



CARRERA DE INGENIERÍA DEL SOFTWARE

PROYECTO DE GRADUACIÓN

Sometida a consideración del Tribunal Examinador para optar por el grado de
Licenciatura en Ingeniería del Software

PROPUESTA DE GUÍA PARA EL ASEGURAMIENTO DEL PROCESO DE CALIDAD DE SOFTWARE PARA LAS EMPRESAS COSTARRICENSES QUE DESARROLLAN SOFTWARE PARA CONSUMO NACIONAL

AUTOR: Alexánder Delgado Castillo

Roosevelt Reyes Pérez

TUTOR: Luis Guillermo Alvarado Quesada

LECTORES: Joaquín Artavia Chaves

Roy Camacho Arroyo

TRIBUNAL EXAMINADOR

Ana Cecilia Odio Ugalde
Directora de Carrera

Luis Guillermo Alvarado Quesada
Tutor

Joaquín Artavia Chaves
Lector 1

Roy Camacho Arroyo
Lector 2

DECLARACIÓN JURADA

Nosotros, **Roosevelt Alejandro Reyes Pérez**, mayor, casado, estudiante de la Carrera de Ingeniería del Software, de la Universidad Técnica Nacional, domiciliado en Alajuela, Barrio Monserrat, portador de la cédula de identidad número 2-0599-0062 y **Alexánder Delgado Castillo** mayor, casado, estudiante de la Carrera de Ingeniería del Software, de la Universidad Técnica Nacional, domiciliado en Alajuela Centro, portador de la cédula de identidad número 2-0571-0113, en este acto, debidamente apercebido y entendido de las penas y consecuencias con las que se castiga, en el Código Penal, el delito de perjurio, ante quienes se constituyen en el Tribunal Examinador de mi trabajo de Tesis para optar por el título de licenciatura en Ingeniería del Software, juramos solemnemente que nuestro trabajo de investigación titulado: **“PROPUESTA DE GUÍA PARA EL ASEGURAMIENTO DEL PROCESO DE CALIDAD DE SOFTWARE PARA LAS EMPRESAS COSTARRICENSES QUE DESARROLLAN SOFTWARE PARA CONSUMO NACIONAL”** es una obra original que ha respetado todo lo preceptuado por las Leyes Penales así como la Ley de Derechos de Autor y Derechos Conexos, número 6683 de 14 de octubre de 1982 y sus reformas, publicada en la Gaceta número 226 de 25 de noviembre de 1982; incluyendo el numeral 70 de dicha ley que advierte: artículo 70º: Es permitido citar a un autor transcribiendo los pasajes pertinentes siempre que estos no sean tantos y seguidos, que puedan considerarse como una producción simulada y sustancial, que redunde en perjuicio del autor y de la obra original. Asimismo, estamos advertidos que la Universidad Técnica Nacional se reserva el derecho de protocolizar este documento ante Notario Público. En fe de lo anterior firmamos en la ciudad de Alajuela, el día 04 del mes de setiembre del año dos mil diecinueve.

Roosevelt Reyes Pérez

Cédula de identidad: 2-0599-0062

Alexánder Delgado Castillo

Cédula de identidad: 2-0571-0113

DEDICATORIA

A Dios.

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr los objetivos propuestos, por estar siempre a mi lado acompañándome y guiando con su eterna sabiduría a lo largo de la carrera, por darme fuerzas en los momentos en los que creí que estaba débil y por otorgarme una vida saturada de aprendizajes y experiencias, además de todo su amor e infinita bondad.

A mi esposa, hija e hijo.

A mi esposa, Mary Picado, por siempre darme apoyo, a mi hija Kathleen por esperarme siempre durante las largas noches y a mi hijo Ismael por siempre motivarme y darme fuerzas con sus ocurrencias de mi pequeño "Gokú". A ellos porque siempre me estuvieron apoyando en todo momento, por su gran amor, sus palabras, consejos y motivación diaria, lo que ha permitido que me forme como una persona de bien y un gran profesional.

A mis padres.

A mis padres Franklin Reyes y Guiselle Pérez, por los buenos consejos, por enseñarme la perseverancia, los principios morales, porque estuvieron brindándome lo mejor de sí mismos, por hacer sacrificios para poder estudiar y salir adelante en lo que me proponga, pero más que nada gracias a ellos por su insaciable amor.

A mis profesores.

Por todo lo enseñado, y porque nunca desistieron al enseñarme, aun sin importar que muchas veces uno como estudiante se distraía con facilidad, sin embargo, ellos continuaron depositando esperanza en mi persona. A todos ellos y que no se pueden mencionar, o de lo contrario no terminaría, se los agradezco desde el fondo de mi alma. Para todos ellos hago esta dedicatoria.

Roosevelt Alejandro Reyes Pérez

A Dios.

Por haberme permitido llegar hasta este punto y haberme dado salud para lograr los objetivos propuestos, por estar siempre a mi lado acompañándome y guiando con su eterna sabiduría a lo largo de la carrera, por darme fuerzas en los momentos en los que creí que estaba débil y por otorgarme una vida saturada de aprendizajes y experiencias, además de todo su amor e infinita bondad.

A mi esposa.

A mi esposa Ana Victoria Quesada por ser el complemento de mi vida, quien me impulsa a seguirme esforzando, por siempre estar en todo momento y brindarme su apoyo y consejos en cada decisión que tomamos, por la motivación constante que me hace seguir adelante y por el amor que me otorga.

A mis padres y Hermanos

A mis Padres Adrián Delgado Mora y Olga Castillo Arroyo, por ser un ejemplo de perseverancia por luchar cada día para que yo y mis hermanos lográramos concluir los estudios, por tanto, amor que me han brindado y el apoyo incondicional, por ser mi admiración y ejemplo para ser una persona de bien. A mis hermanos que en cada momento están ahí para alentarme y acompañarme de manera incondicional, por ser mi motor y sentir el apoyo de toda una familia.

A mis profesores.

Por todas las enseñanzas, por compartir este proceso y alentarme a seguir y aprender, gracias por cada momento y experiencia vivida.

A todos ellos y que no se pueden mencionar, o de lo contrario no terminaría, se los agradezco desde el fondo de mi alma. Para todos ellos hago esta dedicatoria.

Alexánder Delgado Castillo

08 de julio de 2019

NOTA DE APROBACIÓN DEL TUTOR

Por este medio, hago constar que el documento denominado **“PROPUESTA DE GUÍA PARA EL ASEGURAMIENTO DEL PROCESO DE CALIDAD DE SOFTWARE PARA LAS EMPRESAS COSTARRICENSES QUE DESARROLLAN SOFTWARE PARA CONSUMO NACIONAL”** elaborado por los estudiantes Alexánder Delgado Castillo portador de la cédula de identidad número 2-0571-0113 y Roosevelt Alejandro Reyes Pérez portador de la cédula de identidad número 2-0599-0062, ha cumplido con elementos propios de la especialidad, permitiendo la detección de problemas de índole teórica y práctica, así como el empleo de los conocimientos adquiridos en el transcurso de la preparación profesional.

El enriquecimiento de la carrera y el fortalecimiento de la Universidad, mediante este tipo de aportes investigativos son fundamentales para el fortalecimiento de la Institución dentro del contexto universitario costarricense, contribuyendo con el desarrollo de la comunidad nacional. Por lo anterior, le doy el aval a este documento para que sea trasladado al análisis y revisión exhaustiva por parte de los lectores que la dirección haya definido.



Luis Guillermo Alvarado Quesada
Cédula 107160408
Número colegiado: 875 CPIC – 35 474 CPCECR

Cc.- Archivo

CARTA DE AUTORIZACIÓN DEL LECTOR

Nota de validación del lector

Por este medio, en calidad de lector del documento **“PROPUESTA DE GUÍA PARA EL ASEGURAMIENTO DEL PROCESO DE CALIDAD DE SOFTWARE PARA LAS EMPRESAS COSTARRICENSES QUE DESARROLLAN SOFTWARE PARA CONSUMO NACIONAL”**, realizado por los estudiantes Alexander Delgado Castillo, portador de la cédula número dos – cinco siete uno – uno uno tres (2-0571-0113) y Roosevelt Alejandro Reyes Pérez, portador de la cédula número dos – cinco nueve nueve– cero seis dos (2-0599-0062), doy fe de que los ajustes y variaciones propuestas por mi persona, han sido tomados en cuenta en la consistencia y redacción técnica del documento final.

Por lo anterior es que doy mi anuencia para que el documento pase a la revisión final del tutor para su aval y posterior envío al filólogo.

Sin más por el momento, suscribo



MGT. Joaquín Artavia Chaves

Cédula de identidad: 204900534

Número de colegiado: 1190 CPIC

Cc.- Archivo

CARTA DE AUTORIZACIÓN DEL LECTOR

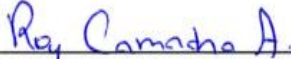
Alajuela, 30 de agosto de 2019

Sra.
MGT. Ana Cecilia Odio Ugalde
Director de la Carrera de Ingeniería del Software
Universidad Técnica Nacional

Estimada señora Directora:

Sirva la presente para saludarle y hacer de su conocimiento mi aprobación, en calidad de lector, del Proyecto de Graduación realizado por los estudiantes **Roosevelt Reyes Pérez**, portador de la Cédula de Identidad No. **2-0599-0062** y **Alexander Delgado Castillo**, portador de la Cédula de Identidad No. **2-0571-0113**, titulado: **"Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para el consumo nacional."** Hago constar que se ha revisado y corregido todos los aspectos referentes a este documento; por lo que manifiesto que el mismo se encuentra listo para ser presentado a la Universidad Técnica Nacional, como trabajo de graduación.

Atentamente,



Lic. Roy Alexander Camacho Arroyo
Lector

CARTA DE REVISIÓN FILOLÓGICA

Alajuela, 03 de setiembre del 2019

Sra.

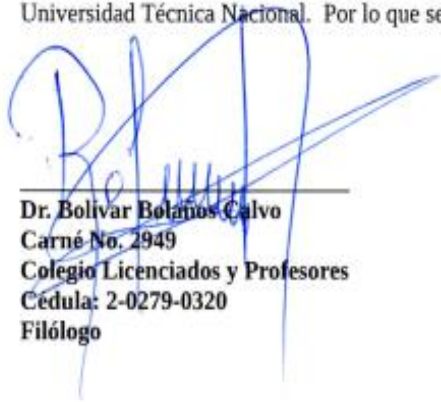
Lic. Ana Cecilia Odio Ugalde

Directora de la Carrera de Ingeniería del Software

Universidad Técnica Nacional

Estimado señora Directora:

Sirva la presente para saludarle y expresar que en mi calidad de Doctorado en Filología Española he revisado la redacción, ortografía y estilo literario del Proyecto de Graduación titulado: **“PROPUESTA DE GUÍA PARA EL ASEGURAMIENTO DEL PROCESO DE CALIDAD DE SOFTWARE PARA LAS EMPRESAS COSTARRICENSES QUE DESARROLLAN SOFTWARE PARA CONSUMO NACIONAL”**, realizado por los estudiantes **Alexander Delgado Castillo** cédula de identidad número 2-0571-0113 y **Roosevelt Alejandro Reyes Pérez** cédula de identidad número 2-0599-0062, para optar por el grado Académico de Licenciatura en **Ingeniería del Software** de la Universidad Técnica Nacional. Por lo que se puede dar fe del correcto español que ésta contiene.



Dr. Bolívar Bolaños Calvo
Carné No. 2949
Colegio Licenciados y Profesores
Cedula: 2-0279-0320
Filólogo

Contenido

Tabla de Ilustraciones	xvii
Tabla de Gráficos	xx
RESUMEN EJECUTIVO	xxii
CAPÍTULO I DEFINICIÓN DEL PROBLEMA	1
1.1 Estado del Arte	2
1.2 Justificación	9
1.3 Planteamiento del Problema	19
1.4 Pregunta Problema	21
1.5 Hipótesis	21
1.6 Objetivos	24
1.6.1 Objetivo General	24
1.6.2 Objetivos Específicos	24
1.7 Alcance y Limitaciones de la Investigación	26
1.7.1 Alcance	26
1.7.2 Limitaciones	26
1.8 Matriz de Congruencia	28
CAPÍTULO II MARCO TEÓRICO	35
2.1 Marco Contextual	36
2.1.1. Desarrollo de Software	36
2.1.2. Fases del Proceso de Desarrollo de Software	37
2.1.3. Proceso para el Desarrollo de Software	40
2.1.3.1 Metodologías Ágiles	40
2.1.3.2. Manifiesto ágil	42
2.1.3.3. Metodologías Ágiles en el Entorno Mundial	46
2.1.3.4. Aseguramiento de la calidad (testeo de pruebas)	55
2.1.4. Herramientas ALM	60
2.1.5. Herramientas de ALM y Metodologías	63
2.2. Marco Legal	66
CAPÍTULO III MARCO METODOLÓGICO	69
3.1. Marco Metodológico	70

3.1.1. Tipo de Investigación.....	71
3.1.1.1. Investigación Exploratoria	71
3.1.1.2. Investigación Explicativa.....	72
3.1.1.3. La Investigación Diacrónica	72
3.1.1.4. La Investigación Cualitativa.....	73
3.1.2. Estudio de Casos	73
3.1.2.1. Estudio Etnográfico	74
3.1.2.2. Estudios Históricos.....	75
3.1.3. Sujetos	75
3.1.4. Población	75
3.1.5. Fuentes de Información.....	80
3.1.5.1 Fuentes Primarias.....	80
3.1.5.2. Fuentes Secundarias	82
3.1.7. Tipo de enfoque.....	84
3.1.8. Matriz Metodológica	86
CAPITULO IV ANÁLISIS DE RESULTADOS Y APORTES	103
4.1. Análisis de Resultados y Aportes.....	104
4.1.1. Sobre la referencia de conocimiento de metodologías de desarrollo.....	107
4.1.2. Sobre las Características del Entorno Empresarial	108
4.1.2.1. Descripción de los roles utilizados.....	110
4.1.2.2. Experiencia de las Personas Encuestadas en Metodologías ágiles.	112
4.1.2.3. Nivel de conocimiento en metodologías ágiles.....	113
4.1.2.4 Uso de Metodologías Ágiles dentro de la empresa a la que laboran	115
4.1.3. Sobre la selección del marco de trabajo	117
4.1.3.1. Planificación de Tareas en las Oorganizaciones	120
4.1.3.2. Visión del Eentorno sobre las metodologías Aplicadas	120
4.1.3.3. Dificultades de implementación de Metodologías Ágiles.....	121
4.1.3.4. Conocimiento en proyectos exitosos utilizando metodologías ágiles	123
4.1.3.5. Razones para adoptar Metodologías Ágiles dentro de una	124
Organización	124
4.1.4. Sobre el Aseguramiento de la Calidad de Software.....	125

4.1.4.1. Consideración para que exista una guía que facilite la Integración de los Procesos.....	127
4.1.4.2. Madurez del proceso de aseguramiento de la calidad de software ..	128
4.1.4.3. Obstáculos presentados al adoptarse el aseguramiento de la calidad del software	130
4.1.4.4. Documentación de los Resultados Obtenidos en las Pruebas de Desarrollo.....	132
4.1.4.5. Creación y Planificación de Planes de Prueba	133
4.1.5. Sobre las Herramientas de Gestión en el Ciclo de Vida del Proyecto.....	135
4.1.5.1. Herramientas para la Documentación de Pruebas	136
4.1.5.2. Beneficios Presentados al utilizar Herramientas para la Gestión de Proyectos	138
4.1.6. Sobre las Entrevistas	139
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES	145
5.1. Conclusiones.....	146
5.2. Recomendaciones.....	154
CAPÍTULO VI PROPUESTA	161
6.1 Introducción	162
6.2 Propósito.....	164
6.3 Objetivo	164
6.4 Alcance.....	164
6.5 Marco Scrum Implementado con Patrones de Aseguramiento de Software	165
6.5.1 Definición de Scrum	165
6.5.2 Pilares de Scrum	166
A. Transparencia	166
B. Inspección.....	166
C. Adaptación.....	166
6.5.3 Roles de Scrum	166
A. Stakeholders (Usuarios).....	166
B. Product Owner (Dueño del Producto)	167
C. Scrum Master (Facilitador)	168

D.	Development Team (Equipo de Desarrollo)	168
E.	Scrum Team (Equipo Scrum)	169
6.5.4	Artefactos del marco de trabajo Scrum y de la Herramienta VSTS	169
A.	Marco de trabajo Scrum	169
B.	Visual Studio Team Services	170
6.5.5	Etapas de planeación	172
A.	Gestión de Requerimientos	172
B.	Criterios de Aceptación	173
C.	Análisis preliminar del backlog	174
6.5.6	Ceremonias SCRUM	175
A.	Sprint Planning (Planeación de Sprint)	175
B.	Daily Scrum (Scrum Diario)	178
C.	Sprint Review (Revisión del incremento)	178
D.	Sprint Retrospective (Retrospectiva del Sprint)	179
6.5.7	Lineamientos Artefactos de Registro en VSTS	180
A.	Lineamientos Atención Épicas	180
B.	Lineamientos atención Historias de Usuario	182
C.	Lineamientos atención FastTrack	186
D.	Lineamientos para Ítems de Seguimiento	190
E.	Lineamientos atención Bugs	191
F.	Lineamientos para el Manejo de Tareas	198
6.5.8	Manejo de Sprint	203
A.	Manejo planificado	203
B.	Manejo No planificado	204
C.	Granularidad de tareas	205
D.	Pesaje de necesidades	205
E.	Velocidad promedio del equipo	206
F.	Manejo capacidad por colaborador	208
G.	Definición de DONE	208
H.	Entrega continua	209
I.	Paralelismo entre QA y Desarrollo	214
6.6	Métodos y Técnicas de Apoyo al Marco de Trabajo Ágil	215

A.	Método SMART	215
B.	Técnica estrella de mar	216
C.	Técnica Planning póker	217
6.7	Procesos de aseguramiento de calidad en el Software	219
6.8	Documentación de Pruebas	220
6.8.1	ISTQB	220
6.8.2	Definición de Aseguramiento de la Calidad	221
6.8.3	Definición de testing	221
6.8.4	Principios del testing	221
6.8.5	Fundamentos de testing	222
A.	Disparadores de errores	222
B.	Errores en especificaciones	222
C.	Es necesario comenzar con testing tan pronto como comenzamos el proceso de desarrollo	222
D.	Las pruebas deben continuar hasta estar seguros de que no habrá fallas graves en el sistema, justo al final del proceso de desarrollo.	222
6.8.6	Fundamentos de Pruebas	222
A.	Los fallos de software pueden llevar a:	222
B.	Los sistemas grandes y complejos, son imposibles de probar de manera exhaustiva	223
C.	Los riesgos	223
D.	La calidad	223
E.	Decidir cuándo ya es suficiente:	224
F.	Depuración (Debugging)	224
6.8.7	Enfoque de técnicas	224
A.	Pruebas estáticas	224
B.	Pruebas dinámicas	225
6.8.8	Técnica de pruebas	225
A.	Técnicas de Cajas Blancas (White-Box Testing)	225
B.	Técnicas de caja negra (Black-Box Testing)	226
C.	Pruebas de caja gris (Gray box testing)	229
6.8.9	Técnicas de diseño de pruebas	230

A.	Identificar condiciones de prueba:	230
B.	Identificar casos de prueba:	230
C.	Identificar procedimientos de prueba:	231
6.8.10	Estimación de Pruebas	231
A.	Enfoque basado en métricas.	231
B.	El enfoque basado en expertos	231
C.	Variantes de estimación.	232
6.8.11	Niveles de pruebas	233
A.	Pruebas de componente	233
B.	Pruebas de integración	234
C.	Pruebas de sistema	234
D.	Pruebas de aceptación	234
E.	Pruebas de aceptación de usuario	234
F.	Pruebas operativas de aceptación	235
G.	Pruebas de Aceptación Contractual y Normativa	235
H.	Pruebas Alfa y Beta	235
6.8.12	Tipos de pruebas	236
6.9	Proceso de Aseguramiento	247
6.9.1	Procedimiento de aseguramiento de software	247
6.9.2	Enfoques y Estrategias de Pruebas	253
6.9.3	Planificación de Pruebas	253
6.9.4	Control, Gestión y Seguimiento del Proceso de Prueba	255
6.9.5	Control de estados de la herramienta Visual Studio Team Services	256
6.9.6	Definición y Aplicación de Métricas	258
6.9.7	Análisis y diseño de prueba	259
6.9.8	Implementación y Ejecución de Pruebas	260
6.9.9	Gestión de Defectos	262
6.9.10	Actividades de Cierre de Pruebas	267
6.9.11	Instalación en Ambiente de QA y Gestión de Incidentes	269
6.9.12	Acompañamiento en Proceso de Pruebas UAT	270
6.9.13	Gestión de Incidentes UAT	270
6.10	Herramientas de Apoyo a Pruebas	270

6.10.1	Microsoft Visual Studio Team Services (VSTS)	271
6.10.2	Microsoft Test Manager (MTM)	271
6.10.3	PL/SQL	271
6.10.4	Toad	272
6.10.5	SQL Server Managment Studio	272
6.10.6	WcfStorm	272
6.10.7	WebServiceStudio	272
6.10.8	Guía de uso Microsoft Visual Studio Team Services	272
6.11	Guía de uso Microsoft Test Manager	281
6.11.1	Configuración de la herramienta	281
6.11.2	Planeamiento y diseño de casos de pruebas	282
6.11.3	La ejecución de casos de prueba	287
6.11.4	Ejecutar casos de prueba	290
6.11.5	Ejecución de pruebas exploratorias	292
6.11.6	Gestión de Dashboard y métricas de evaluación de procesos de desarrollo y calidad de software	293
	Referencias	295
	Anexos	301
	Encuestas	302
	Encuesta presentada	304
	Entrevista	315
	Resultados de las entrevistas	319

Tabla de Ilustraciones

Ilustración 1: Desarrollo del Software y Administración de Proyectos.....	18
Ilustración 2: Ágile Software Development	42
Ilustración 3: Las 5 fuerzas de Porter.....	47
Ilustración 4: Modelo Cascada (Cortés Morales, 2006).....	48
Ilustración 5: Modelo Espiral (Cortés Morales, 2006).....	48
Ilustración 6: Etapa de Planeación (Sprint).....	53
Ilustración 7: Niveles de Confianza para determinar Muestreo de Población	77
Ilustración 8: Fórmula para determinación de Muestreo.....	77
Ilustración 9: Detalle de Muestra de Población.....	77
Ilustración 10: Flujo de Trabajo.....	182
Ilustración 11: Flujo de Estados.....	182
Ilustración 12: Formulario de Historia de Usuario.....	185
Ilustración 13: Flujo de Estados.....	186
Ilustración 14: Formulario de FastTrack.....	189
Ilustración 15: Flujo de Estados.....	190
Ilustración 16: Flujo de Trabajo.....	191
Ilustración 17: Formulario de Bug.....	197
Ilustración 18: Flujo de Estados.....	197
Ilustración 19: Formulario de tareas.....	202
Ilustración 20: Flujo de Estados.....	202
Ilustración 21: Planeación de Sprint.....	203
Ilustración 22: Historias de Usuario	204
Ilustración 23: Capacity.....	208
Ilustración 24: Flujo de Reuniones Diarias.....	211
Ilustración 25: Refinamiento de BackLog.....	214
Ilustración 26: S.M.A.R.T.....	216
Ilustración 27: Técnica Estrella de Mar.....	216
Ilustración 28: Técnica Planning Póker.....	218
Ilustración 29: ISTQB.....	220
Ilustración 30: Pruebas de Caja Blanca.....	226

Ilustración 31: Pruebas de Caja Negra	227
Ilustración 32: Pruebas de Caja Gris.....	229
Ilustración 33: Niveles de Pruebas.	233
Ilustración 34: Proceso Básico de Aseguramiento.	247
Ilustración 35: Tareas Básicas de un asegurador de calidad dentro de SCRUM.	248
Ilustración 36: Scrum Póker.	249
Ilustración 37: Proceso de Recopilación de Métricas de Software.	259
Ilustración 38: Ingreso a VSTS.	273
Ilustración 39: Panel de Épicas.....	273
Ilustración 40: Panel de BackLog Ítems.....	274
Ilustración 41: Crear Nuevo Sprint.	274
Ilustración 42: Crear Sprint.	275
Ilustración 43: Administrador de un sprint.	276
Ilustración 44: Administrador de un Sprint. 2.	276
Ilustración 45: Visualizar Trabajos Pendientes.....	277
Ilustración 46: Visualizador Trabajos Pendientes. Panel.....	277
Ilustración 47: Visualizador de Trabajos Pendientes. Capacidad.....	278
Ilustración 48: Gráficas de Esfuerzo.	278
Ilustración 49: Consulta de Listas Planas.	279
Ilustración 50: Consultas Graficadas.	279
Ilustración 51: Consulta listas de vínculos directos.....	280
Ilustración 52: Consultas de Tipo Árbol.	280
Ilustración 53: Configuración de la Herramienta.	281
Ilustración 54: Seleccionar Team Project.....	281
Ilustración 55: Seleccionar Plan de Pruebas ligado al sprint.....	282
Ilustración 56: Planeamiento de pruebas. Plan.	282
Ilustración 57: Configuración Plan de Pruebas.	282
Ilustración 58: Crear Suite.....	283
Ilustración 59: Creación de Suites en Historias de Usuario.....	283
Ilustración 60: Creación de casos de prueba para Historias de Usuario.....	284
Ilustración 61: Crear Caso de Pruebas	284
Ilustración 62: Agregar Documentos para respaldar pruebas.	285

Ilustración 63: Estado Casos de Pruebas.....	285
Ilustración 64: Consulta Plan de Pruebas.....	287
Ilustración 65: Ejecución Casos de Prueba. Test.....	287
Ilustración 66: Seleccionar Suites según Historia de Usuario	288
Ilustración 67: Casos de Prueba. Run.....	288
Ilustración 68: Selección de la Iteración a Evaluar.....	289
Ilustración 69: Start Test.	289
Ilustración 70: Ejecución de Casos de Prueba.....	290
Ilustración 71: Evaluación de las Expectativas.....	290
Ilustración 72: Resultados de la prueba de la iteración.....	291
Ilustración 73: Continuar con la Siguiete Iteración.....	291
Ilustración 74: Guardar Resultados de la Prueba.....	291
Ilustración 75: Resultado de una Prueba Específica.	292
Ilustración 76: Ejecutar Pruebas Exploratorias.	292
Ilustración 77: Gráficos Auto-Generados por recomendación de VSTS.....	293
Ilustración 78: Ejemplo de Dashboard.	294

Tabla de Gráficos

Gráfico 1: Experiencia en encuestas sobre metodologías de desarrollo	107
Gráfico 2: Situación actual de las empresas en Costa Rica.	109
Gráfico 3: Estimado de colaboradores que laboran en empresas de Software.	109
Gráfico 4: Responsabilidades de trabajo dentro de la organización.	111
Gráfico 5: Años de experiencia en puestos.....	111
Gráfico 6: Conocimiento sobre metodologías ágiles.	112
Gráfico 7: Participación en metodologías ágiles.	114
Gráfico 8: Aplicación metodologías ágiles en el orden laboral y personal.	114
Gráfico 9: Actualmente utilizan metodologías ágiles en las empresas.....	115
Gráfico 10: Utilización de metodologías ágiles en las empresas.	115
Gráfico 11: Localidades que utilizan metodologías ágiles en las empresas.....	116
Gráfico 12: Metodologías ágiles implementadas en las empresas.....	117
Gráfico 13: Responsabilidades definidos en desarrollo de actividades.	118
Gráfico 14: Utilización de métodos para mejorar y desarrollar los procesos actuales.....	118
Gráfico 15: Planificación de tareas en las empresas.....	120
Gráfico 16: ¿La metodología aplicada actualmente es funcional?	121
Gráfico 17: Obstáculos para implementar las metodologías ágiles.....	122
Gráfico 18: Proyectos exitosos con metodología ágil.	123
Gráfico 19: Opiniones sobre incorporación de metodologías ágiles dentro de las empresas.	124
Gráfico 20: Aseguramiento de la calidad del software dentro las empresas.	126
Gráfico 21: Mejora dentro de los procesos de aseguramiento de calidad del software.....	126
Gráfico 22: Existencia de guía para integrar los procesos de aseguramiento en las metodologías actuales.	127
Gráfico 23: Realización de pruebas formales durante el desarrollo.....	129
Gráfico 24: Realización de casos de prueba para documentar pruebas.....	129
Gráfico 25: Obstáculo para adoptar aseguramiento de la calidad del software dentro de las empresas.....	131
Gráfico 26: Documentación de resultados de pruebas en las empresas.....	132
Gráfico 27: Importancia y planificación de planes de prueba.	133
Gráfico 28: Realización de diferentes pruebas durante el desarrollo en ambientes definidos.....	134

Gráfico 29: Utilización de herramientas para la gestión de desarrollo.....	135
Gráfico 30: Herramientas de gestión de proyectos utilizadas en las empresas.	136
Gráfico 31: ¿Las empresas utilizan herramientas para la documentación de pruebas?	137
Gráfico 32: Herramientas para la documentación de pruebas utilizadas en las empresas.....	137
Gráfico 33: Beneficio para utilizar las herramientas para la gestión de proyectos, dentro de las empresas.....	139

RESUMEN EJECUTIVO

El desarrollo de la tesis está orientado al desarrollo de procesos de calidad de software, basado en la mejora del área de ingeniería del software enfocado en empresas nacionales orientadas al desarrollo de software para consumo nacional, siguiendo la tendencia actual en las técnicas que se incorporan y dando fortaleza al proceso de calidad.

