con los requerimientos, identificar el alcance del software y su interacción y elaborar los requerimientos del sistema para derivar los requerimientos de software.
- **Especificación de requerimientos:** se refiere a la confección de uno o más documentos que contenga los requerimientos a detalle. Este puede ser revisado, evaluado y aprobado sistemáticamente.
- **Validación de requerimientos:** Los documentos de requerimientos tienen que someterse a procedimientos de validación y verificación. Con esto se busca garantizar el entendimiento de los requisitos. Además, que el documento cumpla con los estándares de la organización y sea comprensible, consistente y completo.

Los procesos de requerimientos permiten encontrar el motivo raíz por el cual se desarrolla una solución, de esta manera, se puede garantizar que el producto cumple con las expectativas de los interesados.

2.4.2. Requerimientos según ISO/IEC/IEEE 29148:2018(E)

Esta norma detalla el tratamiento unificado de los procesos y productos involucrados en la gestión de requerimientos durante todo el ciclo de vida de los sistemas y el software. Proporciona detalles para la construcción de requisitos textuales bien formados, para incluir características y atributos, en el contexto de la ingeniería de sistemas y software (ISO, 2018).

La ingeniería de requerimientos se enfoca en el levantamiento, desarrollo, análisis, verificación, validación, comunicación, documentación y gestión de requerimientos. (ISO, 2018).

Para los requerimientos individuales este ISO define las siguientes características:

- Necesarios: El requisito define capacidad, característica, restricción y/o factor de calidad esencial.
- Apropriados: La intención específica y la cantidad de detalle del requisito es apropiado para los involucrados.
- No ambiguos: El requerimiento debe tener una única interpretación.
- Completo: Describe suficientemente la capacidad, características, restricción o factor de calidad sin necesidad de utilizar otra información.
- Singular: El requerimiento solo involucra una capacidad, característica, restricción o factor de calidad.
- Factible: El requerimiento tiene que estar dentro de las restricciones del proyecto (por ejemplo, costo o tiempo) y con un nivel de riesgo aceptable.
- Verificable: El requisito está estructurado y redactado de manera que su realización pueda verificarse, a satisfacción del cliente. La verificabilidad se mejora cuando el requisito es medible.
- Correcto: El requisito es una representación precisa de la necesidad de la entidad a partir de la cual se transformó.
- Conforme: Los artículos requerimientos se ajustan a una plantilla y estilo estándar aprobado, cuando corresponda.

Esta norma también indica que los conjuntos de requerimientos deben ser completos, consistente, factible, comprensible y puede ser validado.

Al momento de documentar un requerimiento, esta norma describe los elementos que se deben tomar en cuenta:

- Identificador.
- Número de versión.
- Dueño del requerimiento.
- Prioridad.
- Los riesgos asociados.

- La razón porqué se necesita el requisito.
- Dificultad asociada al requerimiento (por ejemplo, fácil, nominal, difícil).
- Tipo de atributo que caracterice al requerimiento, como funcional o desempeño, interfaz, requerimientos de proceso, calidad, usabilidad, requerimientos de factores humanos, entre otros.

Como guía para documentar los requerimientos de tipo funcional, la norma indica que deben ser redactado de alguna de las siguientes maneras:

- **[condición][sujeto][acción][objeto] [restricción de acción]**: Por ejemplo, en el estado marítimo 1 **[condición]**, el sistema de radar **[sujeto]** detectará **[acción]** objetivos **[objetos]** una distancia de hasta 100 millas náuticas **[restricción de acción]**.
- **[sujeto][acción] [restricción de acción]**: Por ejemplo, el sistema de factura **[sujeto]** debe mostrar las facturas pendientes de los clientes **[acción]** en orden ascendente según la fecha de vencimiento de la factura **[restricción de acción]**.

Esta norma indica que el proceso de requerimientos y sus etapas debe ser iterativa, pues esto permite obtener toda la información necesaria para entender la necesidad real.

2.5. Requerimientos de accesibilidad de software

Antes de hablar de accesibilidad se tiene que hacer una breve mención de otro término llamado **usabilidad**.

La ISO 9241-11 sobre la ergonomía en la interacción humano-sistema., indica que la usabilidad involucra la efectividad, eficiencia y satisfacción de la interacción del usuario con el objeto de interés. Para el caso del software, una usabilidad no adecuada o baja causa que los usuarios no puedan utilizarlo (ISO, 2018).

También, la usabilidad es la inclusión de diseño de la experiencia del usuario, para crear productos que sean efectivos, eficientes y satisfactorios. Esto incluye, elementos generales que afectan a todos y no afectan desproporcionadamente a las personas con discapacidad. La práctica y la investigación de usabilidad a menudo no abordan suficientemente las necesidades de las personas con discapacidad (W3C, 2019).

Con el concepto general de usabilidad, se puede introducir el término **accesibilidad**. No son iguales, pero existe una relación entre ambos.

La accesibilidad o requerimientos de accesibilidad se refieren a aspectos discriminatorios relacionados con la experiencia de usuario equivalente para personas con discapacidad. La accesibilidad significa que las personas con discapacidad pueden percibir, comprender, navegar e interactuar igualmente con sitios web y herramientas (W3C, 2019).

La usabilidad se centra en la experiencia de usuario, este concepto converge con la accesibilidad. Cuando una persona con discapacidad no pueda utilizar una herramienta o software porque no es accesible, este usuario tendrá una muy pobre experiencia en el uso.

La *World Wide Web Consortium* (W3C) brinda recomendaciones para incluir la accesibilidad en las herramientas. Muchos de los requerimientos de accesibilidad mejoran la usabilidad, especialmente cuando existen limitaciones de un usuario. Por ejemplo, utilizar contraste permite a los usuarios utilizar una aplicación en el móvil tanto en la luz del día como en una habitación oscura (W3C, 2019).

Según la W3C los requerimientos de accesibilidad deben estar en dos partes, estas se detallan a continuación:

- Requerimientos técnicos centrados en la construcción del código: Se tiene que asegurar que el producto de software funcione bien con tecnologías de asistencia como los lectores de pantalla o comandos de voz.
- Requerimientos de interacción con el usuario y diseño visual: Un diseño no adecuado puede causar problemas de accesibilidad. Por ejemplo, las instrucciones y comentarios en los formularios son de beneficio para el usuario.

Se busca que los proyectos y productos de software sean de calidad, por lo tanto, se tiene que incluir la accesibilidad para que se puedan cumplir con las expectativas de los usuarios con discapacidad, cuando sean potenciales usuarios de la solución.

2.6. Metodología

Una metodología, en términos generales, consiste en un conjunto de procedimientos o pasos que se ejecutan para alcanzar un objetivo propuesto. Algunas definiciones son las siguientes:

- Conjunto de métodos que se siguen en una investigación científica o una exposición doctrinal (Diccionario de la Real Academia Española, 2019).

- Un conjunto de métodos, reglas y postulados empleados por una disciplina; un procedimiento particular o conjunto de procedimientos (Merriam-Webster Dictionary, 2019).
- Es un sistema de prácticas, técnicas, procedimientos y reglas usados en una disciplina (ISO, 2010).

Las definiciones anteriores hacen referencia al concepto de metodología en general, que se pueden enfocar en diferentes campos como el desarrollo de software.

2.6.1. Metodología de desarrollo de software

Una metodología de desarrollo de software “es un marco para planificar y controlar todo el proceso de construcción de una aplicación o un sistema” y tiene “un conjunto de reglas para planificar todas las etapas del desarrollo de software” (Sivaranjani y Rajeswari, 2017, pp. 1).

Del mismo modo, es un proceso abstracto utilizado para el desarrollo de software, que incluye actividades como: análisis de requisitos, codificación, pruebas, mantenimiento, entre otros (Zima, 2015).

Tomando la definición de metodología, administración de proyectos y software, se puede hacer una relación entre los conceptos y determinar que una metodología de desarrollo de software es la aplicación de conocimiento, habilidades, herramientas, técnicas y procedimientos utilizados para la producción de software.

En el desarrollo de software, a lo largo del tiempo, han surgido diferentes metodologías, pues el éxito en la industria radica en cuál es mejor para unir a las personas, los procesos, la tecnología y las herramientas de manera que se pueda garantizar la calidad. De este modo, una metodología es parte integral del desarrollo de software (Manawadu et al, 2013).

Como parte de las metodologías que existen se pueden mencionar las tradicionales (por ejemplo, cascada) y el enfoque ágil (por ejemplo, Scrum).

2.6.2. Metodologías de desarrollo ágil

Las metodologías de desarrollo tradicionales (cascada, espiral, entre otros) están siendo remplazadas por metodologías de desarrollo de software ágiles. Una metodología ágil está caracterizada por el desarrollo iterativo, integración continua de código y la habilidad para manejar requerimientos de negocios cambiantes (Livermore, 2007).

El tiempo juega un papel importante en el desarrollo de un proyecto, pues entre más tiempo se emplea para su desarrollo se necesitará más de otros recursos (por ejemplo, presupuesto) para finalizarlo. En estas condiciones las metodologías ágiles permiten a los equipos de proyectos de las organizaciones desarrollar proyectos de alta calidad en tiempos reducidos (Livermore, 2007).

Existen diferentes metodologías ágiles, de las cuales se pueden mencionar las siguientes.

- *Extreme programming.*
- Scrum.
- *Crystal methods.*
- *Feature driven development.*
- Kanban.

En una encuesta aplicada a más de 2000 practicantes de enfoques ágiles, llevada a cabo por el Scrum Alliance, reporta que 89% de los encuestados indican que Scrum es el enfoque ágil o al menos uno de los enfoques utilizados en las organizaciones donde se desempeñan (Scrum Alliance, 2016).

Scrum “se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo” (Sutherland y Schwaber, 2013, p. 4).

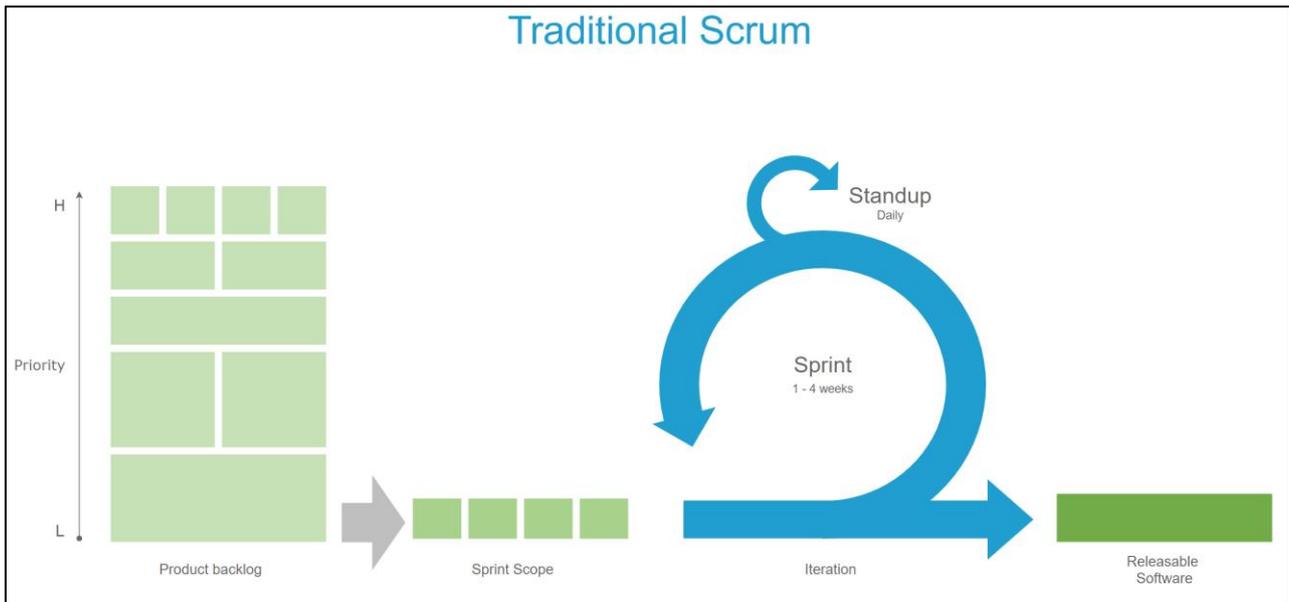


Figura 3. Ciclo de desarrollo con Scrum tradicional. Fuente: Tomado de Requena (2018).

El Scrum Alliance (2017) reporta que los gerentes de proyectos indican que scrum les permite entregar valor al cliente, flexibilidad, capacidad de respuesta, calidad, transparencia, satisfacción y compromiso de los equipos de proyectos, innovación, entre otros.

La teoría de Scrum envuelve una serie de elementos para obtener los beneficios del uso de este enfoque ágil. Scrum involucra equipos, eventos y artefactos. Además, es un enfoque que se puede adaptar según las necesidades de las organizaciones, no existe una forma única de aplicarlo.

Sutherland y Schwaber (2017) detallan los elementos que conforman Scrum, estos componentes se listan a continuación.

- Equipo de scrum: La idea de Scrum es buscar que los equipos sean auto-organizados.
 - Dueño del producto (*Product owner*): Busca maximizar el valor del producto y se encarga de administrar la lista del producto.
 - Equipo de desarrollo (*Development team*): Son los encargados de entregar incrementos terminados del producto que puedan ser potencialmente puestos en producción. Deben ser auto organizados y multifuncionales.
 - Scrum master: es un líder que busca que se lleven a cabo las prácticas de Scrum, además de buscar que se entienda la filosofía.
- Eventos de Scrum: Buscar regularidad y minimizar reuniones no definidas.

- El sprint: Bloque de tiempo de un mes o menos donde se crea un incremento terminado.
- Planificación del sprint: Se define los elementos de la lista del producto (*product backlog*) en que se trabajará en el sprint.
- Objetivo del sprint: es una meta y razón del porqué se desarrolla el incremento.
- Scrum diario (*daily scrum*): Se lleva a cabo todos los días que involucra el sprint. Se enfoca en dar a conocer avances, problemas y trabajo que se realizará en las siguientes 24 horas.
- Revisión de sprint: Es necesario para determinar el estado del incremento al final del sprint. Se busca la retroalimentación y la colaboración.
- Retrospectiva del sprint: Tiempo para que el equipo de desarrollo se inspeccione a sí mismo y puedan determinar oportunidades de mejorar para el siguiente sprint.
- Artefactos de scrum: representa trabajo o valor en diversas formas, apoya principalmente la transparencia y facilita la inspección.
 - Lista del producto (*product backlog*): es la lista donde se acuerda todo lo que debe contener el producto. Son los requisitos del producto.
 - Lista de pendientes del sprint (*sprint backlog*): son los elementos de la lista del producto seleccionados para el sprint.

2.7. Modelo de madurez y capacidad

Los modelos de capacidad y madurez (CMM, del inglés *Capability Maturity Model*) son herramientas que ayudan a analizar y mejorar los procesos internos y el desempeño organizacional que se puede aplicar en una amplia variedad de industrias y disciplinas (Stewart, 2018).

Los CMM se centran en mejorar los procesos de una organización. Contiene los elementos esenciales de los procesos eficaces de una o más disciplinas y describen un camino evolutivo de mejora desde procesos ad hoc e inmaduros a procesos disciplinados y maduros con calidad y eficacia mejoradas (CMMI Institute, 2010).

Un modelo de madurez está dado por diferentes niveles que muestran el camino hacia la mejora de los procesos. Los niveles también indicarán el resultado de evaluar una organización o parte de esta (como equipos de proyectos) (CMMI Institute, 2010).

Los niveles se pueden evaluar en madurez y capacidad (CMMI Institute, 2010). A continuación, se describe cada una:

- Niveles de capacidad: Se centran en un área de proceso específica, esto puede involucrar uno o varios procesos. Implementar mejoras de capacidad en los procesos se conoce como representación continua.
- Niveles de madurez: Se centra en mejorar de manera incremental las áreas de procesos de una organización. Realizar este tipo de mejoras continuas se conoce como representación por etapas.

Las etapas que propone el CMMI son seis, en la Tabla 4. Se muestra la relación entre la capacidad y la madurez.

<i>Nivel</i>	<i>Representación continua Niveles de capacidad</i>	<i>Representación por etapas Niveles de madurez</i>
Nivel 0	Incompleto	
Nivel 1	Realizado	Inicial
Nivel 2	Gestionado	Gestionado
Nivel 3	Definido	Definido
Nivel 4		Gestionado cuantitativamente
Nivel 5		En optimización

Tabla 4. Comparación de los niveles de capacidad y de madurez. Fuente: (CMMI Institute, 2010).

De este modo los niveles de capacidad son consecuencia de mejorar los procesos de un área, y los de madurez son consecuencia de la mejora de procesos en varias áreas de una organización.

EL CMMI detalla los diferentes niveles tanto de capacidad como de madurez. En la Tabla 5 se resume cada nivel.

<i>Nivel</i>	<i>Representación continua Niveles de capacidad</i>	<i>Representación por etapas Niveles de madurez</i>
Nivel 0	Incompleto Procesos que no se realizan o se realizan parcialmente.	No existe registro de madurez y capacidad.

Nivel	Representación continua Niveles de capacidad	Representación por etapas Niveles de madurez
Nivel 1	<p>Realizado</p> <p>El proceso se ejecuta con el trabajo necesario para satisfacer las metas específicas del área del proceso.</p>	<p>Inicial</p> <p>Los procesos son <i>ad hoc</i>. No hay un entorno estable para dar soporte a los procesos.</p>
Nivel 2	<p>Gestionado</p> <p>El proceso se planifica, tiene personal calificado, involucra a los interesados, se monitorea, se controla, se revisa y se garantiza que se ejecutará del mismo modo cuando estén bajo presión.</p>	<p>Gestionado</p> <p>En los proyectos todos los procesos son gestionados. Los proyectos tienen personal calificado, involucra a los interesados, se monitorea, se controla y se revisa.</p>
Nivel 3	<p>Definido</p> <p>El proceso es gestionado y se ajusta al conjunto de procesos estándar de la organización, forma parte del activo de procesos.</p>	<p>Definido</p> <p>Los procesos están bien caracterizados y comprendidos, y se describen en estándares, procedimientos, herramientas y métodos. Los proyectos adaptan sus procesos al conjunto estándar de procesos de la organización.</p>
Nivel 4	<p>Se encuentra en el nivel gestionado cuantitativamente de madurez</p>	<p>Gestionado cuantitativamente</p> <p>La organización y los proyectos poseen objetivos cuantitativos en cuanto a calidad y rendimiento.</p>

Nivel	Representación continua Niveles de capacidad	Representación por etapas Niveles de madurez
Nivel 5	Se encuentra en el nivel optimizado de madurez	En optimización Se implementan acciones de mejorar continua en los proyectos y sus procesos. Las decisiones estas dirigidas con información cuantitativa de los objetivos de negocio y rendimiento.

Tabla 5. Resumen de los niveles de madurez y capacidad según el CMMI. Fuente: Adaptado de (CMMI Institute, 2010).

Realizar evaluaciones de los procesos y proyectos de una organización permite determinar el estado actual, el estado al cual aspira llegar, y es el punto de partida para planificar las actividades que se tienen que realizar para conseguirlo.

2.8. Métricas

La ISO (2003) hace referencia a las métricas como una combinación de dos o más medidas o atributos. La ISO (2010) describe a las métricas como una medida cuantitativa o el grado en el cual un sistema, componente o proceso posee un atributo dado.

Por otra parte, ITIL define las métricas como algo que se mide y se informa para ayudar a administrar un proceso, servicio de TI o actividad (Axelos, 2011).

Tomando en cuenta las definiciones se puede determinar que las métricas se aplican para diferentes campos de estudio, para el contexto actual del desarrollo de software se utilizará el concepto de métricas de calidad principalmente en aseguramiento y control de calidad.

Una métrica de calidad describe un atributo de un proyecto o producto y cómo el proceso de control de calidad lo va a medir. Algunos ejemplos de métricas de calidad incluyen rendimiento a tiempo, control de costos, frecuencia de defectos, tasa de fallas, disponibilidad, confiabilidad y cobertura de prueba (PMI, 2013).

La IEEE (2009) indica que las métricas de calidad de software son una función cuyas entradas son datos de software y cuya salida es un valor numérico único que puede interpretarse como el grado en que el software posee un atributo que afecta su calidad.

Se establecen métricas con el propósito de hacer evaluaciones a través del ciclo de vida del software que permita determinar si los requisitos se alcanzarán. Esto permite disminuir la subjetividad que involucra la calidad, brindando información base para la toma de decisiones dentro del proceso de desarrollo de software (IEEE, 2009).

A continuación, se muestra una serie de beneficios que, según la IEEE (2009), se pueden obtener al implementar métricas de calidad de software:

- Evaluación del cumplimiento de las metas de calidad.
- Establecer desde el inicio requisitos de calidad para un sistema.
- Establecer los criterios de aceptación.
- Evaluar el nivel de calidad alcanzado respecto a los requerimientos establecidos.
- Detectar anomalías o puntos de potenciales problemas.
- Predecir el nivel de calidad que se logrará.
- Monitorear los cambios en la calidad cuando se requiere modificaciones.
- Evaluar la facilidad de cambio en el sistema durante la evolución del producto.
- Validar el conjunto de métrica propuesta.

2.9. Calidad

Según IEEE (2014), la calidad se puede definir como el grado en que un producto o proceso cumple con los requisitos establecidos; sin embargo, la calidad depende del grado en que esos requisitos establecidos representen con precisión las necesidades, deseos y expectativas de los interesados. Del mismo modo, el ISO (2005) define la calidad como “el grado en el que un conjunto de características inherentes cumple con los requisitos” (p. 8).

Los enfoques de la calidad moderna buscan minimizar la variación y entregar resultados a los interesados que cumplan con los requerimientos definidos (PMI, 2013).

Tomando en cuenta las definiciones anteriores se puede determinar que estas tienen objetivos en común; la calidad se mide respecto a las expectativas que se convierten en requerimientos, sin embargo, de la definición del IEEE (2014) se puede concluir que la calidad inicia desde los requerimientos. Con requerimientos que no reflejan los intereses de los involucrados, no se puede lograr la calidad.

La calidad no es trivial, por esta razón el PMI indica que tiene que existir la gestión de la calidad que donde se incluyen los procesos y actividades que determinen políticas, objetivos y responsabilidades de calidad y cómo serán satisfechos (PMI, 2013).

La gestión de calidad tiene tres procesos principales que son: La planificación de calidad, el aseguramiento de calidad y el control de la calidad (PMI, 2013). La responsabilidad de la gestión de calidad será asegurar que se cumplirán los requerimientos.

A partir de las definiciones de calidad, se espera que las organizaciones tengan objetivos de calidad que permitan cumplir con las expectativas de los interesados.

Para cumplir con los objetivos de calidad la ISO (2015) propone una serie de principios que se detallan a continuación:

- **Enfocado al cliente:** No se limita a solo cumplir los requerimientos del cliente sino exceder sus expectativas. Esto permitirá aumentar la satisfacción de los clientes, lealtad y reputación de la organización.
- **Liderazgo:** Para transmitir los objetivos de calidad, los líderes en todos los niveles deben propiciar las condiciones para que las personas se involucren y se comprometan con la organización. Eficiencia y eficacia, mejor coordinación, desarrollo y mejora de las capacidades de la organización son algunos de los beneficios.
- **Involucrar a las personas:** La participación de las personas en los procesos de calidad impacta en la entrega de valor de las organizaciones, por eso es necesario involucrar y comprometer a las personas en todos los niveles para crear ese valor.
- **Enfocar en procesos:** Procesos interrelacionados que funcionan con coherencia permiten lograr resultados consistentes y predecibles. Trabajar de esta manera permite enfocarse en los procesos que son clave e identificar oportunidades de mejora. Además, se puede tener un mejor control en el uso de recursos para lograr los objetivos.
- **Mejora:** La mejora continua es la característica principal de las organizaciones exitosas. Mejorar es importante para mantener los niveles de desempeño, reaccionar ante cambios externos e internos y generar nuevas oportunidades.
- **Decisiones basadas en evidencia:** Generalmente la toma de decisiones es un proceso complicado que involucra incertidumbre y subjetividad. Ante estas situaciones, los hechos, evidencia y el análisis de datos permiten tener posiciones más objetivas al momento de tomar decisiones.
- **Administración de relaciones:** Involucrar a todas las partes involucradas y administrar esas relaciones son clave para alcanzar el éxito en los procesos de

calidad. En general se puede tener un común entendimiento de las metas y el valor esperado por las partes interesadas.

Las descripciones o principios de calidad, mencionados anteriormente, son de carácter general, esto significa que se pueden aplicar en diferentes áreas donde se requiera que las salidas sean de calidad, sean estos proyectos, productos, procesos o servicios.

2.9.1. Calidad de software

Específicamente en el área de software, según IEEE (2014), la calidad es el conjunto entero de atributos que tiene un software para satisfacer los requerimientos expresados o implícitos de los interesados, que generalmente involucra requerimientos funcionales y de desempeño. También, es el grado en que un software posee una combinación de atributos de calidad deseados (IEEE, 2009).

Se pueden entender los atributos de calidad como las características del software, estos atributos de calidad se originan de los requerimientos, con cumplimiento de los requerimientos se logrará cumplir con los atributos de calidad.

El *Foundation Level* del ISTQB (*International Software Testing Qualifications Board*) describe que, en la gestión de calidad del software, se tiene que tomar en cuenta lo que significa calidad para los clientes, desarrolladores, *tester* y cualquier otro involucrado. Es importante que todos establezcan y acuerden las especificaciones y expectativas (Graham et al, 2008). Este es el proceso más importante, pues es la base del resultado esperado.

Algunas características de calidad que presenta el ISQTB son:

- La calidad es medida según los atributos del producto.
- La calidad tiene aspectos subjetivos y no necesariamente cuantitativos, lo que permite que sea ajustable según las expectativas.
- La calidad se basa en un proceso de producción correcto y busca alcanzar requerimientos definidos. Se puede medir aplicando pruebas, inspección y análisis de fallos y defectos.
- La calidad tiene relación con el retorno de la inversión, donde se espera que el tiempo, esfuerzo y costo generen beneficios. La calidad no es trivial, por esta razón el PMI indica que tiene que existir la gestión de la calidad donde se incluyen los procesos y actividades que determinen políticas y objetivos.

2.9.2. Aseguramiento de la calidad (QA) de software

El ISO (2005) indica que aseguramiento de calidad se enfoca en “proporcionar confianza que se cumplirán los requisitos de calidad” (p. 10). El aseguramiento es una de las actividades que se tiene que coordinar en la gestión de la calidad.

A nivel de desarrollo de software, la IEEE (2014) define aseguramiento de la calidad como el conjunto de actividades que definen y evalúan el estado de los procesos de software para proporcionar la evidencia de que estas sean apropiadas y producen productos de calidad adecuada para los fines previstos.

Según CMMI Institute (2010), en el *Capability Maturity Model Integration* (CMMI) el aseguramiento de la calidad tiene que ser parte de un proceso planificado y sistemático para garantizar a las partes interesadas los estándares, prácticas, procedimientos y métodos. El CMMI describe las siguientes actividades de QA:

- Evaluar objetivamente los procesos y los productos de trabajo frente a las descripciones del proceso, los estándares y procedimiento aplicables.
- Identificar y documentar las no conformidades.
- Proporcionar realimentación al personal del proyecto y los gerentes sobre los resultados de las actividades de QA.
- Asegurar que se gestionan las no conformidades.

Además, el CMMI describe que al aplicar un proceso de QA se da soporte a la entrega de productos de alta calidad, esto proporciona a los involucrados de los proyectos, en todos los niveles, la visibilidad apropiada y la realimentación sobre los procesos y los productos de trabajo asociados, en toda la vida del proyecto. QA asegura que los procesos planificados se implementen.

El CMMI recalca que se requiere objetividad en las evaluaciones de QA, pues de esto depende el éxito de los proyectos. De este modo, recomienda que las personas encargadas de realizar actividades de QA, deben estar separadas de quienes desarrollan o dan mantenimiento al producto.

Otra recomendación que indica el CMMI, es que el QA debe iniciar en las fases tempranas de los proyectos, para que se puedan definir los planes, procesos, estándares y procedimientos que aportarán valor al proyecto, satisfará los requisitos y cumplirá con las políticas de la organización.

Sin hacer referencia a un modelo de ciclo de vida del software, IEEE (2014) destaca la importancia de iniciar el proceso de QA desde las fases tempranas del proyecto (Ver Figura 2). Además de garantizar que se ejecutarán en todas las etapas siguientes del proyecto.

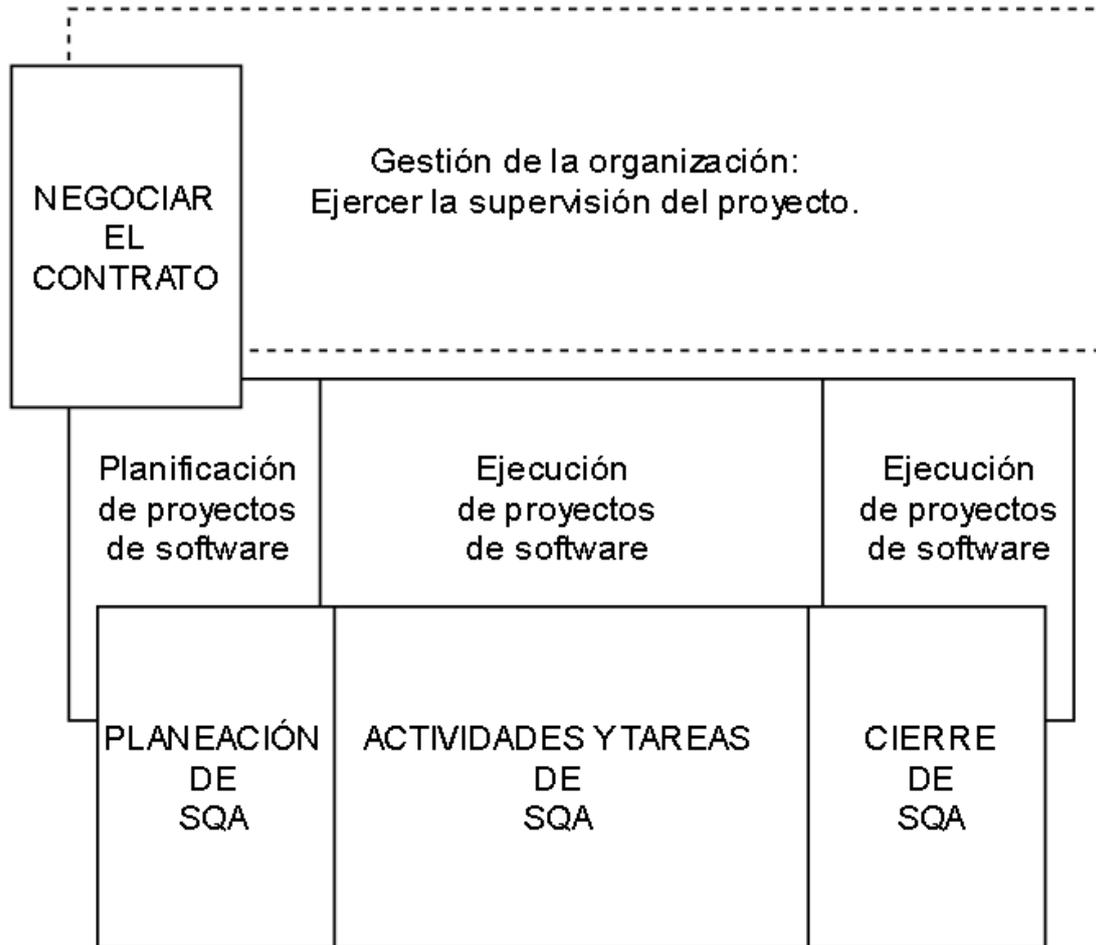


Figura 4. Incorporación de actividades de QA al ciclo de desarrollo de proyectos. Fuente: Tomado de IEEE (2014).

La IEEE (2014) organiza las salidas o resultado de QA en tres grupos.

- Proceso de QA.
 - El proceso es implementado.
 - Se desarrolla la estrategia para guiar el proceso.
 - Se planean y ejecutan las actividades de QA.
 - Se conforma la evidencia y se mantiene.
- Aseguramiento del producto.
 - Se evalúa el cumplimiento de los productos con los requisitos establecidos.
 - Los problemas y no conformidades se identifican y se registran.
- Aseguramiento del proceso.

- Se evalúa el cumplimiento de las actividades con los estándares y procedimientos aplicables.
- La efectividad del proceso es evaluada y se sugieren mejoras.
- Se identifican y registran los problemas y las no conformidades.

2.9.2.1. *Proceso de aseguramiento de calidad de software (SQA) según el IEEE Std 730-2014*

Los requisitos para iniciar, planificar, controlar y ejecutar los procesos de Aseguramiento de calidad del software de un proyecto de desarrollo o mantenimiento de software se establecen en esta norma. Esta norma está armonizada con el proceso del ciclo de vida del software de ISO/IEC/IEEE 12207: 2008 y los requisitos de contenido de información de ISO/IEC/IEEE 15289: 2011 (IEEE, 2014).

La Norma para Proceso de Aseguramiento de Calidad de Software (*IEEE Standard for Software Quality Assurance Processes – IEEE Std 730-2014*), indica que el proceso de aseguramiento de calidad de software busca especificar actividades y tareas que permite a los proveedores de software producir, recopilar evidencia que sirva como base de confianza de calidad, o sea, que se garantice que el producto de software cumple con los requisitos establecidos (IEEE, 2014).

El IEEE Std 730-2014, contiene los aspectos a tomar en cuenta para proponer un proceso de aseguramiento de calidad, para esto menciona que se deben tomar en cuenta tres elementos:

- Implementar un proceso de SQA.
- Aseguramiento del producto.
- Aseguramiento del proceso.

Implementar un proceso de SQA

El propósito de este subproceso de SQA es definir y establecer procesos que se aplicarán a los proyectos de software para garantizar el cumplimiento de los requerimientos. Además, permite tener mediciones de calidad para la toma de decisiones en elementos críticos de un proyecto como costo, cronograma, calidad y riesgo. Para llevarlo a cabo se tienen que definir los roles, métodos y procedimientos (IEEE, 2014).

Al implementar un proceso de SQA, se debe tener como resultado:

- Un rol de SQA.
- Un proceso general de SQA para los proyectos.
- Una política para los roles y responsabilidad de SQA.
- Un método para garantizar que se aplica el proceso de SQA, principalmente en la relación resultado-retroalimentación.
- Un método para compartir lecciones aprendidas.
- Y se define las personas que ejecutarán el rol (es) de SQA.

El proceso de SQA tiene que estar relacionado con los otros procesos relevantes al proyecto para garantizar que se puedan alcanzar los objetivos. Verificación y validación de software siempre están relacionados con el proceso SQA. Hacer estas relaciones permite eliminar actividades y tareas redundantes.

Actividades de aseguramiento del producto

El producto de un proyecto son el software y la documentación relacionada. Un aspecto importante del SQA es garantizar (con evidencias) que esos productos son de calidad.

Las actividades que se deben llevar a cabo dentro del aseguramiento del producto son:

- Evaluar planes de conformidad: Busca que al momento del contrato se haya considerado y documentado todos los elementos que puedan afectar al producto.
- Evaluar la conformidad de producto: Se tiene que determinar el grado en que los productos cumplen con los requerimientos establecidos, de manera interna.
- Evaluar la aceptabilidad del producto: En conjunto con el dueño del producto se evalúa el grado de cumplimiento de los requerimientos y si es aceptable.
- Evaluar el soporte del ciclo de vida del producto: Identificar el nivel de participación que deben tener los involucrados en el ciclo de vida del desarrollo del software.
- Medir los productos: Para garantizar que se alcanzó el objetivo del proyecto.

Aseguramiento del proceso

Se enfoca en evaluar que el proceso utilizado para desarrollar, instalar, operar y mantener el software está conforme al contrato y cualquier regulación impuesta. Además, que es adecuado, eficiente y eficaz, esto significa que el proceso produce software conforme a los requerimientos y consideraciones organizacionales.

Las actividades de aseguramiento del proceso son las siguientes:

- Evaluar la conformidad del ciclo de vida: Revisar si los planes y procesos del ciclo de vida del proyecto se ajustan a los requerimientos de proceso establecidos en el contrato.
- Evaluar la conformidad de los ambientes: Principalmente que cuente con ambientes de desarrollo (*Software Engineering Environment - SEE*) y pruebas (*Software Testing Environment - STE*).
- En caso de uso de subcontratación: Verificar que el subcontratista cumple con los requerimientos de proceso del contrato.
- Medir el proceso.
- Evaluar las habilidades y conocimientos de los colaboradores.

2.9.3. Control de la calidad de software (verificación y validación)

En general el control de calidad gira en torno a las técnicas y actividades operativas que se utilizan para satisfacer los requisitos de calidad (CMMI Institute, 2010). Aplicar control de calidad permite monitorear y registrar resultados de las actividades de calidad, esto permite evaluar el desempeño y aplicar cambios que sean necesarios para cumplir con las especificaciones de un proyecto (PMI, 2013).

Por su parte Galin (2004) menciona que el objetivo de control de calidad (QC) es retener, de cara al cliente o usuarios, cualquier producto que no califique. De este modo, los procesos de QC toman lugar a medida que se completa el desarrollo o fabricación de un producto.

Pressman (2010) indica que el control de calidad involucra un conjunto de acciones que ayudan a asegurar que todo producto de trabajo cumple con las metas de calidad.

El Graham *et al.* (2008), indica que existen dos procesos que se deben tomar en cuenta para identificar el cumplimiento de requisitos y calidad de un software. Estos procesos son:

- **Verificación:** Confirmación, mediante pruebas y la provisión de evidencia objetiva, del cumplimiento de los requisitos especificados.
- **Validación:** Confirmación, mediante pruebas y la provisión de evidencia objetiva, del cumplimiento de los requisitos para un uso o aplicación específico.

Como parte de los beneficios que brinda la implementación de procesos de control de calidad el PMI menciona los siguientes:

- Identificar causas en resultados con calidad deficiente.
- Validar que los entregables del proyecto y el producto cumplen con los requisitos establecidos y esperados.

Hacer pruebas y mediciones acompañado con retroalimentación permite identificar mejoras cuando cualquiera de los productos del trabajo falla en el cumplimiento de las metas de calidad (Pressman, 2010).

Es importante mencionar que para la implementación de control de calidad se tienen que establecer objetivos de calidad de manera que se pueda determinar que las técnicas y actividades de calidad se están ejecutando de manera correcta y terminando con los resultados esperados. En términos generales, los objetivos se establecerán para ver si el proceso de aseguramiento de calidad y su aplicación está cumpliendo con las expectativas de las organizaciones y los otros involucrados claves de los proyectos.

2.9.3.1. Verificación y validación según la norma IEEE para la Verificación y validación de Sistemas, Software y hardware (IEEE Std 1012 -2016)

Los procesos de verificación y validación (V&V) se utilizan para determinar si los productos de desarrollo de una actividad dada cumplen con los requisitos de esa actividad y si el producto satisface su uso previsto y las necesidades del usuario. Los requisitos del proceso del ciclo de vida de V&V se especifican para diferentes niveles de integridad. El alcance de los procesos de V&V abarca sistemas, software y hardware, e incluye sus interfaces. Este estándar se aplica a los sistemas, software y hardware que se están desarrollando, manteniendo o reutilizando. Los procesos de V&V incluyen el análisis, evaluación, revisión, inspección, evaluación y prueba de productos (IEEE, 2016).

Los procesos de verificación y validación de software de esta norma son análisis de los requerimientos, diseño, construcción, integración, prueba de calificación (conformidad con

los requerimientos), pruebas de aceptación (cumplimientos de los criterios de aceptación), verificación, instalación y revisión de software, validación, operación y mantenimiento.

A continuación, se describen los procesos de verificación y validación del software que se consideran en el presente proyecto:

- **Análisis de requerimientos:** Se busca evaluar que los requerimientos sean correctos, completos, precisos, comprobables y consistentes.
- **Pruebas de calificación:** Son pruebas para verificar el cumplimiento de los requerimientos establecidos. El proceso busca evaluar que el software integrado satisface los requerimientos.
- **Pruebas de aceptación:** Se busca evaluar si el software cumple con los criterios de aceptación definidos. Con los resultados de este proceso, el cliente puede determinar si lo acepta o no.
- **Proceso de verificación:** Busca proveer evidencia sobre el cumplimiento de los requerimientos, estándares, prácticas y convenciones en todas las etapas o actividades del ciclo de desarrollo del software. Además, evalúa que se complete con éxito cada actividad del ciclo de vida y satisfaga todos los criterios para iniciar actividades de ciclo de vida posteriores (es decir, el producto se construye correctamente).
- **Proceso de validación:** Busca proveer evidencia de que el producto cumple con los requerimientos establecidos cuando se finalicen todas las etapas o actividades del ciclo de vida del proyecto. En este proceso se determina si se soluciona el problema por el cual se inició el desarrollo del software.

2.9.3.2. *Pruebas de software*

Las pruebas de software no se limitan a determinar si un software está conforme a los requerimientos, sino que es todo un proceso que se aplica durante el ciclo de vida del desarrollo. (Graham *et al.*, 2008).

Según Graham *et al* (2008), el proceso de pruebas de software involucra las siguientes actividades:

- **Planeación:** Se decide lo que se hará antes y después de ejecución. Se tiene que establecer cómo se controlará y reportará el progreso de las actividades de pruebas, entre otras actividades.

- Preparación: Se debe establecer el objetivo de las pruebas y elaborar casos y condiciones para ejecutarlas.
- Evaluación: Se tienen que evaluar los resultados obtenidos en la ejecución de las pruebas y emitir criterio acerca del comportamiento del software. Con base en estos resultados se decide el paso siguiente.

Graham *et al.* (2008) indica que la ejecución de pruebas puede hacerse de dos maneras, la primera es estática, donde no necesariamente se requiere la ejecución de código del software, sino que se hace una revisión de documentos (incluido el código fuente) y análisis estático. Y la segunda es dinámica donde se ejecuta el código y se evalúan según los casos de prueba.

El ISO (2013a) dice que el proceso de pruebas se tiene que iniciar desde el nivel organizacional, donde se debe establecer una política de pruebas y una estrategia organizacional de pruebas. Una vez se cuenta con un proceso de pruebas organizacional, se establece un proceso de gestión de pruebas que se enfoca a nivel de proyectos y por último un proceso de pruebas dinámicas que se centra en las pruebas del software o sistemas como producto.

De este modo, el ISO (2013a) indica que el proceso de gestión de pruebas involucra etapas de planeación, monitoreo y control y finalización de pruebas.

Planeación de pruebas

El propósito del proceso de planificación de pruebas es desarrollar, acordar, registrar y comunicar a las partes interesadas el alcance y el enfoque que se tomarán para las pruebas, lo que permite la identificación temprana de recursos, entornos y otros requisitos de las pruebas (ISO, 2013a).

En la etapa de planeación de pruebas, la ISO (2013a) sobre el proceso de pruebas, indica que hay nueve etapas:

- Entender el contexto: Se identifican los *ítems* de prueba e interesados
- Organizar el desarrollo del plan de pruebas: Definir las actividades de pruebas, los participantes y los horarios.
- Identificar y analizar riesgos: Identificar y clasificar los riesgos. Además. con qué o quienes están relacionados.

- Identificar enfoques de mitigación de riesgos: Se identifican los medios apropiados para tratar los riesgos, en función del tipo de riesgo, la clasificación y el nivel de exposición al riesgo.
- Diseñar las estrategias de pruebas: Se diseñará una estrategia de prueba que incluya fases de prueba, tipos de prueba, características a probar, técnicas de diseño de prueba, criterios de finalización de prueba y criterios de suspensión y reanudación. Se deben tomar en cuenta los riesgos y las limitaciones de organización, proyecto y producto. Además, se identifican las métricas para el monitoreo y control y el ambiente para las pruebas.
- Determinar los participantes y el horario: Se identifican las personas que tienen que llevar a cabo las pruebas y el tiempo en que se ejecutarán.
- Registrar el plan de pruebas: incorporar todas las salidas de las etapas anteriores en el plan de pruebas.
- Formalizar el plan de pruebas: Se debe consensuar el plan de pruebas con los interesados.
- Comunicar el plan de pruebas y habilitarlo.

Monitoreo y control de las pruebas

El propósito del Proceso de Monitoreo y Control de Pruebas es determinar si las pruebas progresan de acuerdo con el plan de Pruebas y con las especificaciones de prueba organizacionales (ISO, 2013a).

Este proceso involucra cuatro etapas:

- Preparar el ambiente de pruebas: Deben identificarse medidas adecuadas para monitorear el progreso del plan de pruebas. Además, para obtener las medidas de monitoreo y control se tiene que realizar actividades de monitoreo como reportes de estado y obtención de métricas de pruebas.
- Monitoreo: Se obtienen las medidas de pruebas. Se evalúa el progreso sobre el plan de pruebas, se identifican los desvíos del plan y se gestionan los riesgos (nuevos o cambiantes).
- Control: Se ejecutan todas las actividades necesarias para llevar a cabo el plan de pruebas. Se asignan responsabilidades, se administran los desvíos del plan de pruebas, se gestionan los riesgos y se lleva a cabo un proceso de aprobación al final de las actividades de pruebas.