El proceso de aseguramiento de la calidad de software es parte fundamental de los procesos de desarrollo modernos, donde el cambio tecnológico y la brecha digital se transforma a gran velocidad y se identifica la necesidad de cuidar la calidad de los productos terminados que se desarrollan, de ahí la importancia de conformando los procesos de aseguramiento de la calidad de software, donde se involucran tres aspectos fundamentales, personas, procesos y herramientas, mismas que se unen para permiten gestionar la calidad durante el desarrollo de un producto de software, importante tomar en cuenta durante la definición de una área de aseguramiento de la calidad es lograr la integración de los tres aspectos mencionados para con ello facilitar la conformación y buen desarrollo del área.

Dado la gran importancia y ser complemento de un proceso robusto de desarrollo, no todas las empresas aplican procesos de calidad de software, esto debido a las dificultades que se presentan donde se identifican limitantes como:

- La falta de experiencia en este campo.
- Falta de información sobre metodologías ágiles.

- La cultura organizacional.
- La resistencia al cambio o adaptaciones de estándares más complejos.

Se identifican beneficios que se adquieren con el cambio como lo son:

- La mejora del trabajo en equipo.
- La habilidad en el manejo de cambios.
- Incremento en la productividad.
- La mejora en la calidad de software.

Al tomar la iniciativa de cambio e identificar la necesidad, es importante para las empresas tener claridad de las limitantes u obstáculos que se presentan al conformar un departamento de calidad de software donde se identifican las siguientes:

- La falta de una guía que structure el proceso.
- La cultura organizacional.
- La falta de experiencia del personal.
- La resistencia al cambio.
- La falta de información sobre el aseguramiento en la calidad de software del lado de los colaboradores.

La presente tesis, tiene como objetivo investigar el área y presentar una guía que muestre un enfoque práctico de cómo se debe de realizar un proceso de

aseguramiento y que pueda guiar a las empresas a crear la estrategia de cambio y administrar la calidad durante el desarrollo de un proyecto de software.

Para construir la guía se realizó una investigación que detallan los factores importantes que conforman el proceso de calidad de software, los cuales serán útiles para realizar la selección del marco de trabajo, herramientas y tener la base de conocimiento para definir la técnica de aseguramiento de la calidad, los tres actores mencionados son valorados factores de integridad que ayuden a unir aspectos importantes que definen una buena ejecución y con ello enfocar las distintas actividades a desarrollar durante la aplicación de aseguramiento de la calidad del software.

En la definición de la metodología se toman en cuenta la opinión de personal que desarrolla labores en el área de desarrollo de aplicaciones de empresas nacionales, esto para medir cual es la tendencia actual de las empresas costarricenses, donde se identifica que se busca la disposición de adoptar metodologías ágiles, logrando definir el marco de trabajo *Scrum*, como la base actual que buscan las empresas nacionales dedicadas al desarrollo de software o que prestan servicios informáticos, basado en el marco de trabajo mencionado.

Se tomó en cuenta detalles importantes que ayudaron a la selección de la herramienta de administración del ciclo de vida y las herramientas de calidad de software, donde se busca que las herramientas cuenten con tendencias ágiles y que integren diseños que permitan el seguimiento a la ejecución y diseño de las labores tanto de trabajo a desarrollar, así como para la organización de planes de pruebas, mismas que se integran y mantienen detalles como la conformación

de *Sprint backlog* (manejo de tareas), que cuenten con dashboard para el seguimiento de tareas y backlog para administración de necesidades y requerimientos.

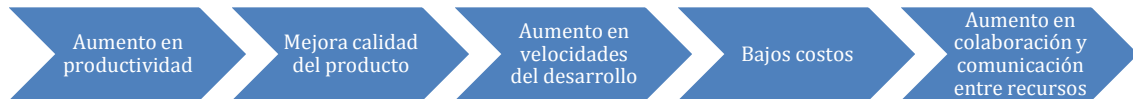
Entre otros aspectos a detalle a esta selección se incorporan aspectos propios de las buenas practica de aseguramiento de la calidad de software donde se adoptan practicas desarrolladas por *ISTQB* (Comité Internacional de Certificaciones de Pruebas de Software), dentro de la guía la técnica de aseguramiento y *testing* (pruebas), integrando procesos de entrega continua donde se identifican etapas que debe realizar un asegurador de calidad para poder desarrollar el proceso.

La guía propuesta también busca ser fuente de conocimiento a lo cual se detallan aspectos de la técnica de aseguramiento de la calidad, clasificación de pruebas, tipos de pruebas entre otros aspectos, en cuanto al marco de trabajo se definen los roles, se detallan reuniones y técnicas que se aplican durante la ejecución del proceso para mejorar los resultados de la misma, también se detallan aspectos de configuración de las herramientas y el uso de las mismas durante la aplicación de los procesos.

La ejecución de la calidad en aquellas empresas dónde su principal actividad económica está representada en el desarrollo de proyectos de software, se ofrece una oportunidad competitiva en la que permite a los representantes de estas asegurar sus productos de calidad, lo que genera un pronto retorno de inversión, permitiendo a las organizaciones optar por un cambio orientado a las metodologías ágiles, dónde se enfrentarán a retos como lo es el aseguramiento de la calidad tanto del producto como del proyecto. Este cambio obliga a las

empresas a realizar un análisis y crear una propuesta en busca de mejorar la calidad en los productos desarrollados donde se deben de tomar en cuenta la metodología, selección de herramientas, procesos y sobre todo mantener siempre buenas prácticas con lo que se diseñará a corto plaza un conjunto de recomendaciones orientado a la gestión de calidad en los proyectos de software.

La guía desarrollada dará una visión de conformación a un proceso de aseguramiento de la calidad de software donde se integran la metodología, los procesos, técnicas de aseguramiento, buenas prácticas del proceso, logrando con la buena aplicación del cambio tener un área de desarrollo de software con un proceso más robusto, moderno, adoptado a nuevas tendencias, para consigo tener los siguientes beneficios:



Importante no dejar de lado la innovación de la academia donde se fortalece la necesidad de tener mayor número de profesionales en el área que ayuden en los procesos de cambio y que conozcan bien las técnicas de aseguramiento de la calidad de software para que las empresas puedan tener más claridad de la gestión del cambio y motivar a más empresas a que realicen innovación de sus procesos y herramientas y con ello ampliar la oportunidad de adoptar nuevas tecnologías que ayuden a innovar y fortalecer sus áreas de desarrollo de software.

CAPÍTULO I DEFINICIÓN DEL PROBLEMA

1.1 Estado del Arte

Para entender más sobre este proceso es importante entender el significado de calidad de software, la palabra calidad posee numerosos significados, se puede definir como aquel criterio que obtiene el cliente final sobre un producto o bien por el cuál brindó en muchos casos una remuneración a cambio. Otra definición que puede brindarse es, También puede considerarse como un conjunto de propiedades, sujetas a un objeto determinado realizando un efecto de visualización u apreciación favorable o desfavorable a los demás objetos que forman su mismo tipo. Una definición más académica es la obtenida por medio del diccionario de la Real Academia Española en su sitio web <http://lema.rae.es/drae2001/srv/search?id=th6fUJ0TbDXX2Xc9jAuL>, dónde se define la palabra calidad como “propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor”.

En cuanto a temas sobre aseguramiento de la calidad, el autor Julián Mera-Paz en su artículo orientado al análisis del proceso de pruebas de calidad de software, publicado en el sitio web <https://revistas.ucc.edu.co/index.php/in/article/view/1482>, menciona que en el país de Argentina, el Instituto Nacional de Tecnología Industrial implementó después del año 2008 un laboratorio de testeo y aseguramiento de calidad de software, que mantiene en firme la política de “***El testeo como parte del proceso de calidad de productos Pruebas de calidad a todo tipo de***

aplicación”, el resultado del proceso de la calidad es el aumento de la calidad al salir a producción, ya que se forma parte del proceso del departamento de calidad. (Mera-Paz, 2016).

Como aporte el mismo autor menciona que, en la Universidad Tecnológica Nacional, en la facultad regional de Buenos Aires, se encuentra el escrito titulado “**Estudio comparativo de los modelos y estándares de calidad del software**” escrito por la Lda. Fernanda Scalone, que permite optar por el título de master en Ingeniería de la calidad, en su escrito determina que el principal instrumento que permite garantizar la calidad en los desarrollos de software es sin lugar a duda el plan de calidad que se basa en normas o estándares completamente genéricos así como en ciertos procedimientos particulares, dónde estos sin lugar a duda van a ser diferenciados según la organización que los ponga en práctica, sin embargo, lo que realmente es importante es la documentación que respaldarán los procesos de la organización y como se verifican que estos sean debidamente ejecutados. (Mera-Paz, 2016)

En España, el estudiante Hugo Alexer Parada Gelvez realiza una investigación que como objetivo final tiene la una tesis doctoral de Ingeniería de sistemas telemáticos, titulado “**Contribución a la gestión de los procesos de software y servicios**”, en la Universidad Politécnica de Madrid, destaca que “son escasas las investigaciones centradas en desarrollar metodologías y técnicas de pruebas en nuevos dominios”, lo anterior según mencionado en el sitio web <https://revistas.ucc.edu.co/index.php/in/article/view/1482> del autor Julián Mera-Paz. (Mera-Paz, 2016)

En España, en la Universidad de Girona, Beatriz Florian el autor del sitio web <https://revistas.ucc.edu.co/index.php/in/article/view/1482> indica que gracias a los aportes brindados por los profesores colombianos Oswaldo Solarte y Javier Reyes se desarrolló un proceso de investigación al cuál fue denominado como “**Propuesta para incorporar evaluación y pruebas de usabilidad dentro de un proceso de desarrollo de software**”, en este se logra obtener como resultado que las pruebas y evaluaciones de usabilidad realizadas durante el desarrollo de un producto de software han mejorado ampliamente la aceptación para verse como estrategia y así mejorar la calidad del producto software. (Mera-Paz, 2016)

En España, en la Universidad de Sevilla, el Ingeniero Rafael Ceballos Guerrero en su tesis doctoral en Lenguajes y sistemas informáticos, presenta el informe nombrado “**Técnicas automáticas para la diagnosis de errores en software diseñado por contrato**”, quién determina la importancia de la calidad en los productos de software. En esta tesis se resalta lo siguiente, “la detección de errores en etapas tardías genera mayores costos en las etapas de desarrollo de software y al final impactan en el precio al cliente”, lo anterior permite enfatizar el uso de automatización para la detección de fallos como parte del proceso de calidad, misma información es localizada en el sitio web <https://revistas.ucc.edu.co/index.php/in/article/view/1482> del autor Julián Mera Paz (Mera-Paz, 2016).

En Estados Unidos, en el Instituto de Tecnología de Massachusetts (por sus siglas en inglés MIT *Massachusetts Institute of Technology*), en el programa ciencia, tecnología y sociedad, se imparte el curso académica

denominado “Introducción al proceso personal de software”, dónde la esencia del mismo curso es el desarrollo del software, sin embargo, según Julián Mera Paz, este curso dedica todo un capítulo enfocado en el proceso de pruebas orientados a la calidad de software. (Mera-Paz, 2016)

En Estados Unidos, en la Conferencia internacional de Ingeniería de Software, la ponencia “**Aseguramiento de la calidad mediante rigurosas especificaciones y pruebas de software: un estudio de caso**” presenta un estudio de caso realizado por varios profesores de la Universidad Estatal de Ball, de la Universidad de Purdue y de la Universidad de Indiana, dónde se determina la importancia del aseguramiento en la calidad del software de una forma temprana, esto según expuesto por Julián Mera Paz en su sitio web <https://revistas.ucc.edu.co/index.php/in/article/view/1482> (Mera-Paz, 2016).

En Suecia, el Instituto de Tecnología Karlskrona y la Universidad de Gotemburgo, realizan una investigación la cual han llamada como “**Interactividad de las pruebas de software basado en búsquedas bajo un experimento controlado**”. Según Julián Mera Paz en su sitio web <https://revistas.ucc.edu.co/index.php/in/article/view/1482>, bajo el nombre de Análisis del proceso de pruebas de calidad de software determina que en dicha investigación se realiza un seguimiento riguroso al sistema de búsqueda interactivo el cual es basado en software de Testeo, mismo ha sido posible por medio de un experimento controlado donde contaron con la participación de 58 estudiantes curso Verificación y validación de software.

Además, se ejecutan pruebas manuales y automatizadas, en estas pruebas se logra evidenciar que las pruebas automatizadas permiten localizar

una considerable cantidad en el número de fallos en comparación a los detectados por las pruebas manuales, por esta razón se decide entre las instituciones reforzar y mejorar este curso con un laboratorio de seguimiento, verificación y validación de software, permitiendo adecuar infraestructura y la adquisición de herramientas para la ejecución de diferentes análisis. (Mera-Paz, 2016)

En Beirut, Líbano, la Universidad Americana de Beirut, en conjunto con los profesores Masri y Zaraket se propusieron a realizar una investigación titulada “Pruebas de software basadas en cobertura: más allá de los requisitos de prueba básicos” (“**Coverage-Based Software Testing: Beyond Basic Test Requirements**”), según Julián Mera-Paz en su estudio denominado “*Análisis del proceso de pruebas de calidad de software*” localizado en su sitio web determina el análisis de la cobertura del software, enfocado en las pruebas de testeo así como los requisitos que deben existir para una prueba extraordinaria, además, se en su escrito se orienta al lector, sobre la importancia que llegan a tener las pruebas de calidad de software, desde el inicio de una etapa temprana y muestra como profundizar en las pruebas de testeo correctamente en el desarrollo de un software con calidad. (Mera-Paz, 2016)

La Universidad Noruega de Ciencia y Tecnología, ubicada en Noruega, según Julián Mera-Paz en su escrito en el sitio web <https://revistas.ucc.edu.co/index.php/in/article/view/1482>, menciona que los profesores Deak, Stalhane y Sindre realizaron una investigación denominada Retos y estrategias para motivar al personal de pruebas de software

(“Challenges and strategies for motivating software testing personnel”);

en esta investigación, al menos doce empresas de tecnología de Noruega, facilitaron el aporte de 36 colaboradores con el fin de identificar los desafíos y las estrategias que permiten motivar al personal a cargo del departamento de pruebas en la calidad de software, y con esto demostrar que el factor humano, el talento humano y las condiciones presentan una repercusión en la calidad de los productos de software que son desarrollados por las empresas. (Mera-Paz, 2016)

En Costa Rica, según SINAES en su sitio web www.sinaes.ac.cr en su documento encontrado en el sitio, habla sobre el hallazgo más destacado ha sido la realización de una Conferencia Magistral de la “Cátedra SINAES Enrique Góngora Trejos Educación Superior y Sociedad”, donde obtuvo la participación la Dra. María José Lemaitre, vicepresidenta de la Red Internacional de Agencias de Aseguramiento de la Calidad de la Educación Superior.

La conferencia llevo el nombre de Aseguramiento de la Calidad ¿Rito o Compromiso? efectuada en Galería Nacional de Arte del Centro Costarricense de Ciencia y Cultura (Museo de los Niños). En esta actividad la distinguida Dra. Lemaitre hizo entrega del Certificado que permite acreditación al SINAES como una de las 4 agencias del mundo que después de haberse sometido a un proceso de evaluación externa, logra cumplir con los requisitos y parámetros de calidad mundial. (SINAES, 2010).

Muy concurrente con la calidad en desarrollo de software está el marco metodológico; el cual incorpora los procesos de calidad de software, en Costa

Rica el Instituto Tecnológico de Costa Rica, publica en la revista Tecnología en Marcha (Vol. 24, Nº. 3, 2011, págs. 38-45) un artículo en enero del 2011 titulado “Consideraciones sobre la administración de proyectos de desarrollo de software, con ciclo de vida ágil, en el contexto costarricense” en este artículo se afirma la adopción de metodologías ágiles durante el ciclo de vida de desarrollo en proyectos costarricenses y algunas interrogantes de las mismas, interrogante a las que se agrega el cómo se realiza un buen proceso de aseguramiento de calidad dentro de un marco metodológico, aplicando una metodología de pruebas que se homologue al proceso.

En el 2016, el periódico el FINANCIERO publica un artículo en el que se hace referencia a las metodologías ágiles, titulado: “**La filosofía ágil llegó a Costa Rica para quedarse**”; donde se menciona la consolidación de las metodologías ágiles en el mercado costarricense, a lo cual se suma los estándares modernos de calidad de software que vienen a incorporarse en el proceso.

Como último elemento a incorporar, se tiene el uso de herramientas que ayuden con la administración del ciclo de vida del desarrollo y herramientas de pruebas que han evolucionado y facilitan el seguimiento y hasta la automatización de las pruebas, permitiendo el crecimiento en los últimos años debido a las ventajas que presentan, estos procesos al incrementar y mejorar la eficiencia en la prestación de servicios que se manifiestan en el departamento de aseguramiento de la calidad del software, esto viene ayudar en la eficiencia y eficacia de los servicios brindados, mediante la mejora de los procesos establecidos.

1.2 Justificación

Desde inicios de la era Industrial, el trabajo se basó en el cumplimiento de algunos elementos, entre lo que destacó la calidad, no como se visualiza actualmente, sino como el control de un proceso de producción específico.

La calidad como propiedad o conjunto de propiedades inherentes a algo, que permiten juzgar su valor es la definición de La Real Academia Española. En su escrito el señor José Cruz Ramírez en su artículo “Historia de la Calidad” presentado en el sitio web http://docencia.fca.unam.mx/~lvaldes/cal_pdf/cal5.pdf menciona que “los primeros estudios sobre calidad se hicieron en Estados Unidos durante los años 30 antes de la Segunda guerra Mundial, la calidad no mejoró sustancialmente, pero se hicieron los primeros experimentos para lograr que ésta se elevara”. (Cruz Ramírez, 2015).

El autor José Cruz Ramírez en el mismo escrito, se transporta al año 1939, y menciona que “con el inicio de la segunda guerra mundial, el control estadístico de proceso presenció un gran cambio convirtiéndola en un arma secreta de la industria; fue así como los estudios industriales elevan la calidad, mediante métodos modernos, con esto se logró llevar a los norteamericanos a crear el primer sistema de aseguramiento de la calidad vigente en el mundo”. (Cruz Ramírez, 2015)

En el año de 1933 el Doctor W. A. Shward, de los Laboratorios Bell, logra aplicar por primera vez, el concepto de control estadístico de procesos con

fines industriales y cuyo objetivo es brindar el mejoramiento en materia de costo-beneficio de las líneas de producción, el resultado obtenido es uso de la estadística, presentándose de manera eficiente en el incremento de productividad y la disminución de errores, estableciendo el análisis específico del origen de las mermas¹, con la intención de elevar la productividad y la calidad, lo anterior es expuesto por José Cruz Ramírez en su escrito denominado "La historia de la calidad". (Cruz Ramírez, 2015).

Entre 1942 y 1945, Edwards Deming contribuyó en el mejoramiento de la calidad de la industria norteamericana dedicada a la guerra, elaborando un seguimiento estadístico de proceso, el cuál es un lenguaje matemático que los operadores pueden entender "**lo que las máquinas dicen**" (Cruz Ramírez, 2015), el señor Deming fue invitado por comandos militares de ocupación estadounidense a tierras japonesas, donde tendría desempeñaría un papel fundamental en cuanto al incremento de la calidad.

El señor Deming, a inicios del año 1947, presenció sus primeros contactos con ingenieros japoneses, en su trayectoria en los años 950, le fue invitado por el presidente de la Unión de Ingenieros Científicos Japoneses hay se dio a conocer como el padre de la calidad japonesa, dónde hizo uso en Japón de su modelo administrativo para el manejo de la calidad.

El ciclo Deming (de Edwards Deming), también conocido como círculo planificar, hacer, verificar, actuar (PDCA por sus siglas en inglés Plan, Do,

¹ Disminución o reducción del volumen o la cantidad de una cosa.

Check, Act) o espiral de mejora continua muy utilizado en las organizaciones, esto permite gestionar aspectos tales como ISO 9000 (referente a la calidad), ISO 14000 (referente al medio ambiente), OHSAS 18000 (referente a la salud y seguridad ocupacional) y ISO 22000 (referente al área alimentaria). (González, 2012)

Los resultados obtenidos en este ciclo permiten a las empresas nacionales que se dedican al desarrollo de software, la mejora integral de la competitividad, de los productos y de los servicios, presentando una mejora continua en la calidad, logrando reducir los costos, logrando optimizar la productividad, incrementando la participación y respuesta al mercado con el beneficio de rentabilidad para la organización. (González, 2012)

Orlando Jorge Franco Murillo y Evelio Martínez en su sitio web <https://www.eveliux.com/mx/El-origen-del-Software-Libre.html> indican que, “durante la primera generación de computadoras a finales de 1940, los programas se escribían de una manera en la que el concepto de software como independiente del hardware”. Estas eran épocas, dónde había carencia del licenciamiento de software, por lo que el software está a disposición de los usuarios y los programas podían ser intercambiados como si se tratase de tarjetas postales. Cuando apareció ARPANET (precursor del Internet) que fue durante la década de los 60 se comenzó la primera comunidad global que se basaba en los valores y principios del software libre. (Franco Murillo & Martínez, 2007).

El crecimiento del desarrollo de software y la complejidad de las aplicaciones provoca la crisis del software, esto hace referencia a los problemas presentados desde sus inicios donde el principal problema se ve reflejado por la baja eficiencia presente al momento de realizar un software. En el año 1968 se realizó la primer conferencia planeada por la OTAN (Organización del Tratado del Atlántico Norte). Friedrich L. Bauer habló por primera vez del conjunto de dificultades o errores ocurridos en la planificación, estimación de los costos, productividad y calidad de un software, o bien, lo que se conoce como la crisis del software, dicho término se le atribuyó a F. L. Bauer aunque ya había sido utilizado por Éger Dijkstra en su libro “El humilde programador”, lo anterior según Javier Garzas en su sitio web <https://www.javiergarzas.com/2013/01/dijkstra-programador-humilde.html>.

Dado a lo comentado en el párrafo anterior fue cuando se dio a conocer una nueva rama de ingeniería, misma que se denominó “La Ingeniería de software”). El Comité de ciencia de la OTAN, realizó la logística para la creación de dos conferencias, que exponen la ingeniería del software. Muchos de los integrantes creen que estas conferencias marcaron el inicio oficial de la profesión de la ingeniería de software.(Garzas, 2013)

La ingeniería de software fue estimulada por la llamada crisis del software de la década de 1960, 1970 y 1980, que identifica muchos de los problemas de desarrollo de software. Muchos proyectos de software sobrepasaron el presupuesto y el tiempo estimados, algunos

proyectos causaron daños a la propiedad y otros proyectos causaron pérdidas de vidas. (Rodríguez F, Clavijo, Chacón Torres, & Valero Pinzón, 2016).

Durante décadas, solucionar la crisis del software fue de suprema importancia para investigadores y empresas productoras de herramientas de software. El artículo científico sobre ingeniería del software presentado en 1986 por Fred Brooks con ocasión del X Congreso Mundial de Informática de la IFIP (Brooks, Proc. IFIP), titulado No hay balas de plata — Esencia y accidentes de la ingeniería del software (*No Silver Bullet — Essence and Accidents of Software Engineering*) menciona que; el costo de propiedad y mantenimiento del software en la década de 1980 fue dos veces más caro que el propio desarrollo del software. Durante la década de 1990, el costo de propiedad y mantenimiento aumentó en un 30% con respecto a la década anterior.

En 1995, las estadísticas mostraron la mitad de los proyectos de manera operacional como no exitosos. El proyecto de software medio sobrepasa su estimación en tiempo en el 50%. Las tres cuartas partes de los grandes productos de software son entregados al cliente con tales fallas que no son usados en absoluto, o no cumplen con los requerimientos del cliente.

Cada nueva tecnología y práctica de la década de 1970 a la de 1990 fue pregonada como una bala de plata para resolver la crisis del software, donde se patrocinan

- Herramientas

- Disciplina
- Métodos formales
- Proceso
- Profesionales

Los defensores de lenguajes como Ada (programación orientada a objetos), o de los procesos software continuaron apostando porque su tecnología sería la que solucionaría la crisis. Sin embargo, hubo gente que dudo el hecho de que no se encontrara una solución única y efectiva.

Si bien es cierto que la búsqueda de una única solución no funcionó, también había que ser consciente de que tampoco existían balas de plata en ninguna otra profesión.

El campo de la ingeniería del software es variado y complejo, como para pensar que una solución resuelva todos los problemas, de ahí nacen un conjunto de buenas prácticas que benefician la calidad de software.

Uno de los problemas presentes en la crisis del software, es la calidad del software la cual desde ese momento ha mantenido en una tendencia de desarrollo mediante nuevas técnicas y la inclusión de guías al proceso; dentro de los marcos de trabajo actual que se definen en el ciclo de vida de desarrollo, el proceso de fusión en muchos de los casos se personaliza al negocio que las implementa, esto debido a que las mismas no son definidas dentro de los marcos de trabajo.

Esta situación se da por la tendencia de utilizar guías, metodologías, herramientas y marcos de trabajo específicos; lo que dificulta la conformación de homologación de procesos entre empresas nacionales de desarrollo de software, debido a que según quien las implemente, va a dar una tendencia diferente en la definición del proceso y no va a mantener un estándar, el cual podrían seguir, teniendo la ventaja de que existen experiencias pasadas y hasta métricas ya establecidas, para poder definir si la aplicación al proceso es exitosa.

En la actualidad, una de las asociaciones que da normativas a los procesos de aseguramiento de la calidad del software, es *ISTQB* (*International Software Testing Qualifications Board* traducido al español Junta internacional de calificaciones para pruebas de software) (*ISTQB. Incorporated society*, 2016), la cual es una asociación sin ánimo de lucro fundada en 2002. Su sede se encuentra en Bélgica y cuenta con constitución, normas y reglamentos propios.

Esta junta *ISTQB* (Comité Internacional de Certificaciones de Pruebas de Software; siglas en inglés) grupo de expertos probadores de nivel internacional que trabajan de manera voluntaria, con más de 400.000 personas certificadas en más de 100 países, es la organización más importante del mundo en la certificación de software de testeo, lo cual ha logrado mediante su esquema de certificaciones “*ISTQB Certified Tester*”, convertirse en referencia para prácticas recomendadas de pruebas.

En su operación en los diferentes países o regiones, ISTQB cuenta con 50 comités que se encargan de manera local del prestigioso esquema de certificación. El comité miembro de ISTQB encargado de la región hispanoamericana es el *HASTQB (Hispanic America Software Testing Qualifications Board* traducido al español como Junta de calificaciones de pruebas de software de hispanoamérica).

ISTQB, crea los Programas y Glosarios (Syllaby and Glossary) (ISTQB. Incorporated society, 2016) que representan el cuerpo común de conocimiento compartido por los miembros en todo el mundo y conforma una guía del proceso de aseguramiento y las características más importantes que lo conforman, guías aplicadas por grandes corporaciones de nivel internacional, mediante miembros certificados alrededor del mundo.

También ISO (Organización Internacional de Estandarización) propone la normativa ISO/IEC 25000, conocida como *SQuaRE (System and Software Quality Requirements and Evaluation* traducido al español como requisitos y evaluación de calidad del sistema y del software), es una familia de normas que tiene por objeto la creación de un marco de trabajo común para la evaluación de la calidad del producto software.

La familia ISO/IEC 2500 durante la evolución el resultado final de normas, especialmente de las normas ISO/IEC 9126, que describe un modelo de calidad del producto software, e ISO/IEC 14598, que abordaba el proceso de evaluación de productos software.

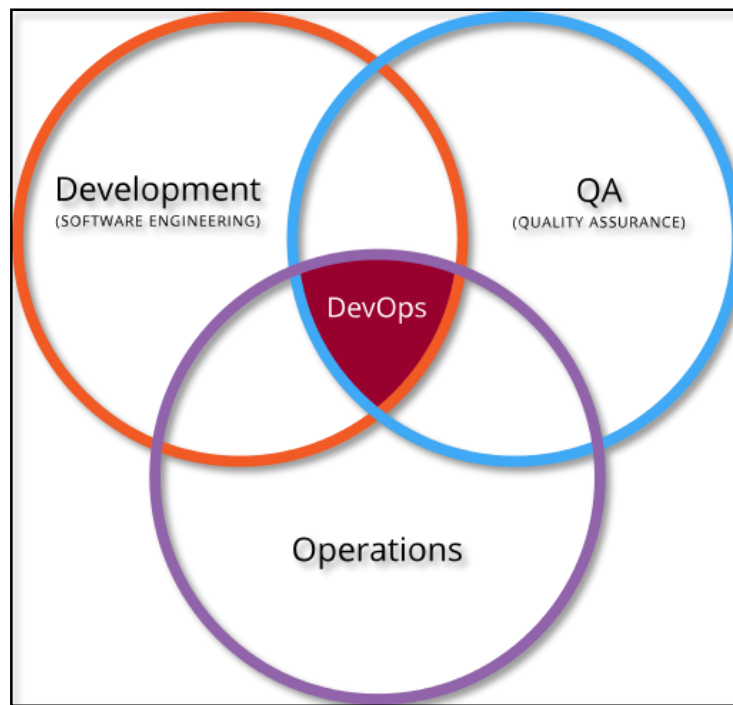
En este punto, la definición de calidad de software es difícil de formular, esto debido a que incluyen aspectos propios de políticas y estándares de los negocios, o se podría tener el caso de empresas estructuradas que tienen su propia definición de calidad para un producto o que el producto cuente con parámetros establecidos que los pueden medir.

Debido al efecto de la necesidad producida por la crisis del software y mediante el nacimiento de las balas de plata, bajo ese concepto donde se diseñan metodologías, procesos, guías y herramientas para la gestión del ciclo de desarrollo de software que han evolucionado, se crea la necesidad de conformar una guía que defina la manera de unir la metodología definida para el proyecto, la guía de aseguramiento a utilizar y las herramientas de gestión del ciclo de vida de las aplicaciones.

Esto con el fin de homologar la receta que se aplica y con ello tener puntos de referencia en mediciones que entre diferentes empresas nacionales de desarrollo de software pueden ser utilizadas para medir el rendimiento, esto debido a que al usar la misma receta para el proceso de ciclo de vida del desarrollo de software se pueden medir parámetros que van a servir como base para las mediciones y con ello lograr el máximo rendimiento de los procesos de desarrollo y poder formular un buen proceso de aseguramiento de la calidad del software.

Sin dejar de lado mecanismos de apoyo a todas esas mediciones como las herramientas de ALM (Application Lifecycle Management, traducido al

español gestión del ciclo de vida de la aplicación), las cuales se integran y ayudan desde la administración del proyecto, el desarrollo de software y procesos de aseguramiento de la calidad del software, conformadas mediante la estructura DevOps, (acrónimo inglés de development y operations traducido al español como desarrollo y operaciones) es un enfoque que promueve la colaboración entre líneas de negocio, desarrollo y operaciones de TI.



*Ilustración 1: Desarrollo del Software y Administración de Proyectos.
Fuente: Devops.svg. (2016, Setiembre 17).*

Es una funcionalidad empresarial que habilita la entrega continua, el despliegue continuo y la supervisión continua de aplicaciones, así como reducir el tiempo necesario para tratar la realimentación (*feedback*) de los clientes, el desarrollo y las operaciones, e incluso las pruebas.

1.3 Planteamiento del Problema

En la actualidad, la gestión del desarrollo de aplicaciones durante el ciclo de desarrollo de los proyectos informáticos, ha estado conformado por marcos de trabajo ágiles, metodológicos de aseguramiento y herramientas de ciclo de vida de desarrollo, utilizadas por diferentes empresas desarrolladoras de software; tomando una temática de personalización según los estándares de la empresa, lo cual origina, que las herramientas se personalicen sin mantener su esencia y que, en algunos casos, terminan personalizando buenas prácticas en sus proyectos o se aplican guías de pruebas de calidad de software para generar buenas prácticas pero las mismas se llevan en paralelo al proceso y no se logra una integración o un estándar, de ahí parte la necesidad de conformar una guía que defina los elementos de integración y la ejecución unificada del proceso.

Este documento, realizará la contextualización dentro de un departamento de desarrollo de software y definir las necesidades de integración dentro del uso de un marco de trabajo ágil y la selección de las herramientas que forman parte de la administración del ciclo de desarrollo y pueda incorporarse dentro de una metodología que gestione el proceso de aseguramiento, donde la ejecución involucre a coordinadores o jefes de departamento, desarrolladores, aseguradores de calidad y usuarios finales, entre otros actores, siempre dentro de empresas nacionales desarrolladoras de software.

El fracaso durante la implementación de un marco de trabajo ágil, se liga con la utilización incorrecta o personalización de los procesos, debido a que

quienes tienen la responsabilidad de utilizarla no logran la integración de manera efectiva en los proyectos.

Por lo anterior, se requiere su incorporación durante el desarrollo de sistemas, mediante la aplicación de buenas prácticas y recomendaciones, que permitan la ejecución adecuada de las técnicas aplicadas, evitando las causas de una inadecuada gestión de proyectos, en ámbitos como el control del avance del proyecto.

Algunos aspectos relevantes, implican el correcto uso de la documentación y la conformación de una guía de trabajo que involucre herramientas para el desarrollo y control, así como la integración de conceptos ágiles que conformen los pasos para una ejecución adecuada y se integren en una guía de aseguramiento de calidad del software, que permee el ciclo de vida del producto.

1.4 Pregunta Problema

Los procesos de calidad actuales han tomado un papel importante en todas las áreas de la ingeniería. En el caso de la ingeniería del software, se acogen procesos de calidad durante el desarrollo de software y con ello aparecen nuevas necesidades en los procesos y la definición de estos.

Dado este panorama, se plantea como base de esta investigación la siguiente pregunta:

¿De qué manera el conocimiento sobre las necesidades de un proceso de aseguramiento de calidad del software permitirá el establecimiento de los parámetros requeridos por una guía específica que mejore el proceso de desarrollo dentro de la calidad del software ajustado a un marco de trabajo ágil?

1.5 Hipótesis

En el análisis del aseguramiento de software, el conocimiento sobre el proceso de aseguramiento de la calidad del software va a permitir crear una guía que mejore el proceso actual, evaluando factores importantes encontrados en los marcos de trabajo ágil y herramientas de ciclo de vida de desarrollo que ayuden a encontrar los puntos importantes para integrar el proceso de desarrollo de aplicaciones y formular la propuesta que mejore el proceso de manera integrada.

La inclusión de herramientas que apoyen al proceso conformará un escenario más claro del trabajo, tomando en cuenta que las mismas son conformadas por herramientas de administración, que contienen gráficos que sirven para el seguimiento y permiten realizar evaluaciones de los procesos.

La guía para el aseguramiento de software debe estar integradas al menos a una herramienta para la administración del ciclo de vida de desarrollo de software y a herramientas para la gestión del proceso de pruebas y un marco de trabajo que ayude en la definición de los procesos durante el desarrollo de la aplicación; por ejemplo, un marco de trabajo ágil o una guía de pruebas que defina las pruebas apropiadas y ayude en la conformación del proceso de aseguramiento de software; todo ello debe integrarse y definirse dentro de la guía y va a conformar la receta que va a mejorar el proceso de aseguramiento de software, logrando definir los pasos a seguir para lograr la guía para el proceso.

Tomando en cuenta que las empresas nacionales de desarrollo de software apliquen la propuesta; cada desarrollo de aplicación logrará resultados que pueden ser medidos y evaluados, logrando con ello, tener parámetros que pueden ser definidos durante la evaluación y así poder evaluar los resultados y maximizar temas como tiempos y esfuerzos realizados por el equipo de desarrollo; además, se van a poder medir cantidad de errores y con ello evaluar la calidad de la aplicación que está en desarrollo.

Por lo anterior es que se plantea la hipótesis guía de la presente investigación de la siguiente manera:

El conocimiento sobre las necesidades de un proceso de aseguramiento de calidad del software permitirá el establecimiento de los fundamentos requeridos para la elaboración de una guía específica que mejore el proceso de desarrollo de software integrando los procesos de aseguramiento.

Además, la propuesta contendrá los aspectos que se deben tomar en cuenta para la selección de una herramienta que se integre al proceso de desarrollo de aplicaciones; así como la selección de la herramienta utilizada en el ámbito nacional con mayor integración al proceso de aseguramiento de software que se integre a un marco de trabajo ágil y que defina los pasos para comprender guías de aseguramiento de calidad de software.

1.6 Objetivos

1.6.1 Objetivo General

- Proponer una guía para el aseguramiento de la calidad de software que contemple el actuar integrado a las metodologías, marcos de trabajo y las herramientas de gestión del ciclo de vida de la aplicación, orientada a la industria de desarrollo del software nacional.

1.6.2 Objetivos Específicos

- Localizar factores técnicos e información detallada que involucra el proceso de aseguramiento de software, para la conformación de una base para el análisis del proceso aplicado por las empresas nacionales de desarrollo de software.
- Recopilar información sobre tendencias aplicadas en procesos de aseguramiento de software tomando en cuenta marcos de trabajo, metodologías ágiles y herramientas, mediante visitas a empresas nacionales de desarrollo de software, para con ello determinar los factores que influyen en el proceso.
- Fundamentar la propensión del aseguramiento de software utilizada en la actualidad por empresas nacionales de desarrollo de software, realizando investigaciones de tendencia y factores técnicos fuera del

ámbito nacional, para la definición de los procesos adoptados en la guía de aseguramiento de calidad del software.

- Detectar la herramienta de gestión de ciclo de vida, marco de trabajo ágil y metodología más apropiado para el proceso, tomando en cuenta los factores de mayor tendencia y aspectos de integración para su aplicación dentro del proceso de aseguramiento de software en empresas de desarrollo de software en el nivel nacional.
- Construir una guía metodológica para el proceso de aseguramiento de calidad del software, aplicando los conocimientos adquiridos durante la investigación y las visitas realizadas a empresas nacionales de desarrollo de software, que permitan la elaboración de una propuesta homogénea.

1.7 Alcance y Limitaciones de la Investigación

1.7.1 Alcance

Definir una guía que contenga los diferentes tipos de pruebas que se realizan durante el proceso de aseguramiento de software; aplicado por el área de desarrollo de aplicaciones en las empresas desarrolladoras de software en Costa Rica.

Esta guía debe contemplar una metodología de desarrollo de software basada en las metodologías (por ej. ágiles, cascada) y herramientas de gestión del ciclo de vida de aplicaciones que mejor se adapten a la definición del proceso, con el fin de proponer una integración que se adecúe y defina el proceso de aseguramiento de software.

1.7.2 Limitaciones

La investigación se limitará a establecer un marco de trabajo enfocado en aseguramiento de calidad del software y todos los procesos que se integren en el mismo.

Se realizará una selección entre múltiples normativas que definen el proceso del aseguramiento para establecer la más adecuada y que contenga las características importantes para realizar la integración con las herramientas de ALM y con la definición en el marco de trabajo de la metodología ágil.

Las herramientas que se definan para el proceso serán seleccionadas de acuerdo con la definición de aspectos que se integren con los patrones de aseguramiento apegado al entorno de trabajo (*framework*)² que sea definido.

La selección del marco de trabajo será definida por su popularidad en el ámbito de empresas de desarrollo de software en Costa Rica.

² Entorno de trabajo o marco de trabajo es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar.

1.8 Matriz de Congruencia

Tema	Titulo	Objetivo	Problema	Hipótesis
Aseguramiento de la calidad de software.	Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.	Proponer una guía para el aseguramiento de la calidad de software que contemple el actuar integrado a las metodologías, marcos de trabajo y las herramientas de gestión del ciclo de vida de la aplicación, orientada para la industria de desarrollo del software nacional.	¿Cuál es la necesidad actual de crear una guía de aseguramiento de la calidad del software que establezca un proceso estructurado?	Diseñar una guía que oriente a las empresas desarrolladoras de software costarricenses en la implementación de procesos de aseguramiento de la calidad del software que homologue y mejore los procesos de desarrollo, integrando los diferentes factores que lo conforman como lo son las herramientas de ciclo de vida y herramientas de aseguramiento de la calidad de software, metodología de aseguramiento y marco ágil de trabajo, para lograr que las empresas costarricenses implementen procesos de aseguramiento que

Tema	Titulo	Objetivo	Problema	Hipótesis
		<p>Localizar factores técnicos e información detallada que se involucra el proceso de aseguramiento de software para conformar una base para el análisis del proceso aplicado por las empresas nacionales de desarrollo de software.</p>	<p>Con base en qué se define el proceso de aseguramiento de la calidad de software</p>	<p>les ayuden a mejorar la calidad de los productos desarrollados en busca de la mejora continua.</p> <p>Recopilar información para poder crear el conocimiento con el que se planteará el proceso que homologue a las diferentes empresas nacionales dedicadas al desarrollo de software, para con ello poder definir las necesidades actuales y solventarlas planteando un proceso completo que ayude a la implementación de la disciplina de aseguramiento de la calidad de software, la investigación busca determinar cuáles son los principales factores que deben ser contemplados en el montaje de la propuesta de implementación para los procesos de</p>

Tema	Titulo	Objetivo	Problema	Hipótesis
		<p>Recopilar información sobre la tendencia actual aplicada a procesos de aseguramiento de software tomando en cuenta Marcos de trabajo, Metodologías ágiles y herramientas, mediante visitas a empresas nacionales de desarrollo de software, para con ello determinar los factores que pueden influir en el proceso.</p>		<p>aseguramiento de la calidad de software. Se utilizará la técnica de recopilación e información basadas en encuestas.</p> <p>Mediante visitas a empresas nacionales dedicadas a desarrollar software se completarán entrevistas a los trabajadores para con ello conocer la tendencia y cada uno de los factores que se integran en el proceso actual de aseguramiento de la calidad de software, logrando medir el cambio que se debe realizar y al lograr generar una propuesta que marque el camino a seguir, conformando una buena aplicación de la técnica de aseguramiento.</p>

Tema	Titulo	Objetivo	Problema	Hipótesis
		<p>Fundamentar la tendencia de aseguramiento de software utilizada en la actualidad por empresas nacionales de desarrollo de software, realizando investigaciones de tendencias y factores técnicos fuera del ámbito nacional, para definir los procesos adoptados en la guía de aseguramiento de calidad del software.</p>	<p>¿Qué define la guía de aseguramiento de la calidad de software y como lo define?</p>	<p>Generar una investigación que ayude a definir los diferentes aspectos que conforman el proceso de desarrollo y que integran con los procesos de aseguramiento de la calidad de software, que permita conocer las tendencias actuales y con ello poder definir la necesidad que existe en el mercado determinando la importancia de aplicar este tipo de procesos y definir la mejor manera de trabajarlos bajo una buena técnica para documentar la tendencia y utilizarla como guía para conocer los pasos a seguir de las empresas para la implementación de los procesos de aseguramiento de la calidad que den mayor aporte a poder identificar la ruta a seguir para la</p>

Tema	Titulo	Objetivo	Problema	Hipótesis
				implementación. Se utilizarán las técnicas de recopilación e información de encuestas y entrevistas, así como búsqueda de información mediante sitios web confiables.
		<p>Detectar la herramienta de gestión de ciclo de vida, marco de trabajo ágil y metodología más apropiado para el proceso, tomando en cuenta los factores de mayor tendencia y aspectos de integración para aplicar en el proceso de aseguramiento de software en empresas de desarrollo de software en el nivel nacional.</p>	<p>¿Definir herramientas de software o documentales propias para aseguramiento de la calidad que se integren al proceso con metodologías de aseguramiento y marcos de trabajo ágil?</p>	<p>Poder identificar cuáles son las herramientas, metodologías de aseguramiento y el marco de trabajo que más son utilizados por las empresas nacionales y que se ajusten e integren al planteamiento del proceso de aseguramiento que se determine como más ajustado a la buena práctica, para formar una guía que defina cuál es la mejor manera de implementación al proceso completo de desarrollo de software. Se utilizará la técnica de recopilación e información</p>

Tema	Titulo	Objetivo	Problema	Hipótesis
		<p>Construir una guía metodológica para el proceso de aseguramiento de calidad del software, aplicando los conocimientos adquiridos durante la investigación y las visitas realizadas a empresas nacionales de desarrollo de software, que permitan la elaboración de una propuesta homogénea.</p>	<p>¿Crear una guía que permita que las empresas de desarrollo de software nacionales homologuen el proceso de aseguramiento de software?</p>	<p>mediante la búsqueda de información confiable.</p> <p>Conformar una guía que ayude a las empresas a conocer la ruta a seguir en la integración de un proceso de aseguramiento de la calidad que tome en cuenta aspectos como la aplicación de metodologías de aseguramiento de calidad del software, marcos ágiles de trabajo y herramienta para administración de ciclo de vida de desarrollo de software. Para poder dar un instrumento que dé como beneficio facilitar la buena utilización de cada una de las técnicas, esto sujeto a la definición del uso de las herramientas para los procesos de aseguramiento de calidad del software, que ayudan a generar un cambio en la tendencia que</p>

Tema	Titulo	Objetivo	Problema	Hipótesis
				ayude a las empresas a estructurar procesos más maduros de desarrollo. Se utilizará la técnica de recopilación e información de encuestas y métodos de búsquedas de información confiable.

CAPÍTULO II MARCO TEÓRICO

2.1 Marco Contextual

2.1.1. Desarrollo de Software

Las funciones de desarrollar software son adoptadas por el programador o una compañía orientada a comercializar servicios informáticos. El cual más ampliamente se orienta a la programación algorítmica.

En informática, un desarrollador o analista-programador, es un especialista capaz de diseñar y crear sistemas informáticos, así como su implementación y etapa de ajuste, utilizando uno o varios lenguajes de programación. Información consultada en el sitio <http://www.capacita-tech.com/Desarrollo.html>

Cuando se va desarrollar un software interactúan diversos actores, el cliente es uno de ellos quien tiene el problema y desea que se solucione, por lo que acude el analista de sistema encargado de entender sus requerimientos y necesidades y elevarlos a los programadores, encargados de realizar el diseño del sistema y la codificación, para que después sea probado e instalado donde el cliente. Es así como intervienen múltiples interesados, ya que una sola persona no determinaría lo necesario, lo más probable es que le haga falta algún requerimiento y entre más involucradas mayor definición, de los requerimientos del sistema.

2.1.2. Fases del Proceso de Desarrollo de Software

Por interés de la investigación, se va a analizar las fases del proceso de desarrollo una vez que el proyecto tiene claro el alcance y es iniciado por los interesados. Según el sitio web https://www.ecured.cu/Desarrollo_de_software del autor Artur Boronat, en su artículo “Desarrollo de Software” indica las fases del proceso de un desarrollo de software dónde destaca que las mismas son:

A. Análisis de requerimientos

La primera etapa para crear un software es definir los requerimientos. El cliente define lo que el software hará requiere de habilidades y experiencia en la ingeniería de software para identificar requisitos mal diseñados, contradictorios, incompletos o ambiguos. El levantamiento de requisitos con el interesado se plasma en el documento ERS (Especificación de requerimientos del sistema) cuya estructura puede ser definida por diversos estándares, un ejemplo CMM-I.

También se realiza la definición del diagrama de Entidad/Relación, en el cual se definen las principales entidades que contribuyen en el desarrollo del software. Una parte crucial del proceso es la toma, análisis y especificación de requisitos, así como la definición de pruebas; Los objetivos finales dependen de esta etapa del proceso. Para realizar la etapa de levantamiento de sistemas se han creado diversos procesos de trabajo incluso modelos de trabajo, hasta el punto no de una manera formal se habla de la Ingeniería de requisitos. La IEEE Std. 830-1998 normaliza la creación de las especificaciones de requisitos software (*Software Requirements Specification*).

B. Diseño y arquitectura

La arquitectura de software determinar el funcionamiento de forma general, incorporar la implementación tecnológica, como el hardware, la red, etc. Para completar las funciones definidas en el sistema se definen los casos de uso y se crean las clases de diseño con las entidades definidas en el análisis de requisitos, obteniendo un modelo cercano a la programación orientada a objetos.

C. Programación

Crear el diseño de código, el cual según los lenguajes de programación seleccionados definirán la complejidad y por ende la duración de esta etapa.

D. Pruebas

Es la definición de etapas y tareas orientadas a la comprobación del software, algunas de estas pruebas son especificadas en la documentación de requerimientos, es buena práctica que las pruebas sean realizadas por una persona diferente al desarrollador, tomando en cuenta que el desarrollador debe realizar sus pruebas, importante se cuente con un área de pruebas con roles específicos a la gestión de pruebas. Se realiza la consulta de información en el

E. Documentación

La documentación de la gestión del proyecto y documentos desarrollados para el desarrollo, como modelaciones (UML), manuales técnicos, Manuales de usuario, diagramas, pruebas, etc.; toda la documentación con el propósito de ser base de conocimiento de eventuales correcciones, futuros mantenimiento, usabilidad, etc.

F. Mantenimiento

EL proceso de mantenimiento y creación de mejoras el software enfrenta errores provocados por nuevos requisitos. El mantenimiento es más complejo y

puede durar más tiempo en comparación con el desarrollo inicial. Una parte de los mantenimientos consiste en arreglar errores, por otra parte, con una porción más grande del trabajo la creación de nuevas funcionalidades o mejoras de las actuales.

G. Metodológicas

El desarrollo de software es complejo por ello existen propuestas metodológicas que inciden en las fases del proceso de desarrollo. Donde las tradicionales se basan en controlar el proceso, estableciendo actividades, herramientas y notaciones utilizadas de manera rigurosa.

Las metodologías han tenido éxito en unos proyectos y presentando problemas en otros, por lo que surgen mitologías centradas en el factor humano y producto; filosofía de las metodologías ágiles, dando mayor enfoque al trabajo colaborativo y el desarrollo iterativo. Enfoque de éxito en requerimientos que sufren constantes cambios, reduciendo tiempo sin sacrificar la calidad.

La tendencia ágil renueva la forma de desarrollar software y genera debate en quienes utilizan metodologías tradicionales.

Los ambientes de desarrollo colaborativo incluso en miembros de equipo con diferentes lugares físicos, la tendencia de lluvia de ideas, el debate, compartir ideas o conocimiento, el trabajo en conjunto para desarrollar tareas.

La colaboración es esencial para cada rama de la ingeniería, pero concretamente, se va a hablar de ingenieros de software en sus tareas de diseñar la aplicación, implementación y mantenimiento de software de alta calidad haciendo uso de la Internet como base para sus interacciones sin importar que estén físicamente separados. (Pérez, 2015).

2.1.3. Proceso para el Desarrollo de Software

La estructura aplicada al desarrollo de un producto de software en el proceso de desarrollo de software o también llamado ciclo de vida de desarrollo de software. Existen modelos a seguir para la definición de los procesos para el desarrollo de software, los cuales definen según el modelo elegido va a tener un diferente enfoque y diferentes actividades a lo largo del proceso. El modelo de ciclo de vida para algunos autores es visto como término generalizado para definir el proceso desarrollo de software.

2.1.3.1 Metodologías Ágiles

Las metodologías ágiles son iterativas e incrementales, que se desarrollan de manera secuencial. (Díaz Agudelo & Olaya Figueroa, 2010).

Cuando se llevan a cabo las iteraciones, se realizan diferentes actividades de desarrollo de software de forma secuencial para lograr a la entrega (reléase), un ejemplo de las actividades es:

Diseño

Implementación

Pruebas

La entrega al final de una iteración (*iteration release*), está orientado a entregar un módulo funcional (conjunto de requerimientos) del sistema final, junto

con la documentación correspondiente desarrollada durante las tareas de desarrollo. También es importante, al iniciar una iteración, se trabaja partiendo de la entrega de la iteración anterior.

Las metodologías ágiles utilizan los fundamentos iterativos, evolutivo lo que implica que los requerimientos, las estimaciones, los planes y las soluciones evolucionan a lo largo de las iteraciones.

Por otra parte, el otro fundamento es el desarrollo adaptativo de acuerdo con la retroalimentación brindada por involucrados en el proceso. Desarrollo adaptativo es similar en concepto al del desarrollo iterativo evolutivo, con la excepción de que el término retroalimentación tiene mayor peso en la evolución (Díaz Agudelo & Olaya Figueroa, 2010), lo anteriores según el sitio web academia.edu en su artículo guía metodológica constru-colectiva.