- Reporte: Se da a conocer el progreso de pruebas, sobre el plan de pruebas, a los interesados y los riesgos identificados.

Finalización de pruebas

El propósito del proceso de finalización de la prueba es poner a disposición activos de prueba útiles para su uso posterior, dejar el entorno de prueba en condiciones satisfactorias y registrar y comunicar los resultados de la prueba a las partes interesadas relevantes (ISO, 2013a).

Este proceso involucra cuatro etapas:

- Guardar los activos de pruebas: Poner a disposición todos los activos de pruebas que puedan ser utilizados luego.
- Limpiar el ambiente de pruebas: El ambiente de pruebas se debe restablecer al estado predefinido luego de terminar las actividades de pruebas.
- Identificar lecciones aprendidas: Se registran las lecciones aprendidas de los procesos de pruebas.
- Reportar la finalización de las pruebas: Se debe obtener la información relevante del plan de pruebas, resultado de pruebas, reporte de estado de prueba, el reporte de finalización de pruebas y reporte de incidentes.

2.9.3.3. Pruebas dinámicas de software

Los procesos de prueba dinámica se utilizan para realizar pruebas dentro de una fase particular de prueba (por ejemplo, unidad, integración, sistema y aceptación) o tipo de prueba (por ejemplo, prueba de rendimiento, prueba de seguridad, prueba de usabilidad) (ISO, 2013a).

La norma ISO (2013a), indica que el proceso de pruebas dinámicas envuelve cuatro subprocesos que son diseño e implementación, preparación del ambiente de pruebas y mantenimiento, ejecución de pruebas y reporte de incidentes de pruebas.

Diseño e implementación de pruebas

El propósito del proceso de diseño e implementación de prueba es preparar los procedimientos de prueba que se ejecutarán durante el proceso de ejecución de la prueba (ISO, 2013a).

Este proceso involucra las siguientes actividades:

- Identificar el conjunto de características: Los ítems o conjuntos de ítems que se desean probar.
- Identificar condiciones de las pruebas: Se determinan todas las condiciones de prueba para las características a probar.
- Los elementos de cobertura de prueba: Se identifican los atributos que deben tener las condiciones de prueba para lograr los criterios de aceptación y finalización especificados en el Plan de Pruebas.
- Casos de prueba: Se identifican las precondiciones, valores de entrada, acciones a ejecutar y los atributos que tiene que cumplir. Estos deben estar priorizados según los riesgos identificados. Todos los casos de prueba deben estar en la especificación de casos de prueba y aprobados por los interesados.
- Conjunto de pruebas: Se agrupan los casos de prueba y las restricciones de ejecución.
- Procedimiento de pruebas: A partir de los conjuntos de casos de pruebas se definen los procedimientos que deben seguir.
- Ítems de información: Se tiene que obtener la especificación de pruebas, requerimientos de datos y ambiente de pruebas.

Preparación y mantenimiento del ambiente de pruebas

El propósito del proceso de preparación y mantenimiento es que el entorno de pruebas cumpla con las condiciones requeridas. Además, se tiene que comunicar su estado a todas las partes interesadas.

Este proceso involucra las actividades de establecer el ambiente de pruebas, según los requerimientos de pruebas, y mantenerlo.

Ejecución de pruebas

El propósito del proceso de ejecución de prueba es llevar a cabo los procedimientos de prueba creados en el entorno de prueba preparado y registrar los resultados.

Esto involucra ejecutar un procedimiento de prueba, comparar el resultado (el resultado real contra el resultado esperado) y registrar el resultado asociado a la prueba.

Reporte de incidente de prueba

El propósito del proceso de reporte de incidentes de prueba es informar a los interesados los casos que requieren acciones a partir del resultado de la ejecución de la prueba.

Este proceso involucra actividades para analizar los resultados obtenidos de la ejecución de pruebas y crear o actualizar los reportes de incidentes.

2.9.3.4. Documentación de pruebas

La ISO (2013a) establece el proceso de pruebas mientras que la ISO (2013b) proporciona una guía para documentar el proceso de prueba. Incluye ejemplos de cómo adaptar la documentación en proyectos ágiles y no ágiles.

A partir del ISO (2013a), el ISO (2013b) indica que de las pruebas se tienen que documentar en tres áreas diferentes, que son a nivel organizacional, de gestión de pruebas y pruebas dinámicas.

A nivel de organización se debe documentar:

- La política de pruebas: Indica los principios de pruebas que se deben seguir en la organización.
- Estrategia de pruebas organizacional: Es un documento técnico que sirve como guía de cómo se deben llevar a cabo las pruebas en la organización.

A nivel del proceso de gestión de pruebas, que se enfoca en los proyectos, se debe documentar:

- Un plan de pruebas: Este documento está conformado por toda la planeación de pruebas de un proyecto (en ocasiones se ocupan más de un plan de pruebas para un solo proyecto) y el documento de administración de pruebas del proyecto. Un plan de pruebas puede ser genérico para todos los proyectos, pero hay proyectos muy específicos que necesitan un plan de pruebas específico.
- El reporte de estado de pruebas: Provee información acerca del estado de prueba ejecutado en un periodo de reporte específico. En metodologías ágiles puede ser discutido en reuniones de iteración y no necesariamente es un documento escrito.
- Reporte de cierre de pruebas: Provee un resumen de las pruebas que se llevaron a cabo según el plan de pruebas.

A nivel del proceso de pruebas dinámicas, se tiene que documentar:

- Especificación del diseño de pruebas: Proporciona pautas para identificar las características que se probarán y las condiciones de prueba que se derivan de cada característica. Se considera el primer paso en la especificación del caso de prueba.
- Especificación de caso de prueba: Proporciona documentación para identificar los elementos de cobertura de prueba y los casos de prueba correspondientes que se derivan de las características.
- Procedimiento de caso de prueba: proporciona un marco para describir los casos de prueba como un conjunto, es decir, agregar casos de prueba, definir un orden de ejecución, identificar la configuración inicial y las actividades posteriores a la ejecución.
- Requisitos de datos de prueba: proporciona una orientación para describir las propiedades y los datos necesarios para realizar los procedimientos de prueba definidos en el documento de especificación de procedimiento de prueba.
- Requisitos del ambiente de prueba: describe las propiedades del entorno de prueba, como el hardware y el software necesarios para realizar los procedimientos de prueba definidos en el documento de especificación del procedimiento de prueba.
- Informe de preparación de datos de prueba: se refiere a la documentación utilizada para describir cada requisito de datos de prueba. Por lo general, este documento incluye los datos durante el período y cuándo se deben completar los requisitos.
- Informe de preparación del entorno de prueba: se refiere a la documentación que describe cada requisito del entorno de prueba. Por lo general, este documento incluye el entorno durante el período y cuándo se deben completar los requisitos.
- Resultados reales: se refiere a la documentación para informar los resultados obtenidos durante la ejecución de casos de prueba definidos. Los resultados obtenidos se comparan con los resultados esperados.
- Resultado de la prueba: Se relaciona con la documentación para registrar los resultados de la ejecución. Los resultados obtenidos se comparan con el resultado esperado y se identifica el estado del caso de prueba.
- Registro de ejecución de prueba: Se refiere a la documentación para registrar detalles del documento de procedimiento de prueba.
- Informe de incidentes de prueba: Proporciona pautas para describir cualquier incidente que se haya identificado durante la ejecución de los casos de prueba. Se

debe desarrollar un informe de incidentes para cada problema detectado durante la ejecución del procedimiento de prueba.

Basado en la ISO (2013b), Eira *et al.* (2018) brinda recomendaciones sobre cómo implementar el proceso de documentación de pruebas en medianas o pequeñas organizaciones.

Eira *et al.* (2018) hace la adaptación partiendo de un proyecto de software que se desarrolla con la metodología Scrum. A partir de esto menciona que se debe crear un proyecto de pruebas al inicio de cada *sprint*, el cual involucra preparación, especificación, ejecución y reporte. El ISO 29119-3 recomienda tener una política de pruebas y una estrategia organizacional de pruebas, sin embargo, por la complejidad Eira *et al.* (2018) determina hacer un plan principal (Master Test Plan). De este modo la estructura de pruebas se muestra en la Figura 5.

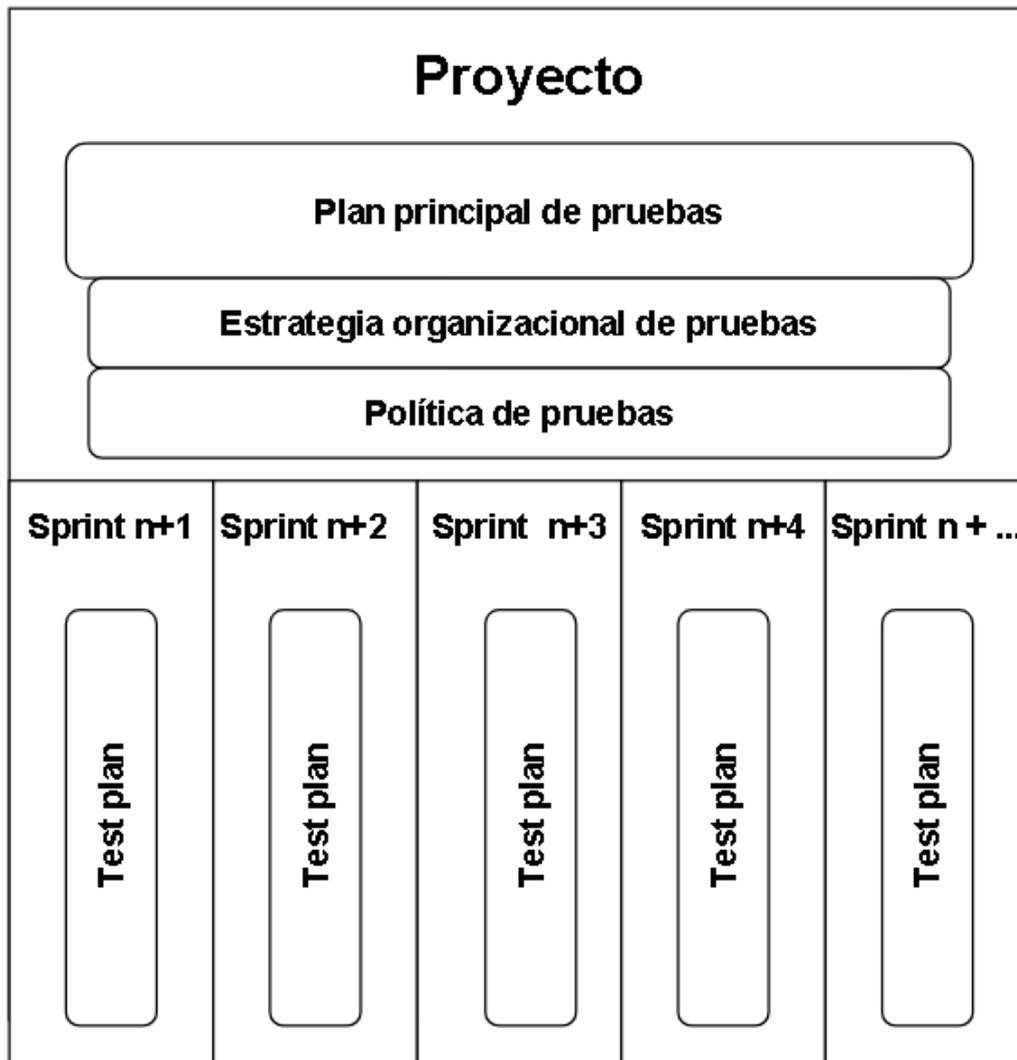


Figura 5. Estrategia de documentación de pruebas según metodología de Scrum. Fuente: Eira et al. (2018).

Por otra parte, Eira *et al.* (2018) dice que los documentos de resultados actuales, resultados de pruebas, registro de ejecución de pruebas e incidentes de pruebas son redundantes con el documento de procedimientos de casos de prueba pues manejan los mismos temas. De este modo, el documento de procedimiento de caso de prueba debe incluir tanto la descripción del caso como el resultado de su ejecución.

En el proceso de adaptación del ISO 29119-3 sobre pruebas de software, específicamente en documentación de software, Eira *et al.* (2018) concluye que se puede manejar con cuatro documentos, que son los siguientes:

- Plan de pruebas: Incluye la estrategia organizacional de pruebas y la política de pruebas.

- La especificación de pruebas: Incluye diseño de los casos de pruebas, especificación de los casos de pruebas y los procedimientos de casos de prueba.
- Requerimientos de pruebas: Incluye requerimientos de datos de prueba, requerimiento del ambiente de prueba, informe de preparación de datos de prueba y el informe de preparación del entorno de prueba.
- Reporte de pruebas: Incluye reporte de estado de la prueba y reporte de cierre de pruebas.

2.10. Modelo de calidad de software

Un modelo de calidad, según ISO (2011), es un conjunto definido de características y de relaciones entre ellas, que proporciona un marco para especificar los requisitos de calidad y evaluar la calidad.

El ISO (2011) indica que existen dos enfoques de modelos de calidad:

- Modelo de calidad en uso: Se refiere al resultado de interactuar con un sistema o software en un contexto específico. Las características que un usuario puede experimentar como efectividad, eficiencia, satisfacción o autonomía de riesgo es calidad en uso.
- Modelo de calidad del producto: Se refiere a las propiedades dinámicas o estáticas del software. Las características que un software pueda tener como cumplir con las funcionalidades, eficiencia en desempeño, compatibilidad, usabilidad, confiabilidad, seguridad, mantenibilidad o portabilidad es calidad del producto.

El ISO (2011) dice que estos modelos son útiles para especificar requerimientos, establecer medidas y desempeñar evaluaciones de calidad. El modelo de calidad del producto tiene las siguientes características:

- Cumplir con las funcionalidades: Incluye completitud funcional, correctamente funcional y apropiadamente funcional.
- Eficiencia en desempeño: Incluye respuesta en tiempo, uso de recursos y capacidad.
- Compatibilidad: Incluye coexistencia con otros productos e interoperabilidad en el intercambio y uso de información.
- Usabilidad: Incluye el grado de demanda al usuario en términos de reconocer si cumple con sus necesidades, facilidad de aprendizaje, manejable, inducción a errores, uso de interfaz y accesible.

- Confiabilidad: Incluye cumplir con el tiempo de operación normal, disponibilidad, tolerancia a fallos y recuperabilidad.
- Seguridad: Incluye confidencialidad, integridad, trazabilidad y autenticidad.
- Mantenibilidad: Un producto debe ser modular, reusable, analizable, modificable y se puede someter a pruebas.
- Portabilidad: Un producto es portable si se puede adaptar, instalar y reemplazar.

2.11. Prácticas de calidad de software

En esta sección se hace un resumen de las mejores prácticas y normas enfocada en calidad utilizadas por la industria del software.

2.11.1. Mejores prácticas

Una buena práctica no se define buena por sí misma, sino que existe evidencia demostrable que esas prácticas funcionan y produce buenos resultados, por lo tanto, se recomienda como modelo. (FAO, 2014). En la Tabla 6, se describen las buenas prácticas consideradas en la base teórica.

Mejor práctica	Enfoque utilizado	Descripción
SWEBOK	Requerimientos y calidad	Contiene las mejores prácticas de la ingeniería de software dividida en 15 áreas. Para este proyecto se toman las áreas de requerimiento (Ver Sección 2.4.1 de Requerimientos de software según SWEBOK).
ISTQB	Pruebas	El ISTQB provee buenas prácticas en el campo de <i>software testing</i> . El nivel <i>foundation</i> se utilizará para efectos del proyecto (Ver Sección 2.9.3.2 de Pruebas de software).
CMMI: DEV	Modelo de madurez y capacidad	Provee las buenas prácticas para evaluar y mejorar procesos relacionados al desarrollo de software (Ver Sección 2.7 de Modelos de madurez).

Tabla 6. Mejores prácticas consideradas en el marco conceptual. Fuente: Elaboración propia.

2.11.2. Normas y estándares

Un estándar es un documento aprobado por un organismo reconocido que proporciona, para uso común y repetido, reglas, pautas o características para productos o procesos y métodos

de producción relacionados, con los cuales el cumplimiento no es obligatorio (DTI, 2019). En la Tabla 7, se describen las normas incluidas en la base teórica.

Norma	Enfoque utilizado	Descripción
ISO/IEC/IEEE 29148-2018 Requerimientos de software	Requerimientos de software	Presenta las consideraciones que se deben tomar en cuenta para el proceso de gestión de requerimientos de software (Ver Sección 2.4.2 de Requerimientos según ISO/IEC/IEEE 29148:2018(E)).
IEEE 730-2014. Procesos de Aseguramiento de la Calidad de Software	Aseguramiento de calidad	Presenta una guía para la planeación e implementación de un proceso de aseguramiento de calidad en el ciclo de vida del desarrollo del software (Ver 2.9.2.1 de Proceso de aseguramiento de calidad de software (SQA) según el IEEE Std 730-2014).
IEEE 1012:2016 Verificación y validación de Sistemas, Software y Hardware	Verificación y validación	Presenta las prácticas que se deben tomar en cuenta al momento de aplicar pruebas de verificación y validación de software (Ver Sección 2.9.3.1 de Verificación y validación según la norma IEEE para la Verificación y validación de Sistemas, Software y hardware (IEEE Std 1012 -2016)).

Norma	Enfoque utilizado	Descripción
IEEE 29119-2 Pruebas de software.	Proceso de pruebas de software	Establece una guía para la planeación e implementación de proceso de pruebas en la organización y los proyectos de desarrollo de software (Ver Sección 2.9.3.2 de Pruebas de software y 2.9.3.3 de Pruebas dinámicas de software).
IEEE 29119-3 Documentación de pruebas	Documentación del proceso de pruebas	Presenta las prácticas que se pueden tomar en cuenta para documentar las pruebas. Para este caso se utilizará una adaptación enfocada en organizaciones pequeñas y medianas (Ver Sección 2.9.3.4 de Documentación de Pruebas).
ISO 25010:2011 Modelos de calidad de sistemas y software	Modelos de calidad	Muestra los modelos de calidad que se pueden tomar en cuenta en desarrollo de producto de software (Ver sección 2.10 de Modelos de calidad de software).

Tabla 7. Normas consideradas en el marco conceptual. Fuente: Elaboración propia.

3. Marco metodológico

El capítulo del marco metodológico describe la metodología de investigación utilizada para desarrollar el presente trabajo.

3.1. Tipo de investigación

Existe la investigación básica, que se enfoca en producir conocimiento y teorías y la aplicada, que se centra en la solución de problemas (Hernández, Fernández y Baptista, 2014). Una investigación que toma lugar en el contexto diario para solucionar problemas específicos de individuos, organizaciones y/o industrias es conocida como investigación aplicada. La investigación aplicada no busca resolver grandes preguntas sin respuesta sobre el universo o la sociedad, sino su objetivo es simplemente generar respuestas para resolver problemas específicos que se está enfrentando en un momento dado (Baimyrzaeva, 2018).

El presente proyecto plantea un problema que se requiere resolver, enfocándose en el campo de la calidad en el desarrollo de software, por lo tanto, se considera que este trabajo es una investigación aplicada.

La investigación aplicada tiene tres enfoques, la primera es **cuantitativa** que es un proceso secuencial y se caracteriza por medir y estimar magnitudes de los fenómenos o problemas de investigación. El segundo es **cualitativo**, que inicia la recolección de datos sin haber definido completamente las preguntas de investigación o hipótesis, y estas puede variar según la etapa. El tercero es **mixto**, que involucran parte del cualitativo y cuantitativo (Hernández *et al.*, 2014).

En este trabajo se identifica un problema y se indaga en información existente (para este caso, norma y buenas prácticas en la calidad de software) que permitan encontrar una solución a partir de la interpretación de la situación actual, por esta razón se considera que la investigación es cualitativa. Este tipo de investigación se centra en la interpretación, donde se busca entender un fenómeno e identificar algo nuevo que pueda ser considerado un éxito (Elliot y Timolak, 2005).

A su vez Hernández *et al.* (2014), menciona que los fenómenos en la investigación cualitativa “se explora desde la perspectiva de los participantes en un ambiente natural y en relación con su contexto” (p. 358) y el propósito “es examinar la forma en que los individuos perciben y experimentan los fenómenos que los rodean, profundizando en sus puntos de vista, interpretaciones y significados” (p. 358). En el presente trabajo se explora el fenómeno de

calidad en el desarrollo de software, y se recaba información de los involucrados para analizar alternativas de solución.

3.1.1. Diseño de la investigación

Hernández et al. (2014) indica que en la investigación cualitativa existen varios diseños del proceso de investigación como: etnográficos, narrativos, fenomenológicos, investigación-acción y teoría fundamentada.

El diseño etnográfico se centra en:

- Explorar, examinar y entender sistemas sociales.
- Producir interpretaciones profundas y significados culturales.
- Describir, interpretar y analizar ideas, creencias, significados, conocimientos y prácticas en sistemas sociales.

El diseño narrativo se centra en:

- Entender una sucesión de hechos, situaciones, fenómenos, procesos y eventos.
- Parte de los pensamientos, sentimientos, emociones e interacciones de quienes lo experimentan.
- Parte de narraciones registradas que describen eventos o conjunto de eventos de manera cronológica.

El diseño fenomenológico se centra en:

- Explorar, describir y comprender lo que los individuos tienen en común según la experiencia ante un fenómeno específico.
- Involucra sentimientos, emociones, razonamientos, visiones, percepciones, entre otros aspectos.

La Investigación-acción se centra en:

- Comprender y resolver problemáticas específicas de una colectividad.
- Aplicar las teorías y mejores prácticas según la situación.
- Aportar información que guíe la toma de decisiones para proyectos, proceso y reformas estructurales.

Para el presente proyecto se cuenta con una colectividad de personas que se interrelacionan entre sí para cumplir con el objetivo, que es desarrollar soluciones de software. Sin embargo,

en el conjunto de actividades que llevan a cabo, se identifica una problemática sobre aseguramiento de calidad que se debe atender. Como parte de la propuesta de solución se indagará en prácticas aceptadas por la industria para solventar la problemática encontrada, de este modo, por las características descritas por Hernández *et al.* (2014) sobre investigación- acción y la problemática que se requiere solucionar, se determina que el diseño de la investigación es de tipo “investigación acción”.

Hernández et al. (2014) identifica tres fases esenciales en la investigación-acción:

- Observar: Construir un bosquejo del problema y recolectar datos.
- Pensar: Analizar e interpretar la situación y los datos.
- Actuar: Resolver la problemática y proporcionar mejoras.

Este tipo de diseño generalmente es cíclico, como se muestra en la Figura 3. Los ciclos están compuestos por:

- Identificar el problema de investigación.
- Formulación de un plan para resolver el problema.
- Implementar el plan y evaluar resultados.
- Realimentación, conduce a un nuevo diagnóstico.

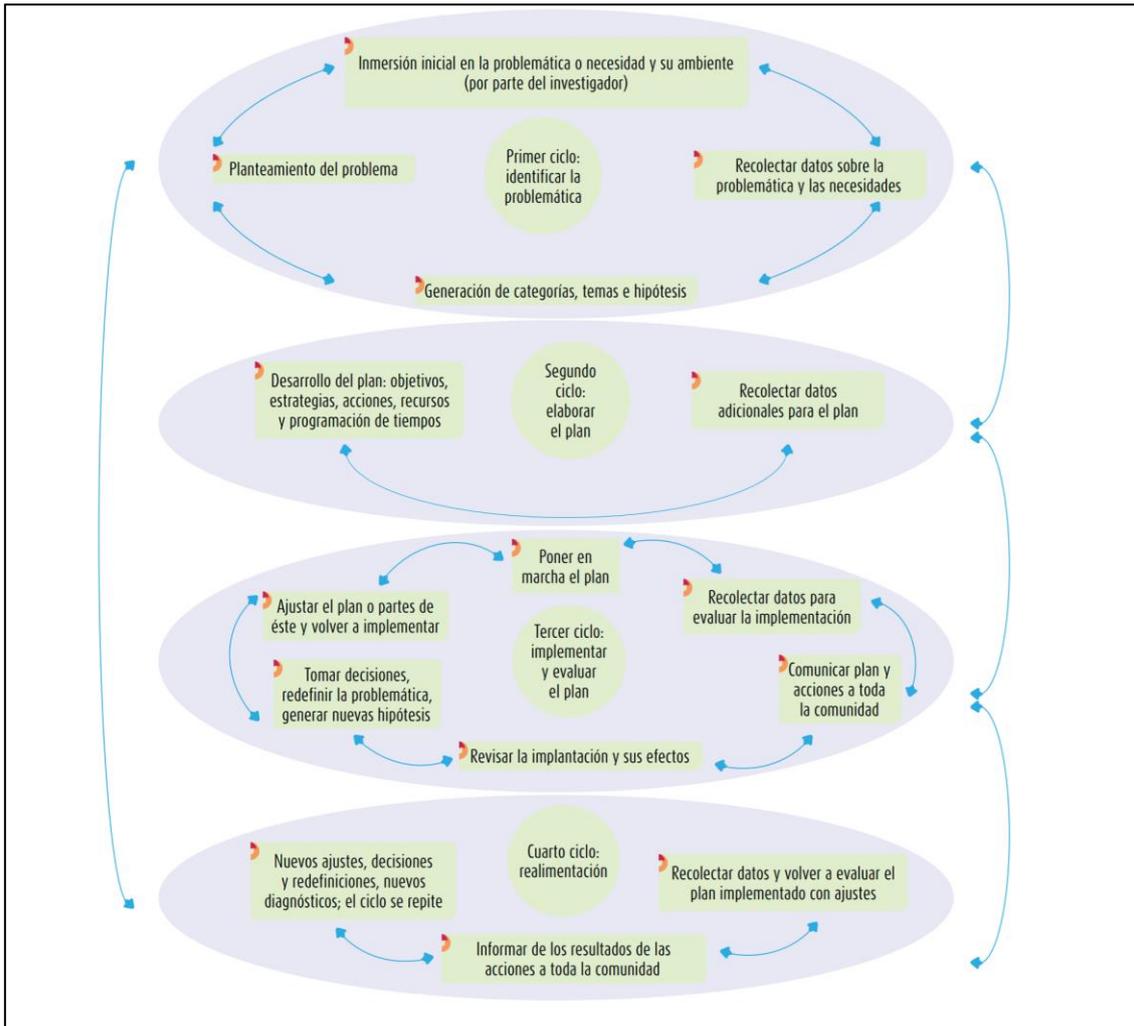


Figura 6. Principales acciones para llevar a cabo la investigación-acción. Fuente: Tomado de (Hernández et al, 2014).

El abordaje que se utiliza para esta investigación-acción será el “práctico”, que según Hernández et al (2014), tiene las siguientes características:

- Estudia prácticas locales del grupo.
- Involucra indagación individual o en equipo.
- Implementa un plan de acción.
- Liderazgo conjunto.

La investigación actual se sitúa en un contexto que involucra un grupo de personas dedicados al desarrollo de software, donde se pretende entender la forma de trabajo actual, partiendo de ahí se quiere hacer una propuesta, en el ámbito de calidad, construida conjuntamente ajustado a sus necesidades que les permita mejorar sus prácticas. Ante esta situación se considera que el enfoque de la investigación-acción que se ajusta es el práctico.

3.2. Sujetos y fuentes de información

Las fuentes y sujetos de información se describen en esta sección.

3.2.1. Sujetos de información.

Los sujetos de información son aquellas "...personas...sobre qué o quienes se van a recolectar datos..." (Hernández y Fernández, 2014). Son quienes colaboran con información necesaria para llevar a cabo el proyecto. Estas personas deben tener relación con los proyectos de desarrollo de software y han generado conocimiento a partir de sus experiencias.

Los sujetos de información consultados para la elaboración de la presente investigación se presentan en la Tabla 8.

Sujeto	Cantidad	Importancia y formación	Relevancia al proyecto
Coordinador de proyecto de software	2	Son los encargados de dirigir y coordinar las actividades de los distintos proyectos de software. Se cuenta con un Diplomado en computación con cuatro años de experiencia desarrollo web y un Ingeniero en computación con 2 años de experiencia.	Al coordinar las actividades del equipo de desarrollo son los que conocen como se ha hecho cada actividad en los desarrollos.
Desarrolladores	5	Son los que llevan a cabo las actividades de desarrollo de software. Se cuenta con Ingenieros y diplomados en computación.	La propuesta no debe afectar el flujo de desarrollo, por eso, ellos tienen que participar en la evaluación de las recomendaciones de la propuesta.

Sujeto	Cantidad	Importancia y formación	Relevancia al proyecto
Evaluadores de accesibilidad	2	Son quienes realizan verificación de accesibilidad del software desarrollado. Son personas con discapacidad visual que tiene conocimientos en uso de aplicaciones de software y tecnologías de asistencia.	Las recomendaciones en cuando a las validaciones de accesibilidad, también se tienen que consultar con estas personas, para que no afectar el flujo normal de su trabajo diario.

Tabla 8. Sujetos de información en Inlutec. Fuente: Elaboración propia.

3.2.2. Fuentes de información

Las fuentes de información son todos los elementos capaces de suministrar información (Barrantes, 2002). Para la investigación se requieren fuentes de información que la sustente, tanto en el proceso de recopilación de la información para clarificar la problemática como para encontrar recomendaciones en la solución de este.

Las fuentes de información se clasifican en primarias y secundarias, según el nivel que aporten a la investigación.

3.2.2.1. Fuentes primarias

Las fuentes primarias envuelven información original reunida para una investigación específica (Hox y Boeije, 2005). Esta información no ha sido procesada en otra investigación, sino que se recogen con procedimientos adecuados según el objetivo de la investigación.

Las fuentes primarias que se identificaron para el presente trabajo se muestran a continuación, en la siguiente Tabla 9.

Fuente	Detalle	Importancia
Documentos de proyectos	Requerimientos, cronogramas, planes de prueba y hallazgos de prueba.	Información de cómo se ha documentado el desarrollo de software,

Fuente	Detalle	Importancia
Reportes	Información y/o reportes de aplicaciones utilizadas para la gestión de proyectos.	Evidencias de ejecución del proceso de desarrollo de software.
Procesos de la organización	Información de procesos establecidos en la organización.	Insumo para determinar el estado actual de la organización.
Procesos ejecutados no documentados	Información que provenga de procesos que no se ha documentado como parte del desarrollo de software.	Insumo para determinar el estado actual de la organización.
Entrevistas	Entrevistas aplicadas a colaboradores del Grupo de interés.	Recabar información que no esté documentada y que sea de importancia para entender la situación actual.
Estándares y buenas prácticas	SWEBOK, PMBOK, ISTQB, CMMI: DEV, ISO/IEC/IEEE 29148, IEEE 730, IEEE 1012, IEEE 29119-2, IEEE 29119-3, ISO 25010	Teoría de calidad que contiene recomendaciones sobre cómo establecer procesos de calidad de software.

Tabla 9. Fuentes primarias de información. Fuente: Elaboración propia.

3.2.2.2. Fuentes secundarias

Las fuentes secundarias se refieren a información para otros propósitos o material creado por otros investigadores que se utilice para el desarrollo de un proyecto de investigación (Hox y Boeijs, 2005).

Las fuentes secundarias que se identificaron para este trabajo son los trabajos de graduación de licenciatura o maestría del Tecnológico de Costa Rica en campos de Administración de Tecnologías de Información y en Maestría de gerencia de proyecto de Ingeniería de Computación.

Estos trabajos tienen enfoques en metodologías o modelos de calidad. La selección de estos trabajos se hizo utilizando el Repositorio de trabajos hecho por investigadores estudiantes y funcionarios del TEC. Los principales trabajos revisados son los siguientes:

- Diseño de sistema para el aseguramiento y el control de la calidad en los proyectos de software del TEC Digital.
- Propuesta de una metodología de control de calidad para los proyectos de automatización, basado en las mejores prácticas de ISTQB, caso SOIN S.A.
- Establecimiento de la metodología para aseguramiento de la calidad en ArcSight QA Costa Rica.

3.3. Metodología de investigación

En esta sección se describe la metodología de desarrollo de TFG, se desglosa el trabajo por etapas para visualizar el flujo de seguimiento (ver Figura 7).

El flujo de trabajo que se siguió está compuesto de las etapas de documentar la situación actual, análisis de brecha y mejoras, elaboración de la propuesta y revisión de herramientas, las cuales estuvieron en constante control y monitoreo. Al final se llevó a cabo una etapa de cierre para la evaluación del trabajo realizado y los resultados obtenidos.

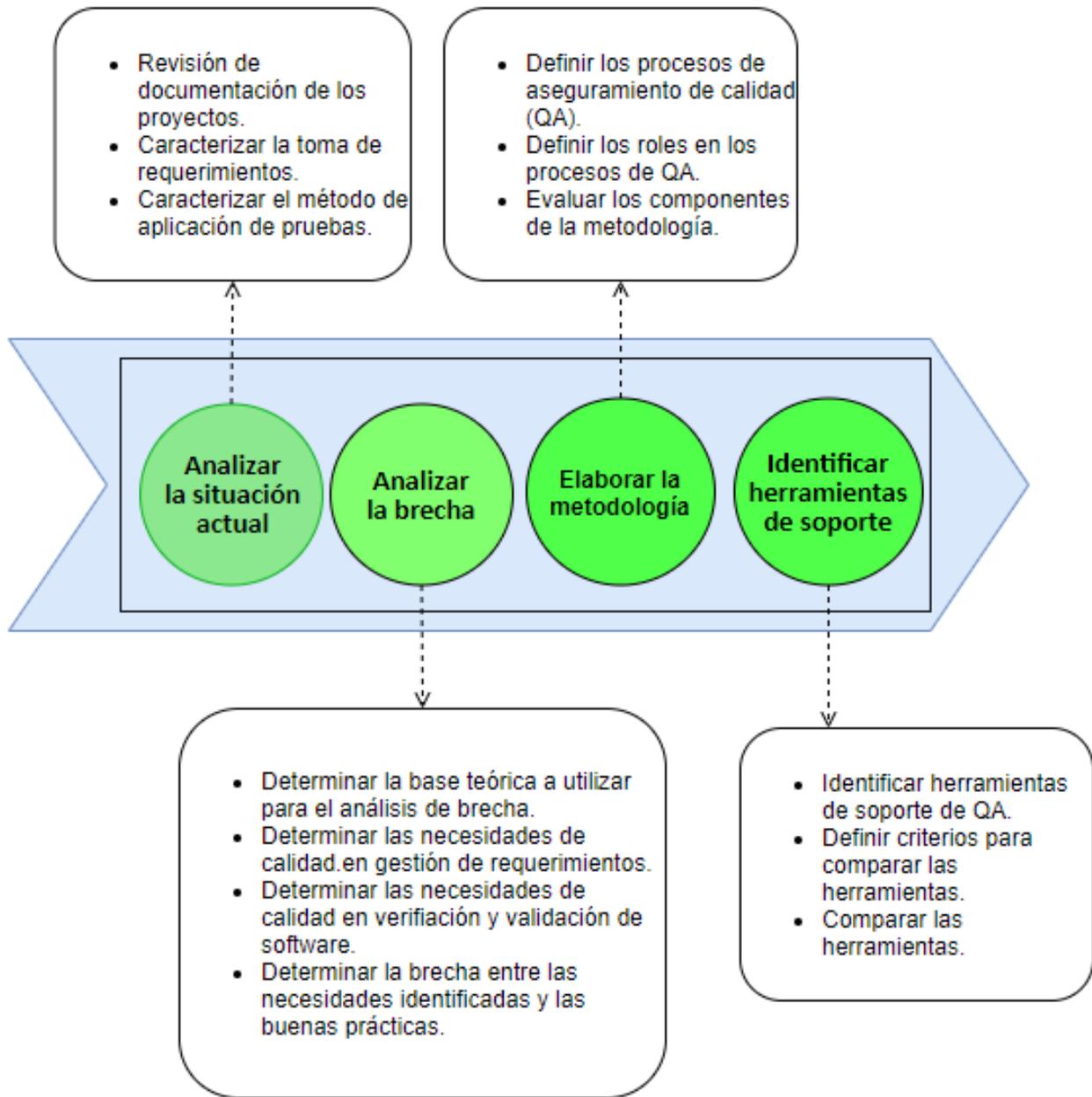


Figura 7. Esquema de la metodología de desarrollo del proyecto. Fuente: Elaboración propia

3.3.1. Analizar la situación actual del proceso de calidad en Inlutec.

Las actividades que comprenden la etapa de documentar la situación actual están enfocadas en comprender el contexto y definir una estructura de la situación actual.

Revisión de documentación de los proyectos de Inlutec

La revisión de la documentación de los proyectos fue necesaria para determinar el proceso de calidad que se ejecuta actualmente en la organización. La información fue obtenida de

los resultados documentados por los involucrados en los proyectos y de entrevistas con los desarrolladores sobre la ejecución de actividades de calidad. Esta información fue uno de los insumos principales para caracterizar el proceso de calidad actual.

Caracterizar el proceso de toma de requerimientos.

Se definió el estado actual del proceso de gestión de requerimientos que se lleva a cabo en Inlutec. Se identificaron los roles y tareas que realizan, así como las condiciones de ejecución. Esta caracterización fue uno de los insumos principales para realizar el análisis de la brecha.

Caracterizar el proceso de aplicación de pruebas.

Se definió el estado actual del proceso de verificación y validación que se lleva a cabo en Inlutec. Se identificaron los roles y tareas que realizan, así como las condiciones de ejecución. Esta caracterización fue uno de los insumos principales para realizar el análisis de la brecha.

3.3.2. Analizar la brecha entre el proceso de calidad actual y las recomendaciones de buenas prácticas y estándares.

A partir de la definición de la situación actual, se hizo una revisión bibliográfica sobre prácticas y normas de calidad. Con estos dos elementos se hizo el análisis de brecha.

Determinar la base teórica a utilizar para el análisis de brecha

La revisión bibliográfica comprende la ampliación del conocimiento sobre aseguramiento de calidad, que permitió entender cómo se realizan estos procesos en la industria del software. Con esta revisión se estableció la base teórica para el proyecto, se seleccionaron las fuentes de información como trabajos similares, estándares, buenas prácticas y/o marcos de referencia sobre calidad que eran aplicables a la organización. El resultado fue el marco conceptual.

Determinar las necesidades y requerimientos de calidad en cuanto a la gestión de requerimientos.

Con la documentación y proceso actual de gestión de requerimientos, recabados con la revisión documental, entrevistas y encuestas, se identifican necesidades de calidad en cuanto al proceso de requerimientos de software.

Determinar las necesidades y requerimientos de calidad en cuanto a la gestión de pruebas.

Con la documentación y proceso actual de verificación y validación, recabados con la revisión documental, entrevistas y encuestas, se identifican necesidades de calidad en cuanto al proceso de implementación de pruebas.

Determinar la brecha entre las necesidades detectadas y las buenas prácticas.

Esta actividad fue la clave para realizar la propuesta. A partir del proceso actual y la base teórica se determinó la brecha que tiene la organización entre sus procesos respecto a las recomendaciones hechas por las prácticas de calidad. Con la brecha identificada, se definieron las estrategias para dar inicio al desarrollo de la metodología propuesta.

3.3.3. Elaborar la metodología

En la etapa de elaborar la metodología, se tomó el resultado del análisis de brecha para realizar la propuesta sobre aseguramientos de calidad para Inlutec.

Definir el proceso de aseguramiento de la calidad acorde con las necesidades detectadas.

A partir de las necesidades y la brecha identificada, se elaboró la metodología de calidad de software para Inlutec. Se definió el proceso, artefactos, métricas y roles necesarios para llevarlo a cabo. Se realizó bajo recomendaciones de estándares y buenas prácticas de calidad.

Evaluar la metodología definida

Durante todo el tiempo de desarrollo de la propuesta, cada componente se sometió a evaluación con la contra parte de la organización, algunos miembros del equipo de desarrollo y profesor encargado. Esto permitió identificar las correcciones que se tenían que aplicar y asegurar que la propuesta correspondía a la necesidad planteada. Se trató de que este proceso fuera cíclico, en la medida de lo posible.

3.3.4. Identificar herramientas de soporte de QA.

Se buscó recomendar herramientas que puedan soportar el proceso de calidad para agilizarlo, para esto se hizo una investigación sobre software que soportan los procesos involucrados en el alcance del proyecto. Se utilizó el *benchmarking* como técnica para comparar las herramientas.

Definición de criterios y comparación de las herramientas

Para la comparación de las herramientas se definieron criterios contextualizados a las necesidades de Inlutec. Estos criterios serán los aspectos que se evaluaron en cada herramienta. Posteriormente se unificó la información para comparar cuál cumple de mejor manera los criterios evaluados.

3.4. Instrumentos de investigación

En esta sección se describen los instrumentos que se utilizaron para la recolección de información pertinente para el desarrollo del proyecto. Como la investigación es cualitativa, la recolección de la información se hizo en el contexto de la problemática y con las personas involucradas.

3.4.1. Observación

La observación “implica adentrarse profundamente en una situación” (Hernández et al, 2014, p.399). Esto permite “comprender procesos, vinculaciones entre personas y sus situaciones, experiencias o circunstancias, los eventos que suceden con el paso de tiempo y los patrones que desarrollan” (Hernández et al, 2014, p.399).

Esta técnica se utilizó para analizar la situación actual y determinar cómo se gestiona el tema de calidad de los proyectos de software en la organización, ya sean roles, procesos, prácticas, instrumentos, herramientas, entre otros, establecidos o realizados de manera empírica.

Hernández (2014), describe cinco formas que el investigador puede utilizar para involucrarse en la observación de la investigación cualitativa:

- No participación, por ejemplo, revisar videos.
- Participación pasiva: Es observador, no interactúa en las actividades.
- Participación moderada: Participa en algunas actividades, no en todas.
- Participación activa: Se involucra en la mayoría de las actividades, sin embargo, no se mezcla con los participantes.
- Participación completa: El observador es un participante más.

Para este proyecto el investigador fue un observador activo, pues se centra en la gestión de calidad del desarrollo de software, pero el rol que juega es de investigador no forma parte del resto de participantes. La plantilla que se utiliza para registrar el resultado de la

observación se encuentra en el Apéndice A en el **Instrumentos1. Plantilla para observación.**

El proyecto se centró en plantear el aseguramiento de la calidad para reforzar la gestión de requerimientos y la validación y verificación del software, por lo tanto, la observación que se hizo es para determinar aspectos que no se obtengan en las entrevistas aplicadas o para reafirmar la situación expresada por lo involucrados.

3.4.2. Entrevistas

Las entrevistas utilizan el principio del diálogo igualitario y horizontal (Carballo, 2001). Esta técnica se emplea para establecer un diálogo con las personas relacionadas al proceso de desarrollo de software. Carballo (2001), también menciona que la entrevista es un modo de descubrir el punto de vista de las personas y recoger información sobre una situación.

Según Hernández *et al* (2014), existen tres tipos entrevistas que son:

- Estructuradas: Se sigue una guía de preguntas específicas.
- Semiestructuradas: Se tiene una guía de preguntas base, pero se pueden incorporar nuevas a medida que se ejecuta la entrevista.
- Entrevistas abiertas: Se tiene un guía base y el entrevistador posee toda la flexibilidad para manejar la guía.

Se utilizó las entrevistas de tipo semiestructuradas, principalmente para determinar prácticas o procesos implementados, referentes a calidad, en los proyectos realizados previos al desarrollo de la investigación. La Tabla 10 muestra los instrumentos y sus propósitos.

Instrumento	Propósito	Sujeto de información	Resultado esperado
Entrevista sobre requerimientos. Ver Apéndice A en el Instrumentos 2. Guía entrevista de requerimientos.	Entender y documentar la forma cómo se gestionan los requerimientos.	Coordinador de proyecto Euler Coordinador de los proyectos web.	El proceso de gestión de requerimientos actual. Lista de las complicaciones en la gestión de requerimientos. Problemas a partir de gestión de requerimientos actual.
Entrevista calidad. Ver Apéndice A en el Instrumentos 3. Guía entrevista de calidad..	Entender y documentar la forma como se gestiona la calidad durante del desarrollo de software.	Coordinador de proyecto Euler Coordinador de los proyectos web	El proceso de gestión de calidad actual. Lista de las complicaciones en la gestión de calidad. Problemas a partir de gestión de calidad actual.

Tabla 10. Instrumentos de entrevistas y su propósito para el proyecto. Fuente: Elaboración propia.

3.4.3. Cuestionarios

Según Casas *et al.* (2003), los cuestionarios se utilizan en la investigación para recoger la información de forma organizada sobre las variables involucradas en los objetivos. Además, Casas *et al.* (2013) menciona que en el cuestionario se pueden incluir preguntas de varios tipos, como:

- Cerradas: Son pregunta con respuestas definidas, y la personas que las responda, debe expresar su opinión bajo esas respuestas.
- Elección múltiple: Son preguntas con una serie de opciones, donde la persona que responda puede seleccionar una o varias opciones para expresar su opinión.

- Abiertas: Se da la libertad a las personas que va a responder las preguntas del cuestionario, que describa con sus propias palabras su opinión.

Para el caso del presente proyecto, se aplicó un cuestionario donde la mayoría de las preguntas son abiertas. En la Tabla 11, se describen los cuestionarios utilizados.