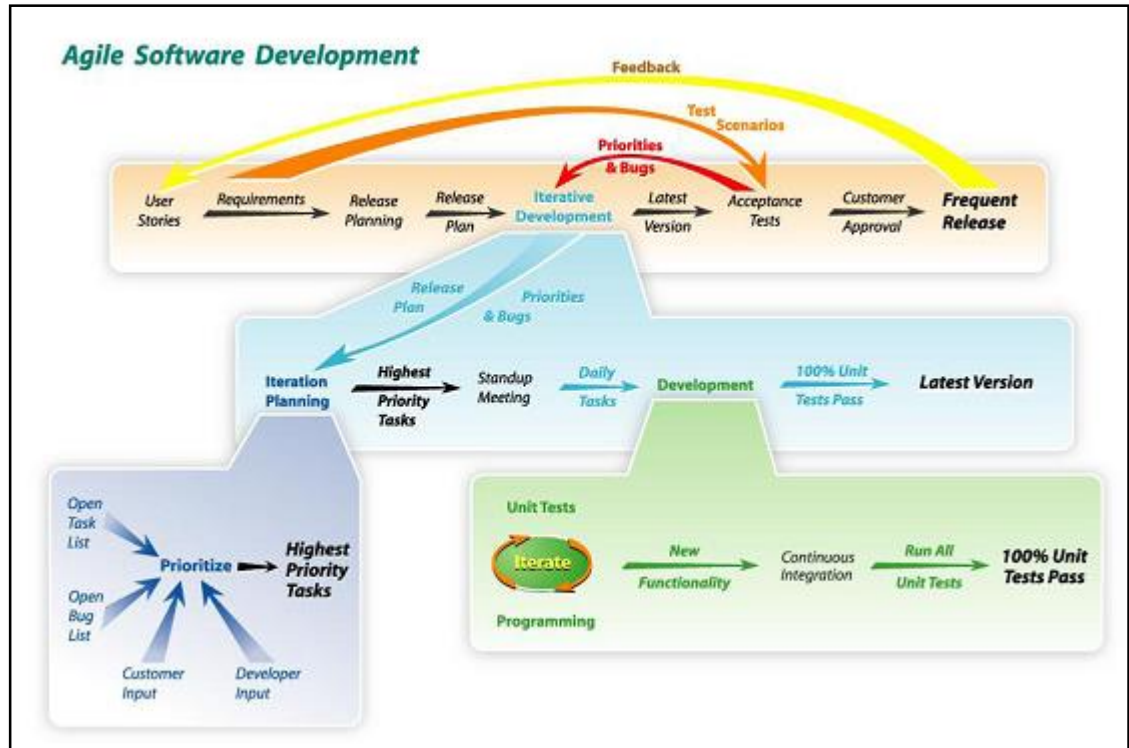


Ilustración 2: *Ágile Software Development*

2.1.3.2. Manifiesto ágil

En febrero de 2001 se creó el manifiesto ágil aplicado al desarrollo de software (Díaz Agudelo & Olaya Figueroa, 2010). Su objetivo fue definir los valores y principios de desarrollo ágil, tomando en cuenta los cambios que se puedan presentar el proyecto (Canos, Letelier, & Penadés, 2003).

El descubrimiento de nuevos métodos de desarrollo de software al hacer y ayudar a otros a hacerlo, donde valoramos de la siguiente lista los elementos de la izquierda más que los de la derecha:

Individuos e interacciones	Procesos y Herramientas
Trabajo en desarrollo del software	Documentación completa
Colaboración del cliente	Negociación según contratos
Respuesta al cambio	Seguir un plan

Los principios del manifiesto ágil se fundamentan los principios donde describir el espíritu ágil (principios número 1 y 2), luego se describen detalles de los procesos (principios: 3 - 8) y se finaliza describiendo al equipo de trabajo (principios 9 - 12). (Díaz Agudelo & Olaya Figueroa, 2010).

Según el artículo escrito por John Edde Díaz y Juan Felipe Olaya Figueroa en su Guía Metodológica para la gestión de proyectos de software, determinan que los principios del manifiesto ágil son los siguientes:

1. **La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.** Un proceso es ágil si a las pocas semanas de empezar ya entrega software que funcione, aunque sea rudimentario. El cliente decide si pone en marcha dicho software con la funcionalidad que ahora le proporciona o simplemente lo revisa e informa de posibles cambios a realizar.
2. **Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.** Este principio es una actitud que deben adoptar los miembros del equipo de desarrollo. Los cambios en los

requisitos deben verse como algo positivo. Les va a permitir aprender más, a la vez que logran una mayor satisfacción del cliente. Este principio implica además que la estructura del software debe ser flexible para poder incorporar los cambios sin demasiado coste añadido. El paradigma orientado a objetos puede ayudar a conseguir esta flexibilidad.

3. **Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo posible entre entregas.** Las entregas al cliente se insisten en que sean software, no planificaciones, ni documentación de análisis o de diseño.
4. **La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.** El proceso de desarrollo necesita ser guiado por el cliente, por lo que la interacción con el equipo es muy frecuente.
5. **Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.** La gente es el principal factor de éxito, todo los demás (proceso, entorno, gestión, etc.) queda en segundo plano. Si cualquiera de ellos tiene un efecto negativo sobre los individuos debe ser cambiado.
6. **El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.** Los miembros de equipo deben hablar entre ellos, éste es el principal modo

de comunicación. Se pueden crear documentos, pero no todo estará en ellos, no es lo que el equipo espera.

7. **El software que funciona es la medida principal de progreso.** El estado de un proyecto no viene dado por la documentación generada o la fase en la que se encuentre, sino por el código generado y en funcionamiento. Por ejemplo, un proyecto se encuentra al 50% si el 50% de los requisitos ya están en funcionamiento.
8. **Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.** No se trata de desarrollar lo más rápido posible, sino de mantener el ritmo de desarrollo durante toda la duración del proyecto, asegurando en todo momento que la calidad de lo producido es máxima.
9. **La atención continua a la calidad técnica y al buen diseño mejora la agilidad.** Producir código claro y robusto es la clave para avanzar más rápidamente en el proyecto.
10. **La simplicidad es esencial.** Tomar los caminos más simples que sean consistentes con los objetivos perseguidos. Si el código producido es simple y de alta calidad será más sencillo adaptarlo a los cambios que puedan surgir.

11. Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos. Todo el equipo es informado de las responsabilidades y éstas recaen sobre todos sus miembros. Es el propio equipo el que decide la mejor forma de organizarse, de acuerdo con los objetivos que se persigan.

12. En intervalos regulares, el equipo reflexiona respecto de cómo llegar a ser más efectivo, y según esto ajusta su comportamiento. Puesto que el entorno está cambiando continuamente, el equipo también debe ajustarse al nuevo escenario de forma continua. Puede cambiar su organización, sus reglas, sus convenciones, sus relaciones, etc., para seguir siendo ágil. (Díaz Agudelo & Olaya Figueroa, 2010).

2.1.3.3. Metodologías Ágiles en el Entorno Mundial

El concepto de metodología ágil de fortalecer después de una reunión realizada en febrero del 2001 en Utah-EEUU, 17 expertos en software, analizan y descubren como dar mayor velocidad al desarrollo de software (Canos, Letelier, & Penadés, 2003) y que a la vez cumpla la expectativa del cliente, siendo flexible al cambio contante durante la construcción del producto y que es centrado en desarrollar software funcional que contenga la documentación necesaria (Canos, Letelier, & Penadés, 2003), constituyendo uno de los principios descritos en el manifiesto de la alianza creada por este grupo. (The Agile Alliance, 2014).

El poder de la negociación es uno de los factores que destacan a las empresas innovadoras y competitivas, según la teoría de las cinco fuerzas de Porter. (Riquelme Leiva, 2015).

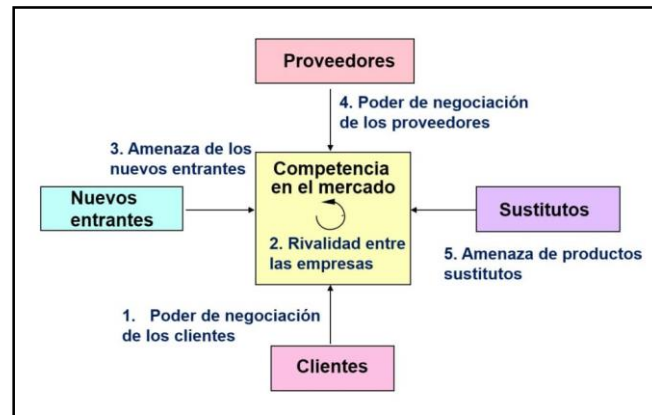


Ilustración 3: Las 5 fuerzas de Porter.

Durante la historia, se pueden ver diferentes modelos que se han creado para el desarrollo de software, entre ellos:

2.1.3.3.1. Metodología Cascada

En 1970 se planteó el **Modelo Cascada** donde se contempla las etapas de: Análisis, Diseño, Codificación, Prueba, Implementación y Mantenimiento (Aparicio, 2012), uno de los problemas que presenta es la falta de comunicación con el cliente, se realiza toda la planeación y desarrollo completo y los errores se detectan en etapas tardías de pruebas, lo cual involucra tiempo debido a que debemos volver al inicio para realizar reparaciones que pueden comprometer grandes partes del desarrollo. El este marco metodológico no se repiten actividades.

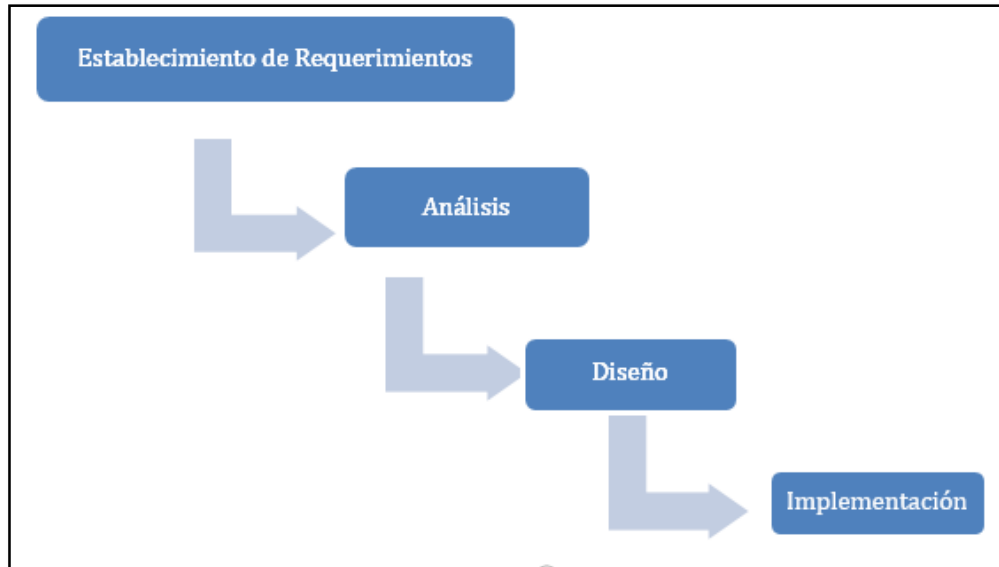


Ilustración 4: Modelo Cascada (Cortés Morales, 2006).

2.1.3.3.2. Metodología Espiral

En 1987 se presenta el **Modelo Espiral**, el modelo agrega la etapa de análisis de los riesgos como una novedad del modelo, realizando ajustes continuos al producto final, el modelo involucra mucho tiempo mayor a los deseados por el cliente.

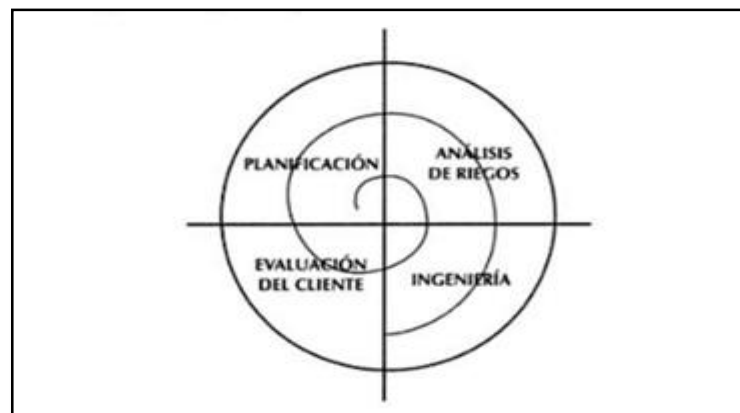


Ilustración 5: Modelo Espiral (Cortés Morales, 2006).

2.1.3.3.3. Programación Extrema (*Extreme Programming*)

Otro modelo de programación propuesto es programación extrema (Extreme Programming “XP”), donde enfoca el marco ágil a la satisfacción del cliente y reducción de costos para atender un cambio, definiendo cuatro variables: tiempo, costo, calidad y alcance, define una estrecha relación con el cliente final en todo el proceso, lo cual hace que el producto final cumpla con todos los requerimientos del cliente; ello puede crear altos tiempos de inversión en desarrollo y problemas para trabajar en equipo, ya que el cliente puede generar retrasos en el proceso.

Conociendo los modelos de desarrollo, describiremos las metodologías de desarrollo ágil que se crearon.

2.1.3.3.4. Metodologías de Crystal (*Crystal Methodologies*)

Elaborada por Alistair Cockburn en 1998 (Garzás, 2012). Metodologías para el desarrollo que se caracterizan por el trabajo en equipo y la reduce el número de artefactos producidos.

Basado en la cooperación y comunicación, limitado por los recursos. El equipo de desarrollo es la clave para definir el esfuerzo y orientar la mejora de habilidades y destrezas, otro factor importante es la definición de políticas para orientar el trabajo en equipo, donde las políticas variaran del tamaño del equipo donde se establece una definición de colores, (Canos, Letelier, & Penadés, 2003), ejemplo:

- Crystal Clear y Orange

2.1.3.3.5. Métodos de desarrollo de Sistemas Dinámicos

(Dynamic Systems Development Methods (DSDM))

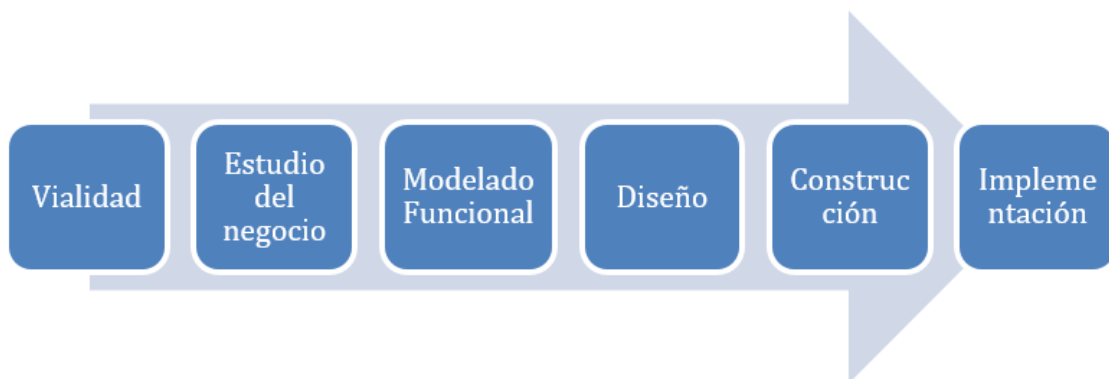
Nace en 1994 con el fin de crear una metodología RAD unificada. Sus principales características son:

Proceso interactivo e incremental

Trabajo en equipo

Trabajar en cooperación con el usuario final

Propone cinco fases:



Las tres últimas son iterativas, además existe realimentación a todas las fases.

(Canos, Letelier, & Penadés, 2003).

2.1.3.3.6. Desarrollo de Software Adaptativo

(Adaptive Software Development (ASD))

Es un desarrollo impulsado por Jim Highsmith. Sus principales características son:

Iterativo

Orientación a componentes de software más que a las tareas realizadas

Anuente a los cambios

Propone un ciclo de vida definido en tres fases esenciales:

- Especulación, colaboración y Aprendizaje

La primera etapa es de inicialización, se realiza la planifican de las características del software; En la segunda etapa de detallan las características y en la tercera etapa de revisa la calidad y entrega al cliente. (Canos, Letelier, & Penadés, 2003)

2.1.3.3.7. RUP³ (“The Rational Unified Process”)

Metodología desarrollada por Rational Corporation (compañía que pertenece a IBM). Enmarca un enfoque de disciplina para asignar tareas y responsabilidades en el proceso y la corporación. Su objetivo es garantizar la calidad del producto satisfaciendo las necesidades del cliente, dentro de la planeación de cronogramas y presupuestos elaborados. (Rational Software, S.F.)

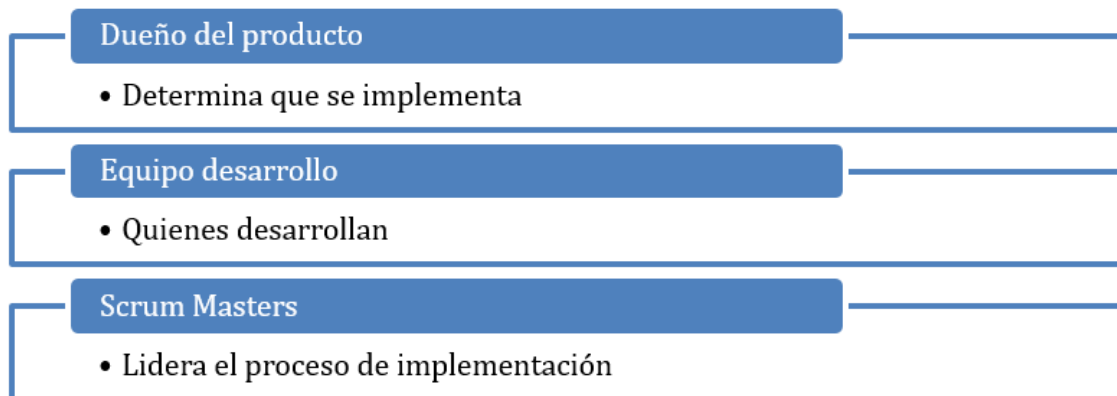
³ El proceso unificado de Rational

2.1.3.3.8. Metodología SCRUM

Es de las más populares sobresale la colaboración con el cliente, entregas continuas e incrementales y la adaptación al cambio, otro factor es la mitigación del riesgo que aparecen en el proceso de desarrollo, lo cual es una ventaja de la implementación de SCRUM, puesto que el enfoque de cambio y entrega rápida es la razón por la que muchas empresas seleccionan esta metodología como tema a investigar e implementar.

Scrum manera ágil de gestionar un proyecto de software (Pérez, 2015), en donde el producto final, se divide en pequeñas partes para poder ser construido forma incremental, donde cada equipo centralizar su atención en una parte.

Para poder trabajar esta metodología es necesario tener al menos 3 roles dentro del equipo:



Al implementar se debe tomar en cuenta; la cultura organizacional, ya se fortalece el trabajo en equipo y la comunicación; además, establece equipos pequeños entre 5 y 9 personas, (Albaladejo, 01), por otra parte, se debe de mantener una buena comunicación con el cliente y con el o los proveedores.

Se compone de 3 fases:

- La primera es donde se realiza el plan de lo que se va a desarrollar, estimación de tiempo, alcance y el diseño de la implementación.
- La segunda es el desarrollo de cada una de las tareas definidas para lograr el producto.
- La tercera es la preparación del producto y evaluación de pruebas (Scrum Manager, S.F.).

En la primera etapa de planeación se realiza una reunión donde se determinan las necesidades del cliente, sus prioridades, y se definen los requerimientos a trabajar (Barrios, y otros, 2011).

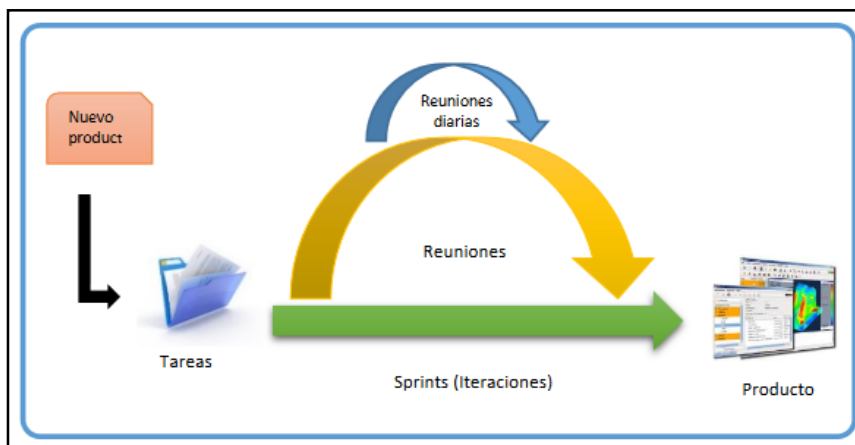


Ilustración 6: Etapa de Planeación (Sprint)

La etapa de desarrollo del software se realizan los *sprint*, esto representa el tiempo a desarrollar los entregables definidos del producto final, se establecen de un mes o menos y deben ser secuencial (Pérez, 2015).

Como método de seguimiento de las iteraciones se define reuniones diarias, la duración es de un máximo de 15 minutos y se realiza el seguimiento de las tareas en ejecución de cada integrante del equipo. Al finalizar el *sprint* se realiza la reunión de presentación del producto con una duración máxima de 4 horas, entrega oficial al cliente, los asistentes son al equipo, el cliente, el gerente y los usuarios; finalmente se desarrollan reuniones de retrospectiva en un proceso de madures del proceso iterativo de desarrollo. (Barrios, y otros, 2011).

Scrum se basa en el Manifiesto para un Desarrollo de Software Ágil, filosofía da mayor valor al individuo, la colaboración con el cliente y al desarrollo iterativo incremental del software basada en un enfoque que muestra efectividad en proyectos al trabajar requisitos cambiantes, logrando la reducción de tiempos de desarrollo, manteniendo la calidad. (Sánchez Freelancer, 2015).

Es importante mencionar los artefactos de *Scrum*:

A. Lista del producto (“*Product Backlog*”)

Lista de productos priorizada (Ticona Yanqui, 2014)

B. Lista del producto en el Sprint (“*Sprint Backlog*”)

Lista de necesidades que conforman un entregable priorizadas y con una definición de tareas para lograrlo (Ticona Yanqui, 2014).

C. Gráfico de avance (“*Burndown Chart*”)

Avance del Sprint mediante una gráfica (Ticona Yanqui, 2014).

D. Gráfico de requerimientos pendientes (“*Sprint Burn Down*”)

Muestra los requerimientos pendientes a trabajar, definidos en las tareas pendientes (*Sprint Backlog*), informe de culminar con éxito el sprint (Tomaselli, Acuña, Estayno, & Lenkovich, 2013).

E. Gráfica de evolución del producto (*Sprint Burn Up*)

Genera un gráfico que permite medir la evolución del producto. El seguimiento se realiza a las tareas mediante los workflow de las tareas:



2.1.3.4. Aseguramiento de la calidad (testeo de pruebas)

Según el sitio web de ISTQB en su página oficial <https://www.istqb.org/> se pueden determinar las siguientes definiciones determinadas al aseguramiento de la calidad.

2.1.3.4.1. Definición de aseguramiento de la calidad

Conjunto de acciones planificadas y sistemáticas, implementadas en el Sistema de Calidad, que son necesarias para proporcionar la confianza adecuada de que un producto satisfará los requisitos dados sobre la calidad.

2.1.3.4.2. Definición de Testeo de Pruebas (“Testing”)

El testeo es la actividad desarrollada para realizar la revisión y evaluar la calidad del producto, con el fin de mejorarlo detectando defectos y problemas que se podrían presentar.

2.1.3.4.3. Principios del Testeo de Pruebas

Los siete principios que establecen pautas generales comunes a todas las pruebas.

A. Pruebas muestran la presencia de errores.

Las pruebas evidencian que hay defectos y reducen la probabilidad de tener defectos, pero no pueden probar que no los hay.

B. Pruebas exhaustivas son imposibles.

Probar todo es imposible, salvo en casos triviales. Se deben analizar los riesgos y prioridades para centralizar el esfuerzo, en lugar de pretender hacer pruebas exhaustivas.

C. Pruebas tempranas.

Las actividades de pruebas se iniciarán lo antes posible para identificar errores de manera temprana, centrándose en objetivos definidos.

D. Agrupación de defectos.

Las pruebas deben centrarse de manera proporcional en la densidad esperada. Con normalidad la mayor parte de los defectos se concentran en un número reducido de módulos.

E. La paradoja de los plaguicidas.

Si repetimos las pruebas, la misma serie de casos de prueba dejarán de encontrar más defectos.

F. Pruebas dependen del contexto.

Las pruebas se llevarán a cabo según el contexto, la evaluación tomará en cuenta aspectos importantes del entorno como factor influyente de las pruebas.

G. Falacia de la ausencia de errores.

Si la construcción no es estable, la detección y corrección de defectos no servirá de nada, se debe de cumplir con las expectativas del usuario.

2.1.3.4.4. Fundamentos del Testeo de Pruebas

A. Disparadores de Errores

- a. Tiempo
- b. Sistemas complejos
- c. Cambios organizacionales

B. Errores en Especificaciones

- a. Diseños basados en especificaciones defectuosos
- b. Construcción sobre aquellos diseños defectuosos
- c. Sistemas mal elaborados
- d. Inicio temprano de pruebas
- e. Las pruebas deben repetirse hasta que el producto esté completamente aprobado.

2.1.3.4.5. Fundamentos de Pruebas

Los fallos de software pueden llevar a:

- Perder dinero, tiempo, reputación de la organización, contaminación ambiental o lesiones graves.

Esto debido a que:

- Realización de pruebas insuficientes, aplicación de pruebas incorrectas, escasez y mejora en escenarios para elaboración de pruebas.

Los sistemas grandes y complejos son imposibles de probar de manera exhaustiva.

- **Los riesgos**

Son parte de todo desarrollo de software, se detallan clasificaciones donde a mayor es el riesgo, mayor debe ser el nivel y detalle de pruebas. Además, se debe tomar en cuenta que entre más complejo es el sistema, existe una mayor probabilidad de fallo.

- **La calidad**

Define que un sistema cumpla con las necesidades referenciadas en historias de usuario que son solicitadas por los interesados, aunque también existen fallos por incumplimiento no funcionales, como la lentitud o la denegación del servicio por causa de muchas peticiones, el tiempo de respuesta puede ser un factor de calidad.

Decidir cuándo ya es suficiente

- Definir cuántas pruebas son suficientes, sabemos que no podemos probar todo.
- Todo sistema tiene un riesgo que determinará hasta dónde llegar.
- Es necesario priorizar los posibles escenarios de prueba y asegurar que las pruebas más importantes se realizaron.
- Se deben definir criterios de aceptación por parte de las partes interesadas, quienes deben indicar el nivel de tolerancia a los defectos.
- **Depuración (*Debugging*)**

Proceso en el cual los desarrolladores identifican las causas de errores o defectos de código y realizar correcciones. Es importante la depuración antes de las pruebas para elevar el nivel de calidad del componente o sistema.

- **Análisis de testing y documentación de pruebas**

Si bien el analista de calidad tiene entre sus características analizar los requerimientos y funcionalidades del software, los tester entre sus tareas también incluyen el análisis de la documentación. Donde parte de las tareas se centran en:

- Entendimiento al cliente y sus necesidades
- Conocimiento funcional del producto

- Determinación correcta de los requerimientos
- Definición de las pruebas y sus respectivos escenarios

Durante el proceso de desarrollo y ejecución de las pruebas, el tester puede detectar y comunicar requerimientos que no están claramente definidos o ambiguos.

Parte de la documentación de pruebas debemos de tener la claridad de documentarlos casos de pruebas que se desarrollan y los set de datos que conforman los scripts de pruebas para formar los escenarios, esta documentación de caso de prueba debe estar referenciada a resultados de pruebas durante la ejecución, los mismos van a guardar la evidencia necesaria para poder respaldar la ejecución de pruebas.

2.1.4. Herramientas ALM

ALM es la sigla para *Application Lifecycle Management*, o Administración del Ciclo de Vida de las Aplicaciones según su traducción al español. El ciclo de vida de un software posee algunas variantes durante su tiempo, entre los más conocidos está el modelo en cascada el cual incluirá el Análisis, diseño, desarrollo, implementación, codificación, pruebas y despliegue.

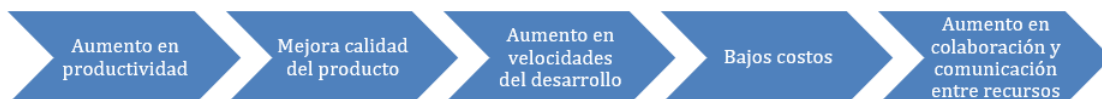
En ocasiones un solo desarrollador o un equipo limitado de desarrollo, deciden realizar una aplicación, es por ello que necesitará algunas herramientas para llevar un control del trabajo que está efectuando, ahora bien, a medida que los equipos crecen o se da un incremento en los requerimientos y tareas no es

suficiente llevar un control manual y es por ello que hay que acudir a herramientas más especializadas.

Estas herramientas poseen un objetivo primordial en dónde se enfocan en crear un repositorio central que junto con sus funcionalidades permitan la automatización del proceso de desarrollo, así como la gran colaboración entre los distintos roles afectados durante todas las etapas del desarrollo.

Durante el desarrollo de un software, este puede verse afectado por tareas: en dónde se establecen requerimientos, se define un diseño, una arquitectura, se llega a crear o modificar el código fuente, se generan pruebas, y comienza la aparición de errores, mismos que deben corregirse. Cuando existe una herramienta ALM controlando lo anteriormente mencionado, todos los elementos se pueden relacionar lo que hace posible conocer cuál es el avance de los requerimientos efectuados en el desarrollo, así mismo estas herramientas indicarán cuales de los requerimientos han sido aprobados, presentan errores, cuales errores están por corregirse, inclusive indica cuáles requerimientos han finalizado completamente, y muchos más que difícilmente se puede llevar de forma manual.

Los beneficios de una puesta en marcha de un ALM son muchos, entre los más destacados están:



Según la compañía Braven IT Consulting, mediante su sitio web <https://www.bravent.net/por-que-implantar-alm-para-gestionar-el-ciclo-de-vida->

de-tus-proyectos en su artículo denominado “*Por qué implantar ALM para gestionar el ciclo de vida de tus proyectos*”, menciona que las herramientas dedicadas al control de la gestión del ciclo de vida de la aplicación tienden ofrecer muchas funcionalidades para obtener un mejor control en el proceso de desarrollo y mantenimiento del mismo, entre sus funcionalidades se encuentran:

- 2.1.4.1 **Administración de requerimientos:** Sirve para registrar todos los requerimientos obtenidos por medio de los casos de uso, historias de usuarios brindadas por quienes conocen el proceso o cualquier otra técnica utilizada para poder realizar el proceso.
- 2.1.4.2 **Administración de tareas:** Sirve para la creación de tareas a cada integrante del equipo, estas tareas deben realizarse con sus prioridades y consideraciones particulares.
- 2.1.4.3 **Control de código fuente:** Sirve para poder proteger el código fuente del desarrollo efectuado, así controlar y llevar histórico de los cambios realizados durante el desarrollo, además permite crear algunas “ramas” el mantenimiento de versiones.
- 2.1.4.4 **Control de errores (*bugs*):** Sirve para llevar un registro de todos aquellos errores o defectos localizados, removidos o que se encuentran aún pendientes.
- 2.1.4.5 **Administración de casos de pruebas:** Sirve para organizar y llevar una correcta documentación sobre las pruebas que estarán realizadas a cada producto desarrollado.

- 2.1.4.6 **Ejecución de pruebas y reporte de errores:** Sirve para controlar todos aquellos casos de pruebas realizados, también indicará cuales de estos presentaron alguna falla durante la ejecución del mismo indicando por medio de reportes los errores encontrados.
- 2.1.4.7 **Herramientas de integración de la aplicación:** Sirve para la realización de integración continua de toda aplicación, permitiendo mejorar altamente la calidad del producto.
- 2.1.4.8 **Administración de versiones:** Sirve para llevar el control de las versiones que han sido liberadas del producto final.
- 2.1.4.9 **Pruebas automatizadas:** Sirve para automatizar o robotizar todas las pruebas y posibles escenarios a generarse y que tienen la posibilidad de generar pruebas de regresión y efectuar carga de estrés al producto.
- 2.1.4.10 **Monitoreo y reportes:** Sirve para generar informes de todo el proceso, generar indicadores, así como utilizar la herramienta para hacer control y auditoría del producto permitiendo poder realizar toma de decisiones en el momento indicado.

2.1.5. Herramientas de ALM y Metodologías

Entre las metodologías existentes al momento de desarrollar una aplicación, hay modelos robustos como RUP (*Rational Unified Process*) y TFS (*Team Software Process*), así como metodologías ágiles actuales como lo es Scrum o XP (*xtreme programming*). Algunas herramientas ALM poseen incorporaciones de plantillas de procesos de las metodologías más utilizadas e

incluso mantienen la facilidad de poder ser modificadas o bien la creación de algunas plantillas personalizadas.

Lo anterior permite llevar un control abierto sobre cada etapa inclusive se pueden definir características particulares de la metodología implementada, un ejemplo de lo anterior sería si se utilizara Scrum como metodología ágil se podría definir un sprint o generar un ítem de la pila de productos.

Entre las herramientas a las que se tiene acceso en el mercado existen Microsoft TFS, HP Application Lifecycle Management, Atlassian JIRA, IBM Rational Team Concert, IBM Rational solution for Collaborative Lifecycle Management, SAP Solution Manager y StarTeam (Coronel, 2013).

Las herramientas ALM por lo general están presentes, desde el nacimiento del proyecto, al control de versiones, los requisitos, las incidencias y el diseño, entre otros. En el 2011 el fabricante líder de las herramientas ALM, fue IBM, seguido por Microsoft y Atlassian. Este último manifiesta estar satisfecho de estar entre los líderes por segunda vez en cuánto a herramientas de planificación ágil empresariales. Esto consolida la creencia de que las herramientas como Jira Software y Portfolio para Jira ayudan a los equipos a practicar con éxito el desarrollo ágil y lanzar un gran software a escala. (Atlassian, 2019)

Por otro lado, el presidente de Atlassian Jay Simons manifiesta:

“Los 119,000 clientes de Atlassian se encuentran en más de 185 países y en casi todas las industrias. Creemos que, si está intentando cambiar el mundo por medio de la innovación de software, Atlassian es el mejor socio que puede elegir.

Sentimos que nuestra creciente presencia en el mercado y nuestra posición continua en el cuadrante de "Líderes" de Gartner es una evidencia de que estamos interrumpiendo el mercado tradicional ágil con nuestros excelentes productos y nuestro modelo único." (Atlassian, 2019)

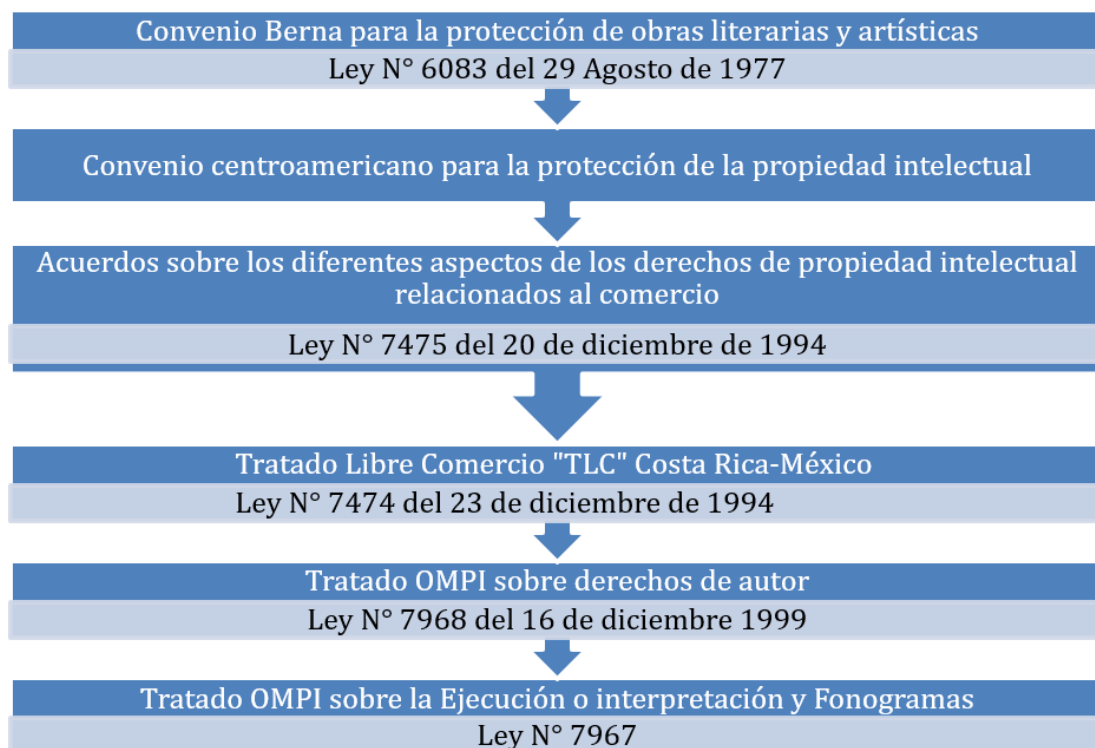
Entre los futuros fabricantes se puede decir que estará HP y actualmente quienes tienen menor capacidad de expansión y crecimiento se encuentran: *TechExcel*, *MicroFocus* o *Rocket Software*; sin embargo, existen algunas organizaciones que son visionarias en el mercado de herramientas ALM, pero lastimosamente mantienen una carencia en medios para poder ejecutarlas, ejemplo de esto es ThoughtWorks o Polarion. (Garzás, 2012).

2.2. Marco Legal

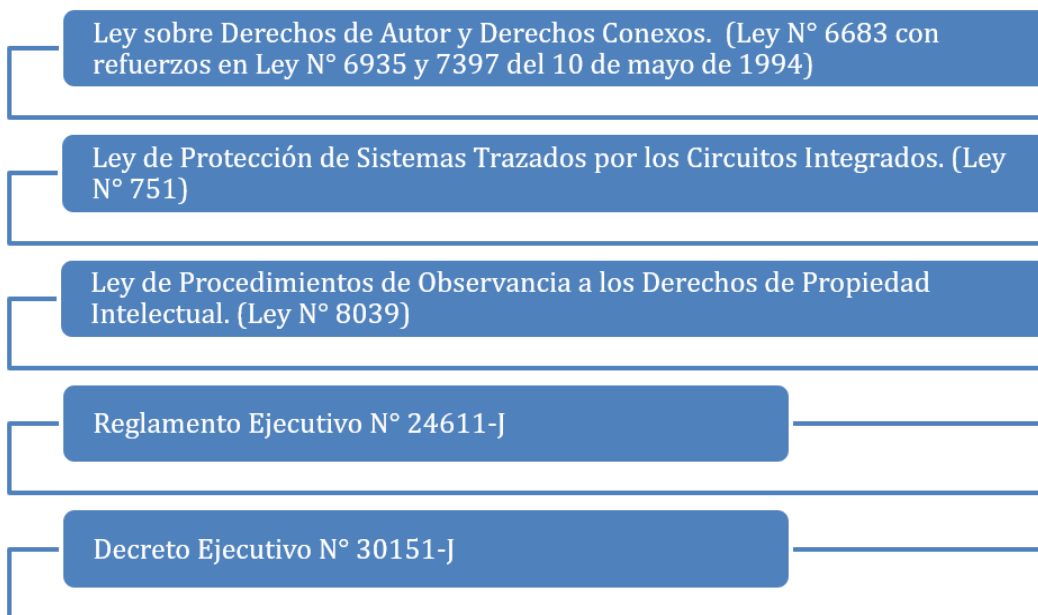
El marco legal se ve envuelto por convenios, normativa legales y leyes que aplican sobre el marco regulativo, mismas que dan potestades y facultades al desarrollo de software dentro de las cuales se pueden mencionar:

- I. Convenios Internacionales ratificados por Costa Rica que permiten regular las aplicaciones de software, así como servicios genéricos.

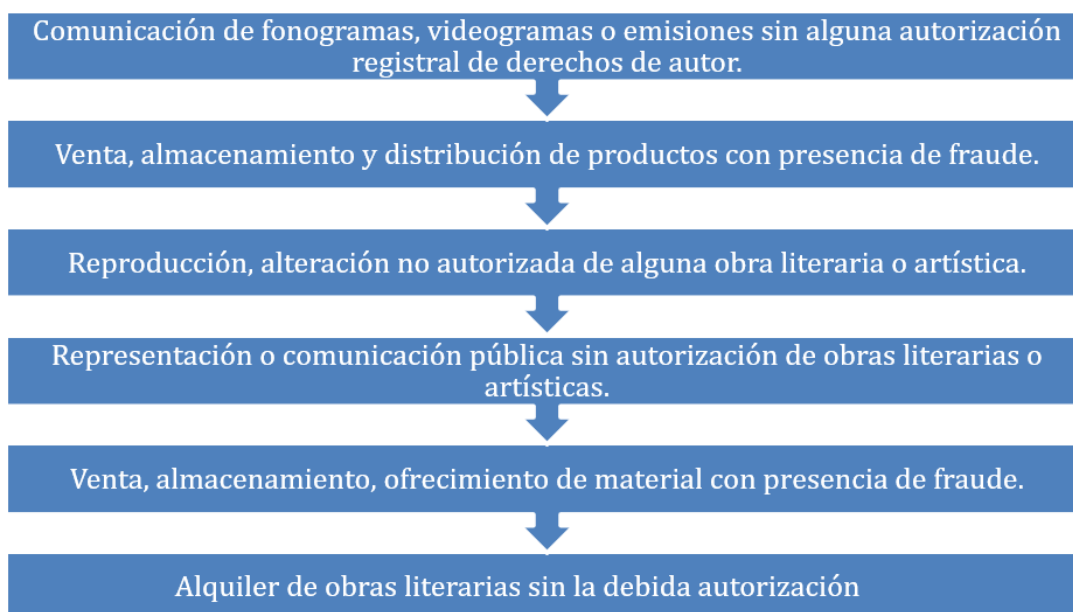
Aa



- II. La normativa nacional que permite regular las aplicaciones de software y otros servicios genéricos:



- III. Ley de Procedimientos de Observancia de Derechos de Propiedad Intelectual. Se debe a los delitos que están vinculados con la protección de las Tecnologías de la Información y Comunicación (TIC), entre estas se pueden citar:



En la actualidad las empresas de desarrollo de software a nivel nacional, mantienen un objetivo de calidad el cual debe ser respaldado por un proceso estructurado de aseguramiento de software, esto con el fin de ofrecer productos de alto estándar de calidad:

El uso de estándares como ISO270013, permite el contribuir en los procesos de reconocimiento y control, de las distintas áreas que se encuentran en una organización, por ejemplo, dentro del área de sistemas, se genera una gran importancia de creación y adaptación de distintos mecanismos y políticas en los procesos, permitiendo asegurar y mejorar la seguridad informática.

Los soportes en los estándares de la actual norma ISO/IEC 270000 pone en firme algunas legislaturas tanto nacionales como internacionales:

ISO/IEC 270014	<ul style="list-style-type: none"> •Especifica los parámetros de seguridad informática. •Apta para desarrollar, implementar y mantener los sistemas de seguridad de información.
ISO/IEC 27001	<ul style="list-style-type: none"> •Determina los requisitos de una implementación
ISO/IEC 27002	<ul style="list-style-type: none"> •Guía de buenas prácticas •Describe controles a seguir
ISO/IEC 27003	<ul style="list-style-type: none"> •Ayuda y orientación sobre la implementación •Incluye el Método de planear, hacer, verificar y actuar •Mejora continua
ISO/IEC 27004	<ul style="list-style-type: none"> •Muestra métricas y técnicas de medición de eficacia •Se aplica en la fase de hacer (DO)
ISO/IEC 27005	<ul style="list-style-type: none"> •Presenta directrices para la gestión de riesgo en la seguridad de la inforamción
Ley 9135	<ul style="list-style-type: none"> •Ley sobre delitos informáticos y la protección de la inforamción.

CAPÍTULO III MARCO METODOLÓGICO

3.1. Marco Metodológico

Existen diversas definiciones sobre los tipos de investigación, estas varían según su tipología y sus características tales como sus estilos, formas enfoques y modalidades. Conceptualmente, estos tipos son definidos para la obtención del conocimiento en una investigación y los tres principales serían: exploratoria, descriptiva y explicativa. (Eyssautier de la Mora, 2002)

La presente investigación definirá un estilo de trabajo basado en un diseño descriptivo, utilizando instrumentos de naturaleza cualitativa, para el análisis de la información de procesos de aseguramiento existentes. Asimismo, se trabajará bajo un diseño exploratorio, aplicándose un instrumento de naturaleza cuantitativa, que facilitará el levantamiento de información respecto de los diversos procesos actuales.

Además, se pretende la identificación de características del contexto de investigación, señalar sus formas de conducta y actitudes, por lo que deberá establecerse su comportamiento concreto, así como descubrir y comprobar la asociación entre las variables de investigación.

Esto será aplicado mediante entrevistas y búsqueda de información suministrada por empresas de desarrollo de software de nivel nacional. Entrevistas conformadas por la búsqueda de información nacional, para la identificación de aspectos importantes a evaluar y con ellos enmarcar los objetivos de cada uno de los instrumentos que serán conformados durante el

proceso, los cuales serán utilizados como objeto de recopilación de información con la que se desarrollará la investigación y se crearán las guías propuestas.

3.1.1. Tipo de Investigación

A continuación, se procede con el detalle de los argumentos que sustentan el tipo de investigación realizado, según su naturaleza de objetivos en cuanto al nivel de conocimiento que se desea alcanzar, se determina que la investigación es de tipo exploratoria y descriptiva.

3.1.1.1. Investigación Exploratoria

La presente investigación se centraliza en analizar e investigar detalles muy relevantes sobre la realidad que aún no han sido analizados en profundidad. Se trata de una exploración que permitirá que investigaciones futuras puedan dirigirse a un análisis de la temática tratada en esta investigación.

Al ser una investigación explorativa y debido a sus características, este tipo de investigación no se enfoca ni parte de las teorías detalladas, por lo contrario, trata de localizar patrones significativos en los datos que deben ser analizados para, que una vez determinado esto partir de estos resultados, crear las primeras explicaciones completas sobre lo que ocurre. (Copyright 2018 Psicología y Mente, 2017).

En el análisis de esta investigación, se beneficiará el encuentro de áreas que requieren mayor análisis investigativo dentro del aseguramiento de la calidad del software, esto para poder lograr el conocimiento para la definición de la guía

de mejora para el proceso, esto se realizará mediante entrevistas y encuestas ejecutadas a los colaboradores que participan en estas áreas.

3.1.1.2. Investigación Explicativa

Este tipo de investigación es utilizado para intentar determinar las causas y consecuencias de un fenómeno cualquiera. Se enfoca en el qué y el porqué de las cosas, y profundiza en el cómo se ha llegado a un estado en cuestión. Para ello pueden usarse diferentes métodos, como el observacional, el correlacional o el experimental. (Copyright 2018 Psicología y Mente, 2017)

El objetivo de ser utilizado en este documento es para poder crear modelos explicativos en el que se puedan observar las secuencias de causa-efecto lineales o no lineales.

Se solicitará la posibilidad de evaluar a los colaboradores, para determinar si aplica la normativa de aplicación de aseguramiento del software (si este existiese) y seguidamente, poder observar el resultado y verificar si el mismo está ejecutado de la manera correcta y permite lograr el mejor resultado.

3.1.1.3. La Investigación Diacrónica

Este tipo de investigación se crea por la naturaleza de la información obtenida para responder al problema de investigación. En este documento se presenta una síntesis diacrónica de las teorías del aprendizaje más conocidas y las cuales se les relaciona con la aparición del software. Dónde se plantea el escenario actual, la problemática vivida a causa de aspectos diversos en la construcción de los proyectos y aplicaciones sin el aseguramiento en la calidad de este.

Para efecto de este documento se plantea que la investigación es cualitativa.

3.1.1.4. La Investigación Cualitativa

Esta investigación según Sampieri, es aquella que estudia los fenómenos en un período largo con el objeto de verificar los cambios que se pueden producir. (Sampieri).

Dentro de esta investigación se podrá citar la investigación participativa, dado que al ser un estudio que nace de un problema originado en la misma comunidad con el propósito del hallazgo de una solución que mejore el nivel de vida de los colaboradores involucrados. Dentro de la investigación participativa se pueden encontrar:

3.1.2. Estudio de Casos

Según el autor Stake en su artículo denominado “Investigación con estudio de casos”, determina que el estudio de casos es una metodología aplicable al análisis de fenómenos sociales y educativos a lo que se refiere como “es el estudio de la particularidad y de la complejidad de un caso singular, para llegar a comprender su actividad en circunstancias importantes”. (Stake, 1998, pág. 11).

Esta metodología está diseñada para investigadores que tienen el deseo de profundizar en el estudio de una situación determinada, permitiendo brindar la oportunidad de estudiar detalladamente una parte de un problema, pero con un tiempo limitado. Puede estar sujeto a un fenómeno, una persona, un evento o

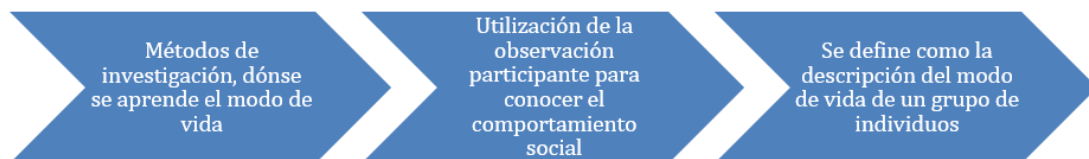
caso muy concreto, de acá que deberá realizarse un análisis del entorno en que se encuentra el objeto de estudio. (Muñoz, 2011).

Esto permitirá realizar una investigación asertiva en cuanto a los tipos de metodologías ágiles que las empresas costarricenses utilizan actualmente, de igual manera permitirá entender el porqué de su utilización.

3.1.2.1. Estudio Etnográfico

Este estudio es el favorito para analizar y enfatizar las particularidades descriptivas e interpretativas de un ambiente sociocultural, dónde es utilizado en aquellos estudios relacionados a la antropología social y la educación, por lo que puede es considerado como uno de los métodos de investigación más relevantes dentro de la investigación humanístico-interpretativa, lo anterior según interpretado por Del Rincón & Latorre, 1992 en su artículo “*Investigación Educativa*”.

En el mismo artículo se dedique que toda investigación etnográfica debe estar recurrida por medio de tres definiciones:



Dentro de la investigación será utilizada para analizar el pensar y actuar de muchos colaboradores que utilizan estas metodologías ágiles, incluso también se podrá analizar si lo consideran necesario, o que tanto resultado brinda a manera de criterio personal, aunque también se abundará sobre el enfoque según la comunidad que pueda utilizar dichas metodologías.

3.1.2.2. Estudios Históricos

La investigación histórica es realizada cuando se desea estudiar de una perspectiva histórica una realidad, apoyado desde luego de las fuentes primarias y fuentes secundarias para el análisis de la misma.

Esto permitirá dar mayor sustento a la investigación, ya que se podrá buscar posibles problemas basados en la experiencia de los usuarios y buscar la mejor metodología ágil para solucionar dicho incidente.

3.1.3. Sujetos

La obtención de información requerida en esta investigación se realizará por medio del Ministerio de Hacienda para determinar el universo de las empresas desarrolladoras de software, se tomará en cuenta a la jefatura de los departamentos de TI en caso de existir, así como a los colaboradores que lo conforman.

3.1.4. Población

La población puede definirse como conjunto total de individuos, objetos o medidas con características comunes observables en un espacio y tiempo determinados.

Empresas desarrolladoras de software de nivel nacional, dentro de las cuales se cuenta con una población en el territorio nacional de 1400 empresas

que realizan alrededor de 30 actividades relacionadas con el sector de servicio de software. Entre estas, las actividades de programación informática, software de red, servicios de diseño y desarrollo de aplicaciones, bases de datos y servicios de alojamiento web.

Sin embargo, el tamaño del mercado tiende a ser mayor, si se considera tomar en cuenta que la Dirección General de Tributación del Ministerio de Hacienda en Costa Rica, registra alrededor de 7.600 contribuyentes vinculados con actividades enfocadas en software. (Sancho, 2013).

Según la población presentada la muestra a determinar para los artefactos sería la siguiente:

Variables	Valor
N: es el tamaño de la población o universo (número total de posibles encuestados).	1,400
k: constante, se ve afectada según el nivel de confianza que sea asignado. Este nivel de confianza indicará la probabilidad de que los resultados de la investigación sean ciertos: <ul style="list-style-type: none"> • 95 % de confianza es lo mismo que decir que nos podemos equivocar con una probabilidad del 5%. 	85%
p: proporción de individuos presentes en la población con características de estudio. Es un dato desconocido, pero se ha llegado a la suposición debido a la literatura que así lo recomienda que puede ser: <ul style="list-style-type: none"> • p=q=0.5 (Considerado como la opción más segura). 	0.5
q: proporción de individuos que no poseen características de estudio: <ul style="list-style-type: none"> • Su cálculo se define como 1-p. 	0.5
e: Error deseado en la muestra de la población. Diferencia entre el resultado obtenido de una muestra de la población y el total de la misma.	5%

Algunos niveles de confianza, más utilizados en el muestreo son:

K	1,15	1,28	1,44	1,65	1,96	2	2,58
Nivel de confianza	75%	80%	85%	90%	95%	95,5%	99%

Ilustración 7: Niveles de Confianza para determinar Muestreo de Población

Por lo tanto, al aplicar la fórmula para la obtención de la muestra se obtiene el siguiente resultado:

$$n = \frac{k^2 * p * q * N}{(e^2 * (N - 1)) + k^2 * p * q}$$

Ilustración 8: Fórmula para determinación de Muestreo.

Por lo que la muestra sería: 181

Utilizando la calculadora del sitio

<https://www.feedbacknetworks.com/cas/experiencia/sol-preguntar-calcular.html>

que permite realizar el mismo cálculo:

N:	<input type="text" value="1400"/>
k:	<input type="text" value="1.44"/>
e:	<input type="text" value="5"/> %
p:	<input type="text" value="0.5"/>
q:	<input type="text" value="0.5"/>
<input type="button" value="Calcular muestra"/>	
n:	<input type="text" value="181"/> es el tamaño de la muestra

Ilustración 9: Detalle de Muestra de Población.

Población objetivo	Ubicación	Tiempo / perfil	Disponibilidad
Empresas nacionales desarrolladoras de software.	Área metropolitana (Mayor concentración de empresas)	Empresas de desarrollo que cuenten con las de 15 empleados y que estén aplicando una metodología para el desarrollo, con al menos 5 años en el mercado.	Se tomó en cuenta la facilidad en la obtención de la información.
Funcionarios de empresas nacionales	Área metropolitana (Mayor concentración de empresas)	<p>Tiempo labor mayor a 2 años en el desarrollo de aplicaciones y se desempeñan en puestos de dirección de equipos.</p> <p>han trabajado bajo alguna metodología de desarrollo que tengan 2 años de experiencia con la metodología y que desempeñen puestos de asegurador.</p> <p>Han trabajado bajo alguna metodología de desarrollo que</p>	Que cuenten con la disponibilidad para responder la entrevista o encuesta de no tener el tiempo para la entrevista.

Población objetivo	Ubicación	Tiempo / perfil	Disponibilidad
		<p>tengan 2 años de experiencia con la metodología y que desempeñen puestos de desarrollador.</p>	
		<p>2 años de experiencia en puestos de planeación de proyecto, puestos similares a analistas de negocio.</p>	
		<p>Recién ingresadas al área de desarrollo que tengan tiempos menores a 2 años de experiencia y que se desarrollen como desarrollador o asegurador.</p>	

3.1.5. Fuentes de Información

Las fuentes de información son los productos, materiales, originales o elaborados, que presentan colaboración con las noticias o testimonios por los que se puede tener acceso al conocimiento. Esta definición permite considerar que una fuente de información puede ser cualquier objeto o situación que permita transmitir algo que, tras su interpretación y elaboración, genere conocimiento.