Instrumento	Propósito	Sujeto de información	Resultado esperado
Encuesta de aseguramiento de calidad. Ver Apéndice A en el Instrumentos 4. Encuesta sobre aseguramiento de calidad en Inlutec..	Conocer el punto de vista de los involucrados en el desarrollo de software sobre un proceso aseguramiento de calidad.	Equipo de desarrollo de proyectos web, que son 6 personas. Coordinador del proyecto Euler.	Percepción de la calidad. Complicaciones en realizar la gestión de la calidad. Problemas a raíz de la gestión de calidad actual. Expectativas del aseguramiento de la calidad. Recomendaciones de un proceso de aseguramiento de la calidad.

Instrumento	Propósito	Sujeto de información	Resultado esperado
Encuestas sobre los requerimientos. Ver Apéndice A en el Instrumentos 5. Encuesta sobre la gestión de requerimientos en Inlutec..	Conocer el punto de vista de los involucrados en el desarrollo de software sobre el proceso gestión de requerimientos.	Equipo de desarrollo de proyectos web conformado por seis colaboradores. Coordinador del proyecto Euler.	Percepción de la calidad de los requerimientos. Percepción de problemas a raíz de la gestión de requerimientos actual. Expectativas de un proceso de gestión de requerimientos. Recomendaciones de un proceso de gestión de requerimientos contextualizado.
Encuesta de evaluación de la propuesta de procesos de requerimientos (Ver Validación 1. Requerimientos en el Apéndice C)	Posterior a una presentación de los procesos propuestos, se pasa una encuesta donde los involucrados indican su conformidad, no conformidad y observaciones respecto a la propuesta de procesos de requerimientos.	Coordinadores de desarrollo Web y Euler.	Los resultados obtenidos sirven para ajustar la propuesta de requerimientos según las observaciones de los involucrados.

Instrumento	Propósito	Sujeto de información	Resultado esperado
Encuesta de evaluación de la propuesta de procesos de requerimientos (Ver Validación 2. Evaluación del proceso de verificación y validación. en el Apéndice C)	Posterior a una presentación de los procesos propuestos, se pasa una encuesta donde los involucrados indican su conformidad, no conformidad y observaciones respecto a la propuesta de procesos de verificación y validación.	Coordinadores de desarrollo Web y Euler.	Los resultados obtenidos sirven para ajustar la propuesta de verificación y validación según las observaciones de los involucrados.

Tabla 11. Lista de cuestionarios aplicados durante el desarrollo del proyecto. Fuente: Elaboración propia.

3.4.4. Revisión documental

La revisión documental se enfoca principalmente en analizar y entender prácticas que se ajusten a las necesidades y aportan valor en la propuesta de solución del problema.

En este trabajo se hace la revisión de los documentos relacionados a las prácticas y procesos actuales utilizados por el equipo de desarrollo de la organización. Esta actividad se realizó para sumergirse en la situación actual y analizar la problemática que se plantea para el proyecto. Para esta revisión se definió la plantilla que está en el Apéndice A en el Instrumentos 6. Plantilla de revisión documental interna. De la aplicación de revisión documental se obtuvo una lista de características de la documentación de requerimientos y de pruebas.

Por otra parte, se hizo la revisión de normas y buenas prácticas como referencia e insumo para analizar y determinar la solución del problema que abarca este proyecto de investigación. Las organizaciones creadoras de normas, marcos de referencias y buenas prácticas (sobre calidad) reúnen conocimiento que ha sido utilizada por la industria y brindan buenos resultados, por esta razón se toman como la base para la propuesta de solución.

Las revisiones de documentos de normas se hicieron con la plantilla establecida en el Apéndice A en el Instrumentos 7. Planilla de revisión documental de normas y buenas prácticas. De la aplicación de la revisión documental se obtuvo una lista de recomendaciones, tanto en la gestión de requerimientos como en la verificación y validación de software, que es unos los insumos para realizar el análisis de la brecha en la organización.

3.4.5. Análisis de los datos

El sentido de este trabajo fue realizar una propuesta basada en las prácticas de la industria sobre calidad que permita mejorar el estado del proceso de calidad en los proyectos de software que desarrolla la organización. De esta manera, la información sobre el proceso actual se analizó de acuerdo con las normas y buenas prácticas, para esto se definió la base teórica y se realizó un análisis de brecha tomando como base a la situación actual, con el Instrumentos 8. Planilla para el análisis de brecha. del Apéndice 1.

Para realizar este análisis, a partir de la situación actual documentada y observada se hizo un resumen de las actividades que realizan, del mismo se tomó las recomendaciones de las normas y buenas prácticas y se contrastó con la situación real. Posteriormente se clasificaron las actividades actuales de la organización según el CMMI para desarrollo. A partir de esa clasificación se definió cuál es el nivel de madurez y capacidad al que puede aspirar la organización en cuanto procesos de requerimiento y calidad.

3.5. Operacionalización de las categorías de análisis

A continuación, en la Tabla 12, se muestran los objetivos específicos con sus respectivas variables o criterios de investigación, esta tabla permite la evaluación de cumplimiento de los objetivos establecidos.

Cuadro de variables				
Objetivo específico	Variable de estudio	Definición conceptual	Indicadores	Instrumentos
Caracterizar el proceso de calidad actual utilizado en Inlutec por medio de observación, consulta con los desarrolladores y revisión de documentación para entender su funcionamiento.	Proceso de calidad	Conjunto de actividades “enfocadas en proporcionar confianza en que se cumplirán los requisitos de calidad” (ISO 9000, 2005)	<ul style="list-style-type: none"> • Diagramas de los procesos actuales. • Lista de diferencias de los procesos actuales según los proyectos. • Lista de características de la documentación de calidad y requerimientos. • Diferencias en la documentación de calidad y requerimientos. 	<ul style="list-style-type: none"> • Entrevistas, con los coordinadores de los proyectos. • Encuesta a los desarrolladores, para entender el proceso actual. • Observación de ejecución del proceso actual • Revisión documental.

Cuadro de variables				
Objetivo específico	Variable de estudio	Definición conceptual	Indicadores	Instrumentos
Analizar la brecha entre el proceso de calidad actual y las recomendaciones de buenas prácticas, estándares y marcos de referencia para determinar las necesidades de calidad evaluando las partes del proceso actual contra las recomendaciones.	Brecha	Comparar el desempeño actual con el desempeño esperado. (Smartsheet, 2019)	<ul style="list-style-type: none"> Listado con la descripción de teorías sobre requerimientos y calidad, aplicables a la organización. Tabla comparativa de los aspectos específicos de la situación actual, recomendaciones de la teoría, brecha que existe y estrategias para abordar las recomendaciones en la organización. 	<ul style="list-style-type: none"> Análisis de brecha. Revisión documental de las prácticas de calidad. Entrevistas a los coordinadores
Elaborar la metodología de aseguramiento y	Metodología	Un conjunto de métodos, reglas y postulados empleados por una disciplina: un	<ul style="list-style-type: none"> Conjunto de actividades de gestión de 	<ul style="list-style-type: none"> Revisión de proyecto similares.

Cuadro de variables				
Objetivo específico	Variable de estudio	Definición conceptual	Indicadores	Instrumentos
control de calidad para los proyectos de software basándose en la brecha entre las recomendaciones de calidad y el proceso actual para ajustarla a las necesidades de Inlutec.		procedimiento particular o un conjunto de procedimientos (Merriam-Webster, 2019)	requerimientos y calidad, interrelacionados entre sí, específicos para el contexto de la organización.	<ul style="list-style-type: none"> • Reuniones de presentación de propuestas con el equipo de desarrollo.
Identificar herramientas de software para soporte de procesos de aseguramiento y control de calidad por medio de un benchmarking para recomendarlas al	Herramientas de software	Herramientas que soportan procesos de calidad, por ejemplo, pruebas.	<ul style="list-style-type: none"> • Lista de herramientas de software revisadas. • Herramientas de software que son aplicables en Inlutec, evaluadas según los criterios definidos. 	<ul style="list-style-type: none"> • Benchmarking. • Revisión documental.

Cuadro de variables				
Objetivo específico	Variable de estudio	Definición conceptual	Indicadores	Instrumentos
equipo de desarrollo de Inlutec.			<ul style="list-style-type: none">• Recomendación justificada de una herramienta o conjuntos de herramientas específicas.	

Tabla 12. Tabla de variables. Fuente: Elaboración propia

4. Análisis de resultados

En el presente capítulo, se presentan los resultados obtenidos de la aplicación de los instrumentos diseñados para trabajar en las etapas de analizar la situación actual del proceso de calidad en Inlutec y analizar la brecha entre el proceso de calidad actual y las recomendaciones de buenas prácticas y estándares.

Se presentan los resultados de la revisión documental, las encuestas, las entrevistas y las observaciones para identificar el estado de la situación actual y recomendaciones de calidad de las normas y buenas prácticas.

4.1. Revisión documental

Esta sección presenta los resultados obtenidos al aplicar la revisión documental de fuentes teóricas de software y calidad e información provista por la organización sobre los proyectos de software.

4.1.1. Base teórica para el proyecto

La base teórica gira principalmente en el aseguramiento de calidad de los proyectos de software, pues el objetivo principal es proponer una metodología que le permita a la organización alcanzar los objetivos de calidad de los proyectos.

A raíz de los proyectos de software, se busca incorporar un proceso de aseguramiento de calidad que involucre la etapa de requerimientos y verificación y validación de software, como se muestra en la Figura 8.

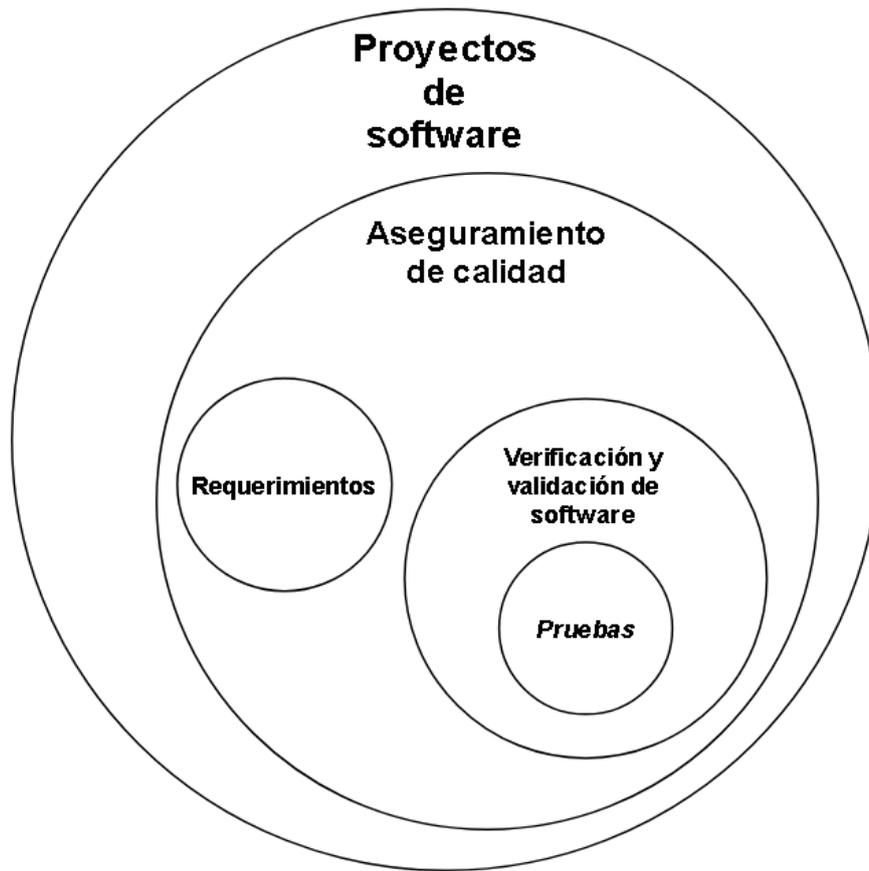


Figura 8. Temas para la revisión documental. Fuente: Elaboración propia.

Del mismo modo la Figura 8 engloba el alcance que se trabajará para el proyecto. A continuación, la Tabla 13 muestra las buenas prácticas y normas consideradas para el proyecto.

Norma o buena práctica	Descripción	Relevancia para el proyecto
ISO/IEC/IEEE 29148-2018 Requerimientos de software	Presenta las consideraciones que se deben tomar en cuenta para el proceso de gestión de requerimientos de software.	<ul style="list-style-type: none"> • Proceso gestión de requerimientos. • Características de requerimientos. • Documentación de requerimientos.

Norma o buena práctica	Descripción	Relevancia para el proyecto
IEEE 730-2014. Procesos de Aseguramiento de la Calidad de Software	Presenta una guía para la planeación e implementación de un proceso de aseguramiento de calidad en el ciclo de vida del desarrollo del software.	<ul style="list-style-type: none"> • Proceso de aseguramiento de calidad durante el ciclo de vida de los proyectos. • Aseguramiento del proceso y del producto.
IEEE 1012:2016 Verificación y validación de Sistemas, Software y Hardware	Presenta las prácticas que se deben tomar en cuenta al momento de aplicar pruebas de verificación y validación de software.	<ul style="list-style-type: none"> • Proceso de verificación y validación de software durante el ciclo de vida del proyecto.
IEEE 29119-2 Pruebas de software	Establece una guía para la planeación e implementación de proceso de pruebas en la organización y los proyectos de desarrollo de software.	<ul style="list-style-type: none"> • Proceso para la gestión de pruebas durante el ciclo de vida del proyecto. • Gestión de pruebas dinámicas del software.
IEEE 29119-3 Documentación de pruebas	Presenta las prácticas que se pueden tomar en cuenta para documentar las pruebas. Para este caso se utilizará una adaptación enfocada en organizaciones pequeñas y medianas.	<ul style="list-style-type: none"> • Documentación del proceso de pruebas para la recolección de evidencia.

Norma o buena práctica	Descripción	Relevancia para el proyecto
ISO 25010:2011 Modelos de calidad de sistemas y software	Muestra los modelos de calidad que se pueden tomar en cuenta en el desarrollo del producto de software.	<ul style="list-style-type: none"> • Modelos de calidad de software. • Modelos de calidad del producto.
SWEBOK	Contiene las mejores prácticas de la ingeniería de software dividida en 15 áreas. Para este proyecto se toman las áreas de requerimiento y calidad.	<ul style="list-style-type: none"> • Proceso de gestión de requerimientos. • Proceso de aseguramiento de calidad.
ISQTB	El ISTQB provee buenas prácticas en el campo de <i>software testing</i> . El nivel <i>foundation</i> se utilizará para efectos del proyecto.	<ul style="list-style-type: none"> • Proceso de gestión de pruebas.
CMMI: DEV	Provee las buenas prácticas para evaluar y mejorar procesos relacionados al desarrollo de software.	<ul style="list-style-type: none"> • Evaluación de los procesos de requerimientos y calidad.

Tabla 13. Normas y buenas prácticas consideradas para el proyecto. Fuente: Elaboración propia.

4.1.2. Revisión, descripción y análisis de la documentación de los proyectos de Inlutec

En esta sección se resumen las características de la documentación proporcionada por Inlutec, sobre gestión de requerimientos y verificación y validación. El propósito es caracterizar las formas en cómo se han documentado los requerimientos y pruebas de los proyectos que han desarrollado y los actuales.

4.1.2.1. *Evaluador de accesibilidad web (EAW) manual y automático*

Descripción: Este es un proyecto de iniciativa Inlutec, el cual busca proveer una herramienta que valide las pautas de accesibilidad digital en los sitios web. Se basa en las recomendaciones de accesibilidad que brinda la W3C por medio del *Web Content Accessibility Guidelines (WCAG)*. El proyecto se desarrolla enfocado a *web* con marcos de desarrollo basados en *javascript* y *node*.

Contenido de la documentación en requerimientos

Como se evidencia en la Respuesta 3 del Apéndice B de la entrevista realizada al coordinador del equipo de desarrollo web, al inicio del proyecto los requerimientos se empezaron a documentar en un archivo escrito, sin embargo, la carga de trabajo en desarrollo no les ha permitido mantenerlo. Esto significa que en cuanto a documentación escrita solo están los requerimientos iniciales y no refleja los cambios que hayan ocurrido. En la Tabla 14, se muestran las características del contenido de los requerimientos identificados en el documento escrito.

Requerimientos funcionales	Requerimientos no funcionales
<p>Las características de los requerimientos funcionales son las siguientes:</p> <ul style="list-style-type: none"> • Código o número de requerimiento. • Nombre del requerimiento. • Tipo de requerimiento. • Prioridad. • Actores. • Detalle del requerimiento. 	<p>Las características de los requerimientos no funcionales son las siguientes:</p> <ul style="list-style-type: none"> • Código o número del requerimiento. • Nombre del requerimiento. • Tipo de requerimiento. • Prioridad <p>Los tipos de requerimientos no funcionales no se encuentran de</p>

Requerimientos funcionales	Requerimientos no funcionales
<ul style="list-style-type: none"> • Restricciones. • Formato de pantalla y reportes. • Reglas de negocio. <p>En cuanto a la redacción del requerimiento, no se sigue alguna sintaxis específica.</p>	<p>manera explícita, pero todos se pueden categorizar como de usabilidad.</p>

Tabla 14. Contenido de la documentación de requerimientos del proyecto EAW. Fuente: Elaboración propia.

Sin embargo, los requerimientos se han manejado por medio de una herramienta de software llamando *ScrumDesk*, que maneja conceptos como historias de usuarios, características y tareas. Lo que permite darle trazabilidad los requerimientos.

El equipo de desarrollo web ha adoptado *ScrumDesk*, el detalle se presenta en términos de inconsistencia con la documentación escrita, pues contiene requerimientos diferentes o desactualizados.

Contenido de la documentación de pruebas

En cuanto a calidad, la práctica principal es la aplicación de pruebas, estas se realizan en términos de pruebas unitarias y de integración y se enfocan en funcionalidad y accesibilidad. No existe un proceso documentado y estándar para ejecutar estas pruebas ni para los resultados.

De la implementación de pruebas, solo se toma los fallos identificados en el momento para solucionarlos y seguir con el desarrollo del software. De la revisión documental, no se encontró evidencia para este proyecto.

4.1.2.2. Sistema costarricense de información sobre discapacidad (SICID)

Descripción: SICID es una plataforma web de servicios que pertenece al Consejo Nacional de la Persona con Discapacidad (Conapdis). También, es web sobre *javascript* y *node*.

Contenido de la documentación de requerimientos

En la documentación de requerimientos del proyecto SICID, se pueden identificar una clasificación de requerimientos tanto funcionales como no funcionales. Como se evidencia en la Respuesta 3 del Apéndice B, de la respuesta a la entrevista aplicada a Víctor Romero,

estos requerimientos involucraron un alto número de cambios que impactaba el desarrollo del proyecto.

A continuación, en la Tabla 15, se muestran las características de los requerimientos documentados.

Requerimientos funcionales	Requerimientos no funcionales
<p>Características:</p> <ul style="list-style-type: none"> • Código de requerimiento. • Precondiciones. • Iniciador (actor). • Resumen. • Descripción de flujo (éxito y alternativo). • Excepción (restricción) <p>Estos requerimientos están formulados como casos de uso.</p>	<p>Para todos los subsistemas que involucra el sistema SICID, se incluyen las siguientes categorías de requerimientos no funcionales.</p> <ul style="list-style-type: none"> • Usabilidad <ul style="list-style-type: none"> ○ Accesibilidad ○ Inclusión ○ Implementación: Requerimientos del proyecto, enfocados en la colocación del producto, metodología de desarrollo y documentación que se debe aportar. ○ Interfaz gráfica • Confiabilidad <ul style="list-style-type: none"> ○ Frecuencia de fallos: Tiempo medio entre fallos (MTBF). ○ Recuperabilidad: Tiempo medio para reparar (MTTR). ○ Precisión: Una solicitud le corresponda la respuesta esperada. ○ Seguridad • Rendimiento <ul style="list-style-type: none"> ○ Disponibilidad ○ Tiempo de respuesta • Soporte

Requerimientos funcionales	Requerimientos no funcionales
	<ul style="list-style-type: none"> ○ Extensibilidad: Implementación de nuevos requerimientos. ○ Adaptabilidad ○ Compatibilidad: requisitos de software. ○ Mantenibilidad ○ Escalabilidad ● Restricciones de diseño: Aspectos de línea gráfica que se tienen que respetar. ● Restricciones de implementación: requerimientos del proyecto, enfocados al tipo y características de las tecnologías de desarrollo que se deben utilizar para crear el producto. ● Interfaces ● Licenciamiento ● Estándares aplicables.

Tabla 15. Contenido de la documentación de requerimientos del proyecto SICID. Fuente: Elaboración propia.

Estos requerimientos sí están documentados, sin embargo, pasa los mismo que el EAW, esos requerimientos no se reflejan los cambios que se dieron durante del desarrollo. Para el cierre de las etapas del proyecto SICID, los desarrolladores han utilizado *ScrumDesk*, para llevar el control del desarrollo.

Contenido de la documentación de pruebas

Del mismo modo, la práctica principal es la ejecución de pruebas unitarias y de integración, enfocadas en funcionalidad y accesibilidad. El único registro de los resultados se obtiene de las pruebas de integración en el documento del plan de pruebas, pero no es una información

que se pueda consultar porque existen algunos documentos no estándares de casos de prueba aislados. De este modo no es posible darles trazabilidad a las pruebas efectuadas.

Proceso	Resultado
Pruebas unitarias	No se registra.
Pruebas de integración	Se registra en el plan de pruebas, pero no se le da trazabilidad. Hay ciertos documentos de planes de prueba, no todos.
Pruebas de accesibilidad	No se registra.

Tabla 16. Contenido de la documentación de calidad del proyecto SICID. Fuente: Elaboración propia.

4.1.2.3. Piels (Plataforma internacional de edición de lengua de señas)

Descripción: Herramienta web basada en *javascript*, *node* y *unity*. Es un proyecto de iniciativa de Inlutec, enfocada en la comunidad de las personas con discapacidad auditiva.

Contenido de la documentación de requerimientos

Para este proyecto, como se evidencia en la Respuesta 3 del Apéndice B de las respuestas a la entrevista aplicada a Víctor Romero, coordinador del equipo de desarrollo web, no se cuenta con requerimientos documentados, se trabajaba en conjunto con personas de la comunidad sorda y se trataba de implementar según las necesidades. Sin embargo, esto causaba un desaprovechamiento del tiempo.

Contenido de la documentación de pruebas

Del mismo modo, no hay documentación de las pruebas efectuadas.

4.1.2.4. Euler

Descripción: Proyecto de iniciativa de Inlutec enfocado proceso de enseñanza y aprendizaje de las matemáticas accesible. Es basado en Python, sin embargo, está en proceso de migración a Java. El producto es una aplicación de escritorio, y se planea pasarlo a una aplicación web.

Contenido de la documentación de requerimientos

En la documentación del proyecto, se puede identificar tanto requerimientos funcionales como no funcionales. En la tabla 17, se muestra las características encontradas.

Requerimientos funcionales	Requerimientos no funcionales
<p>Características:</p> <ul style="list-style-type: none"> • Número de requerimiento. • Nombre del requerimiento. • Descripción. • Prioridad. 	<p>Características:</p> <ul style="list-style-type: none"> • Usabilidad <ul style="list-style-type: none"> ○ Accesibilidad ○ Inclusión ○ Implementación: Requerimientos del proyecto, enfocados en la colocación del producto, metodología de desarrollo y documentación que se debe aportar. ○ Interfaz gráfica • Confiabilidad <ul style="list-style-type: none"> ○ Frecuencia de fallos: Tiempo medio entre fallos (MTBF). ○ Recuperabilidad: Tiempo medio para reparar ante fallos (MTTR). ○ Precisión: Que una solicitud obtenga la respuesta esperada. ○ Seguridad • Rendimiento <ul style="list-style-type: none"> ○ Disponibilidad ○ Tiempo de respuesta • Soporte <ul style="list-style-type: none"> ○ Extensibilidad: Implementación de nuevos requerimientos. ○ Adaptabilidad ○ Compatibilidad: requisitos de software. ○ Mantenibilidad ○ Escalabilidad

Requerimientos funcionales	Requerimientos no funcionales
	<ul style="list-style-type: none"> • Restricciones de diseño: Cuestiones de línea gráfica que se tienen que respetar. • Restricciones de implementación: requerimientos del proyecto, enfocados al tipo y características de las tecnologías de desarrollo que se deben utilizar para crear el producto. • Interfaces • Licenciamiento. • Estándares aplicables

Tabla 17. Contenido de la documentación de requerimientos del proyecto Euler. Fuente: Elaboración propia.

Al hacer la revisión documental y por la experiencia previa de trabajar con el software en desarrollo, se identifican cambios realizados durante el desarrollo del proyecto y la documentación no ha sido actualizada. Incluso al desarrollo actual hay requerimientos del proyecto y del producto no validados y otros que no aplican.

Contenido de la documentación de pruebas

Al igual que en los otros proyectos, la práctica principal de calidad es la aplicación de pruebas unitarias y de integración, enfocados en funcionalidad y accesibilidad. Sin embargo, no hay una documentación de casos de pruebas estándar.

Proceso	Resultado
Pruebas unitarias de funcionalidad	Las pruebas rechazadas se agregan a una lista de hallazgos o <i>issues</i> que tiene tres estados en proceso, pendientes y terminado.
Pruebas de integración de funcionalidad	
Pruebas de accesibilidad	

Tabla 18. Contenido de la documentación de calidad del proyecto Euler. Fuente: Elaboración propia.

4.1.2.5. Resumen de la documentación de requerimientos

Con la revisión de los documentos de los proyectos desarrollados y en desarrollo, se crea la Tabla 19, que muestra un resumen de las características encontradas. La revisión de estos documentos se centró solo en la revisión de requerimientos, cualquier otra información no se revisó, debido a que no está dentro del alcance del del proyecto.

Características de la documentación	Euler	Piels	SICID	EAW
Requerimientos funcionales	Sí	No	Sí	Sí
Casos de uso	No	No	Sí	No
Documentados con una plantilla	Sí	No	Sí	Sí
Requerimientos no funcionales	No ²	No	Sí	Sí
Documentados con una plantilla	No	No	No	Sí

Tabla 19. Comparación entre la documentación de requerimientos de los proyectos involucrados. Fuente: Elaboración propia.

A partir de la documentación de requerimientos funcionales que se encontró en los documentos proporcionados, se hace una comparación de las diferencias que hay entre los diferentes proyectos, esto se muestra en la Tabla 20.

² Si se encuentran requerimientos no funcionales en la documentación del proyecto, sin embargo, la mayoría no es aplicable al proyecto. Excepto las de accesibilidad que son generales a cualquier proyecto.

Características plantilla (funcionales)	Euler	SICID	EAW
Código o número de requerimiento	Sí	Sí	Sí
Nombre del requerimiento.	Sí	Sí	Sí
Tipo de requerimientos	No	No	Sí
Prioridad	Sí	No	Sí
Actores	No	Sí	Sí
Descripción, detalle y/o flujo del requerimiento	Sí	Sí	Sí
Restricciones.	No	Sí	Sí
Formato de pantalla y reportes.	No	No	Sí
Reglas de negocio	No	No	Sí
Precondiciones	No	Sí	No
Resumen	No	Sí	No

Tabla 20. Comparación de la estructura de los requerimientos funcionales de los proyectos involucrados. Fuente: Elaboración propia.

Respecto a los requerimientos no funcionales de la documentación de los proyectos, se determina los siguiente:

- Tanto el proyecto Euler como Sigid, categorizan los requerimientos no funcionales similar a la recomendación del modelo de calidad del producto que hace la ISO 25010 sobre modelos de calidad de software (por ejemplo, confiabilidad, usabilidad y mantenibilidad). No se encuentran requerimientos específicos en cada una de las categorías.
- En el proyecto EAW los requerimientos no funcionales se documentan similar a los requerimientos funcionales, con un código, nombre, tipo y prioridad. Sin embargo, no hay un modelo o plantilla estándar establecido para la documentación de estos requerimientos.

4.1.3. Estado de las pruebas de software

En este apartado se presenta detalles de las pruebas que se realizan en la organización, así como de las herramientas utilizadas y el estado general. Para esto se toma en cuenta los resultados obtenidos de la revisión documental de la sección 4.1.2. de “Revisión, descripción y análisis de la documentación de los proyectos de Inlutec”.

4.1.3.1. Herramientas para pruebas en proyectos web

A continuación, se describen las herramientas que se utilizan, se enfoca principalmente en los proyectos web, pues solo en los proyectos web han sido utilizadas dichas herramientas.

Herramientas para pruebas unitarias o *Code Review* de proyectos web

Como se evidencia en la Respuesta 4 del Apéndice B de las respuestas a la entrevista aplicada a Víctor Romero, coordinador de los proyectos web, indica que el *code review* es un conjunto de actividades que involucra las pruebas unitarias. Las principales actividades que se llevan a cabo son código limpio, alineación a los estándares de código de la organización y aplicación de herramientas (Jasmine, Prittier, Karma y Mocha).

Jasmine es una herramienta que se utiliza para probar el código JavaScript y su comportamiento. Romero (2020) indica que se aplica para crear casos de pruebas del *frontend* que se desarrollan en Angular (utiliza javascript).

Karma se utiliza para crear un ambiente de pruebas, esto permite configurar las condiciones necesarias para ejecutar los casos que se realizan en Angular. Para este caso específico, Karma permite automatizar los casos de pruebas que se establezcan en Jasmine.

Prittier es una herramienta que aplica estándares y reglas de código limpio para typescript. Se utiliza para hacer revisión sintáctica del código desarrollado.

Mocha es un marco para pruebas de javascript basado en *NodeJs*. El desarrollo del Backend, en Inlutec está hecho principalmente con un *framework* de *javascript* y *NodeJs* llamado *loopback*, que permite crear un *application programming interface* (API) para interactuar con la base de datos. Con Mocha se hacen las pruebas de las funciones del API que se desarrollen en *loopback*.

El proceso de cómo se llevan a cabo las pruebas unitarias se encuentra en la Sección 4.3.2.2 (Pruebas unitarias). Romero (2020) indica que, básicamente en el *code review* se hacen las siguientes actividades:

- Revisar la documentación del código.
- Probar la funcionalidad.
- Se aplican las pruebas de *unit testing*, donde se utilizan las herramientas descritas anteriormente.
- Se revisa la lógica y estructura del *feature* desarrollado.
- Revisar el cumplimiento de los estándares de desarrollo.

No se agrega más detalle sobre el código limpio y aplicación de estándares de desarrollo, debido a que no están dentro del alcance del proyecto.

Casos de pruebas de integración

Para las pruebas de integración, se utiliza un elemento que en Inlutec se le ha denominado plan de pruebas, que consiste básicamente en un documento de casos de prueba que tiene la estructura descrita en la Tabla 21.

Campo	Descripción
Historia	La historia indica o hace referencia a la historia de usuario que se ha desarrollado. Para esto, se indica el identificar (Id) y el nombre o descripción del requerimiento.
Escenario	Una historia de usuario puede envolver varios escenarios. Cada uno tiene que someterse a pruebas, para esto se indica el Id y el nombre o descripción del escenario.
Precondiciones	Para cada escenario, se indican cuáles son las condiciones que deben ocurrir antes de ejecutar el escenario. Esto puede involucrar tanto condiciones de la infraestructura (bases de datos en ejecución) como otros escenarios (por ejemplo, ejecutar escenario 1.1.1 del plan de pruebas).
Pasos de prueba	Indican paso por paso cómo se debe ejecutar el caso de pruebas o escenario.
Datos de la prueba	Se indican los datos y el tipo que sean necesario ingresar para poder ejecutar el caso de prueba.

Campo	Descripción
Resultado esperado	Se describe el resultado que se espera del caso de pruebas o escenario, este se compara con el resultado real de la prueba para determinar el éxito o fallo de la prueba.
Fecha y hora de la prueba	Cuando se esté ejecutando el plan de pruebas, para cada caso de prueba o escenario, se debe indicar la fecha para efectos del control.
Resultado real de la prueba	Una vez ejecutada la prueba se debe indicar el resultado obtenido (resultado real). Además, se debe comparar con el resultado esperado asociado al caso de prueba y determinar si se aprueba, rechaza o requiere una mejora.

Tabla 21. Contenido del instrumento de documentación Plan de Pruebas actual. Fuente: Elaboración propia.

4.1.3.2. Estado de la documentación de pruebas

En proyectos como el SICID existen documentos separados donde se ha aplicado el plan de pruebas, principalmente de integración. El resultado de estas pruebas solo se ha utilizado para conocer el estado de las funcionalidades y solucionar errores.

Por otra parte, cuando se reporta un error o se rechaza una prueba, los desarrolladores o el encargado de pruebas crea *issues* o hallazgos para los proyectos respectivos. Para el caso de Euler, se llevan en una hoja de cálculo de Google, que vendría a ser como el *backlog* del producto.

Para el caso de los demás proyectos, utilizan la herramienta ScrumDesk donde manejan todo lo referente al *backlog* y los *sprint backlogs*. En esta herramienta se crean los *issues* para que los desarrolladores con su respectivo coordinador identifiquen en qué deben enfocar el esfuerzo.

En cuanto a resultados de pruebas, no existe documentación formal que se haya obtenido como salida de los procesos de pruebas que se aplican.

4.2. Descripción y análisis de la problemática actual

En esta sección se expone el resultado de las observaciones obtenidas y se presenta un análisis cualitativo de las respuestas obtenidas de las entrevistas y encuestas aplicadas a los sujetos de información.

4.2.1. Observación de los procesos de gestión de requerimientos y verificación y validación.

De los resultados de la observación en cuanto a los procesos de requerimientos y calidad de software, se describe en la Tabla 22 (Ver Respuesta 5 del Apéndice B para más detalle).

Proceso	Observaciones
Requerimientos	<ul style="list-style-type: none">• No se tiene documentación estándar de los requerimientos.• Algunos se llevan en documentos, otros en aplicación web y existen otros que no están documentados en ninguna parte.• No se realiza un proceso de gestión de requerimientos que involucre análisis y especificación.
Calidad	<ul style="list-style-type: none">• Aseguramiento de la calidad generalmente se refiere solo a pruebas del software.• La actividad de calidad principal es la aplicación de pruebas de funcionalidad y accesibilidad, que es enfocado en verificación y validación de software.• La importancia del resultado de pruebas es, principalmente, saber qué partes del software está bien y qué se tiene que solucionar en el momento, por lo tanto, no existen registros ordenados de pruebas aplicadas.

Proceso	Observaciones
En general	<ul style="list-style-type: none"> • No hay comunicación entre los dos principales equipos, para compartir las buenas prácticas implementadas o herramientas utilizada. • Hay diferencias en la documentación, procesos, prácticas y herramientas de los dos principales equipos de desarrollo.

Tabla 22. Resultado de las observaciones realizadas por el investigador. Fuente: Elaboración propia.

4.2.2. Análisis cualitativo de la gestión de requerimientos

La entrevista sobre requerimientos fue aplicada a los coordinadores de los proyectos de software (Ver Respuesta 3 del Apéndice B para más detalle). A partir de esas entrevistas se determina los siguiente:

- Los requerimientos que se han documentado se hacen al inicio del proyecto, estos no se actualizan.
- Algunos proyectos no tienen un documento de requerimientos, esto se da principalmente en los proyectos de iniciativa de Inlutec.
- No se gestionan la documentación de cambios o nuevos requerimientos durante el desarrollo del proyecto.
- Los requerimientos se manejan en las herramientas que cada equipo destina para llevar el control del *backlog* del producto.
- Para algunos proyectos mantener el documento de requerimientos o gestionar los cambios implica tiempo, por lo tanto, lo dejan de lado.
- A raíz de la poca gestión de requerimientos, se ha encontrado con muchos requerimientos ambiguos. En ocasiones esto se puede convertir en retrabajo.
- Los principales problemas, que les representa bajo esfuerzo en la gestión de requerimientos, son expectativas diferentes entre los desarrolladores y los dueños del producto. Además, la incorporación de requerimientos sin analizar que afectan el desarrollo del proyecto e incumplimientos de *deadlines*.
- En cuanto a accesibilidad, se tienen a la mano a los evaluadores que proporcionan los requisitos de accesibilidad, y se ha empezado a hacer revisiones periódicas para validar los requerimientos con los involucrados directos del proyecto.

4.2.3. Análisis cualitativo de la gestión de pruebas

En cuanto a calidad se hizo la entrevista enfocada a entender el proceso de aplicación de pruebas, pues en la observación se identificó que las pruebas es la práctica de calidad de software que se realizan en la organización.

La entrevista sobre pruebas fue aplicada a los coordinadores de los proyectos de software (Ver Respuesta 4 del Apéndice B para más detalles). A partir de esas entrevistas se determina los siguiente:

- En términos generales los coordinadores entienden que aseguramiento de calidad no se enfoca solamente en prueba, sino que este proceso afecta todo del ciclo de vida del desarrollo del software.
- De todo el aseguramiento de calidad solo se realiza verificación y validación de software.
- En cuanto a validación y verificación de software se concluye que las pruebas deben ser un componente importante en el proceso de desarrollo de software.
- Las pruebas que se realizan principalmente son las pruebas funcionales.
- Se realizan pruebas unitarias y de integración manuales. En los proyectos web se ha intentado implementar pruebas automáticas.
- Se trata de aplicar pruebas sobre cada característica que se desarrolla. Sin embargo, en los proyectos web, esto significa tiempo, lo que hace complicado la implementación.
- El resultado de las pruebas no se registra, se centran en identificar defectos o errores para solucionar en el software.
- Desde los inicios del grupo de interés hasta la actualidad, se ha mejora en la metodología de desarrollo, sin embargo, a nivel de aseguramiento de calidad se han identificado oportunidades de mejoras.
- Las pruebas son importantes, porque en ocasiones se presentan errores o defectos en producción que se pudieron detectar con pruebas.
- La definición de requerimientos específicos desde el inicio facilitaría la gestión de pruebas y se puede garantizar que se cumplirán con las expectativas.

4.2.4. Análisis cualitativo de aseguramiento de calidad

En esta sección se realiza el análisis de los cuestionarios, aplicados a los involucrados, sobre un proceso de gestión de requerimiento y aseguramiento de calidad en la organización.

4.2.4.1. Estado de la calidad del software de Inlutec

A seis desarrolladores se les preguntó sobre cómo perciben la calidad del software desarrollado, en términos de alta calidad, media, calidad baja o desconocen el estado (Ver Respuesta 2 del Apéndice B para más detalle). Estas categorías se establecieron de manera subjetiva, debido a que no existe ningún factor de evidencia que las pueda sustentar, son solo para obtener la percepción que tienen los involucrados respecto al software desarrollado. Bajo estas categorías se obtuvo lo siguiente:

- Cuatro personas indicaron que la calidad del software es media o regular.
- Una persona indicó que la calidad del software es baja.
- Una persona indicó que la calidad es buena.

Se puede identificar que el estado de la calidad se percibe diferente, sin embargo, la mayoría concuerda que la calidad del software es regular. Para entender el porqué de la selección de una categoría de calidad, se les preguntó sobre las complicaciones que existen para garantizar la calidad del software y se obtuvo lo siguiente:

- En la metodología de desarrollo actual se requiere apoyo en aspectos de calidad. El tiempo y esfuerzo es dedicado principalmente al desarrollo.
- Necesidad de una guía estándar con métricas que se aplique a los programas, módulos o funciones de software de los proyectos.
- Se necesita realizar pruebas por personas externas al desarrollo.
- Planificar las pruebas para contemplar todas las adecuadas.
- La falta de requerimientos o requerimientos ambiguos están presentes, por lo tanto, afecta la calidad del software.

Del mismo modo, se consultó sobre las mejoras que se necesitan realizar en cuanto a la calidad del software. Los involucrados respondieron lo siguiente:

- Planificar las pruebas e implementarlas en las diferentes etapas del proyecto con una persona que no esté involucrada en el desarrollo.
- Se deben documentar las pruebas y analizar los defectos encontrados.
- Con una gestión de requerimientos las mejoras en cuanto a calidad se podrían solventar.

Tomando en cuenta las respuestas de las personas a quienes se les hizo las preguntas, se puede determinar que:

- Se requiere encontrar una estrategia para probar la calidad del software con los recursos actuales.
- Es necesario un proceso enfocado en calidad, que no afecte la productividad de las actividades de desarrollo.
- Un proceso de gestión de requerimientos puede tener un impacto en la mejora de la calidad del software.
- Las pruebas planificadas, ejecutadas y documentadas pueden ser un insumo para la mejora en la calidad.

4.2.4.2. *Recomendaciones de mejora en el aseguramiento de calidad*

En esta sección se describe el punto de vista del equipo de desarrollo respecto a la propuesta de un proceso de QA. En primera instancia, se preguntó si consideran que la metodología de desarrollo que utilizan actualmente, se le puede incorporar QA (Ver Respuesta 2 del Apéndice B para más detalle).

De los seis desarrolladores, cinco personas indicaron que es posible incorporar un proceso de aseguramiento de calidad a la metodología de desarrollo actual y una persona indicó que es posible, pero también reafirma que se debe analizar la forma de incorporarlo para no afectar las actividades de desarrollo.

Otra pregunta del cuestionario fue, si consideran que un proceso de aseguramiento de calidad aportaría beneficios al proceso desarrollo actual, a continuación, se describen sus respuestas:

- La calidad del software es responsabilidad de los desarrolladores y también es la imagen de una organización que desarrolla soluciones de software.
- Se pueden detectar problemas que no surgen en el ambiente de desarrollo, como aspectos de usabilidad o accesibilidad. Además, se puede tener retroalimentación del equipo de desarrollo e ir adoptando mejores prácticas de programación para minimizar esfuerzos y aplicar lecciones aprendidas.
- Asegurar que el software cumple con las funciones necesarias al momento de exponerlo.
- Reducir la cantidad de errores en el software y facilita concentrar el esfuerzo en los fallos mínimos.
- Garantizar que el software esté cerca de lo esperado por los usuarios.

Una última pregunta se incorporó al cuestionario para identificar las propuestas de mejoras que los desarrolladores visualizan en su contexto, enfocado en un proceso de aseguramiento de calidad. A continuación, se describen:

- Un proceso de QA debe ser ágil, adaptable a las prácticas de scrum y a los horarios de cierre de iteraciones.
- El proceso debe ejecutarse paralelamente al desarrollo de las funcionalidades.
- Debe ser riguroso para detectar los problemas a tiempo, pero sin requerir excesivo esfuerzo de manera que no llegue a ser un cuello de botella. Para esto, debería tener un grado de independencia del desarrollo.
- Debe estar compuesto de procesos concisos y cortos a nivel de desarrollo. Además, analizar cómo hacerlo cuando son proyectos o partes de un proyecto que se tiene que desarrollar rápido.
- Debe adecuarse al flujo de trabajo que tenga el equipo de desarrollo en un momento determinado y aplicarse pruebas cada vez que se libere el estado actual del proyecto.
- Las actividades de QA deben ser planificadas y objetivas.
- También, debe garantizar que se incluye la accesibilidad en los productos desarrollados.
- Las actividades deben ser aplicables, no solo a nivel de un proyecto, sino por etapas del proyecto.

4.2.4.3. *Estado de los requerimientos de software*

A seis desarrolladores se les preguntó sobre cómo perciben el estado de los requerimientos de los proyectos de software, en términos de buena calidad, media, calidad baja o desconocen el estado (Ver Respuesta 1 del Apéndice B para más detalle). Estas categorías se establecieron de manera subjetiva, debido a que no existen ningún factor de evidencia que las pueda sustentar, son solo para obtener la percepción que tienen los involucrados respecto a los requerimientos del software. Bajo estas categorías se obtuvo lo siguiente:

- Dos personas indicaron que la calidad de los requerimientos es media.
- Dos personas indicaron que la calidad de los requerimientos es baja.
- Dos personas indicaron que la calidad de los requerimientos es buena.

Se puede identificar que el estado de los requerimientos se percibe diferente, sin embargo, la mayoría concuerda que la calidad de los requerimientos de software es regular o baja. Para entender el porqué de la selección de una categoría de calidad, se les preguntó las

complicaciones que existen en la gestión de requerimientos del software y se obtuvo lo siguiente:

- Existen requerimientos ambiguos, no documentados y faltantes que causan que otros requerimientos fallen.
- El manejo de cambios de los requerimientos.
- Comunicación de los requerimientos del cliente a los desarrolladores, en ocasiones no se expresa bien la necesidad.
- Falta de rigurosidad por parte del cliente externo en los detalles de la necesidad, esto causa realizar cambios en etapas de liberación o capacitación.
- Comunicación entre el dueño del producto y la persona que hace el levantamiento de los requerimientos.

Del mismo modo, se consultó sobre las mejoras que ellos perciben se debe realizar en cuanto a los requerimientos del software. Los involucrados respondieron lo siguiente:

- Documentación de ideas o necesidades que surgen como requerimientos. Algunos requerimientos quedan como un “*nice to have*” verbales.
- El proceso de tomar y validar requerimientos, por medio de instrumentos adecuados.
- Gestionar los cambios que ocurren posterior al levantamiento de los requerimientos.
- Dedicar tiempo al proceso de levantamiento y validación de requerimientos al inicio del proyecto, para disminuir la cantidad de cambios durante el proceso de desarrollo.
- Hace falta invertir tiempo en prediseños para confirmar con el dueño del producto la necesidad. Generalmente se hace con documentos escritos, que no está mal, pero da espacio para la interpretación.

Tomando en cuenta las respuestas de las personas a quienes se les hizo las preguntas, se puede determinar que:

- Se requiere encontrar una estrategia para disminuir el riesgo de cambios desde el levantamiento de los requerimientos.
- Se requieren estrategias para tomar y validar requerimientos, que representen la necesidad real.
- Se hace necesario evaluar los riesgos que representa los cambios durante del desarrollo del software.

Otra pregunta que se incorporó a cuestionario para identificar las propuestas de mejoras que los desarrolladores visualizan en su contexto, enfocado en un proceso de gestión de requerimientos como parte de un proceso QA. A continuación, se describen:

- Antes de empezar el desarrollo, el proceso debe tratar de identificar y analizar los puntos críticos posibles, que tenga reglas definidas y objetivas.
- Durante el desarrollo del software, el proceso de gestión de requerimientos debe ser corto y conciso que sea fácil de implementar y que no retrase el desarrollo. Esto se puede manejar por medio de revisiones periódicas de avance para la verificación de los requerimientos considerados en el *sprint planning*.
- El desarrollo ágil ya incorpora el manejo de requerimientos durante el desarrollo, pero se requiere reforzar la parte formal.
- Que sea ágil y permita analizar constantemente ante los cambios que surjan.
- Que se incorporen el análisis de requerimientos funcionales, de utilidad y accesibilidad.

Una última pregunta se incorporó para conocer si la herramienta que utiliza, para el manejo de los requerimientos, le es funcional para la trazabilidad de los requerimientos. De los encuestados se identifica que el equipo de desarrollo de Euler utiliza una plantilla de drive (que no facilita la trazabilidad) y los de desarrollo web respondieron que utilizan *ScrumDesk*, una aplicación web. Esto se preguntó para analizar estandarizar la herramienta para todos los proyectos.

4.2.4.4. Recomendaciones de mejora para la gestión de requerimientos de software

En esta sección se describe el punto de vista de los desarrolladores respecto a la propuesta de un proceso de gestión de requerimientos como parte de un proceso de QA. En primera instancia, se preguntó sobre los beneficios que ellos identifican, y respondieron lo siguiente:

- Es útil porque permite a los desarrolladores trabajar en la necesidad real del dueño del producto. Priorizar de acuerdo con las necesidades y reducir la cantidad de cambios y administrar mejor el tiempo.
- Evitar el retrabajo al evitar ambigüedades y tener mejor control de desarrollo en términos de lo desarrollado y lo faltante. Además, estar coordinado con los demás interesados como los evaluadores de accesibilidad y diseñadores.
- Dar seguimiento a los requerimientos y analizar de mejor manera la relación entre lo especificado y lo implementado.

- La entrega final sea lo que el cliente espera.

4.3. Documentación del proceso actual

En esta sección se muestran los diferentes procesos, que tienen que ver con requerimientos y pruebas, llevados a cabo en el desarrollo de software de Inclutec.

Para identificar estos procesos se implementaron las entrevistas a dos involucrados que tienen el rol de coordinador de proyectos en área de desarrollo (Ver Respuesta 2 y Respuesta 3 del Apéndice B para más detalle). Los procesos no están documentados ni estandarizados, por lo tanto, al momento de identificar el proceso actual de requerimientos y pruebas hay diferencias según la forma de trabajo de cada coordinador.

Los procesos que se muestran, en los diagramas por cada sección, ha sido la descripción que los entrevistados brindaron, que refleja la forma en que han intentado realizarlos. Sin embargo, no se aplica para todos los casos.

4.3.1. Proceso de gestión de requerimientos

En esta sección se muestran los diagramas de cómo se realizan los procesos de gestión de requerimientos actualmente.

4.3.1.1. Manejo de requerimientos iniciales

En la Figura 9, se observa el proceso actual de levantamiento de requerimientos para los proyectos de software.

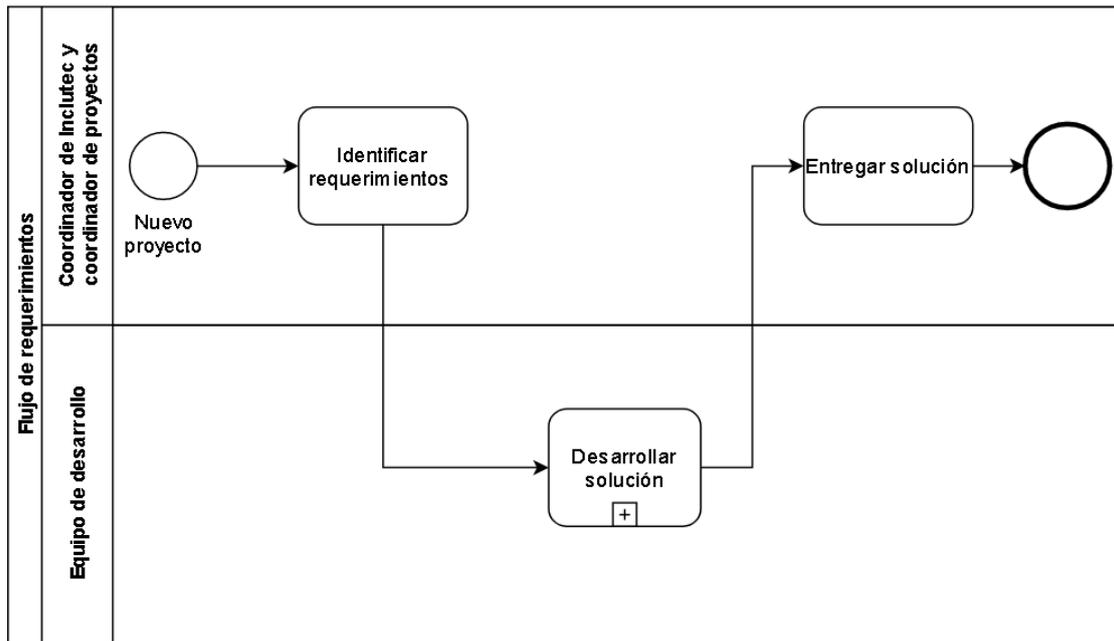


Figura 9. Proceso actual de gestión de requerimientos. Fuente: Elaboración propia.

En la Tabla 23, se muestra la descripción de las tareas del proceso de requerimientos iniciales.

Tarea	Descripción	Datos de entrada	Datos de salida
Identificar requerimientos	Esta tarea involucra identificar y documentar un requerimiento	Necesidades expresadas por el cliente.	Documento de requerimientos (puede ser un documento formal, historias de usuario en <i>ScrumDesk</i> , o cualquier otro elemento).
Desarrollar solución	Es un subproceso que se inicia una vez se tenga el documento de requerimientos.	Documento de requerimientos	Entregas del producto

Tarea	Descripción	Datos de entrada	Datos de salida
Entregar solución	Dar al cliente el producto solicitado	Subproceso de Desarrollar solución ejecutado y terminado	Productos del cliente en el ambiente de producción.

Tabla 23. descripción de las tareas del proceso de requerimientos iniciales. Fuente: Elaboración propia.

En este proceso se ha involucrado dos actores principales, que son:

- Coordinador de Inlutec: Es la persona que hace toda vinculación y atracción de proyectos.
- Coordinador del proyecto: Es el líder que coordina y lleva a cabo el ciclo de vida del proyecto en conjunto con el equipo de desarrollo.

4.3.1.2. Requerimientos durante el proceso de desarrollo

En un proyecto se dan cambios sobre los requerimientos, de este modo, en la Figura 10 se muestra el proceso que involucra cambios sobre los requerimientos.

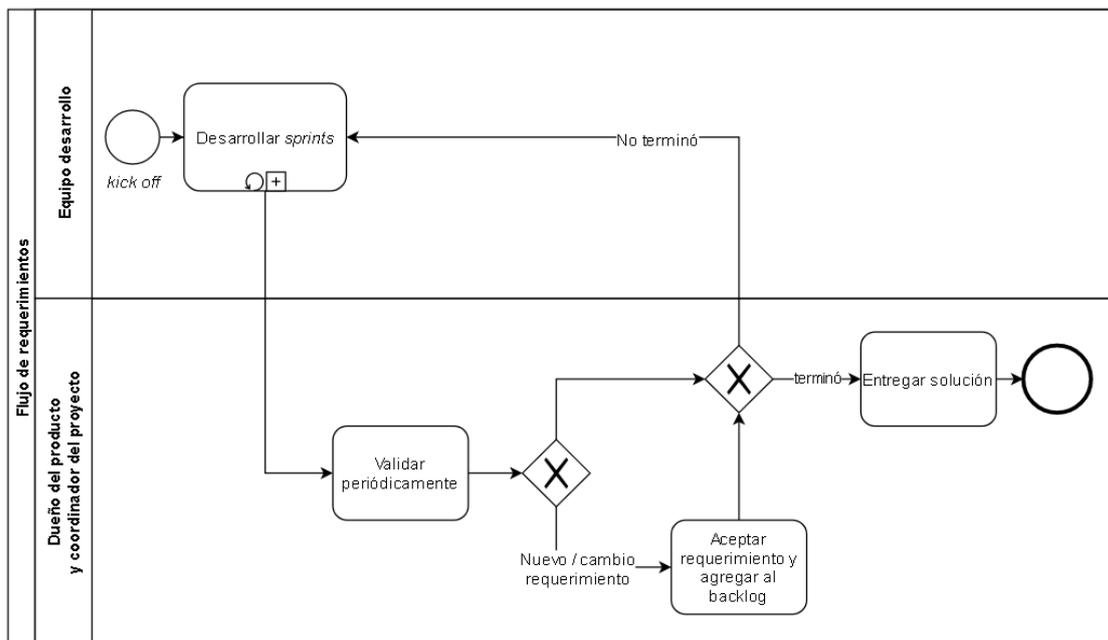


Figura 10. Proceso actual de gestión de cambios de requerimientos. Fuente: Elaboración propia.

En la Tabla 24, se muestra la descripción de las tareas del proceso de requerimientos durante el desarrollo del proyecto.

Tarea	Descripción	Datos de entrada	Datos de salida
Desarrollar <i>sprints</i>	En cada <i>sprint</i> , los desarrolladores hacen incrementos de la solución	Documentos de requerimientos	Incremento de la solución por <i>sprint</i> .
Validar periódicamente	Se hace una revisión con los interesados del proyecto para mostrar el avance	Incremento de la solución	Retroalimentación, cambios y/o sugerencias por parte de los interesados.
Agregar requerimiento al backlog	En una validación pueden surgir cambios o requerimientos, que se deben gestionar.	Cambios o nuevos requerimientos	Documento de requerimientos actualizado.

Tabla 24. Descripción de las tareas del proceso de requerimientos durante el desarrollo del proyecto. Fuente: Elaboración propia.

4.3.2. Proceso de pruebas

La parte de verificación y validación de software es un medio importante en el proceso de garantizar la calidad de los proyecto y software. En Inlutec, en esta línea, existen pruebas unitarias y pruebas de integración.

En el área de desarrollo existe una diferenciación importante en el proyecto Euler y los demás proyectos. Esto se da principalmente porque Euler ha tenido un desarrollo que no es web, a partir de ahí, se ha manejado diferente.

4.3.2.1. Pruebas unitarias Euler

En la Figura 11 se muestra el flujo de tareas que se lleva a cabo para las pruebas unitarias en el proyecto Euler. No hay tareas o pruebas automáticas en este proceso.

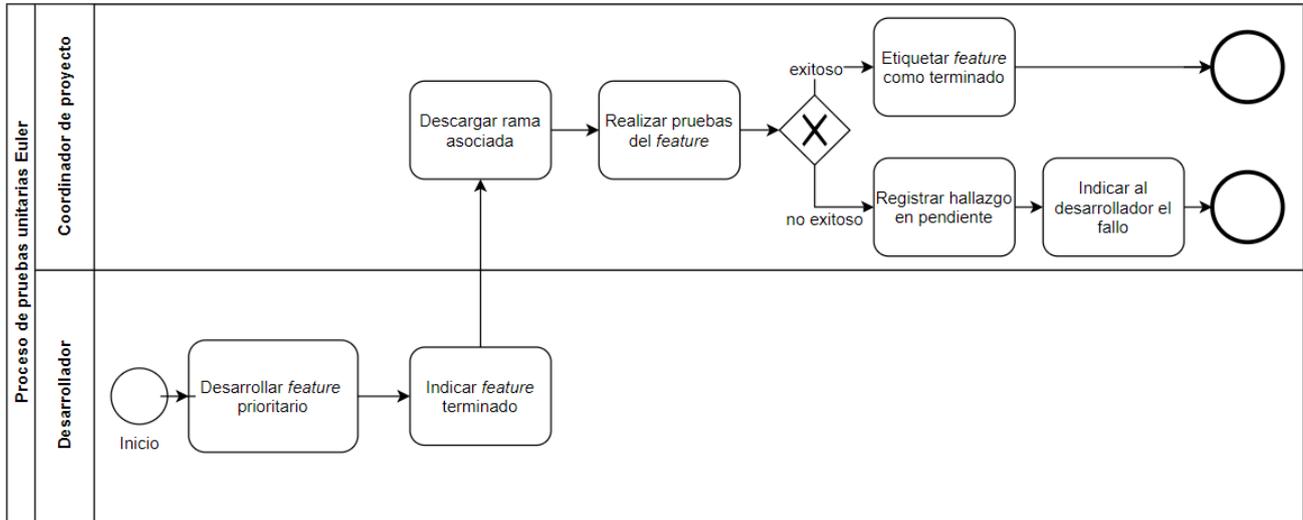


Figura 11. Proceso de pruebas unitarias manuales del proyecto Euler. Fuente: Elaboración propia.

En la Tabla 25, se muestra la descripción de las tareas del proceso de pruebas unitarias de Euler.

Tarea	Descripción	Datos de entrada	Datos de salida
Desarrollar <i>feature</i> prioritario	Los desarrolladores crean un <i>feature</i> según el plan de desarrollo.	Tareas seleccionadas para el desarrollo	<i>Feature</i> desarrollado
Realizar pruebas sobre el <i>feature</i> .	El coordinador del proyecto debe hacer la revisión del <i>feature</i> . Se revisa si el resultado real del <i>feature</i> es el resultado esperado	<i>Feature</i> reportado como terminado.	Cumplimiento o no del resultado esperado.
Etiquetar <i>feature</i> como terminado	Si las pruebas son aprobadas, se cierra el <i>feature</i> .	Cumplimiento o no del resultado esperado.	Lista de <i>feature</i> terminados actualizada

Tarea	Descripción	Datos de entrada	Datos de salida
Registrar hallazgos en pendiente	Si las pruebas no son aprobadas se debe mantener en desarrollo el <i>feature</i> .	Cumplimiento o no del resultado esperado.	Lista de hallazgos actualizada.
Indicar al desarrollador fallo	Si la prueba no se aprueba se indica al desarrollador el motivo	Prueba de <i>feature</i> rechazado.	N/A

Tabla 25. Descripción de las tareas del proceso de pruebas unitarias de Euler. Fuente: Elaboración propia.

4.3.2.2. Pruebas unitarias web

Para los proyectos (excepto Euler) el proceso de pruebas unitarias es como se muestra en la Figura 12.

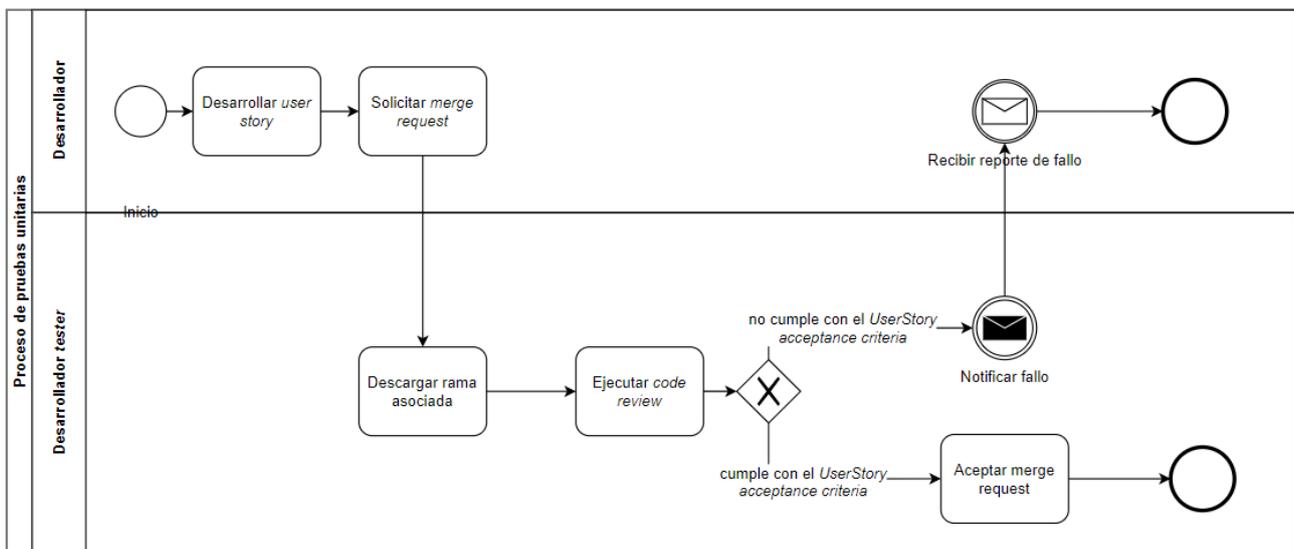


Figura 12. Proceso de pruebas unitarias de los proyectos web. Fuente: Elaboración propia.

En la Tabla 26, se muestra la descripción de las tareas del proceso de pruebas unitarias.

Tarea	Descripción	Datos de entrada	Datos de salida
Solicitar <i>merge request</i>	Al finalizar un <i>feature</i> , los desarrolladores hacen una solicitud para unir su trabajo al resto de solución	<i>Feature</i> terminado sin evaluar.	Merge request realizado.
Compilar localmente	Se compila localmente el incremento	<i>Feature</i> marcado como terminado	Ambiente de prueba local listo para la prueba.
Ejecutar el <i>code review</i>	Llevar a cabo las tareas que involucra el <i>code review</i> .	Incremento compilado localmente.	Resultados reales de la prueba aplicada
Aceptar <i>merge request</i>	El proceso de aceptar el <i>merge request</i> se hace cuando el <i>feature</i> cumpla con el <i>acceptance criteria</i> de la historia de usuario asociada. En caso de que no, se notifica al desarrollador.	Resultados reales de la prueba.	Actualizar la lista de historias de usuario cerradas.

Tabla 26. Descripción de las tareas del proceso de pruebas unitarias de los proyectos web. Fuente: Elaboración propia.

4.3.2.3. Pruebas de integración

El flujo de tareas para la ejecución del proceso de pruebas de integración se muestra en la Figura 13.

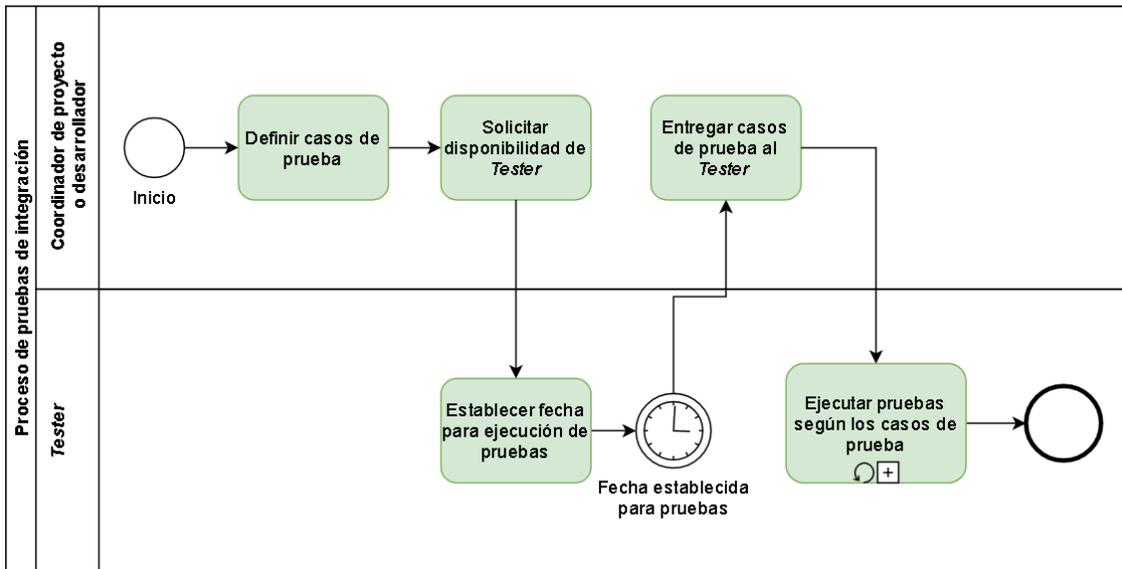


Figura 13. Proceso actual de pruebas de integración de software.

En la Tabla 27, se muestra la descripción de las tareas del proceso pruebas de integración.

Tarea	Descripción	Datos de entrada	Datos de salida
Definir los casos de pruebas	Se prepara la lista de <i>features</i> o funcionalidades para las pruebas.	Incremento o funcionalidades desarrollada a la fecha o según prioridad.	Lista o casos de pruebas.
Solicitar disponibilidad del <i>Tester</i> .	Se solicita disponibilidad a la persona encargada de hacer las pruebas.	N/A	Fecha acordada para pruebas.
Establecer fecha para la ejecución de pruebas	Se analiza la fecha para las pruebas según la carga de trabajo y urgencia.	N/A	Fecha para la ejecución de pruebas.

Tarea	Descripción	Datos de entrada	Datos de salida
Entregar casos de prueba al <i>tester</i> .	Se hace llegar el incremento o lista de funcionalidades a la persona encargada de ejecutar las pruebas.	Lista de casos de prueba	N/A
Ejecutar pruebas según los casos de prueba	Se revisan los resultados reales de cada caso de prueba contra los resultados esperados.	Lista de casos de prueba.	Lista de casos de prueba aceptados y rechazados. Para los rechazados, se indica la razón porque se rechaza.

Tabla 27. Descripción de las tareas del proceso pruebas de integración. Fuente: Elaboración propia.

La figura 14 muestra el subproceso de ejecutar los casos de prueba.

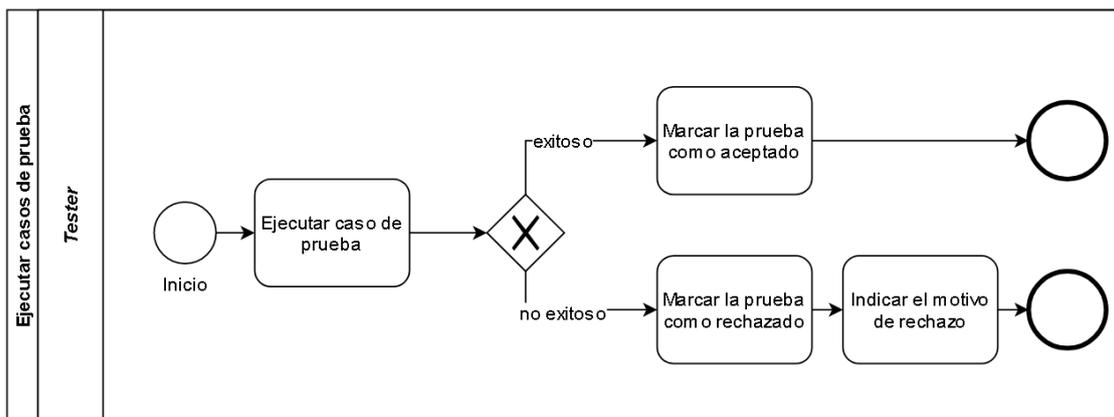


Figura 14. Subproceso actual de ejecución de pruebas. Fuente: Elaboración de propio.

4.3.2.4. Pruebas de accesibilidad Euler

Una parte importante de los proyectos Inlutec es que el software resultado sea accesible. Para lograr este objetivo se someten a pruebas de accesibilidad. Este proceso, en el proyecto Euler, es como se muestra en la Figura 15.

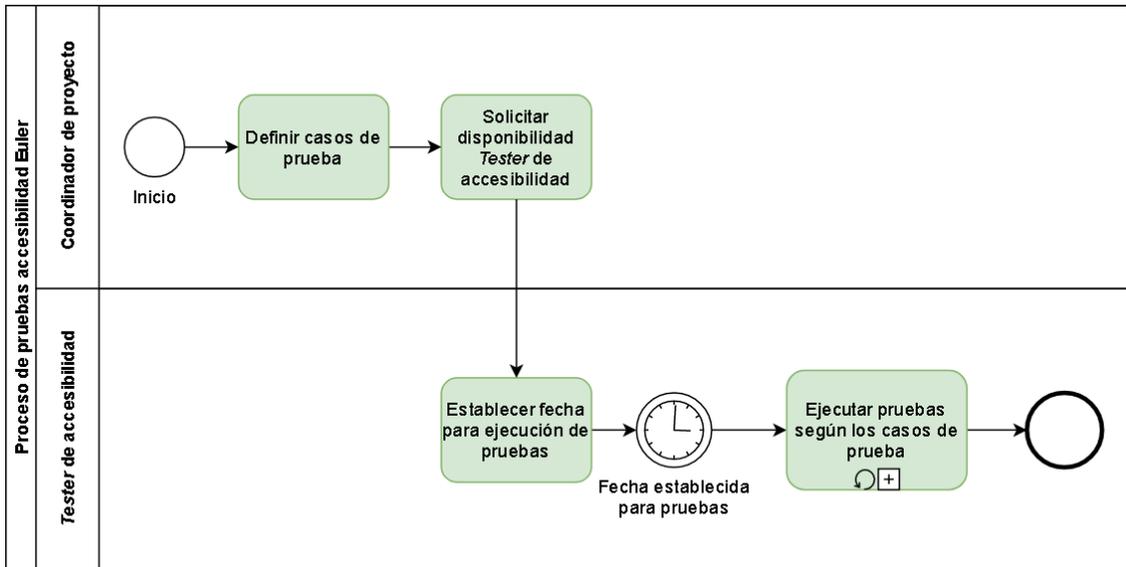


Figura 15. Proceso actual de pruebas de accesibilidad en el proyecto Euler. Fuente: Elaboración propia.

En la Tabla 28, se muestra la descripción de las tareas del proceso pruebas de accesibilidad en el proyecto Euler.

Tarea	Descripción	Datos de entrada	Datos de salida
Definir los casos de pruebas	Se prepara la lista de <i>features</i> o funcionalidades que requieren accesibilidad	Incremento o funcionalidades desarrollada a la fecha o según prioridad.	Lista o casos de pruebas de accesibilidad.
Solicitar disponibilidad del <i>Tester</i> de accesibilidad.	Se solicita disponibilidad a la persona encargada de hacer las pruebas.	N/A	N/A

Tarea	Descripción	Datos de entrada	Datos de salida
Establecer fecha para la ejecución de pruebas	Se analiza la fecha para las pruebas según la carga de trabajo y urgencia.	N/A	Fecha para la ejecución de pruebas.
Ejecutar pruebas según los casos de prueba	Subproceso que indica cómo se ejecutan las pruebas de accesibilidad en Euler	Lista de casos de prueba.	Lista de casos de prueba aceptados y rechazados. Para los rechazados, se registra la razón porque se rechaza.

Tabla 28. Descripción de las tareas del proceso pruebas de accesibilidad en el proyecto Euler. Fuente: Elaboración propia.

En la figura 16, se muestra el flujo del subproceso de ejecución de pruebas de accesibilidad en Euler.

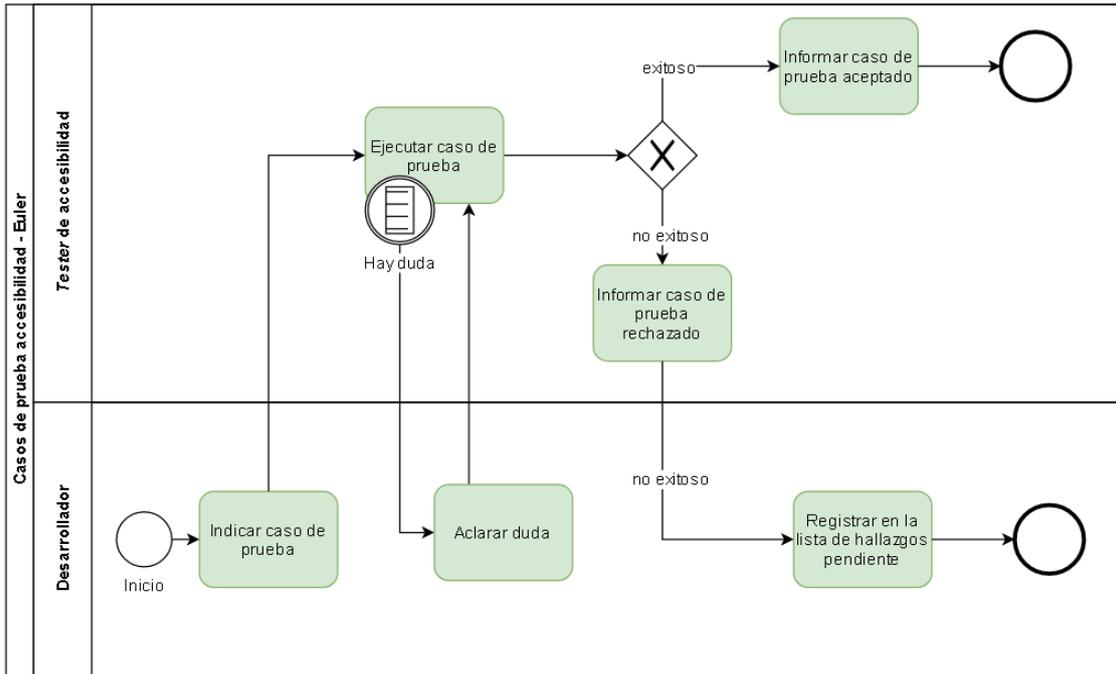


Figura 16. Subproceso de ejecución de pruebas de accesibilidad en el proyecto Euler. Fuente: Elaboración propia.

En la Tabla 29, se muestra la descripción de las tareas del subproceso de ejecutar los casos de prueba de accesibilidad.

Tarea	Descripción	Datos de entrada	Datos de salida
Indicar caso de prueba	Se indica al <i>tester</i> de accesibilidad el caso que tiene que ejecutar.	N/A	N/A

Tarea	Descripción	Datos de entrada	Datos de salida
Ejecutar caso de prueba	Se siguen los pasos de la prueba indicados. En caso de dudas al ejecutar el caso se consulta con el desarrollador que acompaña la prueba.	Caso de prueba de lista.	Resultado real del caso de prueba.
Aclara duda	Si al ejecutar el caso de prueba se tiene alguna duda, se consulta con el desarrollador que acompaña la prueba y este debe aclarar la duda para proseguir con la prueba.	Duda del <i>tester</i> de accesibilidad	Duda aclarada.
Informar caso de prueba aceptado	El <i>tester</i> de accesibilidad compara el resultado real con el esperado y son iguales.	Resultado real	Indicar que se aprueba el caso de prueba.

Tarea	Descripción	Datos de entrada	Datos de salida
Informar caso de prueba rechazado	El <i>tester</i> de accesibilidad compara el resultado real con el esperado y no son iguales	Resultado real	Indicar que se rechaza el caso prueba y el motivo.
Registrar en la lista de hallazgos	Para cada caso de prueba rechazado se registra en la lista de hallazgos del software.	Caso de prueba rechazado y el motivo.	Lista de hallazgos actualizada.

Tabla 29. Descripción de las tareas del subproceso de ejecutar los casos de prueba de accesibilidad. Fuente: Elaboración propia.

4.3.2.5. Pruebas de accesibilidad

Para los proyectos enfocados en web, las pruebas de accesibilidad se han realizado como se muestra en la Figura 17.

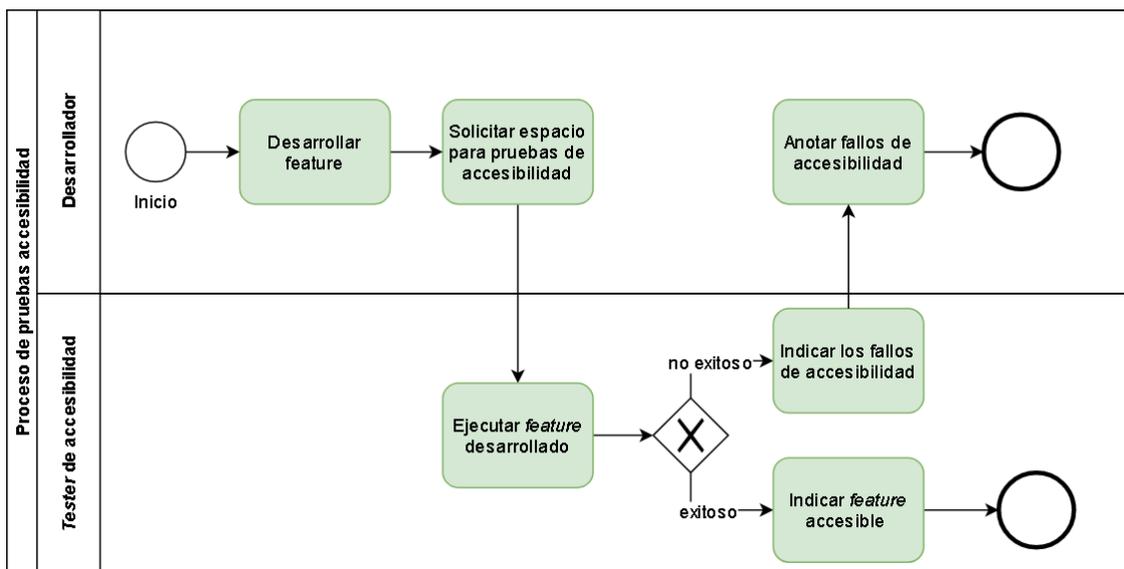


Figura 17. Proceso actual de pruebas de accesibilidad en los proyectos web. Fuente: Elaboración propia.

En la Tabla 30, se muestra la descripción de las tareas del proceso pruebas de accesibilidad en los proyectos web.

Tarea	Descripción	Datos de entrada	Datos de salida
Desarrollar <i>feature</i>	Por cada <i>feature</i> desarrollado el desarrollador a cargo evaluará la accesibilidad.	N/A	N/A
Solicitar espacio para las pruebas de accesibilidad	Desarrollador solicita un espacio con el <i>tester</i> de accesibilidad para realizar las pruebas	N/A	N/A
Ejecutar <i>feature</i> desarrollado.	Se prueba la accesibilidad del <i>feature</i> .	<i>Feature</i> indicado por el desarrollador	Resultado real de accesibilidad.
Indicar <i>feature</i> accesible	El <i>tester</i> de accesibilidad compara el resultado esperado con el resultado real obtenido y son iguales.	Resultado real obtenido y resultado esperado indicado por el desarrollador.	N/A

Tarea	Descripción	Datos de entrada	Datos de salida
Indicar fallos de accesibilidad	El <i>tester</i> de accesibilidad no logra utilizar el <i>feature</i> por fallos de accesibilidad, por lo tanto, el resultado real y esperado son diferentes.	Resultado real obtenido y esperado indicado por el desarrollador.	Fallos de accesibilidad.
Anotar fallos de accesibilidad	El desarrollador debe anotar los fallos de accesibilidad para trabajarlos posteriormente.	Fallos de accesibilidad	N/A

Tabla 30. Descripción de las tareas del proceso pruebas de accesibilidad en los proyectos web. Fuente: Elaboración propia.

4.3.2.6. Análisis de diferencias de los procesos de pruebas en el equipo de desarrollo

Haciendo una revisión de las maneras como se llevan a cabo las pruebas en Inclutec, se identifican ciertas diferencias entre el proyecto Euler y los proyectos web. Se pretende determinarlas para posteriormente, analizar la posibilidad de tener un solo proceso estándar para todos los proyectos.

Las diferencias encontradas en el proceso de aplicación de pruebas se detallan en la Tabla 31.

Euler (Escritorio)	Proyectos Web
En las pruebas unitarias la aceptación o rechazo de pruebas no se hace mediante <i>git</i> .	Utilizan <i>git</i> para la aceptación o rechazo de pruebas unitarias. Además, cuentan con el <i>code review</i> e incursión en pruebas automáticas.
Para las pruebas de integración, solo se proporciona una lista de funcionalidades que se deben probar. Se agrega a la lista de hallazgos lo que haya fallado o los errores encontrados.	Se utiliza el documento “plan de pruebas” donde se indica si se aprueban o rechazan. En ocasiones, se crean <i>issues</i> en <i>gitlab</i> o en <i>scrumdesk</i> .
Como el proyecto está enfocado en personas ciegas, en las pruebas de accesibilidad, los evaluadores hacen las pruebas de integración acompañados de un desarrollador.	Los desarrolladores durante el desarrollo de un <i>feature</i> consultan a los evaluadores sobre la accesibilidad y corrigen en el mismo desarrollo.

Tabla 31. Comparación de la aplicación de pruebas del proyecto Euler y los proyectos Web. Fuente: Elaboración propia.

4.4. Análisis de brecha.

A partir de la documentación del proceso actual, se hace una comparación para identificar la brecha que existe respecto a las prácticas de calidad.

4.4.1. Comparación de la situación actual y las recomendaciones de la teoría

A través de las entrevistas, encuestas, revisión documental y documentación del proceso descritos anteriormente, se puede identificar que el estado actual cuenta con las características que se muestran en la columna “Situación y prácticas actuales” de la Tabla 32, del mismo modo, las recomendaciones teóricas están en la columna “Recomendaciones de las normas o buenas prácticas” de la misma tabla.

Proceso	¿Dónde estamos? ¿Qué tenemos? Situación y prácticas actuales	¿Qué se debería tener? Recomendaciones de las normas o buenas prácticas
Requerimientos	<ul style="list-style-type: none"> • La documentación de requerimientos no es estándar, sin embargo, se cuenta con elementos como identificador, prioridad y tipo. • La especificación de los requerimientos no tiene una forma definida. • De un proceso formal de requerimiento solo se cuenta con el levantamiento donde se hace la documentación inicial. Durante el desarrollo del software, dependiendo la herramienta que se utilice, se actualiza dicha documentación. • Hay proyectos de los cuales no se tiene registro de requerimientos, otros están en herramientas de software que soporta <i>agile</i> y otros en documentos escritos. 	<ul style="list-style-type: none"> • Los requerimientos deben tener un identificador, versión, dueño, prioridad, razón, dificultad y tipo. • Los requerimientos deben especificarse de manera que sean no ambiguos, completos, factibles y verificables. • El proceso de requerimientos debe involucrar levantamiento, análisis, especificación y validación. • Del proceso de gestión de requerimientos debe quedar documentación.

Aseguramiento de calidad	<ul style="list-style-type: none">• No hay un proceso de aseguramiento de calidad establecido. Todo el esfuerzo principal está dedicado al desarrollo.• En cuanto a aseguramiento de calidad se ha ido involucrando al dueño del producto en revisiones periódicas que permita indicar y corregir el rumbo del desarrollo del proyecto.	<ul style="list-style-type: none">• A nivel de proyectos debe existir un proceso de aseguramiento de calidad que contemple todo el proceso de desarrollo del proyecto, desde la planeación hasta el cierre. Además de establecer actividades tanto para aseguramiento del producto como del proceso.• Principalmente en las actividades de aseguramiento del producto, es necesario que se identifique e involucre a los principales interesados, entre ellos el dueño del producto o usuario potencial.
--------------------------	--	---

<p>Verificación y validación</p>	<ul style="list-style-type: none"> • Existe un proceso básico de verificación, que consiste en pruebas de funcionalidades durante el desarrollo del software. Generalmente estas pruebas se hacen a partir de los requerimientos. • No se cuenta con un proceso establecido para la planeación, ejecución, registro y análisis de resultado de pruebas. • En los proyectos de iniciativa de Inlutec se han realizado validaciones, con usuarios potenciales, enfocadas en evaluar la pertinencia de la herramienta, usabilidad y accesibilidad. Sin embargo, se ha aprovechado para levantar nuevos requerimientos y solucionar errores que no detectaron en pruebas. 	<ul style="list-style-type: none"> • Un proceso de pruebas debe involucrar actividades de planeación, preparación y evaluación. • El propósito de aplicar pruebas va más allá de solo evaluar lo desarrollado, también aporta a la mejora continua del desarrollo. Para esto se debe conocer los resultados obtenidos. • Debe haber un proceso de validación del software que permita comprobar que el producto cumple con el objetivo por el cual fue creado.
----------------------------------	--	---

<p>Modelos de calidad</p>	<ul style="list-style-type: none"> • No se hace referencia directa a un modelo de calidad, sin embargo, se realizan algunas actividades que se pueden asociar a un modelo de calidad implícito. • Las pruebas de verificación de funcionalidad, las verificaciones y validaciones de usabilidad y accesibilidad, son las características que se asocian a un modelo implícito de calidad del producto. 	<ul style="list-style-type: none"> • Hay tanto modelos de calidad de uso como del producto, dependiendo de la necesidad se utiliza uno o ambos. Para el caso de este proyecto se hace bajo el modelo de calidad del producto. • El modelo de calidad del producto involucra los requerimientos funcionales y no funcionales.
---------------------------	--	--

Tabla 32. Comparación del estado actual con las recomendaciones de las normas y buenas prácticas. Fuente: Elaboración propia

4.4.2. Matriz FODA del contexto actual

La Figura 18, muestra una matriz FODA resumida de las observaciones, entrevistas, encuestas, revisión documental y documentación del estado actual. Se realiza con el objetivo de identificar estrategias para pasar el estado actual al nivel deseado.



Figura 18. FODA de la situación actual. Fuente: Elaboración propia.

4.4.3. Estado actual y estado esperado

El aseguramiento de la calidad se trata de esas actividades que se realizan durante el ciclo de desarrollo para garantizar que el producto final cumple con las expectativas de los interesados, el enfoque principal del proyecto está en la gestión de los requerimientos y la verificación y

validación del software. De este modo, se realiza una colocación de la situación actual bajo las escalas de CMMI según los procesos involucrados en el proyecto.

De la documentación de la situación actual y la brecha que existe con las recomendaciones básica y principales, la Tabla 33, muestra el estado de los procesos y el nivel que se espera lograr en esos procesos. Para la escala esperada se selecciona el nivel 2, porque esta se obtiene cuando existe un proceso definido que se puede ejecutar periódicamente incluso cuando se esté bajo presión, entonces, este sería el nivel que se espera tener si se implementa la propuesta a desarrollar.

Proceso	Estado actual	CMMI Actual	CMMI esperado
Desarrollo de requerimientos	El proceso se realiza parcialmente.	Capacidad: 0 Madurez: N/A	Capacidad: 2 Madurez: 2
Gestión de requerimientos	El proceso o no se realiza o se realiza parcialmente.	Capacidad: 0 Madurez: N/A	Capacidad: 2 Madurez: 2
Verificación y validación	El proceso se realiza parcialmente	Capacidad: 0 Madurez: N/A	Capacidad: 2 Madurez: 2
Aseguramiento de la calidad	El proceso se realiza parcialmente	Capacidad: 0 Madurez: N/A	Capacidad: 2 Madurez: 2

Tabla 33. Estado actual de los procesos de la organización y estado esperado. Fuente: Elaboración propia.

A partir del resultado actual, el estado esperado en la organización es implementar mejoras en los procesos de requerimientos y verificación y validación, llevando a cabo actividades de aseguramiento de calidad, que les permita tener control de la calidad de los proyectos de software y sus productos.

4.4.4. Estrategias para alcanzar el nivel esperado.

A partir del análisis de la brecha, se determinan las siguientes estrategias:

- Incorporar al equipo desarrollo para evaluar actividades propuestas para los procesos de requerimientos y aseguramiento de calidad. Por medio de la colaboración con los equipos de desarrollo, se puede lograr definir un proceso adecuado que agregue valor al desarrollo actual.
- Enfocar las propuestas a una metodología ágil.
- Simplificar las actividades sin perder objetividad, para que la productividad del equipo de desarrollo no se vea disminuida.
- Estandarizar las actividades para poder centralizar la información, dar trazabilidad y evaluar resultados generales.

5. Propuesta de solución

En este capítulo se define la propuesta de solución a la problemática y necesidad que se evidenció en el Capítulo 4 denominado Análisis de resultados.

5.1. Aspectos generales

Esta sección presenta las generalidades y consideración que involucra la propuesta de solución a la problemática. Para la propuesta de la metodología se considera lo siguiente:

- En primera instancia, en términos de metodología se tiene conocimiento que involucra procedimientos, técnicas, además, la teoría de aseguramiento de calidad indica que debe existir roles definidos para llevar a cabo los procedimientos. Por lo tanto, la metodología tomará como base esos tres elementos principales.
- El proyecto está enfocado en el proceso de aseguramiento de calidad, en el contexto actual esta metodología engloba el proceso de gestión de requerimientos y verificación y validación (control de calidad) de la organización.
- Se definirán roles, técnicas y procedimientos para el proceso general de aseguramiento de calidad y para la gestión de requerimientos y verificación y validación.
- El enfoque de la propuesta de solución será ágil, de manera que se puede implementar en cualquier proyecto de software de Inlutec. No será específica para un determinado proyecto.
- Los procedimientos, técnicas y roles no deben afectar la metodología de desarrollo actual. Esto significa que los procedimientos propuestos deben ajustarse a las etapas que ejecutan actualmente.