En esta investigación se utilizará la metodología cualitativa, la cual requiere la recolección de datos para consumir comparaciones y llegar a posibles contrastes; estos datos se recopilan por medio de documentos relacionados con el tema e información de referencia recolectada desde internet; se hará investigación de campo con entrevistas al jefe del departamento de algunas empresas desarrolladoras de software y encuestas al resto de colaboradores por medio de cuestionarios, cuyo propósito es la descripción de las tareas y los métodos aplicados durante las actividades diarias.

Las fuentes de información se dividen en dos: las fuentes Primarias y las fuentes secundarias.

3.1.5.1 Fuentes Primarias

Las fuentes primarias son aquellas referidas a los portadores originales de la información que no han retransmitido, grabado, o documentado en cualquier medio

la información primordial. Para la extracción de los datos de esta fuente se utilizarán los métodos de encuesta, entrevista y observación.

La entrevista, es una comunicación interpersonal establecida entre el investigador y uno o varios sujetos relacionados con el contexto de estudio a fin de obtener respuestas verbales a una serie de interrogantes planteadas que se relacionan con el tema investigado.

Una encuesta descriptiva es aquella que documenta actitudes o condiciones presentes en un contexto determinado, estas describen en qué situación se encuentra determinada población en el momento en que se realiza la encuesta.

La observación es aquel procedimiento estructurado de recolección de datos que se dedica a explorar, describir, comprender, identificar y crear hipótesis sobre el ambiente de estudio definido.

En esta investigación se realizarán varias entrevistas físicas o bien telefónicas a la jefatura del departamento de las empresas desarrolladoras de software, así como una encuesta completa enviada de forma digital a los colaboradores del Departamento, para la extracción de los datos requeridos con el fin de obtener del análisis del entorno de trabajo exacto en el que se debe aplicar la normativa que se desea construir.

Se tomará en cuenta las historias de usuario utilizadas, interés de la investigación, dado que representan los eventos que encaminará al equipo de

desarrollo hacia el objetivo y que se considera la definición de lo que se quiere desarrollar.

3.1.5.2. Fuentes Secundarias

Según Eyssautier de la Mora en artículo “Metodología de la Investigación Desarrollo de la inteligencia” se refiere a fuentes secundarias como “Todos los portadores de datos e información que han sido previamente retransmitidos o plasmados en cualquier documento, y que utilizan el medio que sea” (Eyssautier de la Mora, 2002). Esta información se encuentra a disposición de todo investigador que la requiera.

Para esta investigación se analizará literatura disponible y actualizada sobre el tema, como el documento de Metodologías Ágiles actuales, así como extractos de información de diversas fuentes como internet, libros, artículos de revista, informes, tesis, entre otras, relacionada con la temática y que pueda ser considerada de carácter formal, posterior a un análisis crítico se extrajo todo conocimiento considerado útil para alcanzar los objetivos de la investigación.

- 3.1.6. Técnicas de Recolectar nformación

Se intenta recolectar la mayor cantidad de información posible con respecto al estado actual, estudios o proyectos que tengan relación con el análisis de riesgos y en general en materia de seguridad informática en la institución.

El tipo de técnica utilizada para la recolección de información es:

A. **Encuesta**

Se fundamenta en un cuestionario o conjunto de preguntas para obtener información de las personas.

B. **Entrevista**

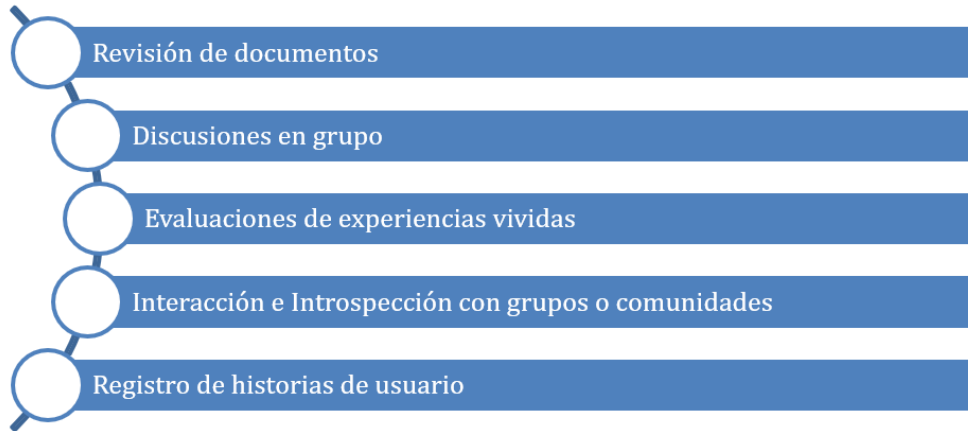
Técnica utilizada frecuentemente en la ingeniería debido a que es de gran conocimiento y de la más utilizada dentro de las metodologías de desarrollo ágil, por lo que brinda un gran acercamiento con los clientes y usuarios para lograr el cometido con estas.

Es utilizado en todas las metodologías ágiles, y se considera que la conversación directa con el “cliente”, también es el método más eficaz para licitar u obtener la información suficiente misma que servirá como soporte para lograr los objetivos propuestos y brindar una satisfacción al “cliente”, esto puede llegar reducir las mal interpretaciones del negocio por parte del equipo de desarrollo.

C. **Reuniones de Análisis**

Utilizado en algunas metodologías ágiles durante el proceso de desarrollo del producto, se puede concluir que es transversal a este último, permitiendo generar una retroalimentación constante de lo que se va trabajando, su técnica consiste en mantener reuniones que permitan realizar análisis de elementos presentados en la etapa actual del proceso, cada reunión debe ser documentada con el fin de que pueda ser de apoyo para reuniones futuras, también sirve para involucrar a todos los colaboradores implicados en el desarrollo proyecto.

Otras formas de recolección de datos son:



3.1.7. Tipo de enfoque

Dentro de una investigación, después de haber elaborado el planteamiento del problema, las preguntas, los objetivos específicos y principales, así como la hipótesis, se deberá seleccionar la muestra a la que se aplicará el estudio de acuerdo con el enfoque elegido, posteriormente se recolectaron los datos pertinentes sobre variables, eventos y objetos involucrados.

Todo trabajo de investigación debe ser basado en dos enfoques principales: el enfoque cuantitativo y el enfoque cualitativo, los cuales de manera conjunta forman un tercer enfoque llamado mixto.

El enfoque cualitativo es aquel cuyo objetivo se centra en la descripción de las cualidades de un fenómeno, mientras que el cuantitativo es aquel que permite examinar información de manera numérica, especialmente en el campo de la Estadística.

El enfoque mixto utiliza métodos de los dos enfoques descritos dentro de un mismo estudio y puede involucrar la conversión de datos cualitativos en cuantitativos y viceversa.

Esta investigación se considera de tipo mixta, porque en primera instancia su estrategia va dirigida a reconocer hechos, procedimientos, estructuras y personas de una manera integral y no mediante la medición de algunos de sus elementos.

Sin embargo, con el transcurrir de la misma, específicamente en su etapa de análisis, se obtienen datos cuantitativos como lo son el número de colaboradores dentro del departamento de TI, cuales metodologías ágiles utilizan y porqué, entre otros datos; por lo que terminó convirtiéndose en un enfoque mixto.

3.1.8. Matriz Metodológica

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
Localizar factores técnicos e información detallada que se involucra el proceso de aseguramiento de software para conformar una base para el análisis del proceso aplicado por las empresas nacionales de desarrollo de software.	Factores técnicos.	Son el conjunto de aspectos que detallan el proceso de aseguramiento de la calidad.	Tiempo de entrega.	Horas/Minutos	Consultar tesis y documentos que contenga información del proceso de aseguramiento.
			Se aplica el aseguramiento de software con un marco metodológico.	SI/NO	Investigación de información (Internet)
			Criterios de aceptación.	Se Hace/No se hace	Investigación de información (Internet)
			Enfoque	Se aplican enfoques (Pasivos, reactivos, sistemático)	Investigación de información (Internet)
			Plan de pruebas	Se crean (SI/NO)	Consultar tesis y documentos que contenga información del proceso de aseguramiento.
			Herramientas.	Se utilizan/No se utilizan	Investigación de información (Internet)
	Información del proceso de aseguramiento	Documentos y artículos que contengan información que documenten el proceso de aseguramiento	Localizar información a nivel internacional	Existe procesos de aseguramiento (Si/No)	Investigación de información (Internet)
			Documentación técnica.	Hay una definición o documentación que respalde el proceso. (Si/No).	Consultar tesis y documentos que contenga información del proceso de aseguramiento.

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
	Base de análisis de procesos.	Recopilación de información del proceso que se aplica en aseguramiento para que sea analizada.	Análisis de las fases del proceso	Cuántas caras (1,2,3,4,5,6...)	Consultar tesis y documentos que contenga información del proceso de aseguramiento.
Identificar las metodologías que se involucran al proceso			Cuales son (Lista de metodologías)	Consultar tesis y documentos que contenga información del proceso de aseguramiento.	
Análisis de la información			Existe necesidad de mejora en el proceso (Si/No).	Consultar tesis y documentos que contenga información del proceso de aseguramiento.	
Recopilar información sobre la tendencia actual aplicada a procesos de aseguramiento de software tomando en cuenta Marcos de trabajo, Metodologías ágiles y herramientas, mediante visitas a empresas nacionales de desarrollo de software, para con ello determinar los factores que pueden influir en el proceso.	Empresas nacionales de desarrollo de software.	Grupo de Empresas nacionales dedicadas al desarrollo de software, que cuenten con las de 15 empleados y que estén aplicando una metodología para el desarrollo.	Poder obtener la experiencia de líderes de equipos.	Existe un proceso de aseguramiento. (Si/No) Hay una definición o documentación que respalde el proceso. (Si/No). Existe necesidad de mejora en el proceso (Si/No). ¿Cuál metodología de desarrollo aplica actualmente?	Visitas a empresas de desarrollo, entrevistas a personas que tengan un tiempo mayor de 2 años aplicando una metodología de desarrollo.

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
				<p>Mencione 3 aspectos que usted cambiaría del proceso actual de aseguramiento o del marco de trabajo que aplica.</p> <p>Que técnicas de aseguramiento aplican en el proyecto que desarrolla actualmente.</p>	
	<p>Empresas nacionales de desarrollo de software.</p>	<p>Grupo de Empresas nacionales dedicadas al desarrollo de software, con más de 15 empleados y que cuenten con una técnica para el aseguramiento de software ya sea mediante la aplicación de una guía o una técnica madura desarrollada por la empresa.</p>	<p>Poder obtener la experiencia de líderes de equipos.</p>	<p>Existe un proceso de aseguramiento. (Si/No)</p> <p>Hay una definición o documentación que respalde el proceso. (Si/No).</p> <p>Existe necesidad de mejora en el proceso (Si/No).</p>	<p>Visitas a empresas de desarrollo, entrevistas a personas que tengan un tiempo mayor de 2 años aplicando guías o técnicas de aseguramiento de la calidad.</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
				<p>¿Cuál metodología de desarrollo aplica actualmente?</p> <p>Mencione 3 aspectos que usted cambiaría del proceso actual de aseguramiento o del marco de trabajo que aplica.</p> <p>Que técnicas de aseguramiento aplican en el proyecto que desarrolla actualmente.</p>	
	<p>Información suministrada por personas involucradas en el proceso</p>	<p>Realizar una investigación dentro de las empresas nacionales que ayuden a identificar las características que conforman e integran el proceso de aseguramiento de la</p>	<p>Funcionarios de empresas nacionales con tiempo laboran mayor a 2 años y se desempeñan en puestos de dirección de equipos.</p>	<p>Existe un proceso de aseguramiento. (Si/No)</p> <p>Hay una definición o documentación que respalde el proceso. (Si/No).</p>	<p>Entrevista Scrum Master, jefe de departamento)</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
		<p>calidad de software a nivel nacional.</p>		<p>Existe necesidad de mejora en el proceso (Si/No).</p> <p>¿Cuál metodología de desarrollo aplica actualmente?</p> <p>Mencione 3 aspectos que usted cambiaría del proceso actual de aseguramiento o del marco de trabajo que aplica.</p> <p>Que técnicas de aseguramiento aplican en el proyecto que desarrolla actualmente.</p>	

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
			<p>Personas que han trabajado bajo alguna metodología de desarrollo que tengan 2 años de experiencia con la metodología y que desempeñen puestos de asegurador.</p>	<p>Existe un proceso de aseguramiento. (Si/No)</p> <p>Hay una definición o documentación que respalde el proceso. (Si/No).</p> <p>Existe necesidad de mejora en el proceso (Sí/No).</p> <p>¿Cuál metodología de desarrollo aplica actualmente?</p> <p>Mencione 3 aspectos que usted cambiaría del proceso actual de aseguramiento o del marco de trabajo que aplica.</p> <p>Que técnicas de aseguramiento aplican en el proyecto que</p>	<p>Entrevista (experto, en aseguramiento de la calidad)</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
				<p>desarrolla actualmente.</p>	
			<p>Personas que han trabajado bajo alguna metodología de desarrollo que tengan 2 años de experiencia con la metodología y que desempeñen puestos de desarrollador.</p>	<p>Existe un proceso de aseguramiento. (Si/No)</p> <p>Hay una definición o documentación que respalde el proceso. (Si/No).</p> <p>Existe necesidad de mejora en el proceso (Si/No).</p> <p>¿Cuál metodología de desarrollo aplica actualmente?</p> <p>Mencione 3 aspectos que usted cambiaría del proceso actual de aseguramiento o del</p>	<p>Entrevista (experto, desarrollador)</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
				<p>marco de trabajo que aplica.</p> <p>Que técnicas de aseguramiento aplican en el proyecto que desarrolla actualmente.</p>	
			<p>Personas con 2 años de experiencia en puestos de planeación de proyecto, puestos similares a analistas de negocio.</p>	<p>Existe un proceso de aseguramiento. (Sí/No)</p> <p>Hay una definición o documentación que respalde el proceso. (Si/No).</p> <p>Existe necesidad de mejora en el proceso (Sí/No).</p> <p>¿Cuál metodología de desarrollo aplica actualmente?</p> <p>Mencione 3 aspectos que usted cambiaría del proceso actual de</p>	<p>Entrevista (Analista de negocio)</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
				<p>aseguramiento o del marco de trabajo que aplica.</p> <p>Que técnicas de aseguramiento aplican en el proyecto que desarrolla actualmente.</p>	
			<p>Personas recién ingresadas al área de desarrollo que tengan tiempos menores a 2 años de experiencia y que se desarrollen como desarrollador o asegurador.</p>	<p>Existe un proceso de aseguramiento. (Si/No)</p> <p>Hay una definición o documentación que respalde el proceso. (Si/No).</p> <p>Existe necesidad de mejora en el proceso (Si/No).</p> <p>¿Cuál metodología de desarrollo aplica actualmente?</p>	<p>Entrevista (Desarrollador o Analista de la calidad)</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
				<p>Mencione 3 aspectos que usted cambiaria del proceso actual de aseguramiento o del marco de trabajo que aplica.</p> <p>Que técnicas de aseguramiento aplican en el proyecto que desarrolla actualmente.</p>	
	Tendencia de aseguramiento de la información en empresas nacionales	Define la tendencia de aseguramiento aplicada por las empresas nacionales	Se aplica con un marco metodológico.	Sí/NO	Entrevista (experto, desarrollador, jefe de departamento)
			Se aplica clasificación acorde al tipo de prueba aplicada	Sí/NO	Entrevista (experto, Asegurador de la calidad)
			Aplican guías de aseguramiento	íí/NO	Entrevista (experto, desarrollador, jefe de departamento)

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
	Proceso de aseguramiento de la calidad de software en empresas nacionales	Conocer los aspectos relevantes dentro de las empresas	Existen documentación que defina como se aplica el proceso.	Sí/NO	Entrevista (Administrador de proyectos, jefe de departamento)
Dentro de la política de la empresa se mencionan los procesos de desarrollo de aplicaciones			Sí/NO El proceso está bien definido en las políticas de la compañía (SI/NO)	Entrevista (Administrador de proyectos, jefe de departamento)	
Medidas de control aplicada por las empresas			Lista de medida de control aplicadas al proceso	Entrevista (Administrador de proyectos, jefe de departamento)	

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
<p>Fundamentar la tendencia de aseguramiento de software utilizada en la actualidad por empresas nacionales de desarrollo de software, realizando investigaciones de tendencias y factores técnicos fuera del ámbito nacional, para definir los procesos adoptados en la guía de aseguramiento de calidad del software.</p>	<p>Fundamentar la tendencia de aseguramiento de la calidad</p>	<p>Investigar las tendencias aplicadas por la industria nacional de desarrollo de software, para obtener un mayor conocimiento de estas</p>	<p>Que influye en el proceso de aseguramiento aplicado por las empresas nacionales fabricantes de software.</p>	<p>Lista de factores que influyen en el proceso. Herramientas Marco metodológico Guías de aseguramiento. Técnicas del proceso</p>	<p>Búsqueda en la información obtenida de las entrevistas y la investigación.</p>
			<p>Que influye en el proceso de aseguramiento aplicado por empresas importantes de desarrollo de software a nivel internacional.</p>	<p>Lista de factores que influyen en el proceso. Herramientas Marco metodológico Guías de aseguramiento. Técnicas del proceso</p>	<p>Búsqueda de información en internet (Artículos y páginas de empresas dedicadas al aseguramiento de software)</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
	<p>Identificar y fundamentar los Procesos empíricos de aseguramiento que aplican las empresas nacionales de desarrollo de software</p>	<p>Fundamentar procesos empíricos que utilizan las empresas nacionales de desarrollo de software</p>	<p>Identificar los procesos</p>	<p>Existen procesos de aseguramiento empíricos. (SI/NO). Lista de factores del proceso empírico.</p>	<p>Información obtenida de las entrevistas aplicadas durante la visita a empresas.</p>
			<p>Fundamentar los procesos encontrados</p>	<p>Se puede definir un proceso empírico con la información obtenida. (Si/No) Definir cada factor encontrado que se involucra en el proceso.</p>	<p>Información obtenida de las entrevistas aplicadas durante la visita a empresas.</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
<p>Detectar la herramienta de gestión de ciclo de vida, marco de trabajo ágil y metodología más apropiado para el proceso, tomando en cuenta los factores de mayor tendencia y aspectos de integración para aplicar en el proceso de aseguramiento de software en empresas de desarrollo de software en el nivel nacional.</p>	<p>Definir el top de mayor tendencia en procesos de aseguramiento que pueden ser aplicadas por las empresas nacionales.</p>	<p>En la información recabada, definir un proceso de aseguramiento y las tendencias que ayudarían a tener un mejor proceso de aseguramiento de la calidad.</p>	<p>Definir y fundamentar la metodología a utilizar.</p>	<p>¿Cuál es el top 2 de metodología seleccionada?</p>	<p>Crear una tabla de Excel que permita la revisión de la información obtenida y agrupada según patrones similares como: Tipo de metodología.</p>
			<p>Definir y fundamentar la guía de aseguramiento de la calidad a utilizar.</p>	<p>¿Cuál es la guía de aseguramiento seleccionada?</p>	<p>Crear una tabla de Excel que permita la revisión de la información obtenida y agrupada según patrones similares como: Proceso de aseguramiento aplicado.</p>
			<p>Definir y fundamentar la herramienta de ciclo de vida de software.</p>	<p>¿Cuál es el top 2 de herramienta de ciclo de vida de software seleccionada?</p>	<p>Crear una tabla de Excel que permita la revisión de la información obtenida y agrupada según patrones similares como: Herramientas de ciclo de vida.</p>
			<p>Definir y fundamentar la herramienta de la calidad de software.</p>	<p>¿Cuáles son el top 2 de las herramientas de aseguramiento de la calidad seleccionadas?</p>	<p>Crear una tabla de Excel que permita la revisión de la información obtenida y agrupada según patrones similares como: Herramientas de aseguramiento de la calidad.</p>

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
	Definir la integridad de las metodologías ágiles y las herramientas de ciclo de vida.	Seleccionar la metodología y la herramienta de ciclo de vida del desarrollo de software, que se integran.	Cuales contienen una mayor integración.	Se integran las tendencias top a la herramienta de ciclo de vida de desarrollo y las metodologías aplicadas (Si/No) Cuales mantienen una mayor integración.	. Tablas de Excel que permitan la selección, mediante parámetros de integración
	Definir la integridad de las guías de aseguramiento y herramientas de aseguramiento de la calidad de software.	Seleccionar la guía de aseguramiento de la calidad y las herramientas de aseguramiento, que se integran.	Cuales contienen una mayor integración.	Se integran las tendencias top a la herramienta de aseguramiento y las guías de aseguramiento de la calidad. (Si/No) Cuales mantienen una mayor integración.	Tablas de Excel que permitan la selección, mediante parámetros de integración
Construir una guía metodológica para el proceso de aseguramiento de calidad del software, aplicando los conocimientos	Describir el marco metodológico seleccionado	Se describe el procedimiento que se establece en el marco metodológico seleccionado.	¿Cuál es el marco metodológico seleccionado?	Existe un marco metodológico bien definido para el proceso. (Sí/NO) Se puede describir el mismo (Sí/NO)	Se describe mediante la información recabada en la investigación.

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
adquiridos durante la investigación y las visitas realizadas a empresas nacionales de desarrollo de software, que permitan la elaboración de una propuesta homogénea.	Describir la guía de aseguramiento seleccionada	Se describe el procedimiento que se establece la guía de aseguramiento seleccionada.	¿Cuál es la guía de aseguramiento seleccionada?	Existe una guía de aseguramiento bien definida para el proceso. (Sí/NO) Se puede describir la misma (sí/NO)	Se describe mediante la información recabada en la investigación.
	Describir uso de la herramienta de ciclo de vida de software.	Se describe el uso de la herramienta establecido.	¿Cuál es la herramienta ciclo de desarrollo de software que mantiene más integración en el proceso de aseguramiento definido?	Se puede crear un manual de uso apegado al proceso definido (SI/NO)	Búsqueda de información en internet del uso de la herramienta
			Definir los gráficos que servirán para medir el proceso integrado.	¿Se pueden crear gráficos con información relevante?	Según la información recopilada.
	Describir el uso de las herramientas de aseguramiento seleccionadas.	Se describe el uso de las herramientas establecido.	¿Cuál es la herramienta de aseguramiento de la calidad de software que más se integra al proceso?	Se puede crear un manual de uso apegado al proceso definido (SI/NO)	Búsqueda de información en internet del uso de la herramienta
			Definir los gráficos que servirán para medir el proceso de aseguramiento	¿Se pueden crear gráficos con información relevante?	Según la información recopilada.

Objetivo Específicos	Variable	Definición Conceptual	Dimensión	Medición	Definición Instrumental
	Establecer los indicadores para una buena aplicación del proceso.	Se crean los indicadores para poder medir el proceso establecido.	¿Cuáles son los indicadores que servirán para medir el proceso definido?	Se pueden definir indicadores en el proceso. (SI/NO) Lista de indicadores para realizar las mediciones del proceso.	Análisis de la información recopilada de las empresas nacionales de desarrollo de software.

CAPITULO IV ANÁLISIS DE RESULTADOS Y APORTES

4.1. Análisis de Resultados y Aportes

En este capítulo, se detalla el análisis de los resultados obtenidos de las entrevistas, búsqueda de información, la observación y encuesta realizada a los colaboradores de distintas organizaciones desarrolladoras de software en Costa Rica, quienes han colaborado de manera atenta para realizar esta investigación.

Se detalla información relevante, mediante la obtención de datos e información clave, que permite determinar el estado de la situación actual.

Se toma en cuenta factores como: conocimiento y experiencia de los colaboradores, puesto que desempeña, así como estructura organizacional de los departamentos para los que laboran, políticas de la empresa, normativa legal y filosófica de cada una de las organizaciones, lo anterior permite conocer lo que se requiere para la toma de decisiones durante la conformación de la guía y así definir un proceso integrado de aseguramiento de la calidad de software.

Se determina, además, la importancia identificada por las empresas nacionales de mantener un proceso de aseguramiento de la calidad de software en sus procesos de desarrollo.

De las personas encuestadas, la mayor parte indica mantener procesos de aseguramiento, los cuales se estructuran de acuerdo con la experiencia de sus colaboradores, en algunos casos, de manera empírica, ya que no cuentan con aseguradores de software que les colaboren en el proceso. En consecuencia, muchas empresas proponen su proceso de aseguramiento y lo adecuan teniendo como resultado que los miembros de la organización identifican la necesidad de

mejora dentro del mismo, mas no identifican la ruta a seguir para implementar el proceso y darle continuidad para tener los mejores resultados.

Durante la búsqueda de información, una de las falencias que indican es la falta de una guía que estructure dicho proceso e indique el camino a seguir para una buena implementación e integración con los demás procesos y herramientas paralelos, además de permitir la gestión y el seguimiento al proceso de desarrollo de software.

Mediante los instrumentos de recopilación de la información se ha alcanzado el acceso a datos de interés, dónde destaca el conocimiento del personal en cuánto a interés individual, características, herramientas o técnicas utilizadas, esto con el fin de identificar, lo que actualmente poseen las empresas desarrolladoras de software en el territorio costarricense.

Se pretende, además, que, a partir de los datos obtenidos, se muestre la verdadera necesidad de las empresas desarrolladoras de software respecto de un marco de trabajo o bien, una guía que controle sus funciones en cuanto al aseguramiento en la calidad del software.

Con la información obtenida de las encuestas, es clara limitante que, para la adopción de procesos de aseguramiento de la calidad de software, la poca información e inexperiencia del personal, unida a una cultura organizacional, cambiaría solo con un instrumento que guie en la conformación de un buen proceso y que permita el conocimiento a los miembros de la organización.

En este sentido hay factores clave como: reuniones de implementación, así como factores técnicos tales como definición de tiempos de entrega, conformación de sprint y el planteamiento de roles y actividades para cada uno

de los miembros del equipo multidisciplinario planteados por Scrum (en este caso).

La propuesta plantea la homologación de la práctica de aseguramiento que permita la definición en el actuar de cada uno de los miembros del equipo frente a las labores que desarrolla, además la adopción de prácticas ágiles, donde se incluyan y definan procesos bajo conceptos de entrega continua; donde se definan prácticas multidisciplinarias.

Otro factor es la definición de procesos de planificación de trabajo, donde cada miembro del equipo define sus labores para que mediante diversas técnicas se realice la sincronización de tareas que permitan el desarrollo participativo de cada uno de los miembros del equipo.

Para ello es importante contar con una herramienta que permita visibilidad del proceso completo, configurando repositorios de necesidades y agrupando elementos de trabajo, el cual es propuesto en el marco de trabajo como sprint y el mismo contiene lo que se está ejecutando, así como su avance; dando mayor visibilidad a procesos de entrega continua, que permita la gestión de la planificación de tareas y que sea visible por el equipo de trabajo y lograr su sincronización.

Otro aspecto importante es identificado por las entrevistas realizadas a expertos del área, donde se identifica a *ISTQB* (Junta Internacional de Calificación de Pruebas de Software por sus siglas en inglés) como uno de los entes más importantes dentro de procesos de certificación de personal de aseguramiento en la calidad del software, dónde la entidad da como base de

conocimiento el documento llamado “*syllabus*”, el cual contiene información en los procesos de pruebas de software.

4.1.1. Sobre la referencia de conocimiento de metodologías de desarrollo

De la encuesta se resalta la información relativa al conocimiento que poseen los colaboradores sobre el tema de metodologías de desarrollo, aspecto de importancia en el producto final. Para la aclaración sobre este concepto, se solicitó información a la población, quienes emitieron su respuesta según se observa en el gráfico número 1.



Gráfico 1: Experiencia en encuestas sobre metodologías de desarrollo

Tal y como se puede apreciar en el gráfico 1, se destaca que, casi la mitad (48%) de los colaboradores encuestados han participado en encuestas referentes a metodologías de desarrollo, este resultado permite medir la experiencia y el conocimiento de la población.

Además de medir la importancia que ha tomado en las áreas de desarrollo de software costarricenses la implementación de metodologías ya que si se han realizado este tipo de encuestas en anterioridad se podría deducir que ha sido en busca de conocer aspectos relevantes que ayuden a definir la selección de una metodología acorde con el desarrollo de software de nivel nacional.

Esta experiencia beneficia directamente esta investigación, asumiendo que el responder este tipo de encuesta, influye en la búsqueda de información de las personas participantes en las mismas y da iniciativas en la búsqueda de información.

4.1.2. Sobre las Características del Entorno Empresarial

De la información recopilada de las encuestas, sobre la madurez que poseen las representadas en el entorno empresarial, aspectos de importancia para el desarrollo del producto final, se detallan en el gráfico número 2, de igual manera el número 3, aportará la información que ayuda en la definición del tamaño de las representadas en la que participan los colaboradores.

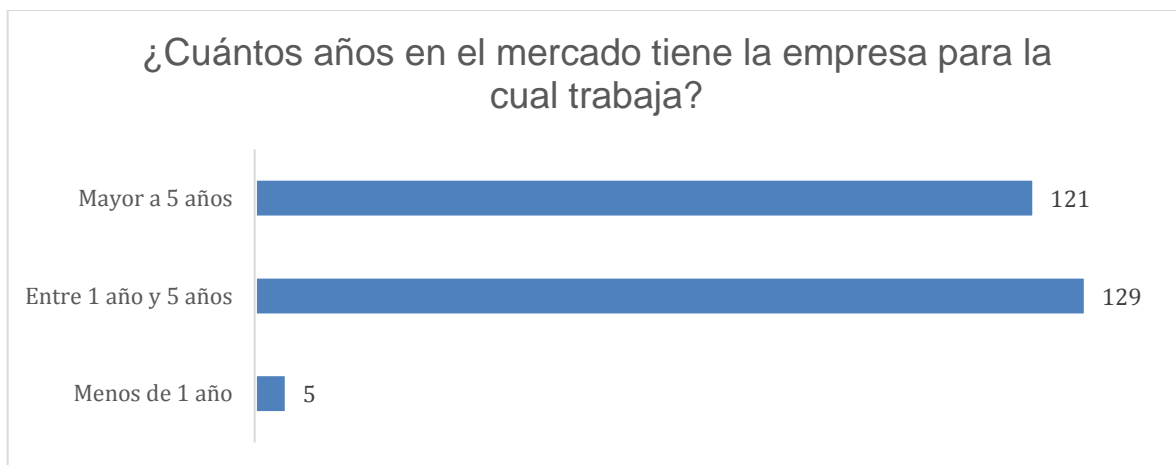


Gráfico 2: Situación actual de las empresas en Costa Rica.

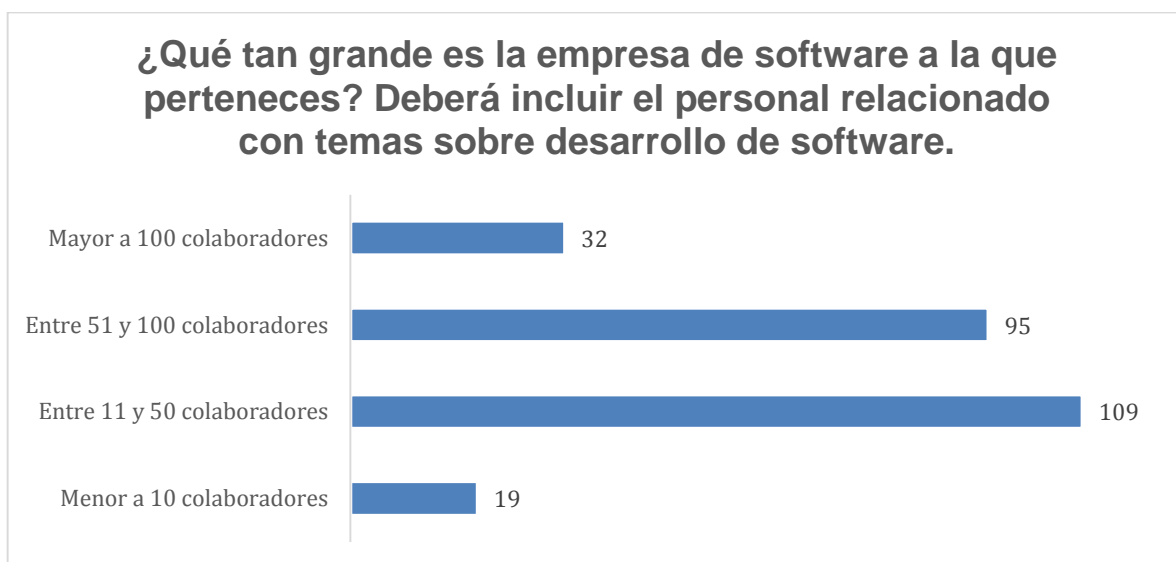


Gráfico 3: Estimado de colaboradores que laboran en empresas de Software.

Como se observa, en el gráfico número 2, las empresas cuya experiencia está entre 1 y 5 años en el mercado laboral (50% de las empresas de los encuestados), son empresas con mayor trayectoria en el desarrollo de software a nivel nacional; sin embargo, un gran porcentaje 49% de las compañías para las que laboran los encuestados, manifiestan que son empresas que poseen mayor tiempo implementando el desarrollo de software en el mercado costarricense.

El gráfico número 3, se indica que las empresas que poseen colaboradores con personal relacionado en temas de desarrollo, (empresas que cuentan entre 11 y 100 colaboradores), son empresas medianas involucradas en los procesos de desarrollo de software, que dan valor importante para la investigación, ya que existen muchas empresas “medianas” que participan en el proceso de desarrollo de software en el territorio nacional, lo cual da una línea de experiencia con mucha fortaleza y la importancia de incorporar mejoras en el corto plazo, ya que este tipo de empresa cuenta con la particularidad de cambio en el corto plazo y tendrían un apoyo importante con el aporte realizado en esta investigación.

4.1.2.1. Descripción de los roles utilizados

De las encuestas se resalta la información relativa a los roles utilizados dentro de las empresas desarrolladoras de software en Costa Rica, aspecto que se destaca por cuanto posee gran influencia en el producto final de esta investigación. Para brindar un mejor concepto, se ha solicitado información a la población, quienes emitieron su respuesta según se muestra en la gráfica número 4, mientras que en el gráfico número 5 se indica la cantidad en años que poseen estos colaboradores en el rol asignado.

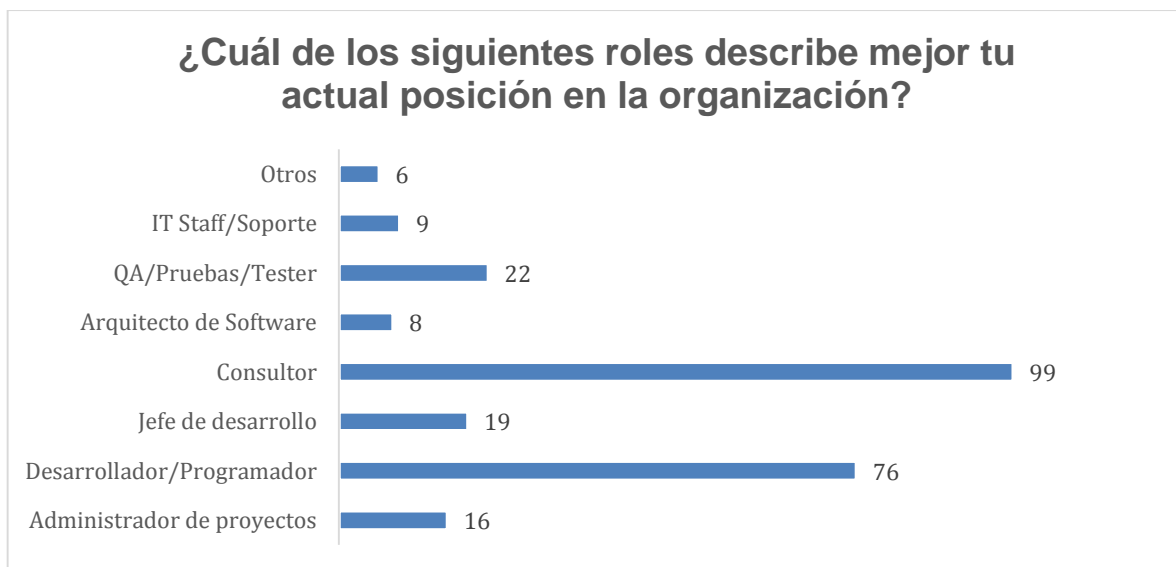


Gráfico 4: Responsabilidades de trabajo dentro de la organización.

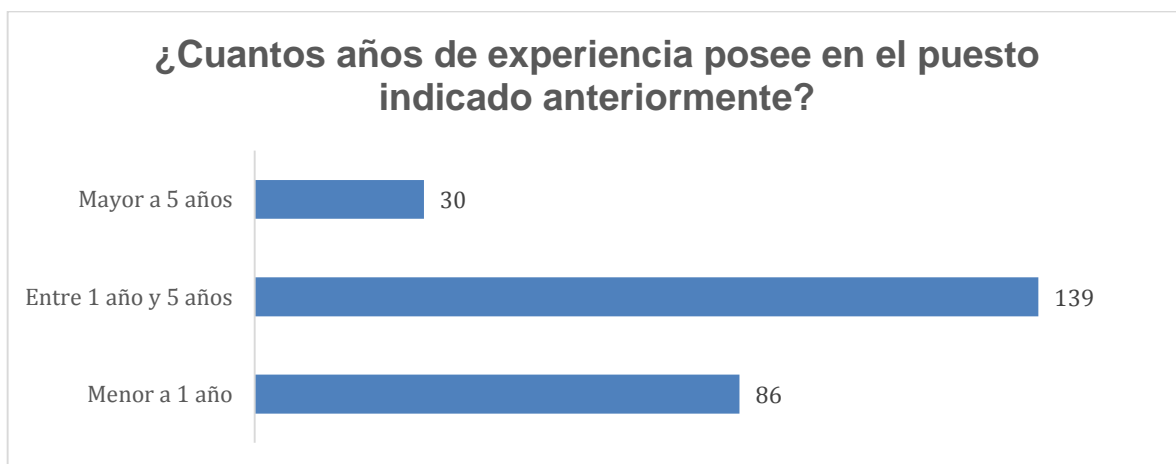


Gráfico 5: Años de experiencia en puestos.

En el gráfico 4 se observa que la mayor concentración de información apunta hacia la consultoría de información con un 39%, luego están los desarrolladores quienes representan un 30%, lo que podría indicar que muchas de las empresas desarrolladoras de software se especializan en la venta de servicios en temas de consultoría, es decir aunque poseen equipos dedicados al desarrollo de software, poseen además dentro de su catálogo de “productos” el brindar consultoría a los departamentos de desarrollo de otras organizaciones.

Por este motivo, la aplicación de buenas prácticas de desarrollo de software e implementación de procesos de aseguramiento de la calidad de software vienen a reforzar el producto final. Lo anterior tomando como referencia la transformación a la cual se enfrentan las empresas en la implementación de *outsourcing*, permitiendo mantener una definición adecuada de los procesos de desarrollo de software, ya que éstas se enfrentan a retos de mayor rotación de personal y cuidar el producto final ya que puede estar expuesto a procesos legales.

4.1.2.2. Experiencia de las Personas Encuestadas en Metodologías ágiles

De las encuestas realizadas se destaca con gran importancia aspectos relacionados con la experiencia que poseen los colaboradores en temas de metodologías ágiles, misma información que se observa en el gráfico número 6.

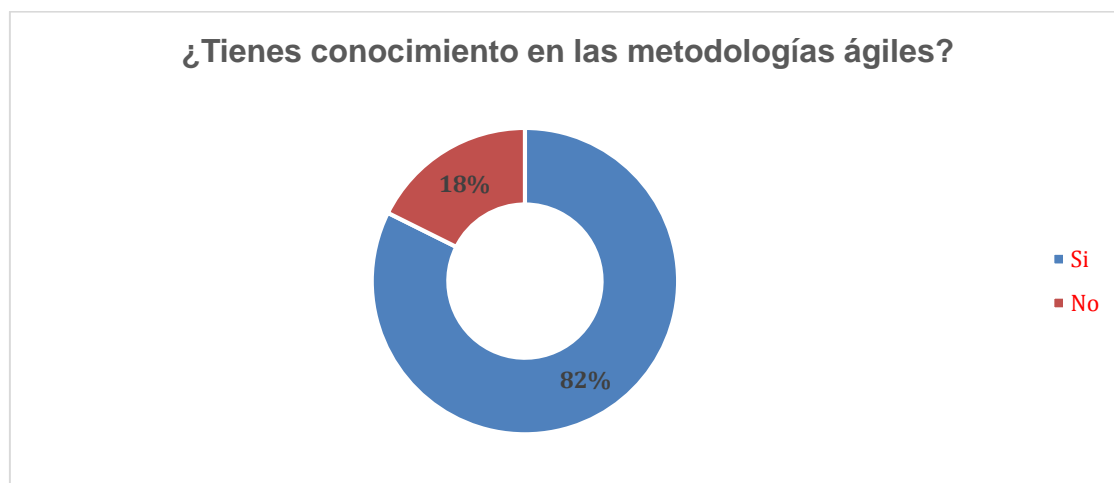


Gráfico 6: Conocimiento sobre metodologías ágiles.

La mayor parte de la muestra (82%) indica tener conocimiento en metodologías ágiles, lo cual representa la importancia y fortaleza que han tomado

los marcos de trabajo ágil en el mercado de desarrollo de software en Costa Rica, donde según los resultados obtenidos se observa claramente la fortaleza y liderazgo de los procesos ágiles en los departamentos de desarrollo de software.

Con base en los resultados se valora cual es el grado de madurez que podrían tener los departamentos en la implementación de metodologías ágiles, basados en esa experiencia y la importancia que se logra identificar, se logra definir que los procesos de aseguramiento de software deben orientarse a la implementación en conjunto con metodologías ágiles.

Sin embargo, la población restante (18%) aunque no poseen conocimiento en las metodologías ágiles lleva su trabajo de manera “tradicional”, basados en prácticas de otros marcos de trabajo.

4.1.2.3. Nivel de conocimiento en metodologías ágiles

De la encuesta se resalta la información relativa al nivel en el conocimiento adquirido en las metodologías ágiles, aspecto que se destaca y que posee gran influencia para el producto final de esta investigación. Para la aclaración sobre este concepto, se solicitó información a la población, quienes emitieron su respuesta según se puede observar en la gráfica número 7. Sin embargo, en la gráfica número 8 se ha solicitado indicar la cantidad en años que poseen sobre el conocimiento indicado.

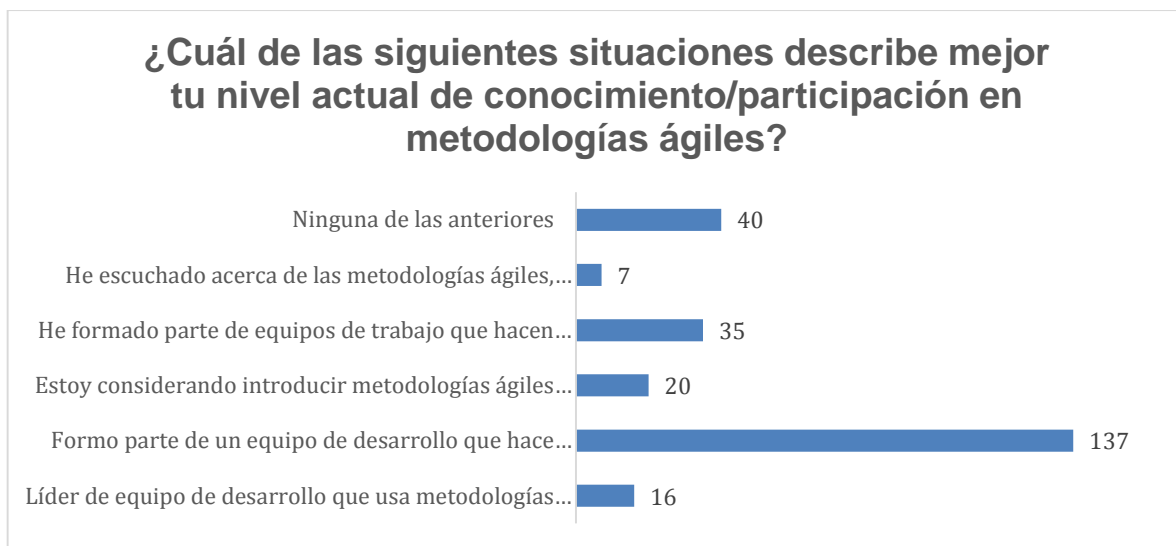


Gráfico 7: Participación en metodologías ágiles.



Gráfico 8: Aplicación metodologías ágiles en el orden laboral y personal.

La mayor parte de la población encuestada indica que forma parte de un equipo que implementan o han implementado metodologías ágiles en sus trabajos, pero se observa que se indica que la mayor parte de la población ha aplicado la misma por un tiempo no mayor a un año a lo cual se identifica la tendencia de cambio que han tenido muchas empresas en los últimos años.

4.1.2.4 Uso de Metodologías Ágiles dentro de la empresa a la que laboran

De la encuesta se resalta la información relativa al uso de metodologías ágiles dentro de las empresas para las que laboran actualmente, aspecto que se destaca por cuánto se relevante en el producto final. Para aclarar estos conceptos, se solicitó la información correspondiente a la población, quienes emitieron su criterio según lo demuestra los gráficos número 9, 10 y 11 siguientes.

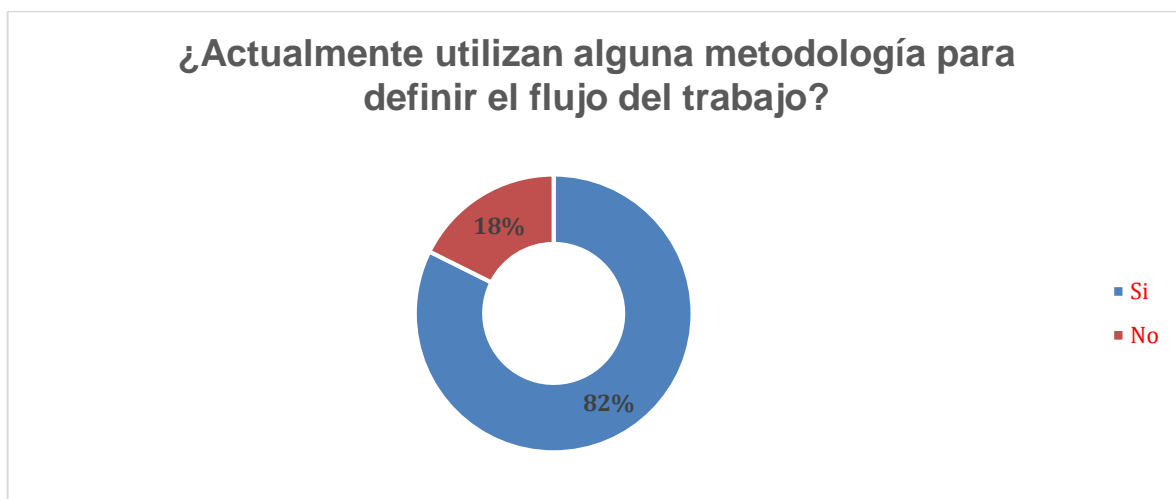


Gráfico 9: Actualmente utilizan metodologías ágiles en las empresas.

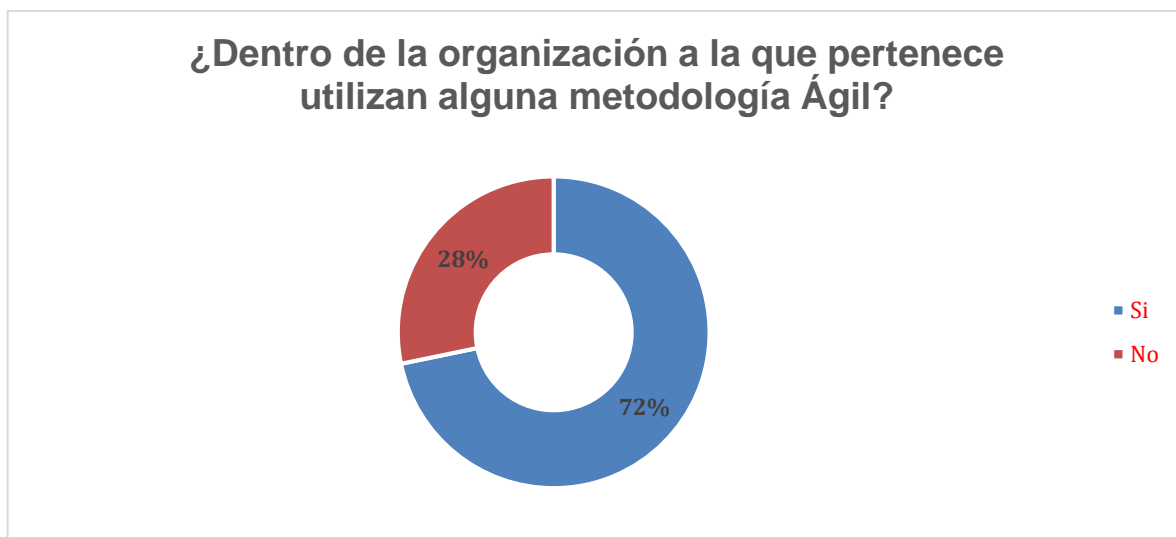


Gráfico 10: Utilización de metodologías ágiles en las empresas.

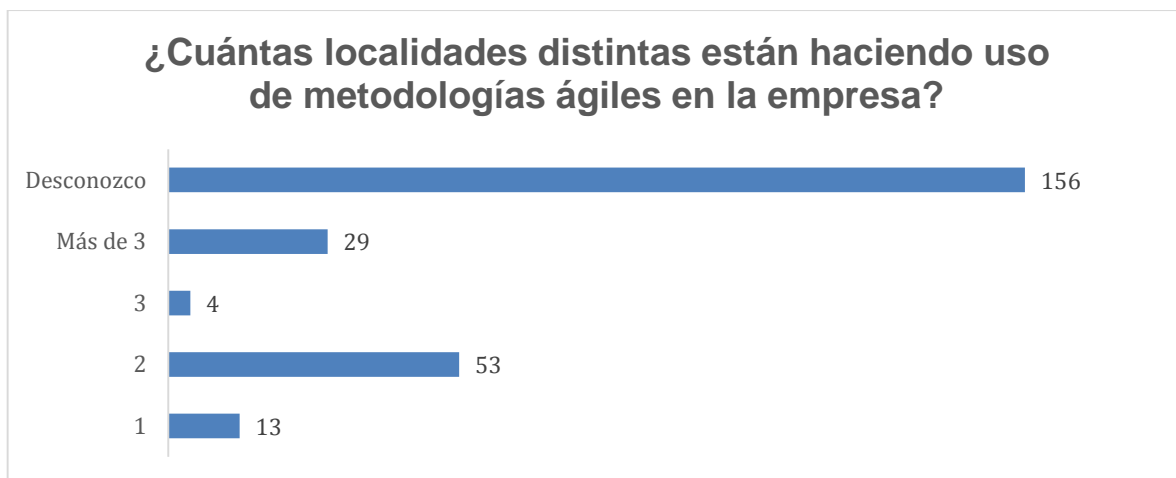


Gráfico 11: Localidades que utilizan metodologías ágiles en las empresas.

La mayor parte de la población (82%) manifiesta que, actualmente, están utilizando alguna metodología ágil, lo hacen para definir el flujo de trabajo lo que permite suponer que es de interés especial para los profesionales de hoy contar con este conocimiento, lo que permite avanzar de manera eficiente en el desarrollo de los proyectos.

En la gráfica número 10, se demuestra que un 72% de la población indica que utilizan las metodologías ágiles, por lo que es de suponer que un gran porcentaje de las empresas están contribuyendo en el aprendizaje de estas técnicas para el desarrollo de procesos, valor que da fortaleza al resultado obtenido, ya que permite conocer que hay empresas costarricenses que fomentan el uso de una metodología específica.

En la gráfica número 11, poco más de la mitad de los colaboradores (62%) indica que desconoce las localidades en las que la empresa está haciendo uso de las metodologías ágiles, lo que deja suponer que solamente tienen

conocimiento sobre su área de trabajo, o bien, se puede presentar el factor de comunicación que puede ser escaso o de mala calidad.

4.1.3. Sobre la selección del marco de trabajo

En búsqueda de información relacionada con los marcos de trabajo que se aplican en las empresas desarrolladoras de software costarricense, se solicitó información a la población encuestada, quienes emitieron su respuesta según se muestra en el gráfico número 12 donde se busca identificar las metodologías más utilizada en el mercado costarricense.

En cuanto a la mejora continua, el gráfico número 13 determina el conocimiento en cuánto si los funcionarios manejan roles específicos para el desarrollo de las actividades, mientras que en el gráfico número 14 se busca la información para determinar si las empresas utilizan algún método para mejorar y desarrollar el proceso actual.

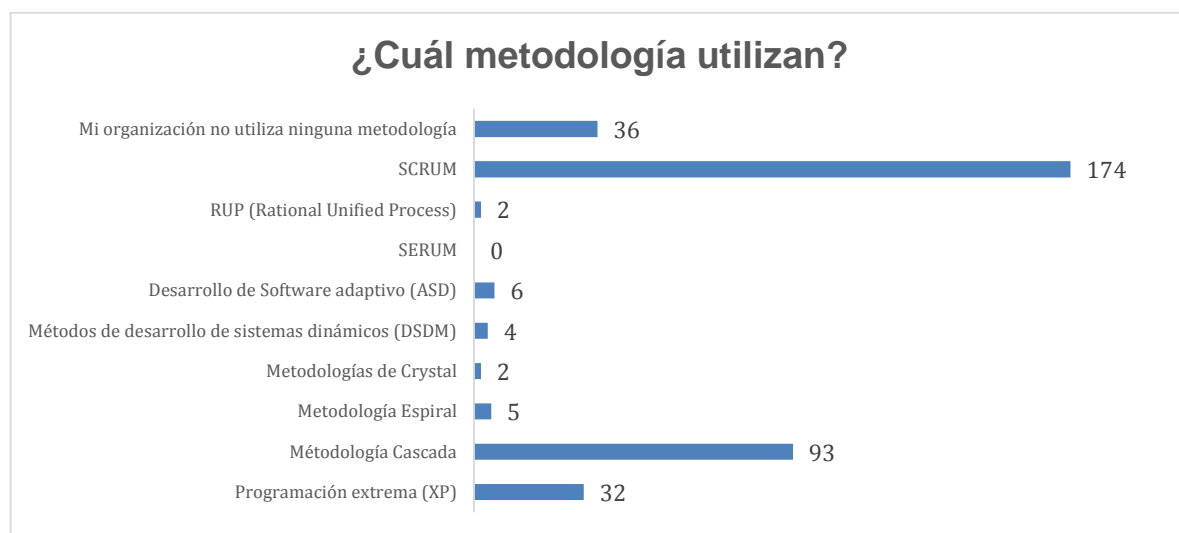


Gráfico 12: Metodologías ágiles implementadas en las empresas.