5.1.1. Procedimientos

Serán todas aquellas actividades relacionadas entre sí que se requieran llevar a cabo como parte del proceso de aseguramiento de calidad. Los procedimientos definidos deben realizarse sobre todos los proyectos de software que estén en ejecución.

En términos generales, los procedimientos definidos serán el flujo de la metodología de aseguramiento de calidad dentro del alcance establecido.

5.1.2. Técnicas y artefactos

Serán todos los instrumentos que se necesiten para que los procedimientos se ejecuten correctamente. Dentro del grupo de técnicas y artefactos se incluyen, entre otros, las plantillas y documentos que se requieran en la ejecución del flujo de los procedimientos.

5.1.3. Roles

Serán las funciones y responsabilidades que debe realizar la persona involucrada en los procedimientos definidos. Los roles establecidos deberán velar por cumplir sus tareas y garantizar que el procedimiento siga el curso esperado, sin afectar a los demás involucrados o los resultados

El principal rol propuesto es un administrador de aseguramiento de calidad (QAM del inglés *Quality Assurance Manager*), que será la persona responsable de garantizar que se ejecuten los procesos de gestión de requerimientos, gestión de cambios de requerimientos y gestión de verificación y validación. Además, de recopilar los resultados para brindar evidencia.

Adaptado al contexto de la organización esta persona también tendrá que desempeñar un papel dentro de los procesos. En requerimientos, tiene que controlar la documentación y métricas. En pruebas debe planear, ejecutar y monitorear y reportar los resultados.

5.1.4. Conceptualización de la propuesta de aseguramiento de calidad

En la Figura 19, se muestra un diagrama general de la metodología de desarrollo de software actual en Inlutec. La propuesta de solución se debe ajustar a esta forma de trabajo, por lo tanto, los procesos propuestos deben ajustarse al modelo actual.

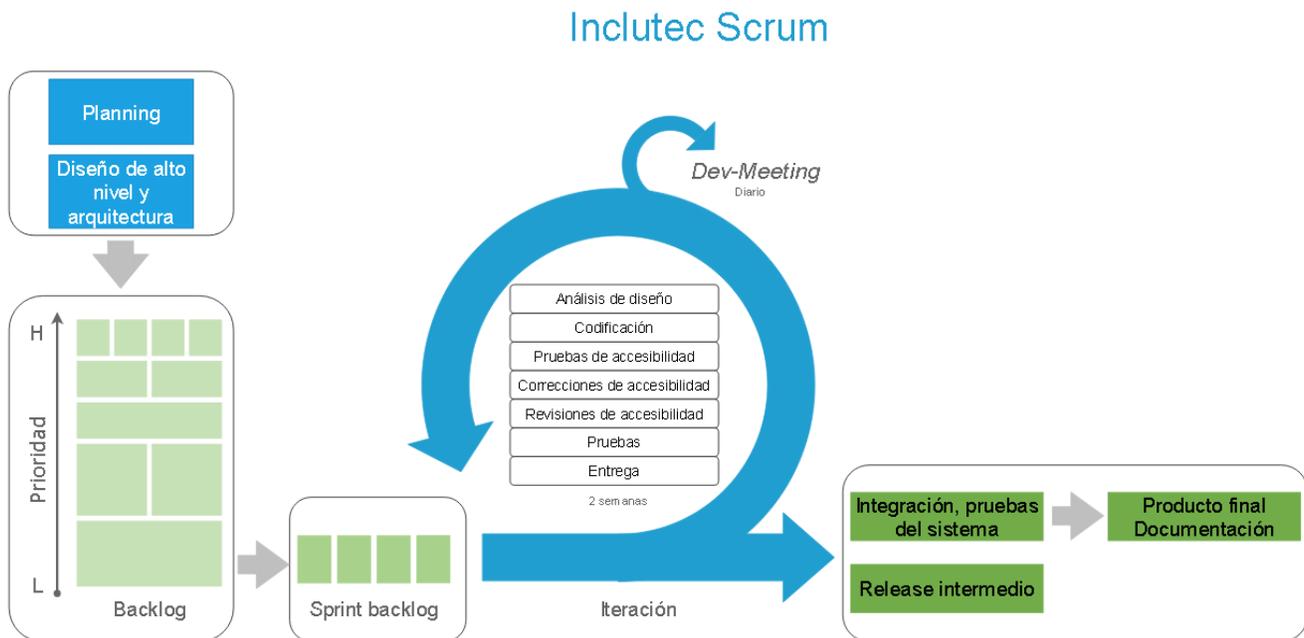


Figura 19. Metodología de desarrollo de software de Inlutec. Fuente: Adaptación de (Romero et al., 2019).

5.1.4.1. Incorporación de un proceso de gestión de requerimientos.

En el proceso de análisis de la situación actual, se identificó que la gestión de requerimientos se centra prácticamente solo en la **obtención**, dejando fuera las actividades de análisis, especificación y validación. Por lo tanto, se busca integrar un proceso para el manejo de los requerimientos, como se muestra en la Figura 20.

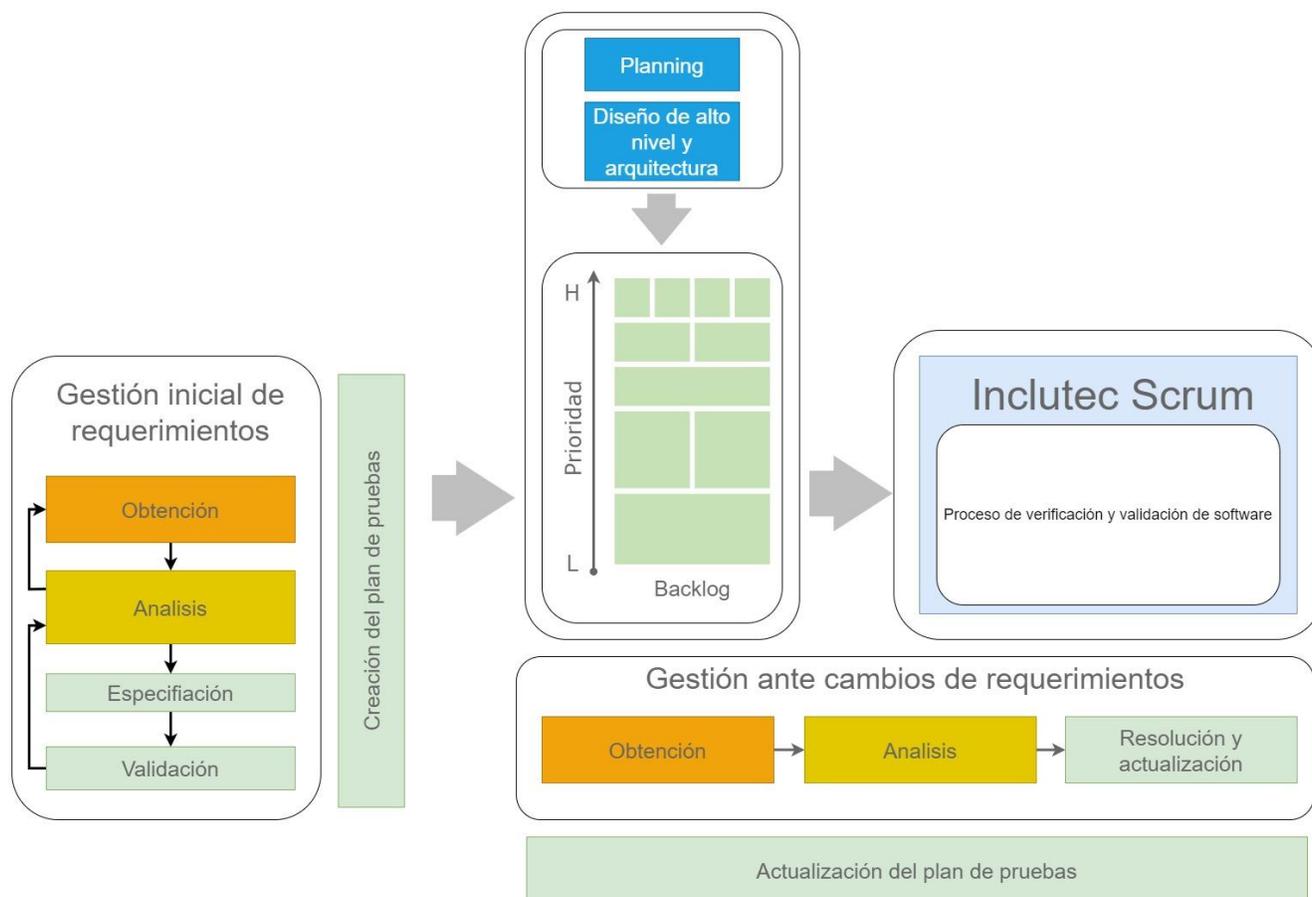


Figura 20. Incorporación de los procesos de aseguramiento al proceso de desarrollo. Fuente: Elaboración propia.

En la gestión inicial de requerimientos tanto la ISO/IEC/IEEE 29148:2018 Ingeniería de requerimientos y el SWEBOK, recomiendan que se debe aplicar un procedimiento que involucre obtención, análisis, especificación y validación de requerimientos. Estas prácticas tienen la intención de asegurar la calidad de los requerimientos y así tener un panorama claro para priorizar y planificar el desarrollo del software.

5.1.4.2. Reforzamiento del aseguramiento de calidad por medio de verificación y validación de software

Las metodologías de desarrollo ágil han sido las alternativas para la entrega de software de calidad en tiempos más reducidos y flexibles ante cambios, durante el desarrollo. Esto les permite a los equipos tomar el control del desarrollo, fragmentar y asegurar que cada *feature* terminado cumplirá con los requerimientos.

La metodología de desarrollo actual, por su naturaleza ágil, contempla la calidad al final de cada ciclo de desarrollo que se ejecute, sin embargo, del capítulo anterior sobre el análisis de resultados y la situación actual, se identifican problemas en la ejecución de las actividades de calidad, lo que genera incertidumbre de la idoneidad de los *features* desarrollados.

Por cada *sprint*, se debería finalizar con pruebas, correcciones y pruebas de integración. En la propuesta de aseguramiento de calidad, se propone reforzar esas actividades por medio de la planeación, implementación de artefactos y recopilación de evidencia, pues son actividades que actualmente no se realizan.

En las siguientes secciones de este capítulo se detallarán los elementos de la propuesta.

5.2. Definición de política de calidad de software

El IEEE Std 730-2014 recomienda que tiene que existir una política de aseguramiento de calidad. Al Inlutec ser una organización pequeña, se define los lineamientos de aseguramiento de calidad con el fin de lograr que los procedimientos se lleven a cabo y pueda obtenerse el valor.

Política de aseguramiento de calidad de software de Inlutec

En esta sección se define una política de orden interno para los involucrados en el proceso de desarrollo de software. Con involucrados se refiere al coordinador y personal administrativo que se relacione con el proceso de desarrollo, evaluadores de accesibilidad, diseñadores y el equipo de desarrollo de software.

- a. Lineamientos del proceso de gestión de requerimientos: Estos lineamientos tienen el propósito de comunicar el proceso que se realizará para la especificación de requerimientos previos al desarrollo del software. A continuación, se enuncian:
 - Al momento del levantamiento de requerimientos debe estar involucrado el coordinador del equipo de desarrollo.
 - Se comunicará a todos los involucrados de la contraparte que se llevará a cabo un proceso que está conformado por las etapas de obtención, análisis, especificación y validación de requerimientos, esto para garantizar que se entienden las necesidades.

- En caso de proyectos por demanda, se debe informar a la contraparte que obtener todos los requerimientos de manera perfecta en la etapa inicial es complejo, de manera que en la primera etapa se identificará el tamaño del sistema, los subsistemas (sus componentes y subcomponentes) para efectos del contrato. Se deben priorizar los subsistemas y componentes y se detallarán los primeros en la lista de prioridad. Los demás se deben cubrir con reuniones periódicas con la contraparte (preferiblemente al final del *sprint*, o según común acuerdo de ambas partes) con el propósito de identificar los detalles de los requerimientos de los siguientes componentes o del siguiente subsistema.
 - Comunicar a los involucrados que durante la ejecución de las etapas de obtención y validación de requerimientos debe estar al menos una persona de la contraparte que tenga conocimiento sobre el proyecto, su propósito y criterio para tomar decisiones.
 - En la validación, los requerimientos especificados tienen que ser revisados por todas las personas directamente involucradas.
 - En el proceso de levantamientos de requerimientos se tienen que identificar el propósito del software y los requerimientos, tanto funcionales como no funcionales.
 - Todos los requerimientos deben documentarse y ser controlados de manera formal durante el desarrollo del proyecto, es decir, en el inicio, durante y al cierre. De este modo cualquier petición o solicitud de cambios que haga el dueño de los productos o cualquier otro involucrado debe documentarse, de lo contrario no puede ser considerado como requerimiento.
- b. Lineamientos del proceso de verificación y validación: Estos lineamientos tienen el propósito de comunicar el proceso que se realizará para el proceso de verificación y aceptación del software. A continuación, se enuncian:
- En las reuniones de planeación y diarias de Scrum deben participar, tanto los desarrolladores como los involucrados en pruebas (integración y accesibilidad) del *sprint* anterior y del *sprint* siguiente. Esto con el fin de determinar el avance en cuanto a pruebas o planear las de la próxima iteración

- Las pruebas que se planeen deben estar basadas en los requerimientos establecidos por la contraparte.
- Se debe cumplir con la planeación, ejecución y reporte de las pruebas unitarias, de integración y accesibilidad.
- Se recomienda analizar en las reuniones de planeación de *sprint* las acciones a tomar basado en los resultados obtenidos del *sprint* anterior.

5.3. Propuesta de la metodología de aseguramiento de calidad.

Esta sección contiene todos los elementos de la metodología propuesta para aseguramiento de calidad, tanto en la gestión de requerimientos como en la verificación y validación de software.

5.3.1. Proceso de gestión de requerimientos de software

Este apartado contiene los procesos de aseguramiento de calidad en la gestión de requerimientos, los roles involucrados y los artefactos propuestos.

5.3.1.1. *Propuesta de proceso de gestión de requerimientos iniciales.*

Dado que la situación actual del proceso de gestión de requerimientos solo se centra en la obtención de los requerimientos y especificación, se propone el proceso que se muestran en la Figura 21.

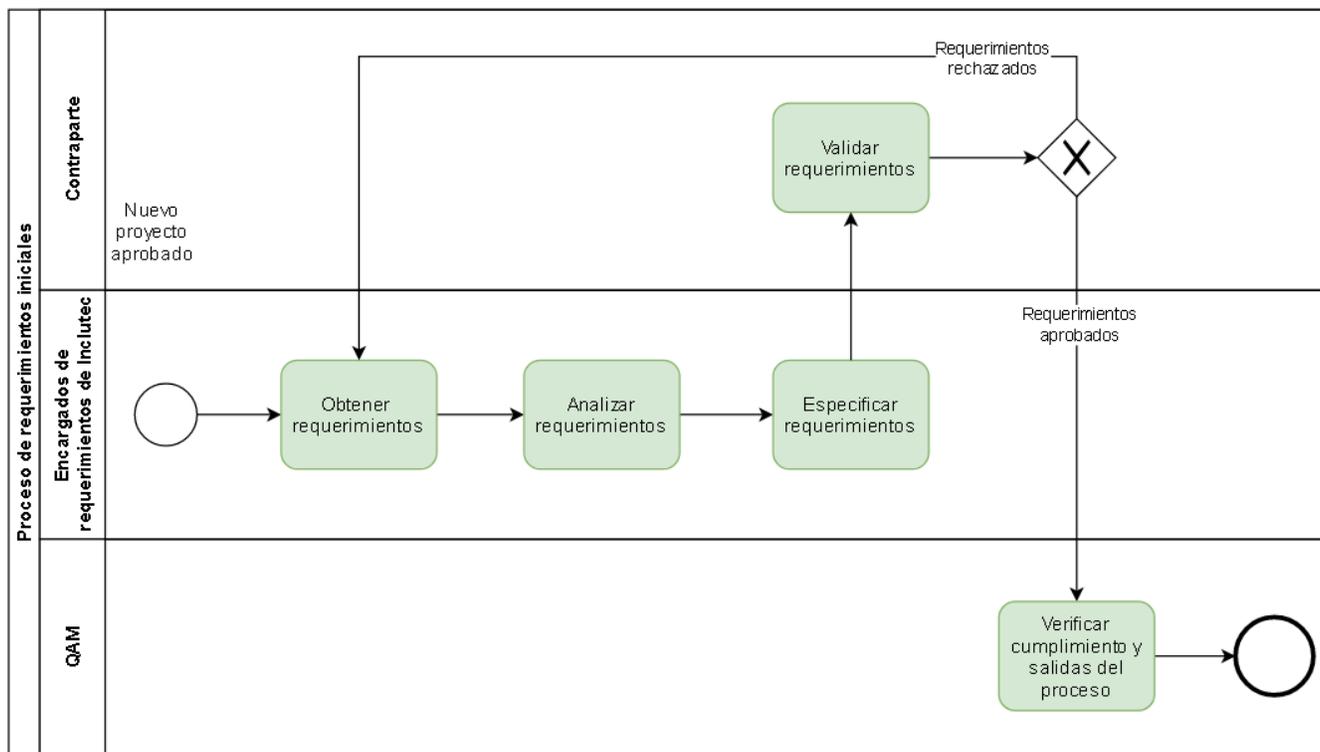


Figura 21. Proceso de recolección y especificación de requerimientos. Fuente: Elaboración propia.

Los actores fundamentales del proceso de requerimientos son los siguientes:

- **Contraparte:** Se refiere a los involucrados que hayan sido designados. En los proyectos de iniciativa de Inlutec, se recomienda como mínimo un miembro del equipo de desarrollo de Inlutec y el dueño del producto o el usuario directo (que serán quienes tenga la necesidad y los requerimientos). En los proyectos por demanda se recomienda como mínimo un encargado de la contraparte y un miembro del equipo de desarrollo. Todos estos involucrados (los mínimos u otros que se involucren) deben conocer el proyecto, su propósito y tener capacidad para emitir criterio en el proceso.
- **Encargados de requerimientos de Inlutec:** Son los designados de Inlutec. Como mínimo se espera que sean quienes participaron con los usuarios, dueños del producto o contraparte.
- **QAM:** Son las personas que llevarán a cabo el desarrollo de la solución especificada.

En la Tabla 34, se describen las tareas para obtener, analizar, especificar y validar requerimientos.

Tarea	Descripción	Datos de entrada	Datos de salida
Obtener requerimientos	<p>Acercamiento inicial y obtención de requerimientos expresados por el dueño del producto.</p> <p>Artefacto (Ver Artefactos en la gestión de requerimientos)</p> <ul style="list-style-type: none"> • Entrevista • Escenarios 	<p>Información del propósito del proyecto e importancia, requerimientos funcionales y no funcionales.</p>	<p>Conjunto de requerimientos no estructurados.</p>
Analizar requerimientos	<p>Se toman los requerimientos iniciales no estructurados, se clasifican, se agrupan y se limpian.</p> <p>Artefacto (Ver Artefactos en la gestión de requerimientos)</p> <ul style="list-style-type: none"> • Clasificar requerimientos. 	<p>Conjunto de requerimientos iniciales no estructurados.</p>	<p>Conjuntos de requerimientos, agrupados y clasificados.</p>

Tarea	Descripción	Datos de entrada	Datos de salida
Especificar requerimientos	<p>Se crea el documento de requerimientos siguiendo los estándares definidos en la organización.</p> <p>Artefacto (Ver Artefactos en la gestión de requerimientos)</p> <ul style="list-style-type: none"> • Especificar requerimientos 	Conjuntos de requerimientos, agrupados y clasificados.	Documento formal de requerimientos.
Validar requerimientos	Se valida el contenido del documento formal de requerimientos y se emite criterio de aceptación o rechazo por las partes interesadas	Documento formal de requerimientos.	Criterio de aceptación o rechazo sobre el documento formal de requerimientos
Verificar cumplimiento y las salidas del proceso	El QAM tiene que verificar que los resultados obtenidos de la especificación y validación de software se hayan completado correctamente.	Documentación de requerimientos	Observaciones del proceso

Tabla 34. Descripción de las tareas del proceso de recolección y especificación de requerimientos. Fuente: Elaboración propia.

5.3.1.2. Propuesta de proceso de validación de requerimientos por sprint

Se propone un proceso de validación periódica de los requerimientos con la contraparte donde se puedan verificar los requerimientos que trabajarán en el siguiente *sprint*, esto servirá para identificar cambios de requerimientos o requerimientos que no se habían identificado.

Se recomienda que este proceso se lleve a cabo periódicamente, en reuniones con la contraparte, para que se pueda asegurar la captura de los detalles faltantes de los requerimientos, que sean necesarios para iniciar el desarrollo del *sprint* siguiente. El proceso propuesto se encuentra en la Figura 22.

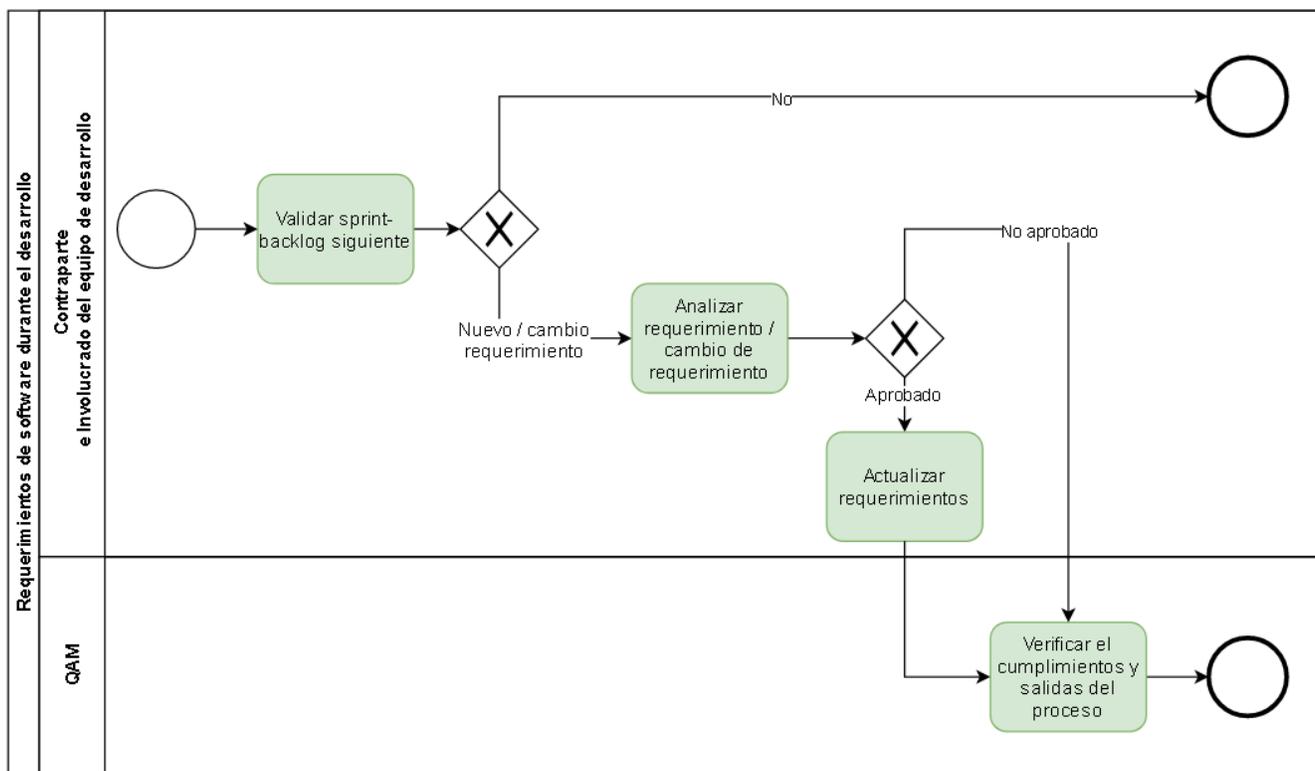


Figura 22. Proceso de gestión de requerimientos durante el desarrollo. Fuente: Elaboración propia.

Los actores principales son los siguientes:

- Contraparte: Es la persona usuario o cliente que valida los incrementos o avances realizados por el equipo de desarrollo de Inlutec.
- Involucrado del Inlutec: Es la persona designada para presentar el incremento o avance a la contraparte. Puede ser el coordinador del proyecto o el desarrollador designado.

En la Tabla 35, se describen las tareas que se deben llevar a cabo si surgen nuevos requerimientos o cambios de requerimientos en las validaciones periódicas con la contraparte.

Tarea	Descripción	Datos de entrada	Datos de salida
Validar <i>sprint backlog</i> siguiente	El encargado debe presentar el <i>sprint backlog</i> a detalle (todos los requerimientos involucrados) que se trabajará en las siguientes semanas y recabar los detalles faltantes.	Requerimientos incluidos en el <i>sprint backlog</i> .	Aceptación de requerimientos correctos o cambios solicitados por la contraparte.

Tarea	Descripción	Datos de entrada	Datos de salida
<p>Analizar requerimiento / cambio de requerimiento</p>	<p>En caso de que haya cambios (por la razón que sea) se debe analizar (con el instrumento de cambios) con el fin de considerar todos los riesgos involucrados. Para lo nuevos requerimientos aplicar el proceso de gestión de requerimientos.</p> <p>Artefacto (Ver Artefactos en la gestión de requerimientos)</p> <ul style="list-style-type: none"> • Cambio de requerimiento 	<p>Sugerencias de cambios</p>	<p>Riesgos identificados por análisis del cambio.</p>

Tarea	Descripción	Datos de entrada	Datos de salida
Actualizar requerimientos	Si considerando los riesgos se aprueba el cambio, entonces se debe actualizar el documento de requerimientos agregando el requerimiento nuevo o actualizar la versión del requerimiento actual.	Cambios o requerimientos nuevos aprobados	Documentos de requerimientos actualizado.
Verificar el cumplimiento y salidas del proceso	El QAM tiene que verificar que los resultados obtenidos de la validación por <i>sprint</i> de software se hayan completado correctamente.	N/A	Observaciones del cumplimiento y justificaciones para los incumplimientos. Estos resultados deben servir para la toma de decisiones en el proceso de mejora continua.

Tabla 35. Descripción de las tareas del proceso de requerimientos durante el desarrollo. Fuente: Elaboración propia.

5.3.1.3. Propuesta de proceso de gestión de cambios de requerimientos

Del mismo modo, para los cambios que se soliciten durante el desarrollo de un *sprint*, se propone el proceso que se muestra en la Figura 23.

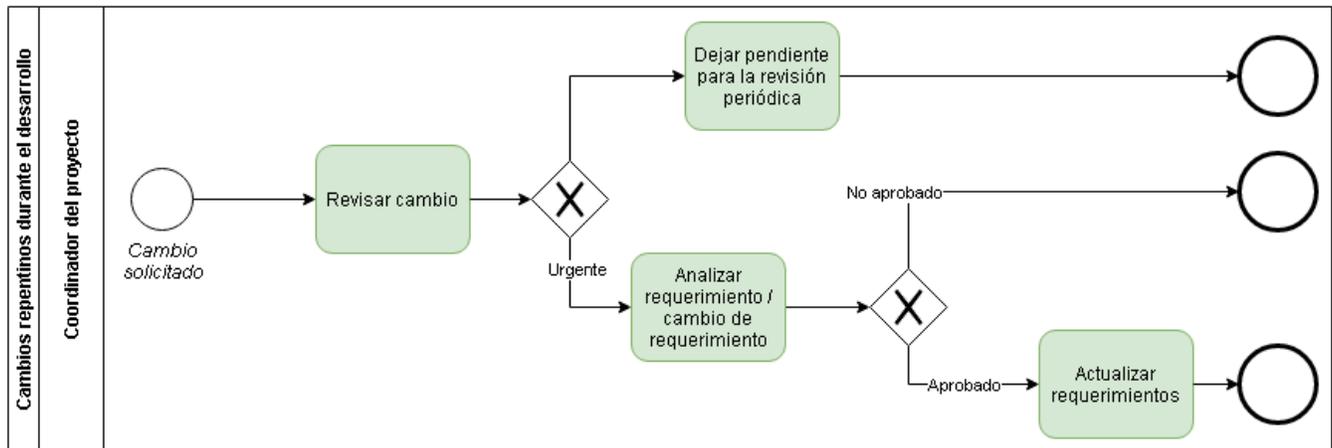


Figura 23. Proceso para la gestión de cambios de requerimientos. Fuente: Elaboración propia.

El actor principal para este proceso es el coordinador de proyectos, pues tienen que tomar decisiones sobre la situación del cambio y cómo impacta al desarrollo.

Según ITIL (2011), hay tres tipos de cambios que deben considerar, estos son los cambios *standard*, los cambios de emergencia y los cambios normales. Para el proyecto actual se propone incorporar y evaluar los cambios normales y de emergencia, para efectos de agilidad.

A continuación, se describen los dos tipos de cambios:

- Cambios normales: serán cambios que no impactan tanto al proyecto, estos pueden ser tanto de bajo como de alto riesgo, pero no tienen que ser atendidos de inmediato.
- Cambios urgentes: Son cambios que pueden ser de bajo o alto riesgo, pero que tienen que ser atendidos de inmediato.

Para la clasificación del cambio, se define la Tabla 36.

Riesgo de Inlutec	Prioridad de la contraparte	
	Alta	Baja
Alto	Urgente	Normal
Bajo	Urgente	Normal

Tabla 36. Matriz de clasificación de cambios de requerimientos. Fuente: Elaboración propia.

En la Tabla 37, se describen las tareas que se deben llevar a cabo si surgen cambios de requerimientos durante un ciclo de *sprint*.

Tarea	Descripción	Datos de entrada	Datos de salida
Revisar solicitud de cambio	El coordinador del proyecto tiene que revisar de identificar si lo analiza de una vez o si puede esperar la revisión periódica con la contraparte.	Cambio solicitado por la contraparte.	N/A.
Analizar requerimiento / cambio de requerimiento	<p>Si el cambio se considera urgente se debe analizar (con el instrumento de cambio.</p> <p>Artefacto (Ver Artefactos en la gestión de requerimientos)</p> <ul style="list-style-type: none"> • Cambio de requerimiento 	Cambio urgente.	Riesgos identificados por análisis del cambio.

Tarea	Descripción	Datos de entrada	Datos de salida
Actualizar requerimientos	Si considerando los riesgos se aprueba el cambio, entonces se debe actualizar el documento de requerimientos agregando el requerimiento nuevo o actualizar la versión del requerimiento actual.	Cambios o requerimientos nuevos aprobados	Documentos de requerimientos actualizado.
Dejar en pendiente hasta la revisión de <i>sprint</i>	Si el cambio no es urgente, se espera a hasta la revisión periódica al final del sprint para negociar con la contraparte.	Cambio normal.	Lista de cambios pendientes.

Tabla 37. Descripción de las tareas del proceso de cambios. Fuente: Elaboración propia.

5.3.1.4. Artefactos en la gestión de requerimientos

En esta sección se describen todos los artefactos propuestos para apoyar el proceso de gestión de requerimientos.

Para cada proyecto deberá existir una carpeta organizada, para agregar toda la documentación que se genere del proyecto y de los artefactos que se utilicen.

Entrevista

El documento genérico de entrevistas para los requerimientos se define en esta sección. El SWEBOK recomienda varias herramientas para la recolección de requerimientos, entre estas, las entrevistas.

Para definir la pila de preguntas que conforman la guía, se utiliza la recomendación que hace Pressman (2010), sobre la recolección de requerimientos, donde indica que se deben hacer preguntas centradas en:

- El cliente y otros participantes, en la metas y beneficios.
- Entender el problema y las percepciones de los involucrados.

Estas preguntas son recomendadas para el primer abordaje al identificar necesidades. Para la aplicación de la entrevista se deben tomar las siguientes consideraciones:

- Las preguntas que se definan son solo una guía, se recomienda que la entrevista sea abierta, el propósito es escuchar lo que tiene que aportar el involucrado.
- Se debe solicitar permiso al entrevistado para grabar la reunión, esto con el fin de recopilar cada detalle que sea importante para el proyecto.
- Esta guía considera preguntas tanto para un proyecto formal por demanda como los proyectos de investigación, por lo tanto, antes de utilizarla se debe identificar las preguntas que aplican para el entrevistado y cuáles no. Además, dependiendo del entrevistado modificar el lenguaje de las preguntas.
- Por la complejidad de recolectar todos los requerimientos de manera perfecta en un solo proceso al inicio, en esta etapa se trata de identificar principalmente el tamaño del sistema y subsistemas (sus componentes y subcomponentes del sistema).
- Se deben priorizar los subsistemas y sus componentes, y enfocarse en recopilar detalles de los primeros en la lista de prioridad.
- Para los siguientes componentes en la lista de prioridad, se debe aplicar este mismo proceso, en reuniones periódicas con los involucrados para garantizar requerimientos más exactos y pertinentes.

Con las consideraciones anteriores se define la guía.

- A. Definición formal del involucrado.
 - I. Nombre
 - II. Organización
 - III. Rol dentro del proyecto.
 - IV. Responsabilidades.
- B. Definir los requerimientos del negocio (Si la persona es dueño del producto).
 - I. ¿Cuál es el propósito de realizar el proyecto?
 - II. ¿Qué esperan lograr con este proyecto?
 - III. ¿Hay solo un área involucrada o afectada hay otras áreas involucradas?
 - IV. ¿Están las áreas relacionadas o son partes y procesos aislados?
 - V. ¿Quiénes son las personas afectadas o usuarias? Perfil, descripciones, funciones.
 - VI. ¿Cuáles es el o los problemas o situación que se quiere solventar?
 - VII. ¿Cuál es la solución adecuada o esperada por el entrevistado?
 - VIII. Para cada subproblema, ¿Por qué se da? ¿Cómo los resuelven ahora?
¿Cómo espera que el proyecto lo solucione?
 - IX. ¿Cuál es el valor que representa el proyecto?
 - X. ¿Puede describir el contexto en que se utilizaría el producto o la solución?
- C. Definir perfil y contexto del involucrado como usuario
 - I. ¿Cuál es la experiencia con software o sistemas informáticos?
 - II. De la experiencia actual con sistemas informáticos o software (web, móviles o escritorio, dependiendo del contexto). ¿Cuáles son las principales limitaciones?
 - III. ¿Cuál es la problemática o situación (como punto de vista del usuario)?
¿Cómo se involucra en la problemática o situación?
 - IV. ¿Cuál es el proceso que sigue para ejecutar sus responsabilidades?
 - V. ¿Cuáles son los productos, salidas o entregables de sus responsabilidades?
 - VI. ¿Hay otros usuarios involucrados en sus responsabilidades? Perfil, descripciones, funciones.

- VII. ¿Cómo identifica que está ejecutando bien sus responsabilidades y entregables?
 - VIII. ¿Qué factores afectan o dificultan que realice sus responsabilidades correctamente? ¿Qué lo facilitaría?
 - IX. Según la problemática o situación ¿Cómo espera que sea la solución de un sistema informático?
 - X. ¿Qué expectativas tiene sobre la facilidad de uso?
- D. Aspectos técnicos (Dependiendo del involucrado entrevistado y a modo de inducción)
- I. Expectativas sobre el entrenamiento.
 - II. Expectativas sobre la seguridad.
 - III. Expectativas de la plataforma (web, web *responsive*, móvil, escritorio).
 - IV. Expectativas de automatización.
 - V. Expectativas de rendimiento (En caso de que no se haya expresado en la solución propuesta).
 - VI. Expectativas del mantenimiento (Si en los proyectos por demanda se acordó en el contrato el mantenimiento y los proyectos de investigación analizar si es aplicable).
 - VII. Aspectos de instalación y configuración (Para los proyectos por demanda o si es aplicable para proyectos de investigación).
 - VIII. ¿Qué aspectos regulatorios están involucrados?
- E. Para el componente prioritario del subsistema prioritario de momento.
- I. Propósito.
 - II. ¿Roles que interactúan (Involucrados)? Perfil, descripción, funciones.
 - III. Flujo.
 - IV. Condiciones de éxito, de fallo y resultado esperado ante fallos.
 - V. Instrumentos o artefactos que utilizan.
 - VI. Datos de entrada (Tipo, tamaño, requerido).
 - VII. Tomar los apuntes según el artefacto de escenario.
- F. Recapitular (Para el entrevistador).

- I. Indicar en sus propias palabras las situaciones o problemáticas, procesos, afectados o involucrados y expectativas de solución identificados.
- II. Abordar aspectos de solución que el involucrado no haya expresado técnicamente. Por ejemplo, “Estos aspectos (características expresadas) se pueden abordar o resolver de esta manera ... ¿Qué le parece?”.

Escenarios

En esta sección se describe el segundo artefacto de la etapa de obtención de requerimientos, que se refiere a la documentación de escenarios, que es una recomendación que Pressman (2010) indica para el entendimiento de los requerimientos.

Para hacer uso de esta herramienta se debe considerar lo siguiente.

- Se aplica en la etapa inicial para los primeros componentes de la lista de prioridad y en las reuniones periódicas sobre los siguientes componentes prioritarios en el sistema.
- Se aplica si el componente involucrado tiene procesos que tienen muchas condiciones que tienen que quedar claras (Queda a consideración del involucrado en la obtención de requerimientos determinar si lo utiliza o no, según su criterio)

La plantilla o estructura para aclarar una situación por medio de escenarios se muestra en la Tabla 38.

Requerimiento asociado	
Nombre del escenario	
Actores	
Secuencia	
Excepciones	

Tabla 38. Plantilla para la documentación de escenarios de requerimientos. Fuente: Elaboración propia.

La información requerida para el uso de la plantilla son los siguientes.

- **Nombre del escenario:** Con el nombre se espera describir el objetivo del escenario. Por ejemplo, pagar trámite con tarjeta de débito.
- **Actores:** Las personas o componentes informáticos involucrados. Por ejemplo, cliente y el sistema de cobro.
- **Secuencia:** Indica la secuencia del escenario desde el inicio hasta que termina. Por ejemplo, ingresar credenciales de la tarjeta y el sistema indica procesa la información y el pago.

- **Excepciones:** Indica todos los casos en que el sistema debe detener o abortar el proceso. Por ejemplo, la fecha de la tarjeta no es válida, el código de seguridad no es válido, entre otros.

Clasificación de requerimientos

La principal actividad para realizar el análisis será ordenar la información obtenida del abordaje hecho en la etapa de recolección de requerimientos. Para esto se deben tomar la siguiente consideración:

- Esta actividad se hará sobre los requerimientos en la etapa inicial, para definir el tamaño del sistema y para efectos del contrato (si es aplicable).
- Esta actividad se realizará posterior a cada reunión periódica con los involucrados.

Para realizar el ordenamiento se define la siguiente estructura:

- Primera clasificación según el enfoque: Se identifican los requerimientos que son del producto y del proceso.
- Clasificación según el modelo de calidad del producto: Esta clasificación involucra si son funcionales y no funcionales. Para los no funcionales se clasifican en desempeño, compatibilidad, usabilidad, confiabilidad y seguridad, además, se define el alcance, es decir, si es para una funcionalidad, componente específico o para todo el proyecto.
- Tercera clasificación según la dependencia: Se identifican los requerimientos son derivados de otro requerimiento de alto nivel.
- Cuarta clasificación según la prioridad: Se define cuáles fueron los aspectos del sistema que el involucrado determinó como prioritario.

Especificación de requerimientos

La especificación de requerimientos es la actividad de formalizar los requerimientos, será la documentación formal del proyecto.

Para documentar los requerimientos se define la siguiente estructura:

- Identificador: Un identificador único que identifique a un requerimiento.

- Descripción: Por el tipo de metodología se establece que el requerimiento tendrá dos componentes, la historia de usuario y las condiciones de aceptación.
- Datos: Si el requerimiento involucra entrada de datos, se tiene que identificar todos los datos y requerimientos de esos datos.
- Número de versión: Es un consecutivo iniciando en 1 (por cada requerimiento en cada proyecto), este debe aumentar si hay cambios en el requerimiento.
- Dueño del requerimiento: Se refiere al usuario que tendrá interacción con el requerimiento en cuestión.
- Prioridad: Es la prioridad que se asigna en el *backlog* y *sprint backlog*.
- Tipo: Tipo requerimiento según el modelo de calidad del producto.

La Tabla 39 y la Tabla 40 son dos ejemplos del uso de la plantilla de especificación de requerimientos.

Identificador	123
Descripción	<p>Historia de usuario</p> <p>Como <i>administrador de la plataforma</i> <role> quiero <i>ver la lista de usuarios</i> <meta> para <i>identificar los activos y no activos</i> <beneficio>.</p> <p>Restricciones</p> <ol style="list-style-type: none"> 1. <i>Si se selecciona ordenamientos</i><condición> <i>se actualizará</i><acción> <i>la lista en el orden seleccionado</i> <restricción de acción>. 2. <i>Si hay más 20 usuarios</i><condición> <i>se mostrará una paginación</i><acción> <i>y en cada página tiene 20 elementos</i> <restricción de acción>.
Datos	N/A
Versión	1
Dueño	Administrador de la plataforma
Prioridad	Alta
Tipo	Funcional

Tabla 39. Ejemplo #1 del uso de la plantilla para la especificación de requerimientos. Fuente: Elaboración propia.

Identificador	124
Descripción	<p>Historia de usuario</p> <p>Como asesor <role> quiero registrar solicitudes digitales <meta> para tener acceso más rápido y ordenado<beneficio>.</p> <p>Restricciones</p> <ol style="list-style-type: none"> 1. Si el campo de nombre no está completado<condición> se indicará <acción> que el espacio de nombre se debe completar<restricción de acción>. 2. Si el número de cédula ya está registrado<condición> se indicará <acción> que el número de cédula actual ya se encuentra registrado<restricción de acción>.
Datos	<p>Dato 1: Cédula</p> <p>Tipo: Número</p> <p>Formato: En nacional x-xxxx-xxxx o e Dimex xxxxxxxxxxxxxx</p> <p>Obligatorio: Sí.</p> <p>Dato 2: Nombre</p> <p>Tipo: Texto.</p> <p>Formato: Tamaño máximo 30 letras.</p> <p>Obligatorio: Sí.</p> <p>Dato 3: Dirección</p> <p>Tipo: Texto largo.</p> <p>Formato: N/A.</p> <p>Obligatorio: No</p>
Versión	1
Dueño	Asesor
Prioridad	Alta
Tipo	Funcional

Tabla 40. Ejemplo #2 del uso de la plantilla de especificación de requerimientos. Fuente: Elaboración propia.

Uso de *ScrumDesk* durante el desarrollo

En la encuesta se identificó que hay diferencias al gestionar los requerimientos durante el desarrollo, principalmente en herramientas que soportan la trazabilidad del requerimiento. En el equipo de desarrollo web han adoptado *ScrumDesk* que funciona bajo los conceptos de Scrum, esta herramienta les facilita la gestión durante el ciclo de vida del desarrollo.

Tomando en cuenta lo anterior se propone que la plantilla de especificación de requerimientos se utilice en la documentación escrita de los proyectos, sin embargo, esa documentación reflejará todos los cambios y estado que se gestionen en *ScrumDesk*.

El uso de *ScrumDesk*, no está generalizado para todos los proyectos. Como parte de la propuesta en la gestión de requerimientos, se plantea que esta herramienta pase a formar parte de la documentación formal de los requerimientos, para lo tanto, se debe considerar lo siguiente:

- Deberá considerar los elementos propuesto en la plantilla de especificación de requerimientos.
- Para realizar algún cambio en algún requerimiento, se debe pasar por el proceso de cambio de requerimiento establecido en esta propuesta.
- Para los elementos que se especifiquen en *ScrumDesk* se debe aplicar la misma política de calidad establecida, para requerimientos, en esta propuesta.

Cambio de requerimiento

En la organización cuentan con una plantilla, esta se creó en los inicios del grupo de interés, sin embargo, no hay registro donde se haya utilizado la plantilla. Por lo tanto, la propuesta involucra modificar la plantilla inicial y socializar la existencia de esta plantilla y la importancia de considerar los elementos que la componen.

Para la gestión de cambios es importante tomar en cuenta los siguiente:

- Utilizar la plantilla de cambios cada vez que exista una solicitud de cambio sobre los requerimientos.
- Cada cambio de requerimientos analizados o aprobados se debe reflejar en la documentación de requerimientos. Esto debe ser realizado por el responsable del proceso.

La plantilla de cambios contiene lo siguiente

- Fecha: La fecha de análisis del cambio.
- Proyecto: El proyecto afectado.
- Coordinador del proyecto o la persona designada: El nombre de la persona que hace el análisis del cambio, generalmente es el coordinador del proyecto o la persona debidamente autorizada.
- Solicitante del cambio: Es el nombre de la contraparte que solicita el cambio.
- Descripción de la solicitud de cambio: Son los aspectos que componen la solicitud del cambio.
- Impacto y riesgos: Los impactos y riesgos en términos de usabilidad y accesibilidad, cronograma, costos, alcance y calidad. En cada aspecto se incluye toda la implicación que conlleva el cambio.
- Aprobación: En la aprobación se incluye el nombre y rol de las personas involucradas que aprobaron el cambio.

La Tabla 41 es la plantilla para el análisis del cambio.

Fecha	
Coordinador del proyecto	
Solicitante del cambio y organización	
Descripción del cambio	
Impacto y riesgos	<u>Usabilidad y accesibilidad</u> <u>Cronograma</u> <u>Costos</u> <u>Calidad</u> <u>Alcance</u>
Aprobación (Nombre – Rol)	

Tabla 41. Plantilla para la gestión de cambios. Fuente: Adaptación del Anexo II.

5.3.1.5. *Métricas de requerimientos*

El ISO 29148 sobre requerimientos de software, indica que las métricas en requerimientos deben servir para determinar la calidad, cantidad y volatibilidad de estos. En la Tabla 42, se describen las métricas de requerimientos por *sprint* ejecutado.

Métrica	Descripción	Cálculo
Progreso de requerimientos	<p>Para cada <i>sprint</i>, se definen una cantidad de requerimientos (en historias de usuario) para desarrollar, se mide el progreso de ese trabajo al final del <i>sprint</i></p> <p>Beneficio</p> <ul style="list-style-type: none"> - Control del cronograma del proyecto 	$\frac{\textit{Historias usuario terminadas}}{\textit{Historias usuario planeadas}}$
Porcentaje de cambios sobre los requerimientos aprobados	<p>De todos los cambios solicitados, se quiere saber cuántos afectan a los requerimientos. Eso después de ejecutar pruebas de aceptación.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Identificar fallos al momento de especificar requerimientos y mejorar para eliminar ambigüedad e incomprensiones. 	$\frac{\textit{Cambios sobre requerimientos aprobados}}{\textit{Total de cambios solicitados}}$

Métrica	Descripción	Cálculo
Requerimientos documentados	<p>Generalmente surgen cambios sobre los requerimientos que pueden generar otros requerimientos, se quiere saber si la documentación de requerimientos corresponde a los requerimientos reales.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Tener control del alcance del proyecto en términos de los requerimientos. 	$\frac{\text{Requerimientos documentados}}{\text{Requerimientos reales}}$

Tabla 42. Métricas para la gestión de requerimientos. Fuente: Elaboración propia.

De las métricas de la Tabla 42, para la primera etapa de implementación de los procesos de QA se trabajarán solamente:

- Progreso de los requerimientos.
- Porcentaje de cambios aprobados de requerimientos.

El QAM es el responsable de recolectar esta información periódicamente en cada cierre de *sprint* con aquellos requerimientos que hayan involucrado cambios y con las características terminadas que hayan sido probadas oportunamente.

5.3.2. Proceso de validación y verificación de software

En esta sección se definen los procesos de aseguramiento de calidad en la gestión de verificación y validación de software, los roles y artefactos propuestos.

La verificación y validación tienen varios enfoques, para la presente propuesta se contempla el análisis de requerimientos (este proceso se describe en la sección 5.3.1 definida como **Proceso de gestión de requerimientos de software**) documentación de pruebas, pruebas de calificación, pruebas de aceptación y proceso de verificación y validación.

5.3.2.1. Documentación de pruebas

En este apartado se muestra la documentación de verificación y validación que se gestionará durante el desarrollo del proyecto. En la Figura 24, se muestra un diagrama de la estructura de las etapas de documentación de pruebas de verificación.

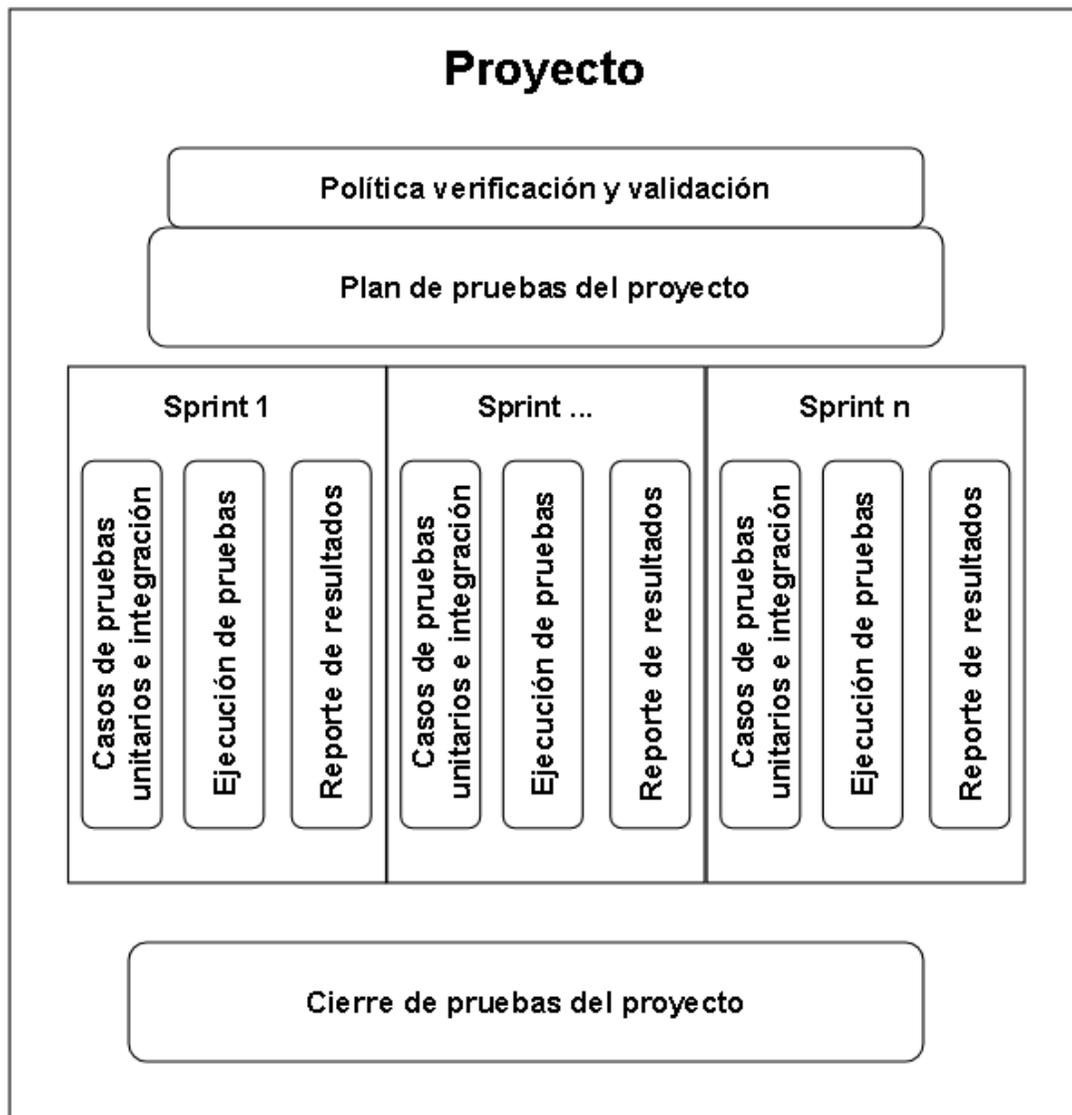


Figura 24. Organización de las pruebas y documentación en el proyecto. Fuente: Elaboración propia.

Plan de pruebas

Según la teoría y la adaptación sobre la documentación de pruebas, se recomienda tener un plan para abordar las pruebas del software desde el inicio hasta el cierre del proyecto.

Se propone que el documento del plan de pruebas debe contar con las siguientes secciones:

- Portada: Para esto se tomará el formato definido para los documentos de la organización, así como la tabla para el control de cambios.
- Introducción: Introduce al proyecto en cuestión, en términos de alcance y componentes.

- **Objetivos:** Describir todas las acciones que se buscan realizar y lograr con el plan de pruebas.
- **Alcance:** Se define lo que se cubrirá con las pruebas y cómo se logrará.
- **Pruebas:** En esta sección se definen las pruebas unitarias y pruebas de integración. También, se documentan las pruebas a partir de los aspectos definidos en el modelo de calidad. Para cada tipo de pruebas se definirá lo siguiente:
 - Definición.
 - Involucrados.
- **Requisitos:** En esta sección se definen los requisitos en términos de software, *frameworks* de desarrollo, almacenamiento y hardware que sean necesarios. Estos requisitos se definen principalmente para el ambiente de pruebas.
- **Supuestos:** Se identifican y definen los elementos con los que se cuenta para inicio del proyecto.
- **Riesgos:** Se definen los posibles riesgos que se vayan a presentar en el proceso de pruebas. Para cada riesgo, se define la estrategia de tratamiento.
- **Aprobaciones:** El contenido de este documento se evaluará con todas los involucrados principales en el desarrollo. Cada involucrado deberá firmar cuando esté de acuerdo.

Especificación de casos de prueba

Para cada tipo de prueba definida en el plan de pruebas, se documentará cada caso de prueba que se defina durante el proyecto. Como los proyectos se gestionan con una metodología de desarrollo basada en *Scrum*, este documento se estará actualizando según cada *sprint* que se planee y ejecute.

Reporte de pruebas

En cada reunión de *sprint planning*, el encargado debe presentar un reporte de pruebas del *sprint*. Para este reporte se tomarán los resultados registrados en los artefactos de prueba y se hará un resumen en términos de las métricas definidas para el proceso (Ver sección **5.3.2.4 Métricas de verificación** y **5.3.2.6 de Métricas de aceptación**).

Además, en cada revisión periódica con la contraparte, se deben realizar reportes de los resultados de los avances del proyecto, bajo las métricas de pruebas de aceptación definidas (Ver sección **5.3.2.6 de Métricas de aceptación**).

5.3.2.2. Propuesta de procesos de pruebas de calificación

Las pruebas de calificación tendrán dos enfoques, las pruebas unitarias y las pruebas de aceptación. Estas son realizadas para verificar que el software se va desarrollando conforme a los requerimientos establecidos, además, que la integración de cada elemento que lo compone responde según lo esperado.

Las principales dificultades identificadas en la organización para definir e implementar un proceso de pruebas es la falta de planificación y documentación de estas. De este modo la propuesta se enfocada definir esos detalles del proceso.

Pruebas unitarias, integración y accesibilidad

Las pruebas unitarias y de integración son las actividades principales para recolectar evidencia sobre el software desarrollado. En cada etapa o ciclo del desarrollo se deben implementar para asegurar el cumplimiento de los requerimientos.

En Inlutec el desarrollo de software se trabaja por *sprint*, previo a cada *sprint* se realiza la planeación donde se acuerda cuáles elementos del *backlog* se incluirán. De este modo, se propone incorporar el planeamiento y preparación de las pruebas asociadas a los requerimientos del *sprint*. Además, se deberá informar los resultados obtenidos de las pruebas en ciclo anterior, para tomar decisiones sobre el desarrollo si es necesario.

La Figura 25 muestra la propuesta para organizar los *sprint planning*, tomando en cuenta las pruebas de software.

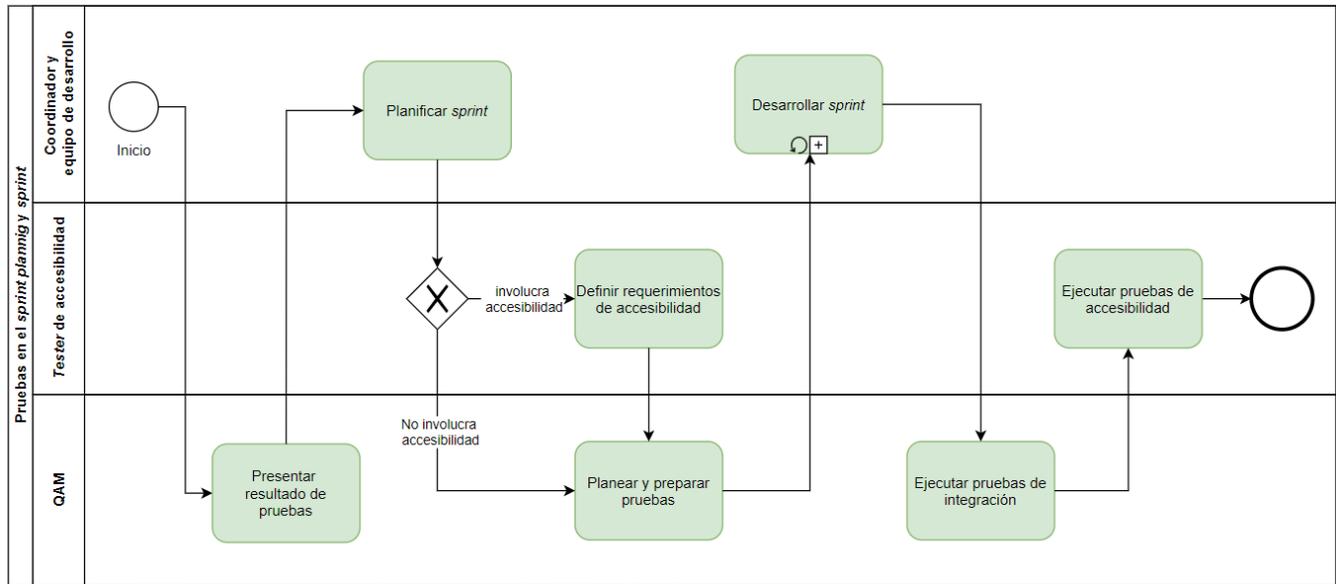


Figura 25. Proceso de planificación y ejecución de pruebas de verificación. Fuente: Elaboración propia.

Los actores principales son:

- **QAM:** Es la persona que comprobará el software (principalmente las pruebas de integración) que se realizarán durante el desarrollo según se planee.
- **Tester de accesibilidad:** Es la persona que comprobará el software (en los casos de prueba donde se involucre la accesibilidad), esto se ejecutará durante el *sprint* según se planee.
- **Equipo de desarrollo:** Son el coordinador del equipo y los desarrolladores que colocan su carga de trabajo para la etapa siguiente.

La Tabla 43 describe las tareas que involucra el proceso de pruebas en el desarrollo de software.

Tarea	Descripción	Datos de entrada	Datos de salida
Presentar resultado de pruebas	La persona encargada presenta el reporte de todas las pruebas ejecutadas, se hace un resumen, con el fin de comunicar algún detalle que aporte al equipo de desarrollo.	Resultado de pruebas	N/A.
Planificar <i>sprint</i>	Los desarrolladores harán su planificación tradicional, tomando en cuenta cualquier hallazgo que se haya identificado en las pruebas.	<i>Backlog</i> y hallazgos de pruebas	<i>Sprint backlog</i>
Definir requerimientos de accesibilidad	El <i>tester</i> de accesibilidad debe definir los requerimientos de accesibilidad tomando como base el <i>sprint backlog</i> seleccionado.	<i>Sprint backlog</i>	Requerimientos de accesibilidad del <i>sprint</i>

Tarea	Descripción	Datos de entrada	Datos de salida
Planear y preparar las pruebas	Se identifican las pruebas de integración que se aplicarán y se estima el tiempo en que deben estar listas para ejecutarlas. Además, los desarrolladores tomarán en cuenta las pruebas unitarias necesarias con sus respectivos criterios de aceptación.	<i>Sprint backlog</i>	Documento de pruebas de integración. Historias de usuarios con sus criterios de aceptación.
Desarrollar el <i>sprint</i>	Se desarrolla las historias de usuario y se aplican las pruebas unitarias (manuales y automáticas) y pruebas de integración.	<i>Sprint backlog</i> y pruebas planeadas.	Funcionalidades y características planeadas.

Tarea	Descripción	Datos de entrada	Datos de salida
Ejecutar pruebas de integración.	Se ejecutan las pruebas de integración que se planearon y se registran los resultados. Si hay requerimientos de accesibilidad involucrados se debe utilizar el EAW y reportar los resultados.	Historias de usuario desarrolladas e integradas	Resultado de pruebas ejecutadas
Ejecutar pruebas de accesibilidad	Se ejecutan las pruebas que involucran accesibilidad, que se planearon y se registran los resultados.	Historias de usuario desarrolladas e integradas que involucran accesibilidad	Resultado de pruebas ejecutadas que involucran accesibilidad

Tabla 43. Descripción de las tareas del proceso de planificación y ejecución de pruebas de verificación. Fuente: Elaboración propia.

El subproceso del desarrollo del *sprint* se muestra en la Figura 26, en este proceso se incorpora la ejecución y documentación de resultados pruebas.

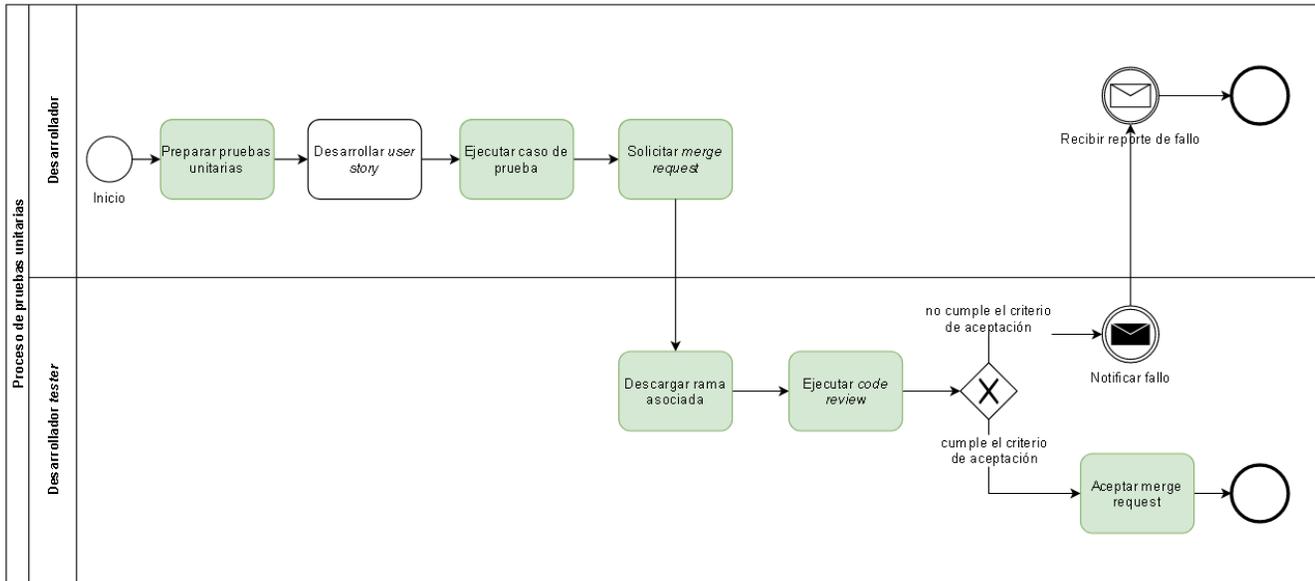


Figura 26. Proceso de la planificación y ejecución de pruebas unitarias. Fuente: Elaboración propia.

Los actores principales de este proceso de desarrollo con pruebas unitarias son los siguientes:

- **Desarrollador:** Es la persona encargada de desarrollar la historia de usuario asignada.
- **Desarrollador tester:** Es el desarrollador encargado de revisar y ejecutar la prueba unitaria.

En la Tabla 44 se describe las tareas principales del proceso.

Tarea	Descripción	Datos de entrada	Datos de salida
Preparar pruebas unitarias	<p>La recomendación es tener las pruebas programadas (en caso de pruebas automáticas) o casos de prueba antes de iniciar el desarrollo, y aclarar cualquier duda al respecto.</p> <p>Artefacto Pruebas unitarias (Ver sección 5.3.2.3 de Artefactos de pruebas de verificación)</p>	Historias de usuario	Casos de pruebas unitarias.
Ejecutar casos de prueba	El mismo desarrollador ejecutará sus casos de prueba para verificar que puede solicitar el <i>merge request</i> .	Historia de usuario desarrollada	N/A
Solicitar <i>merge request</i>	Solicitar que el trabajo sea agregado en integración continua.	Historia de usuario terminado	N/A
Descargar rama	El desarrollador <i>tester</i> , tomará el trabajo y lo ejecutará en su ambiente local.	Rama con una historia de usuario terminada	N/A

Tarea	Descripción	Datos de entrada	Datos de salida
Ejecutar <i>code review</i>	<p>Probar la funcionalidad, ejecutar las pruebas unitarias (automáticas o manuales), registrar el resultado de las pruebas unitarias y levantar <i>issues</i> en Gitlab para los casos fallidos. Además, llevar a cabo las otras actividades que involucra el <i>Code review</i> en el equipo de desarrollo de Inlutec.</p> <p>Artefacto Pruebas unitarias (Ver sección 5.3.2.3 de Artefactos de pruebas de verificación) <i>Gitlab issue tracker</i></p>	Código compilado para pruebas	Resultado de las pruebas unitarias.
Aceptar <i>merge request</i>	Si todas las pruebas fueron aprobadas se aprueba el <i>merge request</i> .	Funcionalidad sin fallos identificados	N/A

Tabla 44. Descripción de las tareas del proceso de planificación y ejecución de pruebas unitarias. Fuente: Elaboración propia.

Pruebas de accesibilidad

Las pruebas de accesibilidad se realizan principalmente cuando se trata la interfaz de usuario y la accesibilidad para personas con baja visión y ceguera se vea involucrada. Cuando existan estas situaciones el equipo de desarrollo y pruebas debe involucrar a los evaluadores de accesibilidad en la planeación de sprint como se muestra en la Figura 24.

Las pruebas de accesibilidad manuales se harán al mismo tiempo que las pruebas de integración (durante el desarrollo según el resultado de la planeación de *sprint*). También es responsabilidad del encargado de las pruebas de integración ejecutar los casos de prueba y reportar los resultados, que involucren accesibilidad, en la herramienta de Inlutec EAW.

Pruebas de integración

La propuesta para las pruebas de integración incluye lo siguiente:

- Recolectar la información obtenida del resultado de las pruebas, que se presentará en los reportes, y servirá de insumo para analizar el avance del proyecto, tener información de calidad, cobertura del software y tomar decisiones de mejora en el proceso de desarrollo.
- Involucrar a la persona encargada de las pruebas, tanto en las reuniones de planeación de *sprint* como en las reuniones diarias del equipo para llevar el control del avance de las pruebas y otras actividades de aseguramiento de calidad.
- Las pruebas de integración se ejecutarán durante el *sprint* para no centrar el esfuerzo al final de la etapa.

5.3.2.3. Artefactos de pruebas de verificación

Para cada proyecto deberá existir una carpeta organizada que sirva para agregar la información del proyecto y los artefactos que se utilicen.

Pruebas unitarias.

Para registrar las pruebas unitarias, se propone la Tabla 45 como plantilla.

Id Caso de prueba	Requerimiento asociado	Descripción	Método	Prerrequisitos
Situación	Dato de entrada	Resultado esperado	¿Aceptado?	Observaciones
1			No	
2			No	
3			No	
4			No	
5			No	

Tabla 45. Plantilla para la documentación de casos de prueba unitarias. Fuente: Elaboración propia.

Los datos que se requieren para completar la plantilla son:

- **Id Caso de prueba:** Es un identificador con la estructura PU-XXXX, donde XXXX es un número consecutivo de las pruebas unitarias de un proyecto.
- **Requerimiento asociado:** Se debe indicar el requerimiento al que corresponde esta prueba.
- **Descripción:** Una breve descripción del caso (partir de la historia de usuario).
- **Método:** Los métodos definidos hasta el momento son el manual y automático.
- **Prerrequisitos:** Se indican las consideraciones que se deben tomar en cuenta para ejecutar el caso de prueba.
- **Datos de entrada:** Para cada situación del caso de prueba, se indica el dato de entrada.
- **Resultado esperado:** Para cada situación y dato de entrada, se indica el resultado que se debe obtener.
- **Aceptado:** El aceptado será un “Sí” o un “No”, esta sección la completa el desarrollador que hace la revisión de par.
- **Observaciones:** Dados los resultados esperados y el resultado real, el desarrollador que hace la revisión de par, indicará las observaciones de éxito o de rechazo de la prueba.

Pruebas de accesibilidad manuales y de integración

En Inlutec se ha utilizado un instrumento para ejecutar las pruebas de integración. Se seguirá utilizando la misma estructura tanto para las pruebas de integración como para las pruebas de accesibilidad que realizan los evaluadores.

No se harán cambios en el instrumento, al menos que haya que organizarla diferente para que sea accesible para los evaluadores de accesibilidad. Esto principalmente porque se encuentra en una hoja de cálculo y en ocasiones este tipo de documentos no es accesible para personas con ceguera o baja visión. Sin embargo, el instrumento será el mismo (Ver **Anexo IV. Plantilla para los casos de pruebas manuales e integración.**).

5.3.2.4. Propuesta de métricas de verificación

Las métricas de verificación se validarán en cada *sprint*, por lo tanto, al final de cada *sprint* se debería tener un reporte de estas. En la Tabla 46, se describen las métricas.

Métrica	Descripción	Cálculo
Progreso de pruebas de integración	<p>En cada <i>sprint</i> se hace un planeamiento de pruebas de integración, se determina del porcentaje del progreso de las pruebas planeadas.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Tener control de las pruebas y tomar decisiones ante riesgos por retrasos. 	$\frac{\text{Pruebas integración realizadas}}{\text{Pruebas integración planeadas}}$

Métrica	Descripción	Cálculo
Progreso de pruebas unitarias	<p>En cada <i>sprint</i> se hace un planeamiento de pruebas unitarias, se determina del porcentaje del progreso de las pruebas unitarias planeadas.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Tener control de las pruebas y tomar decisiones ante riesgos por retrasos. 	$\frac{\textit{Pruebas unitarias realizadas}}{\textit{Pruebas unitarias planeadas}}$
Progreso de pruebas de accesibilidad	<p>En el <i>sprint</i> se pueden definir pruebas de accesibilidad, se determina del porcentaje del progreso de las pruebas de accesibilidad planeadas.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Tener control de las pruebas y tomar decisiones ante riesgos por retrasos. 	$\frac{\textit{Pruebas accesibilidad realizadas}}{\textit{Pruebas accesibilidad planeadas}}$

Métrica	Descripción	Cálculo
<p>Porcentaje de pruebas de integración aprobadas</p>	<p>En cada <i>sprint</i> se hace un planeamiento de pruebas de integración, se determina el porcentaje de pruebas de integración aprobadas.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Identificar causas si el porcentaje de aprobación es baja y plantear mejorar en términos de pruebas unitarias. 	$\frac{\text{Prueñas integración aprobadas}}{\text{Pruebas integración planeadas}}$
<p>Porcentaje de pruebas unitarias aprobadas</p>	<p>En cada <i>sprint</i> se hace un planeamiento de pruebas unitarias, se determina el porcentaje de pruebas unitarias aprobadas.</p> <p>Beneficios</p> <ul style="list-style-type: none"> - Identificar causas si el porcentaje de aprobación es bajo y tomar decisiones sobre el desarrollo. - Identificar riesgos e impactos sobre el cronograma y desarrollo del proyecto. 	$\frac{\text{Pruebas unitarias aprobadas}}{\text{Pruebas unitarias planeadas}}$

Métrica	Descripción	Cálculo
<p>Porcentaje de criterios de aceptación aprobados</p>	<p>Cada historia de usuario tiene un criterio de aceptación asociado, se determina cuantos de los <i>merge request</i> solicitados se aprueban por conformidad con el criterio de aceptación, por <i>sprint</i>.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Sirve para tomar saber el avance del desarrollo en términos de funcionalidades aceptadas. - Sirve para insumo de comparación con los resultados de aceptación. 	$\frac{\textit{Merge request aceptados}}{\textit{Merge request solicitados}}$

Métrica	Descripción	Cálculo
<p>Porcentaje de pruebas de accesibilidad aprobadas</p>	<p>En el <i>sprint</i> se pueden definir pruebas de accesibilidad, se determina del porcentaje de pruebas de accesibilidad aprobadas.</p> <ul style="list-style-type: none"> - Identificar causas si el porcentaje de aprobación es baja y tomar decisiones sobre el desarrollo. - Identificar riesgos e impactos sobre el cronograma, desarrollo del proyecto y la accesibilidad para los usuarios. 	$\frac{\text{Pruebas accesibilidad aprobadas}}{\text{Pruebas accesibilidad planeadas}}$

Métrica	Descripción	Cálculo
<p>Densidad de defectos en pruebas de integración</p>	<p>Por cada componente que se someta a pruebas, se pueden encontrar defectos. Se busca identificar la relación entre la cantidad de defectos y las pruebas de integración especificadas.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Permite determinar el grado de robustez del software desarrollado. 	$\frac{\text{Cantidad de defectos encontrados}}{\text{Casos de prueba planeados}}$
<p>Densidad de defectos en pruebas de accesibilidad</p>	<p>Por cada componente que se someta a pruebas, se pueden encontrar defectos. Se busca identificar la relación entre la cantidad de defectos de accesibilidad y las pruebas de accesibilidad especificadas.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Permite determinar el grado de robustez en accesibilidad del software desarrollado. 	$\frac{\text{Cantidad defectos accesibilidad}}{\text{Casos de prueba accesibilidad planeados}}$

Tabla 46. Métricas del proceso de verificación. Fuente: Elaboración propia.

En la primera etapa de implementación de los procedimientos solo se trabajarán las siguientes métricas:

- Progreso de pruebas de integración.
- Progreso de pruebas de accesibilidad
- Progreso de pruebas de unitarias.
- Porcentaje de pruebas de integración aprobadas.
- Porcentaje de pruebas de accesibilidad aprobadas.
- Porcentaje de pruebas unitarias aprobadas.

El responsable de recolección de datos para estas métricas es el QAM (Gestor de aseguramiento de calidad) y se hará periódicamente al cierre de *sprint* que haya involucrado pruebas unitarias, de integración o accesibilidad.

Las otras métricas de la Tabla 46, se irán recolectando a medida se vaya madurando la adopción en los equipos.

5.3.2.5. *Propuesta de proceso de pruebas de aceptación*

Las pruebas de aceptación se harán en cada reunión de revisión periódica que se haga con la contraparte, para esto se deben tomar las siguientes consideraciones:

- En las reuniones periódicas debe participar como mínimo el encargado de pruebas de verificación del proyecto, un miembro del equipo de desarrollo y la contraparte.
- Definir un conjunto de casos de prueba tomando como base los requerimientos y los casos de pruebas funcionales e integración.
- El encargado de la organización y equipo de desarrollo debe presentar los casos a la contraparte o la misma contraparte debe ejecutar los casos de prueba seleccionados.
- Se utilizará el mismo artefacto de las pruebas de integración para llevar el control de los resultados actuales.

El proceso general de pruebas de aceptación periódicas se muestra en la Figura 27.

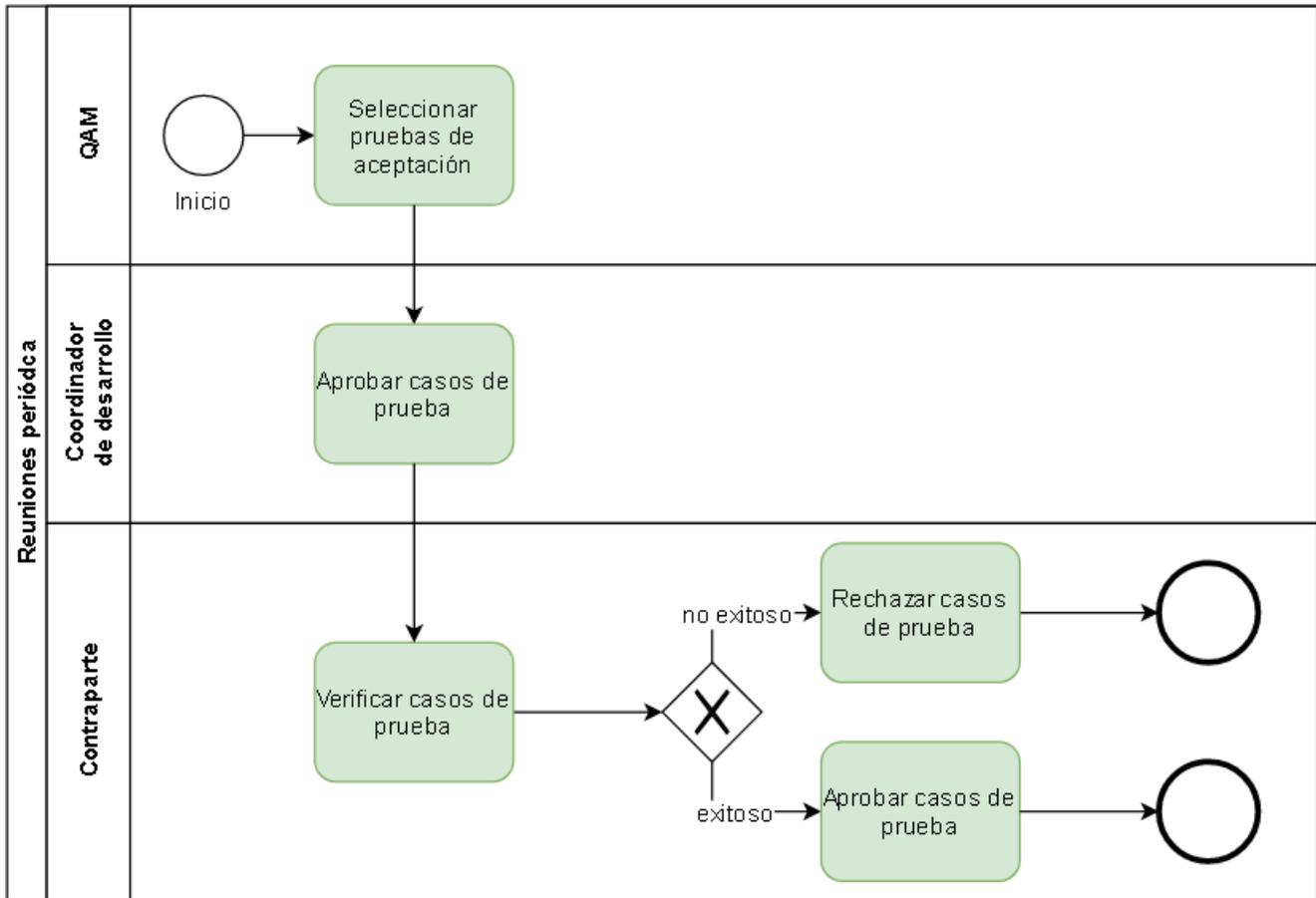


Figura 27. Proceso de pruebas de aceptación. Fuente: Elaboración propia.

Las tareas del proceso se describen en la tabla 47.

Tarea	Descripción	Datos de entrada	Datos de salida
Seleccionar casos de prueba	<p>Se definen los casos de prueba aplicables para las pruebas de aceptación.</p> <p>Artefacto Instrumento de pruebas (Ver Anexo IV. Plantilla para los casos de pruebas manuales e integración.).</p>	Casos de pruebas definidos para el <i>sprint</i> .	Casos de pruebas de aceptación.
Aprobar casos de prueba	Se validan los casos de prueba para verificar que sean aplicables.	Casos de pruebas de aceptación.	Casos de pruebas de aceptación aprobados.
Verificar casos de prueba	La contraparte evalúa el software, ya sea, que lo ejecute el mismo o que se haga una demostración.	Casos de prueba de aceptación aprobados	N/A
Rechazar casos de prueba	A partir de los resultados de las pruebas, la contraparte debe decidir si acepta el resultado o no, con las respectivas observaciones.	Ejecución de casos de prueba	Casos de prueba rechazados y observaciones.

Tarea	Descripción	Datos de entrada	Datos de salida
Aceptar casos de prueba	Según lo determine la contraparte, decide si acepta el conjunto de pruebas y acepta el entregable.	Ejecución de casos de prueba	Aprobación de los casos de prueba y el entregable.

Tabla 47. Descripción de las tareas del proceso de pruebas de aceptación. Fuente: Elaboración propia.

5.3.2.6. Métricas de aceptación

En la Tabla 48, se describen las métricas de aceptación, estas sirven para medir el grado de aceptación de la contraparte y para evaluar la eficacia de las pruebas de verificación que se estén aplicando.

El responsable de registrar estas métricas será el QAM (Gestor de Aseguramiento de Calidad), y se hará entrega y validación con la contraparte.

Métrica	Descripción	Cálculo
Porcentaje pruebas aceptadas	<p>En la demostración periódica con la contraparte, se identifican la cantidad de pruebas aceptadas.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Tomar decisiones en cuanto a las pruebas de verificación aplicadas, en caso de que el porcentaje sea bajo. 	$\frac{\text{Pruebas aceptación aprobadas}}{\text{Pruebas aceptación especificadas}}$

Métrica	Descripción	Cálculo
Densidad de defectos de <i>sprint</i>	<p>Es la relación entre la cantidad de defectos encontrados en las pruebas de aceptación con las pruebas especificadas.</p> <p>Beneficio</p> <ul style="list-style-type: none"> - Tomar decisiones sobre las pruebas unitarias y proceso de aceptación de <i>merge request</i>, en caso que el resultado sea alto. 	$\frac{\text{Cantidad de defectos encontrados}}{\text{Casos de prueba planeados}}$

Tabla 48. Métricas del proceso de validación (aceptación). Fuente: Elaboración propia.

5.3.3. Costos estimados de la implementación de la propuesta

La propuesta realizada implica cambios en las prácticas actuales, y esto a su vez generan costos de implementación en la organización. Por lo tanto, en esta sección se hace un resumen de lo principales costos que puede significar la puesta en marcha de esta propuesta.

Dentro de los costos considerados se incluyen las horas de capacitación e inclusión de personal. Además de la implementación de los nuevos procesos, poner en práctica la propuesta involucra la definición de un nuevo rol llamado Gestor de Aseguramiento de la Calidad, para esto, se hace esta aproximación de los costos que representaría para Inlutec.

Los costos se separan en dos escenarios principales, el primero es la inclusión y capacitación de personal nuevo y el segundo caso es capacitación del personal actual. En las siguientes secciones se muestran los detalles.

5.3.3.1. Costo de contratación de personal nuevo

En este apartado se muestran los costos aproximados correspondiente al crecimiento del equipo de desarrollo al incluir personal nuevo, principalmente desarrolladores y el rol de Gestor de Aseguramiento de Calidad.

Para la obtención de los costos se consideró lo siguiente:

- El salario aproximado de obtuvo del sitio: <https://tusalarario.org/costarica/tu-carrera-profesional/costa-rica-funcion-y-salario/costa-rica-desarrolladores-de-software>.
- Las cargas sociales aproximadas se obtuvieron de: <https://www.ccss.sa.cr/calculadora>.

Las horas de capacitación considera tanto la metodología de desarrollo actual como los procesos de aseguramiento de calidad propuestos. En la Tabla 49, se muestra los costos base salariales de los roles.

Perfil	Salario estimado	Cargas sociales	Total
Desarrollador junior	750,000	37.00%	1,027,500
Desarrollador senior	900,000	37.00%	1,233,000
Director de proyectos	1,200,000	37.00%	1,644,000
Quality Assurance Manager	1,200,000	37.00%	1,644,000

Tabla 49. Costos de salario aproximado de las personas en caso de contratación. Fuente: Elaboración propia.

En la Tabla 50, se muestra las horas requeridas en capacitación para los roles definidos y el costo que representa.

Cantidad de horas	Cantidad personas	Horas inducción	Horas por costo	Costo total
Desarrollador junior	1	20	6,422	128,875
Desarrollador senior	1	20	7,706	154,125
Director de proyectos	1	20	10,275	205,500
Quality Assurance Manager	1	20	10,275	205,500
Total				693,563

Tabla 50. Costos por horas de capacitación del personal contratado. Fuente: Elaboración propia.

En esta sección solo se consideran los costos de inclusión de nuevo personal, si se toma esta alternativa se tiene que sumar las horas y costo de capacitación de los desarrolladores que se pueden visualizar en la siguiente sección

5.3.3.2. Costo de capacitación del personal actual

En este apartado se muestran los costos aproximados correspondiente a la capacitación del personal de los equipos de desarrollo actuales.

En la Tabla 51, se muestra los costos salariales aproximados de los roles.

Perfil	Salario estimado	Cargas sociales	Total
Desarrollador junior	750,000	37.00%	1,027,500
Desarrollador senior	900,000	37.00%	1,233,000
Director de proyectos	1,200,000	37.00%	1,644,000
Quality Assurance Manager	1,200,000	37.00%	1,644,000

Tabla 51. Costos de salarios aproximados del personal actual. Fuente: Elaboración propia.

En la Tabla 52, se muestra las horas requeridas en capacitación para los roles definidos y el costo que representa.

Cantidad de horas	Cantidad personas	Horas capacitación	Horas por costo	Costo total
Desarrollador junior	6	15	6,422	577,969
Desarrollador senior	3	15	7,706	346,781
Director de proyectos	1	15	10,275	154,125
Quality Assurance Manager	1	15	10,275	154,125
Total				1,233,000

Tabla 52. Costos por horas de capacitación del personal actual. Fuente: Elaboración propia.

5.3.4. Validación de la propuesta.

Las propuestas en cuanto a los procesos de gestión de requerimientos y verificación y validación han sido revisadas con los principales coordinadores de desarrollo y con algunos desarrolladores. Principalmente para medir la pertinencia de los elementos propuestos a la situación actual y agilidad de implementación al proceso de desarrollo.

Debido a la situación actual, de la pandemia, esta se ha hecho con presentaciones, vía Zoom, y con encuestas de Google. Para ver la respuesta de los involucrados, respecto a la aceptación de los elementos propuestos, revisar los Apéndices H e I.

5.4. Análisis de herramientas de QA

En esta sección se realiza una comparación entre herramientas que puedan soportar los procesos de aseguramiento y control de calidad de software, principalmente en verificación y validación, en Inclutec.

5.4.1. Criterios de análisis

En esta sección se describen los criterios para hacer la comparación de herramientas enfocadas en aseguramiento de calidad.

Antes de describir los criterios, se describen las herramientas de software que utilizan actualmente para llevar el control del desarrollo de software durante todo el ciclo de vida:

- Scrum desk: Herramientas para el llevar el control de los requerimientos en desarrollo, tienen un enfoque ágil (Scrum), por lo tanto, involucra la teoría de *sprint*, *backlog*, *user stories* y *tasks*.
- Gitlab: Es una herramienta que se enfocada en llevar el control de versiones del software en desarrollo, implementa la integración continua (CI) y tiene funcionalidades de seguimiento de *bug/issues*.

Tomando en cuenta la existencia y uso de las herramientas anteriores y los procesos propuestos, se definen los criterios de funcionalidad que se utilizará para el análisis de las herramientas, estos se describen a continuación:

- Proyecto: Permitir crear proyectos para llevar el control por separado.
- Requerimiento: Proveer de algún módulo o apartado donde se puedan enlazar casos de prueba a los requerimientos.
- Casos de prueba: Crear casos de prueba con requisitos y casos de éxito.
- Enfoque ágil: Cuenten con alguna forma de organizar las pruebas que se ajuste al proceso de desarrollo actual.
- Ejecución de pruebas: Permitir el cambio de estado y resultado de los casos de prueba.

- Seguimiento de resultados: Dar seguimiento a los resultados de pruebas que se hayan ejecutado.
- Roles: Asignación de roles en la ejecución de pruebas.
- Reportes/ Estadísticas: Poder visualizar en forma de reportes y gráficos los resultados integrados de las pruebas.
- *Bug/Issue tracker*: Que se pueda integrar con la herramienta GitLab para llevar el control de los *issues* o *bugs*, que se encuentren en las pruebas.

En la Tabla 53, se clasifican los criterios de funcionalidad bajo el marco axiológico definido por Naranjo (2020) en un enfoque alternativo de investigación, que indica para tomar una decisión basada en criterios de selección debe ser pesados según el nivel de importancia.

Criterios	Valor único
Proyectos	15%
Requerimientos	5%
Casos de prueba	15%
Enfoque ágil	5%
Ejecución de pruebas	15%
Seguimiento de resultados	15%
Roles	10%
Reporte/Estadísticas	15%
Bug Issue tracker (Integración Gitlab)	5%

Tabla 53. Pesos de los criterios de funcionalidad para el benchmarking. Fuente: Elaboración propia.

Además, se definen criterios técnicos y de precios de las herramientas, estas se describen a continuación:

- Precio: Es el costo en término monetario para la adquisición de la herramienta.
- Prueba: La posibilidad de utilizar la herramienta en una versión de prueba gratis.
- Instalación: Tomar en cuenta si es un *Software as a Service* (SaaS) o se necesita un instalar en un servidor propio.
- Capacitación: Son aquellos materiales sobre el software, que estén disponibles para aprender a utilizar la herramienta.
- Soporte: Son los servicios que el proveedor ofrece de soporte a la herramienta.

En la Tabla 54, se clasifican los criterios técnicos y de precio bajo el marco axiológico definido por Naranjo (2020) en un enfoque alternativo de investigación.

Criterios	Valor	Opciones
Precio ³	20%	Gratis: 20% \$1-\$25: 10% \$25 o más:5%
Prueba gratuita	10%	Valor único
Instalación	20%	Nube: 20% Servidor propio: 15%
Capacitación	15%	Una fuente:5% Dos fuentes:10% Tres o más fuentes 15%
Soporte	15%	Valor único
Accesibilidad ⁴	20%	Valor único

Tabla 54. Peso de los criterios técnicos y de precio para el benchmarking. Fuente: Elaboración propia.

³ Para el cálculo del puntaje se hace por el precio por usuario al mes.

⁴ Para esto se toma en cuenta si hay referencia de accesibilidad con lectores de pantalla.

5.4.2. Herramientas seleccionadas

En esta sección se describen las herramientas seleccionadas por sus características y los criterios definidos (Ver Sección 5.4.1)

Para seleccionar las herramientas, se hace una revisión Gartner Inc y Forrester, sin embargo, los proveedores que están en el “cuadrante mágico” (Gartner) y en las “olas” (Forrester) son herramientas enfocadas en equipos grandes y con procesos maduros. Por lo tanto, se replantea la búsqueda en Google, el sitio Capterra y un reporte de EasyQA sobre herramientas de gestión de pruebas y ciclo de vida de aplicaciones de software.

Las herramientas seleccionadas se encuentran en la Tabla 55.

Herramienta	Sitio
Testrail	https://www.gurock.com/testrail/
Testlink	http://www.testlink.org/
PractiTest	https://testcollab.com/
TestCollab	https://testcollab.com/
Test lodge	https://www.testlodge.com/
QACoverage	https://www.qacoverage.com/
TestCaseLab	https://testcaselab.com/
EasyQA	https://geteasyqa.com/

Tabla 55. Herramientas consultadas. Fuente: Elaboración propia.

Se hace una revisión de la documentación de las herramientas para conformar una tabla comparativa, con el motivo de analizar los criterios establecidos e identificar cuál de las herramientas se ajusta más a las condiciones y necesidades de la organización.

En la Tabla 56, se hace la comparación de las herramientas a partir de los criterios de funcionalidad definidos.

Criterios / Herramientas	Test Rail	Testlink	Practi Test	TestCollab	TestLogdge	QA Coverage	TestCaseLab	EasyQA
Proyectos	X	✓	✓	✓	✓	✓	✓	✓
Requerimientos	X	✓	✓	X	✓	✓	X	✓
Casos de prueba	✓	✓	✓	✓	✓	✓	✓	✓
Enfoque ágil	X	X	✓	X	X	✓	X	✓
Ejecución de pruebas	✓	✓	✓	✓	✓	✓	✓	✓
Seguimiento de resultados	✓	✓	✓	✓	✓	✓	✓	✓
Roles	✓	✓	✓	✓	✓	✓	✓	✓
Reporte/Estadísticas	✓	✓	✓	✓	✓	✓	X	✓
Bug Issue tracker (Integración Gitlab)	✓	X	✓	✓	X	X	X	✓
Puntos obtenidos	75%	90%	100%	90%	90%	95%	70%	100%

Tabla 56. Comparación bajo los criterios de funcionalidad. Fuente: Elaboración propia.

En la Tabla 57, se indican los precios y aspectos técnicos de las herramientas seleccionadas.

Criterios / Herramientas	Precio	Prueba gratis	Instalación	Capacitación	Soporte
TestRail	Nube: \$32/mes/usuario Instalado: 334/ año/ usuario	Sí 30 días	Cloud y server propio	<i>Webinars</i> y documentación	Online en el sitio y Comunidad
TestLink	No tiene	N/A	Server propio	Documentación	Foro
PractiTest	\$147/mes/3 <i>tester</i>	Sí 14 días	Cloud	<i>Webinars</i> y documentación	Foro y online en el sitio
TestCollab	\$0 /200 casos /400 casos ejecutados/ 3 usuarios/ mes \$25/ usuario/ mes \$10/ 50+ usuarios/mes	Sí 14 días	Cloud y Docker	Tutorial en el sitio	Blog

Criterios / Herramientas	Precio	Prueba gratis	Instalación	Capacitación	Soporte
TestLodge	\$49/ 150 planes de prueba/ 600 casos de prueba/ 300 ejecuciones de prueba/ mes \$99/ 500 planes de prueba/ 3000 casos de prueba/ 1000 ejecuciones de prueba/ mes	Sí 30 días	Cloud	Tutorial en el sitio	Blog y online en el sitio
QACoverage	Cloud Administrador del ciclo de vida: \$19/ usuario/ mes Administrador de casos de prueba: \$9/ usuario/ mes Instalado Administrador del ciclo de vida: \$119/ usuario/año Administrador de casos de prueba: \$59/ usuario/ año	Sí 30 días	Cloud y server propio	No se encontró referencia	Contacto vía correo electrónico

Criterios / Herramientas	Precio	Prueba gratis	Instalación	Capacitación	Soporte
TestCaseLab	<i>Basic:</i> \$48/ 500 casos de prueba/ mes <i>Essential:</i> \$99/ 1000 casos de prueba/mes <i>Advanced:</i> \$149/ 3000 casos de prueba/ mes	Sí 30 días	Cloud	<i>Webinars</i>	Online en el sitio, blog y vía correo electrónico
EasyQA	<i>Cloud:</i> \$10/ Usuario/ Mes	Sí	Cloud	<i>Webinars</i>	Online en el sitio, blog y vía correo electrónico

Tabla 57. Comparación bajo precio y criterios técnicos. Fuente: Elaboración propia.

En la Tabla 58, se muestran el puntaje obtenido en los criterios técnicos y de precio definidos.

Herramienta/Criterio	Precio	Prueba gratuita	Instalación	Capacitación	Soporte	Accesibilidad	Total obtenido
Testrail	5%	10%	20%	10%	15%	0%	60%
Testlink	20%	10%	15%	5%	15%	0%	65%
PractiTest	5%	10%	20%	10%	15%	0%	60%
TestCollab	20%	10%	20%	5%	15%	0%	70%
Test lodge	5%	10%	20%	5%	15%	0%	55%
QACoverage	10%	10%	20%	0%	15%	0	55%
TestCaseLab	5%	10%	20%	5%	15%	0%	55%
EasyQA	10%	10%	20%	5	15	0%	65%

Tabla 58. Puntaje asignado a las herramientas según los criterios técnicos y de precio. Fuente: Elaboración propia.

Con los valores establecidos en el marco axiológico, se determina que, en cuanto a las funcionalidades, EasyQA cumple con las características que se necesitan para soportar los procesos propuestos, al obtener el 100% de los puntos. Por otra parte, en cuanto los otros criterios técnicos y de precio está a 5% por debajo de TestCollab, pero se toma la decisión de recomendar easyQA, debido a que TestCollab tiene una versión gratuita, pero en el caso en que el uso exceda esa capa gratis el costo es mayor al de easyQA. TestCollab sería la alternativa inmediata de easyQA en caso que se considere que no se va a exceder la cantidad gratuita o TestLink si se considera que es mejor instalar propio a un costo monetario de 0.

Con eso resultados obtenidos, se determinar que la herramienta a recomendar a Inlutec es EasyQA. Además, se les proporciona las valoraciones de las demás herramientas para que puedan considerar otras según se requiera.

6. Conclusiones

En este capítulo se muestran las conclusiones determinadas a partir del desarrollo de los objetivos planteados.

Acerca de (Caracterizar el proceso de calidad actual utilizado en Inlutec por medio de observación, consulta con los desarrolladores y revisión de documentación para entender su funcionamiento.)

Las conclusiones determinadas para este objetivo se obtuvieron de la observación del proceso y las entrevistas y encuestas aplicadas a los coordinadores y equipo de desarrollo de software de Inlutec, se definen las siguientes:

- En cuanto a requerimientos no hay un proceso estándar para su gestión. A partir del acercamiento y documentación inicial de requerimientos, se inicia el desarrollo, y durante el ciclo de vida del software se desconoce el estado de la documentación y sus cambios.
- Los cambios de requerimientos no se gestionan, esto implica que para cada cambio no se evalúan los riesgos o restricciones que puedan afectar el desarrollo del proyecto y el cumplimiento de los requisitos.
- Los coordinadores están de acuerdo en que una gestión de requerimientos es esencial para entregar un producto de calidad, por lo tanto, indican que un proceso de gestión de requerimientos aportaría positivamente al equipo desarrollo. No importa el proceso de pruebas si los requerimientos no están bien especificados.
- Los coordinadores indican que la metodología de desarrollo actual, por su flexibilidad, permite la incorporación de un proceso de QA. Sin embargo, debe ser corto y objetivo que se enfoque tanto al inicio del proyecto como los cambios que surjan durante el ciclo de vida del proyecto.
- Se realizan actividades de pruebas, sin embargo, no se puede encontrar evidencia concreta de su ejecución. Del mismo modo que en requerimientos, las actividades de pruebas no están estandarizadas ni controladas.
- La carga de trabajo de los desarrolladores le quita prioridad a la implementación de pruebas, siendo el tiempo el principal motivo de reducir esfuerzos en las pruebas. Por lo tanto, se piensa en definir un rol para que pueda gestionar este tipo de procesos.

- A falta de información sobre requerimientos y pruebas, se desconoce la calidad del software desarrollado.
- Se requiere al menos un rol externo al equipo de desarrollo, que se encargue de las actividades de aseguramiento de calidad, para este caso, del control, monitoreo, y ejecución y análisis de pruebas.
- Los encuestados indican que es necesario un proceso para la gestión de verificación y validación (pruebas) que se iterativamente en conjunto con la metodología de desarrollo. Además, que sea ágil en términos de tiempo y flexibilidad según la carga de trabajo.

Acercas de (Analizar la brecha entre el proceso de calidad actual y las recomendaciones de buenas prácticas, estándares y marcos de referencia para determinar las necesidades de calidad evaluando las partes del proceso actual contra las recomendaciones.)

Las conclusiones determinadas para este objetivo se obtuvieron de la revisión documental sobre prácticas de la industria de software en cuanto a calidad y la comparación o análisis de brecha, a partir de la teoría y el proceso caracterizado.

- Existen buenas prácticas y normas por parte de la IEEE, ISO, CMMI, PMI y ISTQB, en cuanto a requerimientos y verificación y validación de software.
- El nivel de madurez en cuanto a aseguramiento de calidad, gestión de requerimientos y verificación y validación es cero. Establecer algún proceso de estos tipos sería el primer acercamiento para definir y estandarizar procesos.
- De los elementos que las prácticas de calidad recomiendan, la organización posee algunos que se necesitan estandarizar y otros que necesitan ser planteados e implementados.
- La estrategia para elaborar la propuesta es ordenar los elementos existentes en la organización e incorporar recomendaciones de las prácticas de calidad, de esta manera establecer los procesos de gestión de requerimientos y verificación y validación.
- La metodología de desarrollo de software no se debe modificar, sino incorporar los procesos propuestos a la forma de trabajo actual.

Acerca de (Elaborar la metodología de aseguramiento y control de calidad para los proyectos de software basado en la brecha entre las recomendaciones de calidad y el proceso actual para ajustarla a las necesidades de Inlutec).

Las conclusiones determinadas para este objetivo se obtuvieron del desarrollo de la propuesta. A partir de la brecha identificada, se proponen los procesos de gestión de requerimientos y verificación y validación.

- La propuesta desarrollada representa el primer acercamiento a establecer formalmente un proceso de aseguramiento de calidad en la organización. Se compone de procesos que se relacionan entre sí, con el objetivo de garantizar la calidad de los productos de software de Inlutec.
- La metodología propuesta complementa el proceso de desarrollo de software actual, desde previo al inicio del desarrollo hasta las pruebas periódicas de verificación y validación.
- Los procesos, roles y artefactos que conforman la metodología, están bajo las recomendaciones de las buenas prácticas y normas sobre ingeniería de software.
- La metodología está diseñada a alto nivel para que sirva como el punto de inicio para trabajar en la mejora continua de calidad en la organización. Tomando como base la información que se obtenga de la ejecución, se puedan tomar decisiones para mejorar en las actividades de desarrollo de software.

Acerca de (Recomendar herramientas de software apropiadas que se ajusten a la metodología de aseguramiento y control de calidad propuesta, esto por medio de un benchmarking para que puedan ser utilizadas por el equipo de desarrollo de Inlutec).

Las conclusiones determinadas para este objetivo se obtuvieron del desarrollo de la propuesta. A partir de la brecha identificada, se proponen los procesos de gestión de requerimientos y verificación y validación.

- Los equipos de desarrollo utilizan algunas herramientas de software como *ScrumDesk* u hojas de cálculo para llevar el control del desarrollo de software.

- En la etapa de pruebas principalmente, no se tiene información centralizada de las ejecuciones. Esto impide las acciones de mejora continua basada en información y se desconoce el estado o progreso real del software.
- Los criterios para la evaluación y recomendación de herramientas se basaron en funcionalidad y precio.
- Los criterios de funcionalidad considerados involucran principalmente el proceso de pruebas de verificación y validación. En menor medida se consideran los requerimientos, esto porque, junto a la documentación de requerimientos escritos, los desarrolladores han adoptado *ScrumDesk* para el manejo de las historias de usuario.

7. Recomendaciones

A partir de las conclusiones descritas en el Capítulo 6, se describen las recomendaciones tomando como base todo el proceso de desarrollo del proyecto.

- Se parte del SWEBOK para definir un proceso básico de gestión de requerimientos con sus respectivos roles y artefactos, este proceso involucra requerimientos iniciales, validaciones periódica y cambio de requerimientos. Se debe seguir este proceso para dar trazabilidad y cobertura a los requerimientos durante el desarrollo de proyecto.
- La gestión de requerimientos debe ajustarse al proceso propuesto siguiendo las tareas y artefactos asociados, para que, al momento de aplicar el proceso de pruebas, se puedan validar correctamente.
- Los procesos propuestos están estructurados y descritos a alto nivel, para que sea aplicable a cualquier tipo de proyecto de software de la organización. Las tareas se centran en controlar y generar información útil para el proyecto, sobre requerimientos y pruebas. Se recomienda tomar en cuenta estos procesos para abordar los nuevos proyectos.
- Se incorpora un rol de Administrador de Aseguramiento de Calidad (QAM), esta persona se debe encargar de validar el cumplimiento de los procesos establecidos, así como cumplir con las responsabilidades que tenga asignadas en cada proceso propuesto.
- Tanto los procesos de gestión de requerimientos y verificación y validación no son definidos en una sola etapa del desarrollo en específico, sino que se deben realizar de manera iterativa conforme se desarrolla el proyecto.

- Los procesos están diseñados para la fácil adopción incorporando elementos existentes con nuevas prácticas, de manera que, se puedan empezar a utilizar desde los proyectos actuales.
- Si se quiere trabajar en la mejora continua de las actividades y procesos basada en información, se debe implementar los procesos y verificar el cumplimiento de estos, con sus respectivas salidas.
- Se hizo una indagación de herramientas para soportar los procesos propuestos y agilizar el trabajo de aseguramiento de calidad. De esas herramientas se determina que EasyQA cubre todas las necesidades de la organización, por lo tanto, se recomienda analizar esa opción.
- Utilizar una herramienta de software, que soporte principalmente la verificación y validación de software, permite recopilar información y estadísticas para medir la calidad del software y tomar decisiones en el proceso de desarrollo de los proyectos.

8. Bibliografía

- Aguilar, A. (2018). *Establecimiento de la metodología para aseguramiento de la calidad en ArcSight QA Costa Rica* (Tesis de maestría). Tecnológico de Costa Rica, Cartago, Costa Rica.
- Axelos (2011). ITIL® glossary and abbreviations English.
- Baimyrzaeva, M. (2018). *Beginners' Guide for Applied Research Process: What Is It, and Why and How to Do It?* University of Central Asia.
- Barrantes, R. (2002). *Investigación un camino al conocimiento un enfoque cualitativo y cuantitativo*. San José: EUNED.
- Bourque, P. y Fairley, R. E. (2014). *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0*. IEEE Computer Society Press.
- Carballo, R. F. (2001). La entrevista en la Investigación cualitativa. *Pensamiento actual*, 2(3).
- Casas J., Repullo J.R., Donado J. (2003). La encuesta como técnica de investigación. Elaboración de cuestionarios y tratamiento estadístico de los datos. *Atem Primaria. Volumen* (31), 527-38.
- CMMI Institute. (2010). CMMI® para Desarrollo, Versión 1.3. Software Engineering Institute.
- Desarrollo, C. (2010). CMMI® para desarrollo, versión 1.3. C. Para Desarrollo, Version 1.3, 23.
- DTI (Department of Trade and Industry). (2019). *Standards and Standardization*. Philippine:PMI. Recuperado de <http://www.bps.dti.gov.ph/index.php/standards>.
- Eira, P., Guimarães, P., Melo, M., Brito, M. A., Silva, A. y Machado, R. J. (2018). Tailoring ISO/IEC/IEEE 29119-3 Standard for Small and Medium-Sized Enterprises. In *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 380-389). IEEE.
- Eira, P., Guimarães, P., Melo, M., Brito, M. A., Silva, A., y Machado, R. J. (2018, April). Tailoring ISO/IEC/IEEE 29119-3 Standard for Small and Medium-Sized Enterprises. In *2018 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)* (pp. 380-389). IEEE

- Elliott, R. y Timolak, L. (2005). Descriptive and interpretive approaches to qualitative research. A handbook of research methods for clinical and health psychology, 1(7), 147-159.
- FAO (Food and Agriculture Organization of the United Nations). (2014). *Good practices template*. FAO, pp.1.
- FAO (Food and Agriculture Organization of the United Nations). (2014). *Good practices template*. FAO.
- Galín, D. (2004). *Software quality assurance: from theory to implementation*. Pearson Education India.
- Graham, D., Veenendaal E., V., Evans, I. y Black, R. (2008). *Foundations of software testing: ISTQB certification*.
- HERNÁNDEZ, R., Fernández, C., & Baptista, P. (2014). *Metodología de la investigación*, Sexta Edición, Editorial McGraw-Hill. Pág. 10, 115.
- Hox J., J. y Boeije H., R. (2005). *Data collection, primary versus secondary*.
- Institute of Electrical and Electronics Engineers (2009). *IEEE Standard for a Software Quality Metrics Methodology* (IEEE Std 1061-1998 (R2009)).
- Institute of Electrical and Electronics Engineers (2014). *IEEE Standard for Software Quality Assurance Processes* (IEEE Std 730-2014).
- Institute of Electrical and Electronics Engineers (2016). *IEEE Standard for System, Software, and Hardware Verification and Validation* (IEEE Std 1012-2016).
- International Organization for Standardization. (2003). *Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting practices manual* (ISO 20926).
- International Organization for Standardization. (2005). *Sistemas de gestión de calidad – Fundamentos y vocabulario* (ISO 9000:2005).
- International Organization for Standardization. (2010). *System and software engineering - Vocabulary* (ISO 24765).

- International Organization for Standardization. (2011). *System and software engineering – System and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models* (ISO/IEC 25010-1).
- International Organization for Standardization. (2013a). *Software and system engineering – Software Testing – Part 2: Test Process* (ISO/IEC/IEEE 29119-2).
- International Organization for Standardization. (2013b). *Software and system engineering – Software Testing – Part 3: Test Documentation* (ISO/IEC/IEEE 29119-3).
- International Organization for Standardization. (2015). *Sistema de Gestión de la Calidad – Requisitos* (ISO 9001).
- ISO 9000. ISO Central Secretariat, Ginebra, Suiza, 2005.
- Kerzner, H. (2002). Strategic planning for project management using a project management maturity model. John Wiley & Sons.
- Kerzner, H. (2017). Project management: a systems approach to planning, scheduling, and controlling. John Wiley & Sons.
- Lauesen, S. y Younessi, H. (1998). Six Styles for Usability Requirements. In *REFSQ* (Vol. 98, pp. 155-166).
- Livermore, J. A. (2007, March). Factors that impact implementing an agile software development methodology. In *Proceedings 2007 IEEE SoutheastCon* (pp. 82-86). IEEE.
- Malan, D. (2020). Top 60+ Software Development Companies in 2020. Techreviewer. Recuperado de <https://techreviewer.co/top-software-development-companies/>.
- Manawadu, C., Johar, M. G. M. y Perera, S. J. E. C. (2013). An Evaluation of Software Development Methodology Adoption by Software Developer in Sri Lanka. *Editorial Committees*, 84.
- Methodology. (2019). *En Merriam-Webster diccionario en línea* (Onceava Edición). Recuperado de <https://www.merriam-webster.com/dictionary/methodology>.
- Metodología. (2019). En *Diccionario de la Lengua Española en línea*. Recuperado de <https://dle.rae.es/metodolog%C3%ADa>

- Methodology. (2019). En Merriam-Webster Online Dictionary. Recuperado de <https://www.merriam-webster.com/dictionary/methodology>
- Meulen, R. (2017). Analytics and Lean/Agile Practices Are Revolutionizing Application Development. Gartner. Recuperado de <https://www.gartner.com/smarterwithgartner/prepare-for-major-shifts-in-enterprise-software-development/>
- Naranjo, L. (2019, noviembre 13). *Entrevista investigador Includtec* [Entrevista personal].
- Naranjo, L.C (2020). Investigación en informática: el enfoque alternativo. *Technology Inside*. Volumen (5), 4-14
- Object Management Group (OMG) (2011). Business Process Model and Notation (BPMN). Recuperado de <https://www.omg.org/spec/BPMN/2.0/PDF/>.
- Ocampo, M. (2017). *Propuesta de una metodología de control de calidad para los proyectos de automatización, basado en las mejores prácticas de ISTQB, caso SOIN S.A* (Proyecto de licenciatura). Tecnológico de Costa Rica, Cartago, Costa Rica.
- Pressman, R. S. (2010). *Software Engineering: a practitioner's approach* McGraw-Hill, New York, 68.
- Project Management Institute (PMI) (2013). Guide to the project management body of knowledge (PMBOK guide). In Project Management Institute (Vol. 5).
- Requena, A. (2018). ¿Qué es un sprint de scrum? *OpenWebinars*. Recuperado de: <https://openwebinars.net/blog/que-es-un-sprint-scrum/>
- Romero, V.;Muir, H.;Rodríguez, J.;Gómez, A., y Chacon, M. (2019). *Avances en tecnologías inclusivas aplicadas en educación en Iberoamérica*, Ciudad de Mexico, Mexico:CONAIC.
- Roshdy, Tarek. (2018) *What is testing?* [Archivo de video]. Recuperado de <https://www.udemy.com/course/istqb-foundation-level-training/learn/lecture/13238830#overview>

- Sánchez, M. (2017). *Diseño de sistema para el aseguramiento y el control de la calidad en los proyectos de software del TEC Digital* (Proyecto de licenciatura). Tecnológico de Costa Rica, Cartago, Costa Rica.
- Scrum Alliance. (2017). The estate of scrum. Recuperado de https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/StateOfScrum_2016_FINAL.pdf
- Scrum Alliance. (2018). The estate of scrum. Recuperado de [https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2017-SoSR-Final-Version-\(Pages\).pdf](https://www.scrumalliance.org/ScrumRedesignDEVSite/media/ScrumAllianceMedia/Files%20and%20PDFs/State%20of%20Scrum/2017-SoSR-Final-Version-(Pages).pdf)
- Sivaranjani. J., Rajeswari. R. (2017). A Study on Software Development Methodologies. *International Journal of Innovative Research in Computer and Communication Engineering*. Volumen (5), 7727-7737.
- Steward, M. (2018). *Conservation Capability Maturity Model. A tool for assessing and improving performance of conservation organizations*.
- Sutherland, J., & Schwaber, K. (2013). The scrum guide. The definitive guide to scrum: The rules of the game. Scrum. org, 268.
- Sutherland, J., & Schwaber, K. (2017). The scrum guide. The definitive guide to scrum: The rules of the game. Scrum. org.
- Tecnológico de Costa Rica. (2019). *Qué es el tec*. Cartago, Costa Rica: TEC. Recuperado de <https://www.tec.ac.cr/que-es-tec>.
- W3C (2019). Accessibility Conformance Testing (ACT) Rules Format 1.0. W3C. Recuperado de <https://www.w3.org/TR/act-rules-format/>.
- Wands, B. (2007). *Art of the digital age*. Thames & Hudson.
- Zima, D. (2015). Modern Methods. *Modern methods of software development*. Task Quarterly. Volumen (19), 481-493.

9. Anexos

Anexo I: Plantilla de minutas.

Plantilla de minuta establecida por Inlutec para la recopilación de resultados de asuntos y acuerdos de las reuniones efectuadas.

Nombre de proyecto
Minuta de reunión

Minuta "XX"	Fecha de minuta: yyyy/mm/dd
-------------	-----------------------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
yyyy/mm/dd	:	:	

Agenda
1.

2. Datos de persona promotora de reunión

Organización	Nombre de Promotor(a)	Correo / Teléfono

3. Participantes

Nombre	Institución/Correo/Teléfono	Iniciales

4. Puntos tratados

<input type="checkbox"/>

5. Acuerdos / Consultas

ID	Descripción	Responsables	Fecha (cumplimiento)
01			

6. Puntos pendientes

<input type="checkbox"/>

7. Firmas de conformidad

Participantes	Firma

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Anexo II: Plantilla de gestión de cambios

Plantilla establecida por Inlutec para la solicitud de cambios en un proyecto determinado.

Solicitud de Cambio Numero X

Proyecto	Nombre del proyecto	Fecha	[yyyy/mm/dd]
Project Manager	Nombre del project manager	Organización	Inlutec
Solicitante de Cambio	Nombre de solicitante	Cliente	
Origen de cambio	Causa u origen de cambio	Tipo de cambio	
Descripción de la solicitud de cambio		Justificación de la solicitud de cambio	
El siguiente cambio consisten en		Por razones	
Impactos			
Impacto en cronograma: ...		Impacto en costos: ...	
Impacto en alcance: ...		Impacto en ...	
Implicaciones			
En recursos: En Interesados: Documentación: Riesgos:			
Aprobación			
Encargado	Rol	Firma	Aceptación
Encargado 1	Rol 1	Firma	Sí/No

Tipos de cambio: Alcance Costo Tiempo Calidad Recursos Procesos Documentación otro	Causa u origen de cambio: Solicitud del cliente Reparación de defecto Acción correctiva Acción preventiva Modificación de documentos Otros
---	---

Anexo III: Plantillas de requerimientos actuales.

Plantilla 1. Requerimientos funcionales del EAW.

Es la forma de documentar los requerimientos, según la documentación revisada.

Lista de requerimientos funcionales

N°	Nombre de requerimiento	Tipo de reque.	Prioridad
001	Obtener la página web y los subpáginas deseadas mediante Web scraping.	Funcional	Alto
002	Obtener archivo html de las páginas web encontradas mediante el web scraping.	Funcional	Alto
003	Identificar-agrupar elementos o componentes de un archivo html.	Funcional	Alto
004	Evaluación de criterios de nivel A.	Funcional	Alto
005	Evaluación de criterios de nivel AA.	Funcional	Alto
006	Listar links para el nombre de cada página web y ver un cumplimiento de criterios más específico.	Funcional	Medio
006.5	Ingresar a criterios de una página de la lista de páginas analizadas.	Funcional	Medio

Especificación de requerimientos funcionales

001. Obtener la página web y los subpáginas deseadas mediante Web scraping.

Actores

Usuario, Sistema.

Detalle del requerimiento

Mediante la técnica de web scraping se podrá ingresar la url de un sitio web que se desea evaluar para poder obtener todas las páginas anexas a este sitio y guardarlas de forma local en el sitio para poder analizarlas y descargarlas después. Además de también obtener el alias de cada página.

Paso 1. El usuario ingresa una url.

Paso 2. El sistema hace scraping a esa url.

Paso 3. El sistema obtiene todas las páginas de esa url.

Paso 4. El sistema obtiene todos los alias de las páginas.

Paso 5. Las páginas se almacenarán localmente en el servidor.

Restricciones

1. El usuario tiene que ingresar una url válida.

Formatos de pantalla y reportes

NA

Plantilla 2. Requerimientos funcionales de Euler.

Es la forma de documentar los requerimientos funcionales, según la documentación revisada.

Número de requerimiento	RF-001
Nombre de requisito	Crear, editar, actualizar y eliminar expresiones matemáticas accesibles
Descripción	<ul style="list-style-type: none">• Hacer lectura de expresiones matemáticas según las posibilidades perceptuales de los usuarios, debe existir lectura en voz y lectura de forma visual.• Se debe contar con símbolos agrupados por categorías para insertar y construir una expresión.• Las categorías son: operaciones elementales, simbología básica, fracciones, paréntesis, raíces, geometría, rectas, flechas, teoría de conjuntos, trigonometría, alfabetos griego en minúscula y mayúscula, funciones y sistema de ecuaciones.• Los símbolos deben ser los soportados por MathML 3.0.
Prioridad del requerimiento	Alta.

Plantilla 3. Requerimientos no funcionales del EAW.

Es la forma de documentar los requerimientos no funcionales, según la documentación revisada.

Lista de requerimientos no funcionales

N°	Nombre de requerimiento	Tipo de reque.	Prioridad
0001	Diseño responsivo	No Funcional	Alto
0002	Mensajes de error informativos		Alto
0003	Debe poseer un menú de configuración visual		Alto
0004	Buen contraste visual		Alto
0005	Cumplir con los criterios de usabilidad descritos en la WCAG 2.1		Alto

Anexo IV. Plantilla para los casos de pruebas manuales e integración.

Casos de prueba manuales para el <<nombre de Sistema>>

Historia		Escenario		Pre- condiciones	Pasos de Prueba	Datos de Prueba	Resultado Esperado	Fecha y hora de realización de la prueba	Resultado real de la prueba		N° en el issue tracker	Estado
ID Historia	Descripción	ID Escenario	Descripción						Descripción	Tipo		

10. Apéndices

Apéndice A. Instrumentos para la investigación

Instrumentos1. Plantilla para observación.

Plantilla del observador

Nombre: _____

Fecha: DD/MM/YY

Proceso	Observaciones
Proceso 1	Observaciones del proceso 1
Proceso 2	Observaciones del proceso 2

Instrumentos 2. Guía entrevista de requerimientos

Guía de entrevista: Requerimientos.

Entrevistador / rol	
Entrevistado / rol	
Fecha	

1. Cuando se inicia el desarrollo de un proyecto ¿Existen requerimientos documentados o registrado para el inicio de este?
2. ¿Cómo se hace el levantamiento de requerimientos para un proyecto de software?
3. ¿Existe documentación de requerimiento de los proyectos desarrollados o que están en desarrollo?
4. ¿Cuáles son los problemas más comunes en el proceso de convertir los requerimientos en producto?
5. ¿Cómo se manejan los cambios en los requerimientos durante el desarrollo del proyecto?
6. ¿Han tenido que lidiar con requerimientos ambiguos? ¿Cómo los gestionan?

Instrumentos 3. Guía entrevista de calidad.

Guía entrevista pruebas

Entrevistador / rol	
Entrevistado / rol	
Fecha	

1. ¿Sabe usted que es aseguramiento de calidad?
2. ¿Qué tan importante son las pruebas en el desarrollo de proyectos de software en Inclutec?
3. ¿Cada cuánto tiempo realizan las pruebas?
4. Se guían con los requerimientos para la realización de pruebas.
5. ¿Qué tipos de pruebas realizan?
6. ¿Se realizan pruebas unitarias y pruebas de integración?
7. ¿Se realizan pruebas automáticas o todo el proceso es manual?
8. ¿Se registran estadísticas sobre los casos de prueba, fallos, aprobados, resultados o comportamientos inesperados?
9. Desde los inicios del desarrollo de software, ¿han implementado mejoras en la aplicación de pruebas?
10. ¿Cómo funciona el proceso para envío de un componente de software a pruebas?
11. ¿Han tenido problemas que se pudieron evitar aplicando pruebas?
12. ¿Cuánto impacto tiene los requerimientos en la aplicación de las pruebas? Por ejemplo, si los requerimientos no son claros, ¿impacta al momento de hacer verificación del software?
13. ¿Tiene conocimiento si los fallos o resultado inesperados en producción, corresponde a los requerimientos o se debió detectar en pruebas?
14. ¿Cómo se autoriza el *deployment*?

Instrumentos 4. Encuesta sobre aseguramiento de calidad en Inclutec.

Encuesta: Aseguramiento de la calidad

Definición: Propósito del aseguramiento de calidad es buscar especificar actividades y tareas que permite a los proveedores de software producir y recopilar evidencia que sirva de como base de confianza de que el producto de software cumple con los requisitos establecidos.

Responda las preguntas tomando en cuenta la definición anterior.

1. El resultado de la calidad de software que se ha tenido hasta ahora, ¿Cómo la clasifica usted?
 - a. Alta
 - b. Media
 - c. Baja
 - d. Desconocido
2. Actualmente ¿cuáles son las principales complicaciones para garantizar la calidad del software desarrollado?
3. Considera que usted que existen problemas o aspectos que mejorar en la calidad del software desarrollado en Inclutec.
4. Según su experiencia en la organización ¿Cuáles son las complicaciones que usted identifica para implementar un proceso aseguramiento de la calidad?
5. Si usted considera que un proceso aseguramiento de calidad es útil en el proceso de desarrollo de software ¿Por qué sería de beneficio?
6. Considera usted que la metodología de desarrollo que utilizan actualmente se le puede adjuntar procesos de aseguramiento de calidad.
7. ¿Qué características debe tener un proceso de aseguramiento de calidad que se ajuste a las características de Inclutec?

Instrumentos 5. Encuesta sobre la gestión de requerimientos en Inclutec.

Encuesta: Gestión de requerimientos

Definición: La gestión de requerimiento, en general, consta de etapas como obtención, análisis, especificación y validación de requerimientos.

Nota: llámese documentación de requerimientos no solamente a un documento escrito sobre los requerimientos sino todo aquello donde se registren los requerimientos de un software (por ejemplo, Kanban, formularios de historias de usuarios, hojas de cálculo, entre otros.)

Responda las preguntas tomando en cuenta la definición anterior.

1. Si tuviera que evaluar grado de calidad de la especificación de requerimientos ¿Cómo la clasifica usted?
 - a. Alta
 - b. Media
 - c. Baja
 - d. Desconocido
2. Actualmente ¿cuáles son las principales complicaciones para realizar la gestión de requerimientos?
3. Considera que usted que existen problemas o aspectos que mejorar en la gestión de requerimientos.
4. Según su experiencia en la organización ¿Cuáles son las complicaciones que usted identifica para implementar un proceso de gestión de requerimientos?
5. Si usted considera que un proceso de gestión de requerimientos es útil en el proceso de desarrollo de software ¿Por qué sería de beneficio?
6. Considera usted que la metodología de desarrollo que utilizan actualmente se le puede adjuntar proceso de gestión de requerimientos.
7. ¿Qué características debe tener un proceso de gestión de requerimientos que se ajuste a las características de Inclutec?

Instrumentos 6. Plantilla de revisión documentar interna.

Plantilla para revisión documental interna

- a. Nombre del proyecto.
- b. Descripción: En qué consisten el proyecto.
- c. Contenido de la documentación en requerimientos: Se describe el contenido en contrato en la documentación de proyecto relacionado a requerimientos y gestión de requerimientos.

Requerimientos funcionales	Requerimientos no funcionales
Características de los requerimientos funcionales	Características de los requerimientos no funcionales

- d. Contenido de la documentación en calidad: Se describe el contenido en contrato en la documentación del proyecto relacionados a prácticas de calidad.

Proceso de calidad	Resultados del proceso
Características de proceso	Características del resultado del proceso de calidad

Instrumentos 7. Planilla de revisión documental de normas y buenas prácticas.

Plantilla para la revisión documental

Norma o buena práctica	Descripción	Relevancia para el proyecto
Norma 1		
Norma 2		

Instrumentos 8. Planilla para el análisis de brecha.

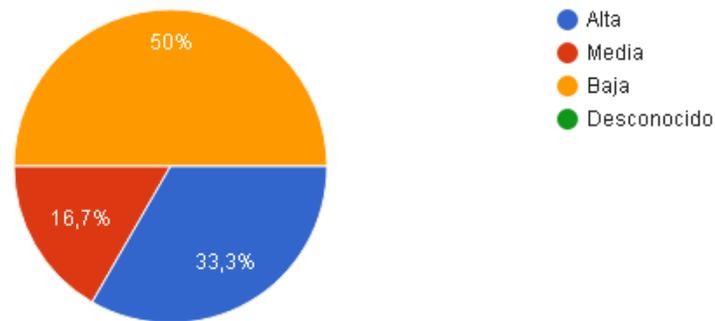
Proceso	¿Dónde estamos? ¿Qué tenemos? Situación y prácticas actuales	¿Qué se debería tener? Recomendaciones de las normas o buenas prácticas
Proceso 1		
Proceso 2		
Proceso n		

Apéndice B. Respuestas de instrumentos.

Respuesta 1: Encuesta sobre requerimientos en Inlutec.

Si tuviera que evaluar grado de calidad de la especificación de requerimientos ¿Cómo la clasifica usted?

6 respuestas



Actualmente ¿cuales son las principales complicaciones para realizar la gestión de requerimientos?

6 respuestas

Falta de rigurosidad por parte del cliente externo en la toma de detalles en la presentación de la información (análisis del negocio), que obliga a hacer cambios considerables en etapas de liberación y capacitación

Para los proyectos de investigación son difíciles de definir

Que no te expliquen bien que es lo que el cliente quiere, o que solo te diga a medias lo que quiere y en el proceso de desarrollo los requerimientos cambien constantemente.

Disponibilidad del cliente, aterrizar un requerimiento del cliente en una funcionalidad y que sea lo que el cliente espera. Manejo de cambios en los sistemas

Requerimientos ambiguos
Falta de documentación
Existen requerimientos que no están que hacen que otros requerimientos fallen

Asegurar el mismo entendimiento entre el cliente y quien toma los requerimientos.

Considera que usted que existen problemas o aspectos que mejorar en la gestión de requerimientos.

6 respuestas

Sí, es una experiencia que siempre requiere de reconocer un nuevo problema en nuevos contextos, por lo que se deben tomar diferentes enfoques base según la necesidad

Sí

Sí, se podría invertir más tiempo en esto o intentar tener una toma de requerimientos bastante buena al inicio de cualquier proyecto para no tener que después cambiarlos o en menor medida.

La manera de tomar o validar requerimientos. Si el instrumento para hacer la gestión de requerimientos es el indicado. Manejo de cambios en el sistema

Surgen ideas que implican cambios en los requerimientos que no se documentan
Algunos requerimientos quedan como un "nice to have" verbales

Usualmente no se presentan pre diseños, aspectos gráficos al cliente a medida de que confirme lo que necesita. Muchas veces se toman documentos sólo escritos, y no es malo, pero algo escrito queda a la interpretación de cada persona y algunas veces varía.

Según su experiencia en la organización ¿Cuáles son las complicaciones que usted identifica para implementar un proceso de gestión de requerimientos?

6 respuestas

La necesidad de un repositorio formal de requerimientos para todos los recursos generados durante un proyecto (dónde encontrar cartas de entendimiento, copias de comunicaciones con clientes externos, minutas, blueprints, diagramas de programación, manuales)

No existe el tiempo debido para hacer una buena ingeniería de requerimientos, la naturaleza de los proyectos

No saber realizar las preguntas adecuadas, no tener tan claro el problema o la necesidad a resolver, o tener algo muy ambiguo.

Herramientas o instrumentos para la gestión de requerimientos. Seguimiento del cumplimiento de los requerimientos. Documentación de cambios y minutas. Manejo de cambios en el sistema

Requerimientos emergentes que se deben satisfacer de inmediato, es decir, no se le da importancia a la documentación de un nuevo requerimiento, es más importante el desarrollo para satisfacerlo

Abarcar lo más posible el problema real, imaginar qué funcionamientos se van a usar realmente o que no falte ninguna característica.

Si usted considera que un procesos de gestión de requerimientos es útil en el proceso de desarrollo de software ¿Por qué sería de beneficio?

6 respuestas

Es necesario porque mantiene sintonizados al grupo de desarrolladores respecto a lo que el cliente espera que el software resuelva, dirigiendo los esfuerzos y tiempos hacia las características que sean más útiles primero y así dar la posibilidad de validarlas con el cliente durante el proceso de desarrollo y minimizar el número y tamaño de los cambios cuando el software entre en etapa de liberación, lo que también puede implicar ahorrar tiempo de desarrollo para dedicarlo a otras tareas del ciclo del software.

Producto de calidad que se ajusta a la necesidad del usuario

Sería de súper beneficio, acortaría los tiempos de desarrollo, evitaria tener que estar rehaciendo trabajo que ya se había hecho pero que no era necesario o era de otra forma, tener más control de lo que se desea desarrollar y tanto el lado del cliente como la de los desarrolladores, diseñadores o demás personas del equipo involucradas en el proyecto se sentirán mejor si se sabe desde el inicio que los requerimientos están bien.

El seguimiento de requerimientos especificado en la toma inicial vs el implementado. Ayudaría a hacer específicamente las funcionalidades que el cliente pidió, se tendría un panorama de todo el proyecto el realizado y el que falta

Sí, pues mantienen el enfo que del proyecto

Intenta evitar un alto porcentaje de cambios en el sistema, pero lo más importante es que es un reflejo de la entrega final.

Considera usted que la metodología de desarrollo que utilizan actualmente se le puede adjuntar proceso de gestión de requerimientos.

6 respuestas

Absolutamente

Sí

Sí, sin ningún problema.

Si, un planeamiento inicial con todas las historias de usuario que reflejen los requerimientos del cliente con actualizaciones en cada cierre de iteración para el control de los mismos

Sí, de igual forma que se realizan validaciones de accesibilidad

Si

¿Que características debe tener un proceso de gestión de requerimientos que se ajuste a las características de Inlutec?

5 respuestas

Antes del desarrollo, debe ser un proceso completo que trate de abarcar todos los casos de uso críticos que se logren identificar, estableciendo reglas sobre los documentos y recursos que se van a generar. Durante la ejecución del desarrollo del software (ya dentro de un proceso de sprints, por ejemplo), debería ser un proceso corto y conciso, que los desarrolladores se "apropien" de este como parte del ciclo de revisión semanal de avances. Parte de estas características están presentes en el proceso de "desarrollo ágil", así que lo que se necesita es adecuar la parte formal (manejo de recursos generados del proyecto)

Agil, que se acople a proyectos de investigación

Ágil, de análisis constante ante diversos cambios

Accesibilidad, funcionalidad y utilidad

Análisis de necesidades principales y una buena investigación sobre las necesidades del cliente.

Si utilizan una herramienta para la documentación de requerimientos ¿Cuál es? ¿Se le puede dar trazabilidad al requerimiento en esta herramienta?

4 respuestas

Scrumdesk ayuda mucho a la visualización de los requerimientos orientados a las "historias de usuario", que se detallan como requerimientos técnicos que se deben realizar para que la historia funcione y tenga un nivel de éxito. Cada requerimiento técnico se le puede asignar recursos (encargado, tiempos estimados de esfuerzo y adjuntar recursos digitales), por lo que es posible realizar una trazabilidad tan específica como se requiera, a nivel de desarrollo del proyecto.

Es un encadenamiento de varias plantillas de documento de drive, scrum desk y git lab

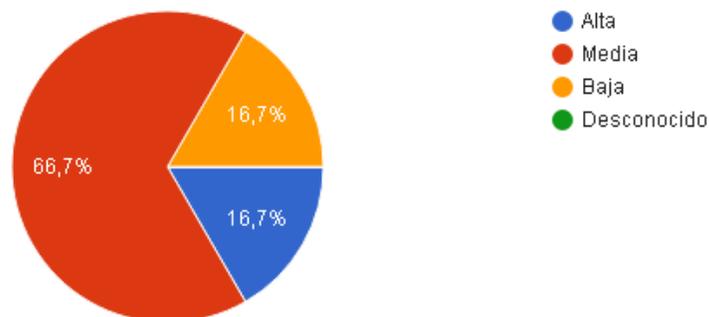
Drive, no :(

Plantilla de requerimientos, cuenta únicamente con una sección de control de cambio, sin embargo, no permite seguir la trazabilidad de los requerimientos

Respuesta 2: Encuesta sobre aseguramiento de calidad en Inlutec.

El resultado de la calidad de software que se ha tenido hasta ahora, ¿Cómo la clasifica usted?

6 respuestas



Actualmente ¿cuales son las principales complicaciones para garantizar la calidad del software desarrollado?

6 respuestas

Tiempo
Requerimientos ambiguos
Falta de requerimientos

Tiempo para realizar pruebas o hacer el proceso de aseguramiento de la calidad, falta apoyo en el aspecto de calidad en la metodología utilizada

Realizar todas las pruebas adecuadas.

Necesidad de un sistema o guía estándar de métricas para cada tipo de programa/módulo/función de un proyecto de software que se desarrolla actualmente, además de una falta de automatización de pruebas que puedan acelerar estas revisiones

El desarrollo de pruebas mediante un equipo externo al de desarrollo.

No se tienen los recursos para efectuarlas (tiempo y gente qa),

Considera que usted que existen problemas o aspectos que mejorar en la calidad del software desarrollado en Inlutec.

6 respuestas

Si existen, pero se solventan cuando se detectan, muchos de estos problemas se originan por requerimientos espontáneos urgentes

Si, todo

Siento que no está mal, sin embargo, podría mejorar.

Sí, después de validar requerimientos, hace falta un proceso rápido para que una o más personas (no necesariamente otro desarrollador) fuera de un proyecto específico en Inlutec pueda hacer de "beta-tester" y pueda realizar ciertas pruebas de estilo "rendimiento" y usabilidad del programa, dependiendo de la etapa en la que se esté trabajando y de un sistema de métricas práctico y automatizable

Documentación de pruebas y análisis de los defectos ya encontrados.

Si

Según su experiencia en la organización ¿Cuáles son las complicaciones que usted identifica para implementar un proceso aseguramiento de la calidad?

6 respuestas

Tiempo

Curva de aprendizaje si la hubiera, tiempos, comunicación con el cliente o jefes sobre la importancia del proceso de calidad y lo poco tangible que es como resultado para de utilidad (generalmente se buscan resultados rápidos y funcionalidades implementadas sin importar la calidad),adaptación del equipo desarrollador

Realizar todas las pruebas necesarias, que aveces no se pueden realizar completamente por el tiempo asignado.

Principalmente adaptar procesos rigurosos de aseguramiento, como las pruebas de estrés u otros que requieren mucho tiempo debido a la cantidad de aspectos que se han requerido para las pruebas del software que habían recaído en una sola persona en el pasado

Los recursos

Poco recursos

Si usted considera que un proceso aseguramiento de calidad es útil en el proceso de desarrollo de software ¿Por qué sería de beneficio?

6 respuestas

Porque acerca el software a la calidad esperada por los usuarios

Un software de calidad es responsabilidad de los desarrolladores, y la imagen de una compañía, la calidad de ese software representa eso.

Ayudaría mucho a saber que el software funciona correctamente, no tiene tantos errores o son menores y para arreglarlos sería más sencillo.

Es beneficioso porque se pueden detectar problemas que dentro del ambiente de desarrollo de software se pueden escapar, no necesariamente relacionados con calidad de código, pero con aspectos de usabilidad y accesibilidad del programa. También es importante para retroalimentación del equipo de desarrollo e ir adoptando mejores prácticas de programación para minimizar tiempos de esfuerzo y evitar caer en los mismos errores

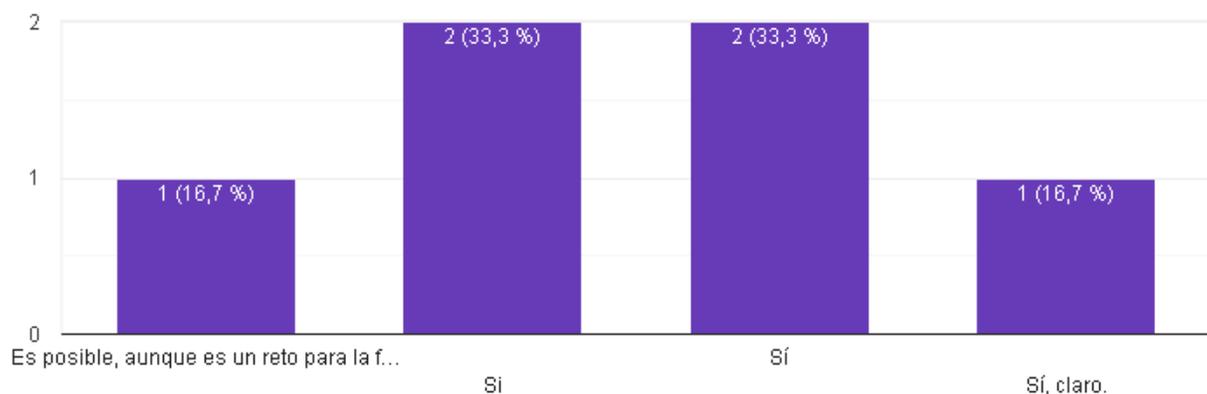
Para asegurar que el programa funcione bajo las condiciones a las que se va a exponer.

Productos que se ajustan a las necesidades del clientes

Considera usted que la metodología de desarrollo que utilizan actualmente se le puede adjuntar procesos de aseguramiento de calidad.



6 respuestas



¿Que características debe tener un proceso de aseguramiento de calidad que se ajuste a las características de Inlutec?

5 respuestas

Que se enfoque en las pruebas de accesibilidad

Que la ejecución del mismo permita que sea aplicado a funcionalidades específicas, es decir que no sea un proceso para todo el proyecto, que sea un proceso que se pueda aplicar en cada etapa del proyecto, en cada funcionalidad

Agil, con disponibilidad de tiempo, adaptable a los procesos de scrum y a los horarios de cierres de iteración

Debería ser un proceso ejecutado en paralelo y ejecutarse justo al terminar una funcionalidad importante. Ser lo suficientemente riguroso para poder detectar problemas lo antes posible, pero sin requerir excesivo esfuerzo por parte de quienes lo ejecutan y que pueda significar un "cuello de botella" para el desarrollo (es decir, debería tener cierto grado de "independencia" respecto al proceso de desarrollo). Lo ideal sería tener procesos concisos y cortos de aseguramiento de calidad a nivel de desarrollo (por ejemplo, programación de tests), pero hay ocasiones donde se deben realizar proyectos muy rápidos y simplemente no se puede dedicar tiempo para ello, hay que analizar qué se puede hacer respecto a este tema. De acuerdo al flujo de trabajo, pruebas generales de calidad deberían realizarse cada vez que se libera el estado actual del proyecto para una demostración o pruebas externas del cliente.

Planificación de actividades para el aseguramiento de la calidad y objetividad.

Respuesta 3: Respuestas entrevistas requerimientos.

Guía de entrevista: Requerimientos.

Entrevistador / rol	Dionisio Palacio Amador/ Investigador
Entrevistado / rol	Víctor Romero Chacón/ Coordinador proyectos web
Fecha	12/02/2020

1. Cuándo se inicia el desarrollo de un proyecto ¿Existen requerimientos documentados o registrado para el inicio de este?

Para el proyecto SICID existe un documento que se siguió para el desarrollo del proyecto, pero no se considera un documento de requerimientos. Son "requerimientos malos".

Para Piels, no se realizaron requerimientos y para el Evaluador se empezaron con requerimientos, pero no se siguió porque fue muy difícil mantenerlo, solo se manejaban los sprint y en ellos se consideraba incorporar o cambiar elementos del proyecto.

2. ¿Cómo se hace el levantamiento de requerimientos para un proyecto de software?

Ha sido diferente por cada proyecto, por ejemplo, para SICID se hicieron posterior al documento de requerimientos. Sin embargo, el desarrollo era aislado del dueño del producto hasta que se hacían las entregas.

Para los otros proyectos que son nuevos, por cada entrega que se hacía, salen nuevos requerimientos, entonces se ha maneja por sprint, pero no hay documentación.

3. ¿Existe documentación de requerimiento de los proyectos desarrollados o que están en desarrollo?

Ver pregunta 1.

4. ¿Cuáles son los problemas más comunes en el proceso de convertir los requerimientos en producto?

Lo principal es la falta de comunicación entre el dueño del producto y el equipo de desarrollo, esto pasaba más a menudo en 2017 y 2018, en 2019 un poco menos porque se empezó hacer reuniones periódicas donde se validaban los requerimientos.

5. ¿Cómo se manejan los cambios en los requerimientos durante el desarrollo del proyecto?

En SICID en 2019, se limitaban a no hacer cambios, pero, en 2017 y 2018 prácticamente se aceptaba cualquier cambio que se propusiera.

En el evaluador, por la naturaleza del proyecto se implementa todo lo que el dueño del producto requiera, incluido los cambio que aparezcan.

En PIELS, se aceptaba todos los requerimientos y cambios que hiciera la comunidad sorda, y esto funcionaba de esta manera al ser un proyecto de investigación.

Esto sí generaba problemas, por ejemplo, en PIELS en ocasiones por los cambios, se tenía que replantear totalmente el desarrollo.

En el caso del evaluador, se ha tratado de mantenerlos en puntos medios, aceptar cambios, pero se busca que no afecte totalmente el desarrollo actual del proyecto.

A raíz de esto, se tenía que trabajar horas extras para cumplir con los deadlines. Solo en Sicid se negociaban los deadlines, al existir cambios se cambiaba el deadline.

6. ¿Han tenido que lidiar con requerimientos ambiguos? ¿Cómo los gestionan?

Sí, por ejemplo, para el catálogo del Sicid, los requerimientos no eran adecuados, se tuvo que hacer reuniones con la contraparte para entender los que se quería, no se utilizó los requerimientos iniciales.

Y ha pasado que la contraparte ha dicho que lo que se le está entregando no es lo que esperaban.

También como no se establecían requerimientos formales, para el caso del SICID, que ha movido el personal a cargo, se da la pérdida de información que afecta también al producto.

Guía de entrevista: Requerimientos.

Entrevistador / rol	Dionisio Palacio Amador/ Investigador
Entrevistado / rol	Verónica Mora Lezcano/ Coordinadora del proyecto Euler
Fecha	12/02/2020

1. Cuándo se inicia el desarrollo de un proyecto ¿Existen requerimientos documentados o registrado para el inicio de este?

Para el caso del proyecto Euler, por el 2018, sí había requerimientos, pero estaban incompletos, algunos no eran aplicables al proyecto, o sea, no se debían satisfacer, y algunos que eran muy necesarios, pero no estaban en el documento de los requerimientos. En general lo que había era un documento con un resumen de las características que se esperaba tuviera la herramienta.

Pero luego un asistente en el proyecto, de la carrera de ATI, les dio un poco de forma a los requerimientos con los que se empezó la nueva etapa del desarrollo.

2. ¿Cómo se hace el levantamiento de requerimientos para un proyecto de software?

Luego que se "arreglaron" los requerimientos del proyecto, los nuevos requerimientos no se han documentado, pero se tiene que hacer para entregarlo como parte del acuerdo en el cierre de etapa.

3. ¿Existe documentación de requerimiento de los proyectos desarrollados o que están en desarrollo?

Actualmente, como dije los requerimientos nuevos no se han documentado, el documento actual es un Word con la descripción de requerimientos.

4. ¿Cuáles son los problemas más comunes en el proceso de convertir los requerimientos en producto?

Al ser un proyecto de investigación se entiende que van a surgir requerimientos repentinos, sin embargo, cuando estos surgen no se analizan respecto al desarrollo actual, tampoco se evalúa la urgencia de desarrollo y la prioridad que se le debe asignar. Esto provoca que los requerimientos aumenten y se puede perder el control del desarrollo e incumplimientos del deadlines.

5. ¿Cómo se manejan los cambios en los requerimientos durante el desarrollo del proyecto?

Los cambios no se gestionan. Ante algún cambio solo se aceptan. Sucede igual que los requerimientos nuevos.

0. ¿Han tenido que lidiar con requerimientos ambiguos? ¿Cómo los gestionan?

Pues sí, y generalmente lo que se ha hecho es tomar la opción más factible.

Respuesta 4: Respuestas entrevistas sobre pruebas.

Guía entrevista

Entrevistador / rol	Dionisio Palacio Amador/ Investigador
Entrevistado / rol	Víctor Romero Chacón/ Coordinador de los proyectos web
Fecha	19/02/2020

1. ¿Sabe usted que es aseguramiento de calidad?

Cumplir con las necesidades del cliente.

2. ¿Qué tan importante son las pruebas en el desarrollo de proyectos de software en Inlutec?

Alta importancia.

3. ¿Cada cuánto tiempo realizan las pruebas?

Previo al *deployment*. Se está tratando que se haga previo a pasar a *develop*, pero es complicado.

4. Utilizan a los requerimientos como guía para para la realización de pruebas.

Antes no, ahora lo que se hace, se hacen las pruebas, y se tiene que cumplir con el *acceptance* criterio de una historia de usuario, porque las historias de usuario son donde se llevan el registro de los requerimientos.

5. ¿Qué tipos de pruebas realizan?
 - a. Funcionales.
 - b. No funcionales: Accesibilidad. Se intentó realizar de desempeño, pero no ha funcionado, pruebas de código (*clean code*).
6. Se realizan pruebas unitarias y pruebas de integración.

Las pruebas unitarias se están tratando de automatizar, eso tiene que cumplir con el requerimiento.

Las pruebas de integración son importantes, porque antes en SICID, se desarrollaba algo y se dañaba otra cosa, entonces estas pruebas han ayudado.

7. Se realizan pruebas automáticas o todo el proceso es manual.

Se aspira que las pruebas unitarias automatizadas se empiecen a utilizar en el evaluador web. Pero por ahora no se hace.

8. Se registran estadísticas sobre los casos de prueba, fallos, aprobados, resultados o comportamientos inesperados.

No se contabiliza los resultados de las pruebas.

9. Desde los inicios del desarrollo de software, ha implementado mejoras en la aplicación de pruebas.

Hasta ahora, en los últimos sistemas es que se han implementado las pruebas.

10. ¿Cómo funciona el proceso para envío de un componente de software a pruebas?

Se hace merge request, se compila en el ambiente de pruebas. En otros casos se pasa a otras personas para que hagan las pruebas respectivas.

11. ¿Han tenido problemas que se pudieron evitar aplicando pruebas?

Sí claro, la aplicación de pruebas ha hecho falta.

12. ¿Cuánto impacto tiene los requerimientos en la aplicación de las pruebas? Por ejemplo, si los requerimientos no son claros, ¿impacta al momento de hacer verificación del software?

Claro que afecta.

13. Tiene conocimiento si los fallos o resultados inesperado en producción, corresponde a los requerimientos no claros o se debió detectar en pruebas.

Han pasado de los dos.

14. ¿Cómo se autoriza el deployment?

- Que compile pruebas.
- Se prueba la funcionalidad específica del motivo del reléase.
- Y Víctor es el encargado de tirarlo a producción.

Guía entrevista pruebas

Entrevistador / rol	Dionisio Palacio Amador/ Investigador
Entrevistado / rol	Víctor Romero Chacón/ Coordinador de los proyectos web
Fecha	18/02/2020

1. ¿Sabe usted que es aseguramiento de calidad?

N/A

2. ¿Qué tan importante son las pruebas en el desarrollo de proyectos de software en Euler?

Sin eso, desarrollamos para nada.

3. ¿Cada cuánto tiempo realizan las pruebas en Euler?

Diario para mejorar nuevas características.

En accesibilidad, dependiendo de la urgencia.

De integración, cuando se libera un conjunto específico de funcionalidades.

4. Se guía con los requerimientos para la realización de pruebas.

Las pruebas son empíricas, se conoce cual debería ser el resultado esperado, y se trabaja con eso.

5. Al equipo de desarrollo que tipos de pruebas realizan.

- a. Funcionales: sí se realizan pruebas funcionales sobre lo que se desarrolla.
- b. No funcionales: No se realizan otras pruebas no funcionales. Solo en ocasiones se somete a revisión de interfaz, pero solo de diseño visual.

6. ¿Se realizan pruebas unitarias y pruebas de integración?

Sí se hacen pruebas unitarias, pero se hacen manuales, no se han automatizado. Y sí se hacen pruebas de integración cuando se requiere revisar toda la solución para alguna entrega.

7. Se realizan pruebas automáticas o todo el proceso es manual.

Todas, las pruebas son manuales.

8. Se registran estadísticas sobre los casos de prueba, fallos, aprobados, resultados o comportamientos inesperados.

Registran los fallos y los resultados inesperados, no se contabilizan en sí, pero se podría hacer. En cuanto a casos de pruebas, no se registran cuantas veces falla algo o cuantos se aprueban.

9. Desde los inicios del desarrollo de software, ha implementado mejoras en la aplicación de pruebas.

Sí se ha mejorado, se hace la revisión de las funcionalidades de cada desarrollador, y además se realiza una prueba de integración donde se prueba todo. Al inicio no funcionaba de ese modo.

10. ¿Cómo funciona el proceso para envío de un componente de software a pruebas?

Yo reviso cada cosa que ellos desarrollan.

Las pruebas de integración se hacen, cuando un nuevo componente va a afectar al resto de funcionalidades, o cuando se van a hacer una entrega de la solución.

11. ¿Han tenido problemas que se pudieron evitar aplicando pruebas?

Sí, por ejemplo, en ocasiones se integran y no se les presta atención y luego resulta en que se han alterado otras funcionalidades.

12. ¿Cuánto impacto tiene los requerimientos en la aplicación de las pruebas? Por ejemplo, si los requerimientos no son claros, ¿impacta al momento de hacer verificación del software?

Sí, porque en el Caso de Euler, si se hicieran pruebas basadas en requerimientos, no van a pasar las pruebas, entonces, sí afecta el estado de los requerimientos en la aplicación de pruebas y sus resultados.

13. Tiene conocimiento si los fallos o resultados inesperado en producción, corresponde a los requerimientos o se debió detectar en pruebas.

Se dan dos casos, en ocasiones al ser de investigación se utiliza defectuoso, y se sabe que puede fallar en ciertos casos, sin embargo, si han surgido fallos que debieron identificar en pruebas.

14. ¿Cómo se autoriza el deployment?

N/A

Respuesta 5: Resultado de la observación.

Platilla del observador

Observador / rol	Dionisio Palacio Amador / Investigador
Fecha	02/03/2020 – 06/03/2020

Proceso	Observaciones
Requerimientos	<ul style="list-style-type: none"> • No se tiene documentación estándar de los requerimientos. • Algunos se llevan en documentos, otros en aplicación web y existen otros que no están documentados en ninguna parte. • No se realiza un proceso de gestión de requerimientos que involucre análisis y especificación.
Calidad	<ul style="list-style-type: none"> • Aseguramiento de la calidad generalmente se refiere solo a pruebas del software. • La actividad de calidad principal es la aplicación de pruebas de funcionalidad y accesibilidad, que es enfocado en verificación y validación de software. • La importancia del resultado de pruebas es, principalmente, saber que partes del software está bien y qué se tiene que solucionar en el momento, por lo tanto, no existen registro ordenados de pruebas aplicadas.
En general	<ul style="list-style-type: none"> • No hay comunicación entre los dos principales equipos, para compartir las buenas prácticas implementadas o herramientas utilizada. • Hay diferencias en la documentación, procesos, prácticas y herramientas de los dos principales equipos de desarrollo.

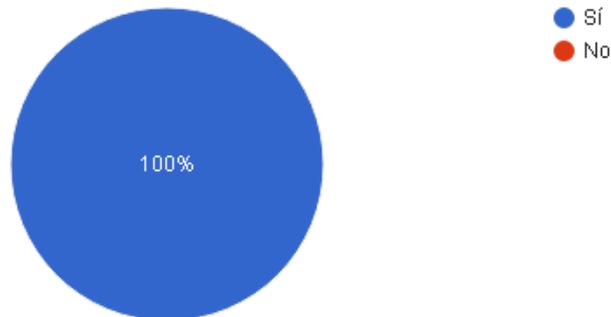
Apéndice C. Validación de la propuesta.

Validación 1. Requerimientos

Proceso general de gestión de requerimientos iniciales.

¿Esta de acuerdo con la estructura propuesta?

6 respuestas



Observaciones

6 respuestas

¿Podrían haber cambios en la etapa de desarrollo de la solución? Cómo afecta a la especificación de los requerimientos? La propuesta me parece genial, solo tomar en cuenta que el alcance podría verse afectado si en el desarrollo surgen imprevistos

La validacion de requerimientos se da en todos los encargados de requerimientos, pero se comprende la idea en el diagrama.

Si se cumple esa estructura los requerimientos van a quedar bien especificados, ya que hay dos entes encargados del proceso, por lo cual supongo que dos personas distintas van a velar por la buena especificación, lo cual permite detectar deficiencias, al tener la perspectiva de otra persona

Me parece bien la estructuración y la secuencia de pasos

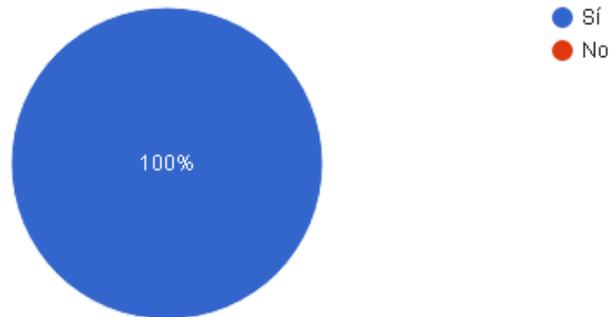
Es clara y facil de entender.

Me parece que la estructura propuesta es la que se debería seguir siempre que se pueda.

Actividades y artefactos para el análisis de requerimientos.

¿Esta de acuerdo con la guía para clasificación y ordenamiento de requerimientos?

6 respuestas



Observaciones sobre la guía.

4 respuestas

no

Muy de acuerdo con el método, sobre todo por que agrupa los requerimientos que no quedan claros de forma que se pueden incluir en la siguiente toma de requerimientos.

Respecto a los requerimientos no claros me surge la duda de cuando se clasifican de esta forma

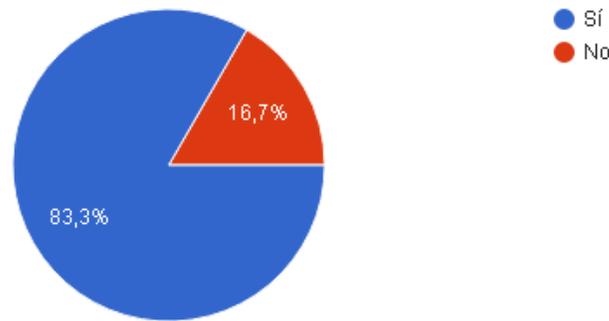
no tengo

Actividades y artefactos para la especificación de requerimientos.

¿Está de acuerdo con la plantilla y recomendaciones para la especificación de requerimientos?



6 respuestas



Observaciones sobre la plantilla y su estructura.

6 respuestas

Es posible adjuntar los escenarios, me parece que los escenarios describen un flujo que debería estar documentado en la especificación

Muy completa, el único detalle es que la dificultad no es tan relevante, ya que asignarla a una tarea depende de todo el equipo y no solo de los expertos en requerimientos.

Debe ser obligatorio respetar la plantilla

no tengo

Bastante completa y me pareció excelente la parte donde se especificaban los campos que se necesitan en ese requerimiento

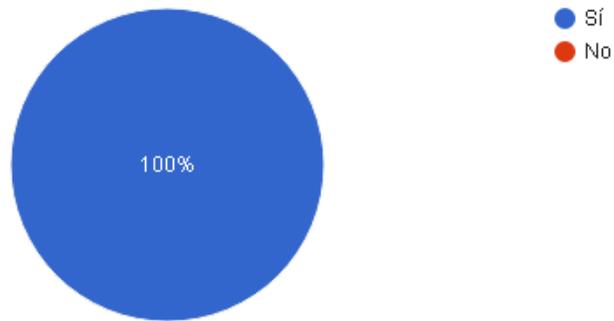
Me parece que la plantilla está bien, pero tal vez podría mejorarse en algunos aspectos.

Validación 2. Evaluación del proceso de verificación y validación.

Proceso de requerimientos durante

¿Esta de acuerdo con la estructura propuesta?

2 respuestas



Observaciones

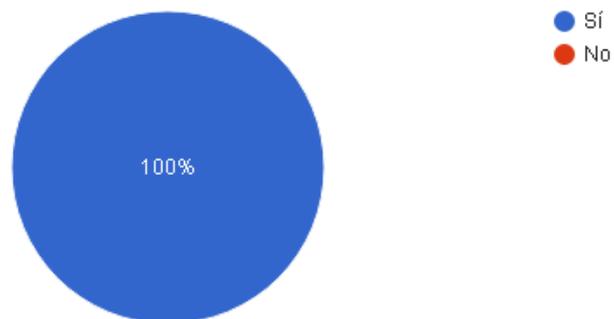
1 respuesta

no hay

Proceso de cambios

¿Esta de acuerdo con la estructura propuesta?

2 respuestas

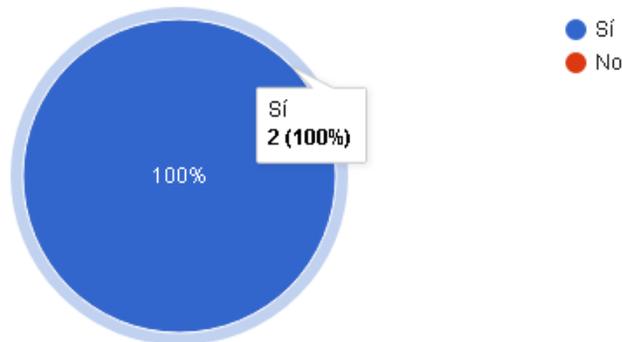


Artefactos para el análisis de requerimientos durante

¿Esta de acuerdo con la plantilla para cambios?



2 respuestas



Observaciones

1 respuesta

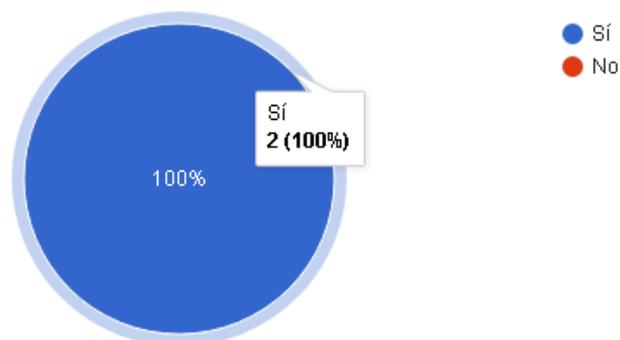
no hay

Proceso general de pruebas unitarias e integración

¿Está de acuerdo con la estructura del proceso general de pruebas? *Nota* La tarea de pruebas de integración esta al final del proceso, sin embargo, esta se realizará durante del desarrollo del sprint.

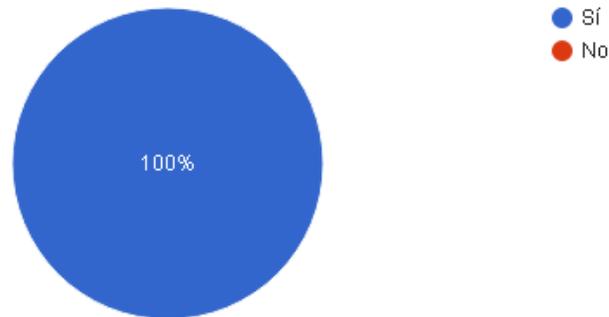


2 respuestas



¿Está de acuerdo con la estructura del proceso de pruebas unitarias?

2 respuestas



Observaciones sobre los procesos de pruebas y pruebas unitarias

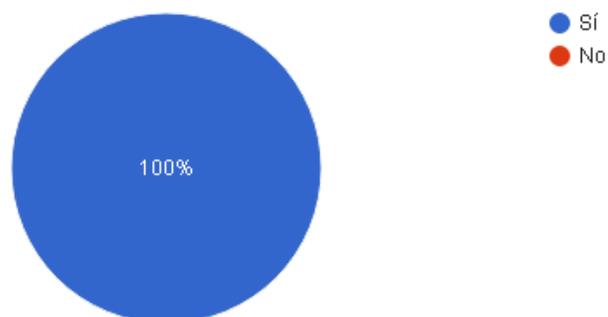
1 respuesta

No hay

Artefacto de pruebas unitarias

¿Está de acuerdo con la estructura del artefacto de pruebas unitarias? *Nota* se utilizará gitlab issue tracker para el reporte de fallos.

2 respuestas



Observaciones del artefacto de pruebas unitarias

1 respuesta

no hay

Apéndice D: Minutas con el profesor

Seguimiento con el profesor tutor.

Minuta 1

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de inlutec
Minuta de reunión**

M-01	26/02/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
26/02/2020	10:00 am	11:00 am	ATI

Agenda
1. Atención de dudas 2. Observaciones sobre el avance.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
ATI	María José Artavia	martavia@itcr.ac.cr

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

<input type="checkbox"/> Observaciones sobre el primer reporte de avance. <input type="checkbox"/> Evacuación de dudas respecto a los capítulos 1,2,3. <input type="checkbox"/> Recomendaciones sobre los capítulos y avances.
--

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 2.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de inlutec

Minuta de reunión

M-02	04/03/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
04/03/2020	10:00 am	11:00 am	ATI

Agenda

1. Atención de dudas
2. Observaciones sobre el avance.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
ATI	María José Artavia	martavia@itcr.ac.cr

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

- Dudas sobre citas APA.
- Dudas sobre firma de minutas.
- Dudas sobre el contenido del marco teórico.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Consultar disponibilidad de reunión para el martes 10/03/2020 a las 11:30am.	DPA	06/03/2020

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 3.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de inlutec

Minuta de reunión

M-03	19/03/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
19/03/2020	10:30 am	11:00 am	Zoom

Agenda
1. Atención de dudas 2. Observaciones sobre el avance.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

<input type="checkbox"/> Dudas los reportes de avance <input type="checkbox"/> Dudas sobre marco metodológico.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 4.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de inlutec

Minuta de reunión

M-04	3/04/2020
------	-----------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
3/04/2020	09:00 am	09:30 am	Zoom

Agenda
1. Observaciones sobre los avances de los capítulos 1, 2, 3
2. Estado del semestre por el Covid 19

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
ATI	María José Artavia	martavia@itcr.ac.cr

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

<input type="checkbox"/> Correcciones sobre los avances. <input type="checkbox"/> Estado de los TFG debido al Covid-19, y cambios en el cronograma.
--

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 5.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de inlutec

Minuta de reunión

M-05	22/04/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
22/04/2020	07:30 am	08:00 am	Zoom

Agenda
1. Dudas sobre capítulo 5. 2. Entrega de avances Capitulo 1,2,3,4.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
ATI	María José Artavia	martavia@itcr.ac.cr

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

<input type="checkbox"/> Evacuación de dudas <input type="checkbox"/> Fechas de entregas de avance.
--

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Entrega del documento con las observaciones	MJA	Próxima semana.

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 6.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de inlutec

Minuta de reunión

M-06	08/05/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
08/05/2020	10:00 am	10:30 am	Zoom

Agenda

1. Comentar avance del capítulo 5
2. Revisar cronograma.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	DPA	Dag1411@gmail.com

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

- Avance del capítulo 5.
- Devolución de los capítulos 1,2 con las observaciones.
- Revisión de los tiempos verbales de los capítulos.
- Establecer fecha para la reunión de visita del tutor a la organización.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Enviar observaciones del capítulo 1,2	MJA	08/05/2020
Fecha para reunión de la visita del tutor	DPA	08/05/2022

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 7.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de inlutec

Minuta de reunión

M-07	27/05/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
27/05/2020	11:30 am	12:00 am	Zoom

Agenda
1. Evacuación de dudas. 2. Revisión de cronograma.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
ATI	MJA	martavia@itcr.ac.cr

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

<input type="checkbox"/> Se resolvieron dudas referentes al capítulo 5. <input type="checkbox"/> Devolución de los capítulos 3,4 con las observaciones. <input type="checkbox"/> Fecha de la segunda evaluación.
--

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Enviar observaciones del capítulo 3,4	MJA	29/05/2020
Reunión de seguimiento el viernes 29 de mayo a las 11:30 am.	MJA	29/05/2020

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 8.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de inlutec

Minuta de reunión

M-08	29/05/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
29/05/2020	11:30 am	12:00 am	Zoom

Agenda

1. Evacuación de dudas.
2. Comentarios de las observaciones el capítulo 3 y 4.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
ATI	MJA	martavia@itcr.ac.cr

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

- Devolución de los capítulos 3 y 4.
- Renombre de las etapas de la investigación basada en los objetivos.
- Comentarios sobre las observaciones.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Enviar capítulo 5, 6 y 7 para revisión.	DPA	29/05/2020

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 9.

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de inlutec
Minuta de reunión**

M-09	05/06/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
05/06/2020	10:30 am	11:30 am	Zoom

Agenda

1. Evacuación de dudas sobre las observaciones de los capítulos 3 y 4.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	DPA	dag1411@gmail.com

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA

4. Puntos tratados

- Evacuación de dudas sobre las observaciones realizadas por el profesor.
- Eliminar las observaciones que no aplican.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Enviar revisión de los capítulos 5,6 y 7	MJA	Siguiete semana

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Apéndice E: Minutas con la organización.

Minuta 1.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de Inlutec

Minuta de reunión

M-01	29/10/2019
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
29/10/2019	9:00 am	11:00 am	Inlutec

Agenda

1. Validación del problema observado.
2. Aclarar dudas.
3. Detalles del problema.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML
Víctor Romero Chacón	vjrch451@gmail.com	VRC

4. Puntos tratados

- Mostrar el problema que se observó a los coordinadores para obtener observaciones y más detalles de la situación.
- Observaciones por parte de los coordinadores acerca del problema.
- Detalles del problema

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
VRC indica los problemas principales en el área de desarrollo web, es que no hay un proceso como tal, solamente se hacen un esfuerzo de realizar pruebas, en ocasiones al final del <i>sprint</i> o al antes de una entrega. Del mismo modo, no se da un seguimiento completo de los resultados de las pruebas que se aplican, por lo tanto, se desconoce la calidad del producto desarrollado.	VRC	N/A
VML indica que también por falta de mecanismo de control de requerimientos, también causa problemas al momento de verificar el software, esto causa problemas con el usuario final, y se tienen que dedicar esfuerzos por corregir software en producción.	VML	N/A
VRC indica que las pruebas son cruciales para definir la calidad del software y hace falta un proceso definido para esto. En la metodología de desarrollo están bien, porque les permite cumplir con las tareas propuestas al final del <i>sprint</i> y han ido mejorando en eso, pero con las pruebas por causa de la carga de trabajo se les ha complicado.	VRC	N/A

Minuta 2.

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de Inlutec
Minuta de reunión**

M-02	12/02/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
12/02/2020	09:15 am	09:45 am	Oficina Verónica

Agenda
1. Entrevista sobre requerimientos

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML

4. Puntos tratados

<input type="checkbox"/> Aplicación de la entrevista sobre requerimientos.
--

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
VLM	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 3.

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de Inlutec
Minuta de reunión**

M-03	12/02/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
12/02/2020	09:50 am	10:15 am	Oficina Víctor

Agenda
1. Entrevista sobre requerimientos

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Víctor Romero Chacón	vjrch451@gmail.com	VRC

4. Puntos tratados

<input type="checkbox"/> Aplicación de la entrevista sobre requerimientos.
--

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Enviar documentos de requerimiento del proyecto Evaluador web y SICID.	VRC	13/02/2020

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
VRC	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 4

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de Inlutec
Minuta de reunión**

M-04	18/02/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
12/02/2020	08:30 am	09:00 am	Inlutec

Agenda
1. Entrevista sobre pruebas de software

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML

4. Puntos tratados

<input type="checkbox"/> Aplicación de la entrevista sobre pruebas de software.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
VLM	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 5

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de Inlutec
Minuta de reunión**

M-05	19/02/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
19/02/2020	4:00 pm	4:30 pm	Inlutec

Agenda
1. Entrevista sobre pruebas de software

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Víctor Romero Chacón	vjrch451@gmail.com	VRC

4. Puntos tratados

<input type="checkbox"/> Aplicación de la entrevista sobre pruebas de software.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
VRC	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 6.

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de Inlutec

Minuta de reunión

M-06	27/02/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
27/02/2020	04:00 pm	04:30 pm	Inlutec

Agenda

- Validación de los procesos actuales de requerimientos y pruebas.
- Recomendaciones y correcciones.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML

4. Puntos tratados

- Validación de los procesos actuales de requerimientos y pruebas.
- Identificar las recomendaciones o correcciones hechas por la involucrada.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Aplicar correcciones a los procesos según las observaciones de la involucrada.	DPA	Siguiente semana

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
VLM	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 7.

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de Inlutec**

Minuta de reunión

M-07	06/03/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
06/03/2020	9:00 am	10:00 am	Inlutec

Agenda
1. Información sobre las pruebas unitarias

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Víctor Romero Chacón	vjrch451@gmail.com	VRC

4. Puntos tratados

<input type="checkbox"/> Herramientas y actividades que se realizan en las pruebas unitarias.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Aplicar correcciones unos cambios en el proceso de pruebas unitarias documentado inicialmente.	DPA	Siguiete semana

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
VRC	
DPA	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 8.

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de Inlutec
Minuta de reunión**

M-08	20/04/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
20/04/2020	11:00 am	12:00 pm	Hangout

Agenda
1. Presentación de propuesta de gestión de requerimientos.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML
Víctor Romero Chacón	vjrch451@gmail.com	VRC

4. Puntos tratados

<input type="checkbox"/> Se presentó al equipo de desarrollo el proceso propuesto de gestión de requerimientos y sus artefactos.
--

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Enviar formulario de evaluación de la propuesta	DPA	Durante la semana

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
VLM	
DPA	
VRC	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 9.

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de Inlutec
Minuta de reunión**

M-09	30/04/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
30/04/2020	3:30 pm	4:30 pm	Hangout

Agenda
1. Presentación de propuesta de gestión de verificación y validación.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/Telefono	Iniciales
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML
Víctor Romero Chacón	vjrch451@gmail.com	VRC

4. Puntos tratados

<input type="checkbox"/> Se presentó al equipo de desarrollo el proceso propuesto de gestión de pruebas y sus artefactos.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Enviar formulario de evaluación de la propuesta	DPA	Durante la semana

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
VLM	
DPA	
VRC	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Apéndice F: Visitas del profesor tutor a la organización
Minuta 1

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de inlutec
Minuta de reunión**

M-01	18/02/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
18/02/2020	01:00 pm	09:45 am	Inlutec

Agenda

1. Cronograma del desarrollo del proyecto.
2. Fechas importantes
3. Aclaración de dudas

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
Inlutec	Dionisio Palacio Amador	dag1411@gmail.com

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML

4. Puntos tratados

- Presentación del cronograma del desarrollo del proyecto a la organización (Evaluación de avances y visitas).
- Control de avances y reuniones periódicas del tutor con el estudiante.
- Recomendación a la organización sobre la revisión de capítulos desarrollados.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
La evaluación periódica por parte de la organización será vía correo.	VML	Según el cronograma
Las reuniones semanales con el tutor será lunes a las 11:30 am, excepto la primera que será martes 25/02/2020.	MJA, DPA	Lunes, semanalmente

6. Puntos pendientes

Pendiente	Responsable (s)
Revisión de filólogo (obligatorio o no)	MJA

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	
VML	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 2

Propuesta de una metodología de aseguramiento y control de calidad para los proyectos de software de inlutec

Minuta de reunión

M-02	11/05/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
11/05/2020	11:30 am	12:00 pm	Inlutec

Agenda

1. Motivo de la segunda visita a la organización
2. Observaciones de los avances por parte de la contraparte de la organización
3. Aclaración de dudas y observaciones.

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
ATI	María José Artavia	martavia@itcr.ac.cr

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML

4. Puntos tratados

- Se comentó sobre la revisión de avances realizados tanto por la contraparte y el profesor tutor.
- Observaciones de la contraparte sobre los avances.
- Aclaración de dudas respecto a las observaciones realizadas en los avances.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)
Enviar formulario de la segunda evaluación.	MJA	Durante la semana
Entrega de lo que falta revisado.	MJA, DPA	Lunes, semanalmente

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	
VML	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Minuta 3

**Propuesta de una metodología de aseguramiento y control de calidad
para los proyectos de software de inlutec
Minuta de reunión**

M-02	24/06/2020
------	------------

1. Información de reunión

Fecha	Hora de inicio	Hora de finalización	Lugar / Medio
24/06/2020	10:00 am	11:00 pm	Inlutec

Agenda

1. Motivo de la segunda visita a la organización
2. Observaciones de la contraparte
3. Aclaración de dudas y observaciones de los capítulos de recomendaciones y conclusiones

2. Datos de persona promotora de reunión

Organización	Nombre de promotor (a)	Correo/Teléfono
ATI	María José Artavia	martavia@itcr.ac.cr

3. Participantes

Nombre	Correo/teléfono	Iniciales
María José Artavia	martavia@itcr.ac.cr	MJA
Dionisio Palacio Amador	dag1411@gmail.com	DPA
Verónica Mora Lezcano	vimora@itcr.ac.cr	VML

4. Puntos tratados

- Se comentó sobre la revisión de avances realizados tanto por la contraparte y el profesor tutor.
- Observaciones de la contraparte sobre los avances.
- Aclaración de dudas respecto a las observaciones realizadas en los avances.

5. Acuerdos / Consultas

Descripción	Responsables	Fecha (cumplimiento)

6. Puntos pendientes

Pendiente	Responsable (s)

7. Firmas de conformidad

Participantes	Firma
MJA	
DPA	
VML	

Nota: 48 horas después de ser enviada por medios digitales la minuta a los y las participantes, se dará por hecho la aprobación de la misma.

Apéndice G: Resumen.
Minutas contraparte

Registro de reuniones con la contraparte

Nombre estudiante	Dionisio Palacio Amador
Contraparte	Verónica Mora Lezcano

Control

Número	Motivo	Fecha
1	Reunión Requerimientos	12/02/2020
2	Reunión Pruebas	18/02/2020
3	Validación de procesos actuales	27/02/2020
4	Presentación-propuesta-requerimientos	20/04/2020
5	Presentación-propuesta-verificación y validación	30/04/2020
6	Primera reunión Profesor tutor	18/02/2020
7	Primera visita Profesor tutor	10/03/2020
8	Segunda visita Profesor tutor	11/05/2020
9	Tercera visita Profesor tutor	24/06/2020

X VERONICA ISABEL MORA LEZCANO (FIRMA) Digitally signed by VERONICA ISABEL MORA LEZCANO (FIRMA) Date: 2020.07.02 07:34:07 -05'00'

Inq. Verónica Mora Lezcano
Contraparte

X 

Dionisio Palacio Amador
Estudiante

Minutas profesor tutor

Registro de reuniones con el profesor tutor.

Nombre estudiante	Dionisio Palacio Amador
Nombre del profesor	María José Artavia

Control

Número	Motivo	Fecha
1	Primera reunión Profesor tutor	18/02/2020
2	Reunión seguimiento	26/02/2020
3	Reunión seguimiento	04/03/2020
4	Primera visita Profesor tutor	10/03/2020
5	Reunión seguimiento	19/03/2020
6	Reunión seguimiento	03/04/2020
7	Reunión seguimiento	22/04/2020
8	Reunión seguimiento	08/05/2020
9	Segunda visita Profesor tutor	11/05/2020
10	Reunión de seguimiento	27/05/2020
11	Reunión de seguimiento	27/05/2020
12	Reunión de seguimiento	05/06/2020
13	Tercera visita Profesor tutor	24/06/2020

X MARIA JOSE
ARTAVIA
JIMENEZ
(FIRMA)

Firmado digitalmente
por MARIA JOSE
ARTAVIA JIMENEZ
(FIRMA)
Fecha: 2020.07.02
08:27:17 -06'00'

Maria José Artavia
Profesor tutor

X 

Dionisio Palacio Amador
Estudiante

Minutas con otros colaboradores de la organización

Registro de reuniones con colaboradores de Inlutec

Nombre estudiante	Dionisio Palacio Amador
Colaborador	Víctor romero Chacón

Control

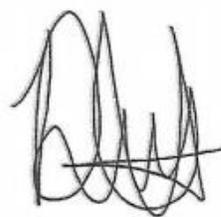
Número	Motivo	Fecha
1	Formulación del problema	29/10/2019
2	Reunión sobre requerimientos	12/02/2020
3	Reunión sobre pruebas	19/02/2020
4	Reunión de aclaración de los procesos actuales	06/03/2020
4	Presentación-propuesta-requerimientos	20/04/2020
5	Presentación-propuesta-verificación y validación	30/04/2020

X



Víctor Romero Chacón
Coord. Desarrollo web

X



Dionisio Palacio Amador
Estudiante

Apéndice F: Guía de implementación.

Guía para la implantación de la metodología de aseguramiento de calidad

El presente documento tiene el propósito de brindar recomendaciones para la implementación de la metodología de aseguramiento de la calidad de software en los proyectos de Inlutec.

Consideración 1. Comunicación y socialización

La metodología propuesta consta de una serie de procesos cortos, que están mapeados y acoplados con la metodología de desarrollo de software actual, en Inlutec. Estos procesos deben ser comunicados a todos los miembros de los equipos desarrollo, de manera que todos tengan conocimientos de la metodología.

Consideración 2. Integración y culturalización

Actualmente, los equipos de desarrollo por los enfoques de los proyectos están separados, sin embargo, la metodología está conformada de manera general para gestionar cualquier proyecto. Por lo tanto, los equipos deben compartir sus conocimientos y prácticas entre sí, de manera que la adopción vaya en conjunto, de esta manera se puede propiciar la mejora continua.

Consideración 3 Autogestión

La metodología cuenta con varios elementos, que son: procesos, actividades, roles y artefactos, tanto para la gestión de requerimientos como para la verificación y validación de software. En primera instancia un rol de administrador de aseguramiento de la calidad es propuesto, sin embargo, esto puede tomar tiempo. Por esta razón, se debe partir de estado actual del proyecto y evaluar la capacidad del equipo para autogestionar los procesos propuestos que apliquen desde ese momento.

Consideración 5. Implementación

Los procesos, por su enfoque iterativo, se pueden abordar en cualquier etapa del proyecto. Por lo tanto, la implementación debe ser como un proyecto, donde cada elemento de la propuesta puede ser tomada como el *backlog* y en cada lapso definido se decidirá el conjunto de elementos que se empezaran a utilizar, con un enfoque progresivo. Además, cada elemento de la propuesta cuenta con las consideraciones para llevarlo a cabo, no deben omitirse.

Consideración 4. Retrospectiva

Una vez se haya iniciado con la adopción de algún proceso, se debe hacer una retrospectiva interna entre el equipo para evaluar el progreso y obtener retroalimentación de la adopción. La metodología tiene la intención de ser flexible, por lo tanto, se puede ajustar a las necesidades siempre y cuando se cumpla con el propósito del proceso. Esta retrospectiva debe ser importante, porque es donde se evalúa lo que se está haciendo bien, qué se puede mejorar, qué no se está haciendo y qué se empezará hacer para continuar etapa.