¿Los funcionarios manejan roles definidos para el desarrollo de las actividades?

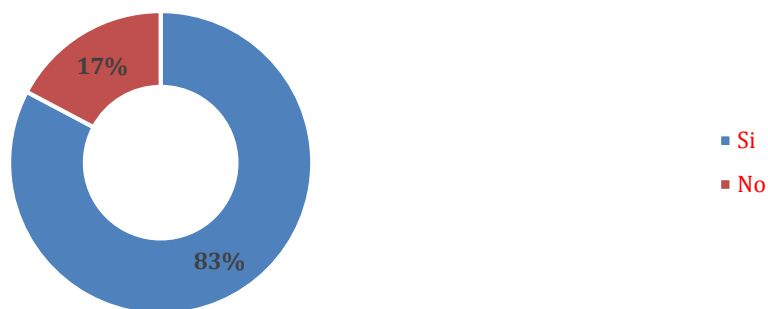


Gráfico 13: Responsabilidades definidos en desarrollo de actividades.

¿Tienen algún método para mejorar y desarrollar el proceso actual, bajo un concepto de mejora continua?

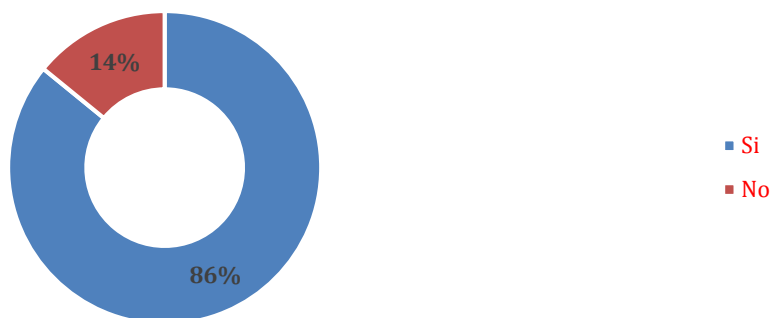


Gráfico 14: Utilización de métodos para mejorar y desarrollar los procesos actuales.

La mayor parte de la población encuestada, indica que el marco de trabajo *Scrum* es el *framework* más utilizado (gráfico número 12), para definir los procesos de desarrollo de software, adoptados por las empresas costarricenses dentro de sus procesos, dónde se destaca la fortaleza sobre las metodologías cascada (37%) mismas que se supone que eran las que poseían mayor fuerza de implementación en las organizaciones nacionales antes del cambio a marcos de trabajo ágiles.

Se realizaron preguntas de tendencias que fortalecen el uso del *framework* “*Scrum*” observando como muchas de las empresas que adoptan la tendencia, fortalecen la aplicación con conceptos modernos de entrega continua, en la ejecución de planificación de labores realizadas, donde se fortalecen temas como el trabajo en equipo, también se identifica que el uso del marco de trabajo es visto como positivo por los equipos que aplican dicha metodología.

En cuanto a si los funcionarios manejan los roles para el desarrollo de actividades (gráfica 13), la población determina en un gran porcentaje (83%) que efectivamente se hace este manejo en las actividades que ejercen y tienen claridad del rol que cumplen; a lo cual se puede suponer que marca un orden de procedimientos y actores que llevan hacia el control de la aplicación del marco de trabajo.

También se realizan consultas sobre la aplicación de procesos de mejora continua que permite tener un mejor control y ejercer métodos para el mejoramiento en el desarrollo de los procesos que actualmente se presentan en las organizaciones bajo el concepto de mejora continua (gráfica número 14), dando como exitosa la concentración de información con un 86% de las personas encuestadas, quienes afirman que manejan el concepto de mejora continua en sus organizaciones, afirmación que refuerza el uso del marco de trabajo Scrum ya que el mismo implementa reuniones orientadas a procesos de mejora continua.

4.1.3.1. Planificación de Tareas en las Oorganizaciones

De la encuesta, resalta la información relativa a la planificación de tareas en las organizaciones, aspecto que brinda valor agregado al producto final de la investigación. Para la aclaración de este concepto, se solicitó la información a la población, quienes generosamente emitieron su criterio según se aprecia en la gráfica número 15.

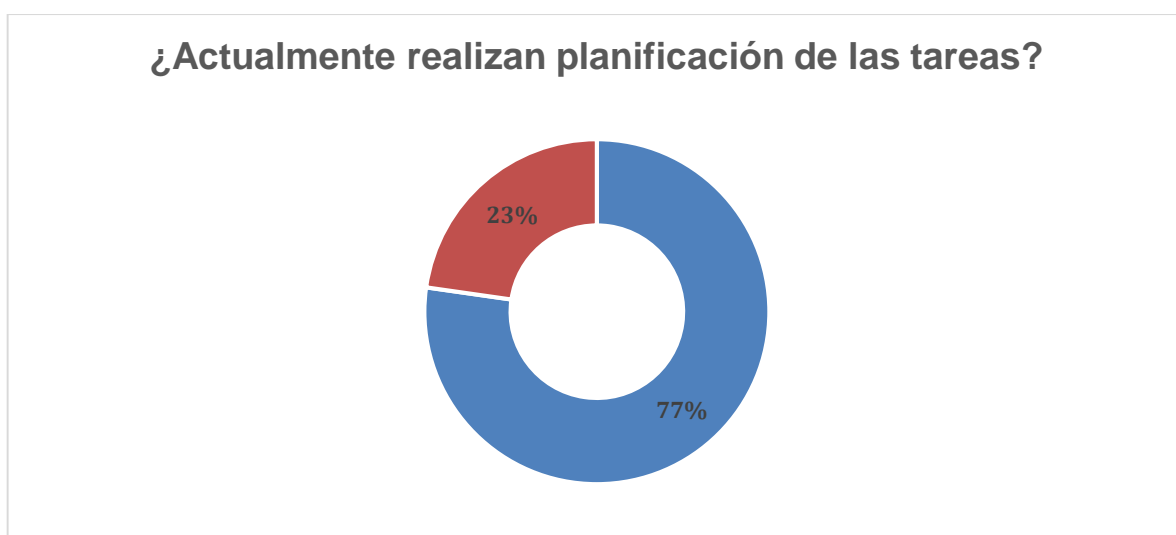


Gráfico 15: Planificación de tareas en las empresas

Basados en este resultado dentro de los procesos de desarrollo de software que actualmente se aplica, se identifica la importancia de la planificación de tareas, dónde las empresas en su mayoría (77%) lo aplican logrando mayor eficiencia dentro del desarrollo de software.

4.1.3.2. Visión del Entorno sobre las metodologías Aplicadas

Se recopila también información relacionada a las metodologías aplicadas en el entorno de desarrollo de software, aspecto que brindar un valor agregado al producto final.

¿Actualmente, el equipo ve como positiva la metodología aplicada?

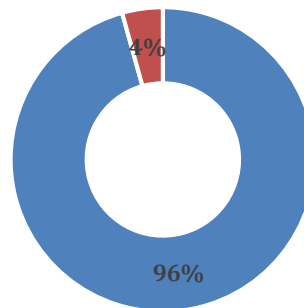


Gráfico 16: ¿La metodología aplicada actualmente es funcional?

Un aspecto importante es como la mayor parte de las personas encuestadas (96%) ven como positivo la metodología aplicada, lo que da fuerza a la aplicación del marco de trabajo ágil “Scrum” mismo que tiene mayor apoyo en la encuesta, otro punto importante es la fortaleza brindada a los procesos de mejora continua, planificación del trabajo y la definición de roles dentro de este.

Por otro lado, se identifica que son poco los encuestados (4%) los que ven la aplicación de otros tipos de marco de trabajo, como buenas prácticas de desarrollo, en algunos casos por el miedo al cambio o falta de información que no les permite llevarlos hacia los procesos de innovación.

4.1.3.3. Dificultades de implementación de Metodologías Ágiles

De la encuesta se resalta la información relativa en cuanto a las dificultades de implementación que presenta la temática de metodologías ágiles, aspecto que viene a brindar un valor agregado al producto final de la investigación. Para la aclaración de este concepto, se solicitó información a la

población, quienes generosamente emitieron sus criterios, según se puede apreciar en la gráfica número 17.

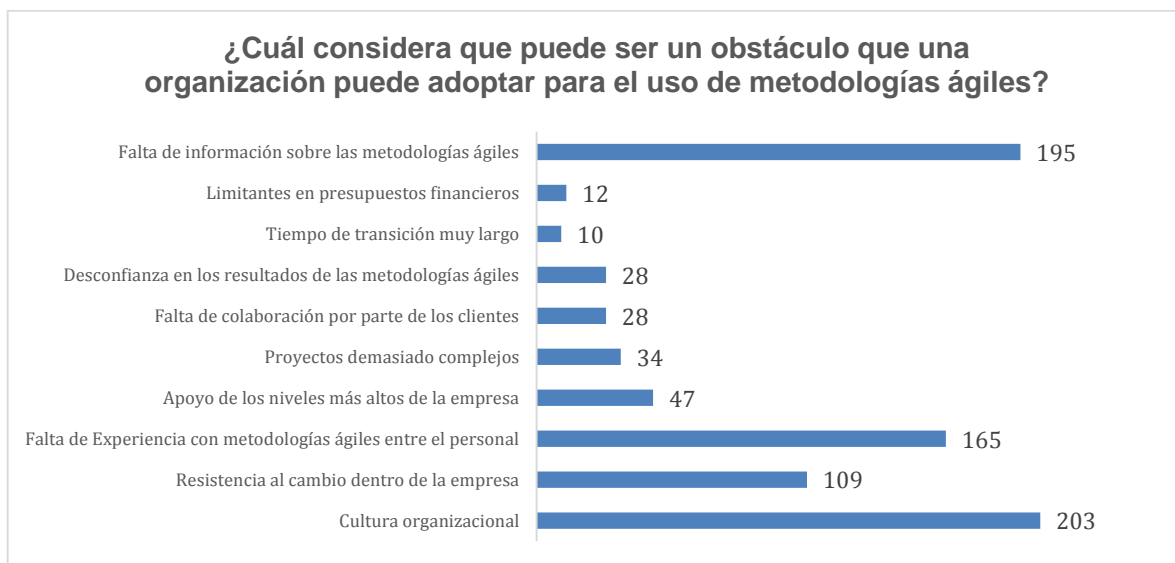


Gráfico 17: Obstáculos para implementar las metodologías ágiles.

Se define la falta de información, la falta de experiencia y la cultura organizacional, como las dificultades presentes durante la implementación de marcos de trabajo ágiles, la ejecución de cambios en organizaciones debe ser algo planificado para lograr el éxito.

En la implementación de metodologías no es sólo el cambio en los procesos, es el cambio cultural que se genera, ya que tanto los procesos como la técnica, es gestionado dentro de los colaboradores para lograr el éxito de la implementación.

Aspectos como la falta de información y la falta de experiencia, son aspectos que se podrían mejorar con el desarrollo de una guía especializada, la cuál sería fuente de información y ayudaría a comprender esos procesos y su definición.

4.1.3.4. Conocimiento en proyectos exitosos utilizando metodologías ágiles

De la encuesta, resalta la información relativa en cuánto el conocimiento de los proyectos exitosos con metodologías ágiles dentro de las organizaciones, aspecto que brindar un gran valor agregado al producto final. Para la aclaración de este concepto, se solicitó la información a la población, quienes generosamente emitieron sus criterios, según se puede apreciar en la gráfica número 18.



Gráfico 18: Proyectos exitosos con metodología ágil.

Gran parte de la población encuestada indica que desconoce el dato de casos de éxito en proyectos trabajados con el marco de trabajo ágil (55%), lo que permite suponer que en muchos de los casos son empresas que recién están iniciando en procesos ágiles o que mantienen una cantidad limitada de proyectos dentro de su marco de implementación, teniendo como objetivo implementar la totalidad de sus desarrollos en su respectivo marco de trabajo en un plazo determinado.

Se identifica que como punto de mejora para las empresas que aplican dichos procesos, es la distribución de la información a los equipos que participan

en ellos, donde se debería mantener informados a los equipos del avance de las implementaciones y los casos de éxito por una parte y la retrospectiva de proyectos, para determinar puntos de mejora en la técnica que se ejecuta con ellos, lo que lograría que el personal se mantenga informado de los resultados de la técnica que aplican.

4.1.3.5. Razones para adoptar Metodologías Ágiles dentro de una Organización

De la encuesta se resalta la información relativa en cuanto a las razones principales para adoptar metodologías ágiles como el conocimiento de los proyectos exitosos con metodologías ágiles dentro de las organizaciones, aspecto que viene a brindar un gran valor agregado al producto final de la investigación. Para la aclaración de este concepto, se solicitó la información a la población, según se aprecia en la gráfica número 19.

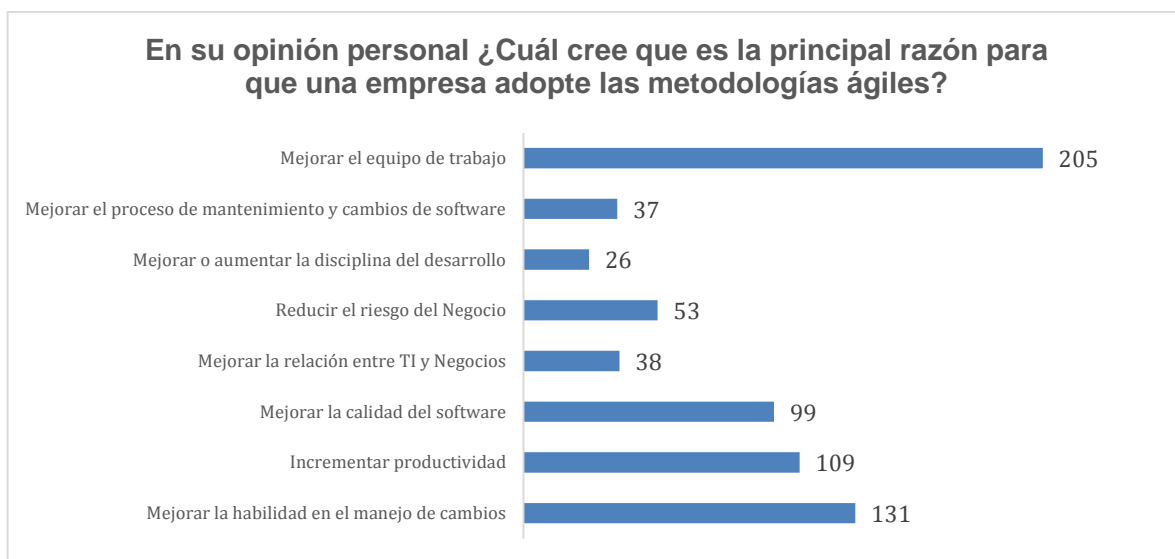


Gráfico 19: Opiniones sobre incorporación de metodologías ágiles dentro de las empresas.

Dentro de las mejoras obtenidas por la aplicación de metodologías ágiles, las mejoras percibidas por el equipo de trabajo según los resultados de la encuesta es la mejora del trabajo en equipo, la mejora en la habilidad del manejo de cambios, productividad y la mejora en la calidad de los productos, aspectos positivos y que se puede deducir, son parte de los elementos que las empresas valoran para tomar la decisión de cambio.

En vista que durante el proceso de cambio realizados por las empresas en la aplicación de metodologías ágiles uno de los cambios definidos en dicha disciplina es la inclusión de algún proceso de aseguramiento de software, los miembros de los equipos asocian como mejora inmediata la calidad del software, la aplicación de la técnica en conjunto.

4.1.4. Sobre el Aseguramiento de la Calidad de Software

Mediante la encuesta se recopiló información sobre el criterio de quienes participaron, que forman parte del proceso de desarrollo de software. De los resultados se identifica que los procesos de aseguramiento de la calidad del software, deben someterse a procesos de mejora continua, aspectos que brindar un gran valor agregado al producto final de la investigación. Los encuestados emitieron sus criterios según se puede apreciar en la gráfica número 20 y 21.

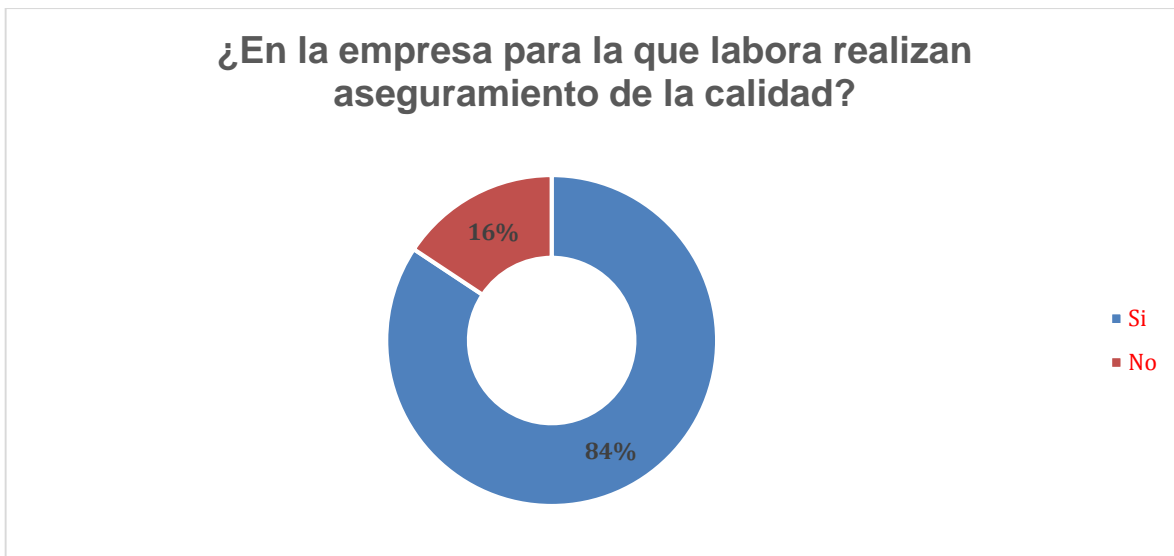


Gráfico 20: Aseguramiento de la calidad del software dentro las empresas.

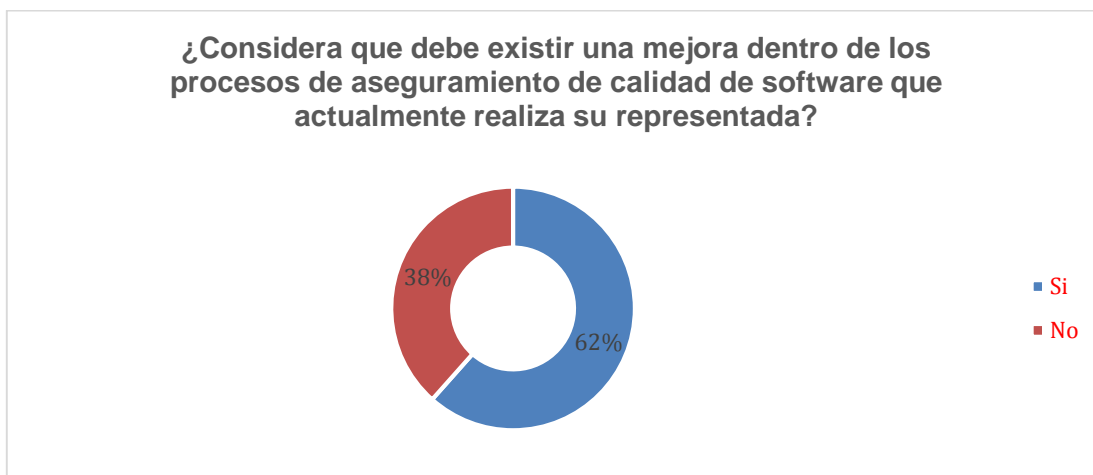


Gráfico 21: Mejora dentro de los procesos de aseguramiento de calidad del software.

Las personas encuestadas en su mayoría (84%) indica que dentro de la organización a la que pertenecen implementan algún proceso de aseguramiento de la calidad de software, pero también en su mayoría (62%) indica que los procesos actuales que aplican, deben implementar una mejora ya que detectan deficiencias en los procesos. Lo anterior permite, suponer que muchos de estos son procesos empíricos, que se adaptan a los procesos que las empresas implementan y a la forma en que se está desarrollando.

Basado en ese indicio se identifica la posibilidad de mejora ya que los mismos no mantienen una estructura de implementación, ni de ejecución. Se basan en los procesos de pruebas donde dan fortaleza las pruebas realizadas por usuarios finales y no la ejecución adecuada de un equipo que realice pruebas formales durante el desarrollo de software.

4.1.4.1. Consideración para que exista una guía que facilite la Integración de los Procesos

De la encuesta se resalta la información relativa en cuanto a si es necesario la existencia de una guía que facilite la integración de los procesos de aseguramiento de calidad del software a las metodologías actuales, aspecto que brindan un gran valor agregado al producto final de la investigación. Para la aclaración de este concepto, se solicitó la información a la población, según se aprecia en la gráfica número 22.

Gráfico 22: Existencia de guía para integrar los procesos de aseguramiento en las metodologías actuales.



La mayor parte de la población concuerda que debe existir una guía que permita la integración del proceso de aseguramiento con los procesos actuales propuestos por las metodologías ágiles en la implementación.

Esto se fortalece dado que en la encuesta se observa que la mayor parte de las personas definen que debe existir una mejora en los procesos de aseguramiento.

Se podría interpretar que muchos de los procesos son empíricos, aspecto que existiendo una instrumento que ayude al fortalecimiento de los procesos actuales y apoye nuevas empresas que quieran unirse al cambio e implementar procesos de aseguramiento de calidad en las áreas de desarrollo de software, daría como resultado procesos que de alguna manera se homologan entre las empresas y el fortalecimiento de la experiencia de las personas que lo aplican; además, de servir de marco de referencia a personas que toman el rol de guía o revisor de los procesos.

4.1.4.2. Madurez del proceso de aseguramiento de la calidad de software

De la encuesta se resalta la información relativa en cuanto a la madurez de los procesos de aseguramiento de la calidad de software dónde se enfoca la realización de pruebas formales y si las mismas son documentadas, aspectos que brindan un gran valor agregado al producto final de la investigación. Para la aclaración de estos conceptos, se solicitó la información a la población, según se aprecia en la gráfica número 23 y gráfica número 24.

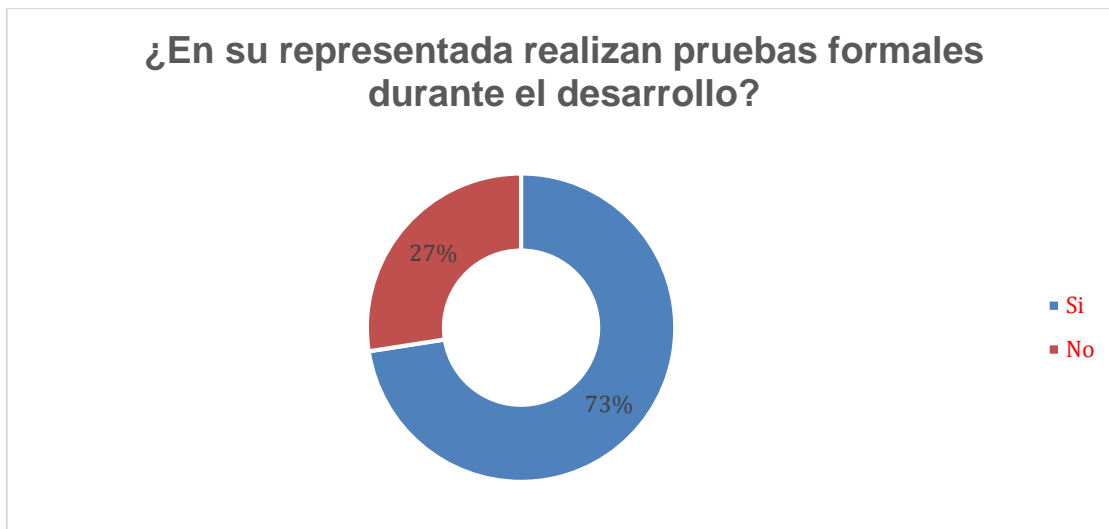


Gráfico 23: Realización de pruebas formales durante el desarrollo.

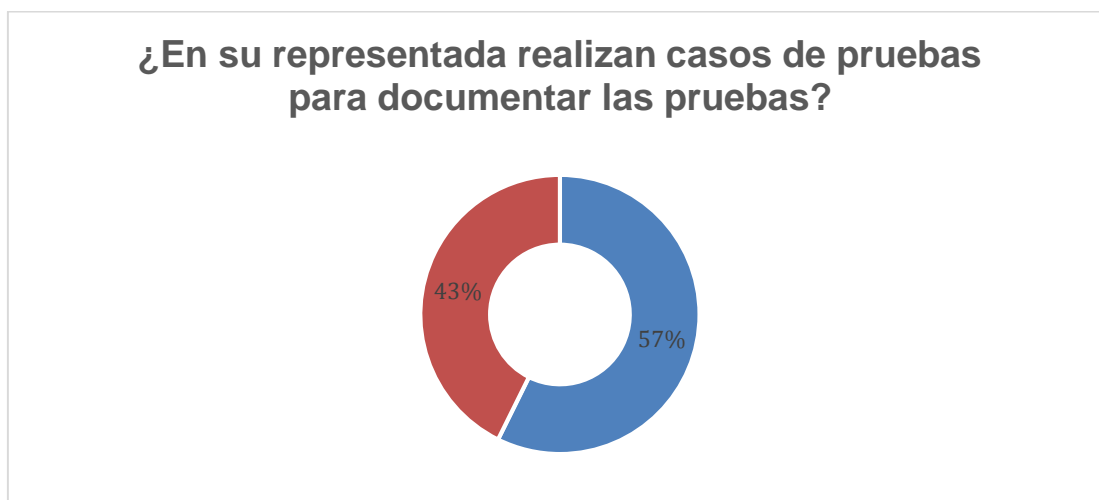


Gráfico 24: Realización de casos de prueba para documentar pruebas.

Se puede decir que casi tres cuartas partes (73%) de la población encuestada realizan pruebas formales durante el desarrollo, sin embargo, solamente un poco más de la mitad de las empresas realizan pruebas sin la factibilidad de poder documentarlas. Esto indica que gran cantidad no realiza el proceso de documentación correctamente, lo que se podría deber a la ausencia de una guía que conforme el proceso y marque el rumbo para tener una cultura de aseguramiento en la calidad del software de forma completa.

Desde luego, en este punto influye el tema de cultura organizacional, resistencia al cambio e inclusive, la inexperiencia por parte del personal a cargo, que a su vez puede estar sujeta a falta de información en cuanto a temas de aseguramiento en la calidad del software, donde se implementa un proceso sin apoyo de perfiles expertos o con métodos de aplicación que no fortalecen la ejecución o herramientas que no benefician la documentación de las pruebas y el diseño de casos de prueba que ayuden a tomar evidencia del proceso de aseguramiento de la calidad.

En cuanto a la documentación de las pruebas, el resultado es favorable (57%), más de la mitad manifiesta que sí realiza la documentación de las pruebas, lo que es un número positivo, porque se considera que en caso de una posible falla se puede verificar la documentación de dicho proceso, más falta definir qué tipo de documentación se realiza y como definen la mismas dentro de herramientas que beneficien el proceso.

4.1.4.3. Obstáculos presentados al adoptarse el aseguramiento de la calidad del software

De la encuesta se resalta la información relativa en cuanto a los obstáculos principales para adoptar el aseguramiento en la calidad del software, aspecto que brinda un valor agregado al producto final de la investigación. Para la aclaración de este concepto, se solicitó la información a la población, según se aprecia en la gráfica número 25.

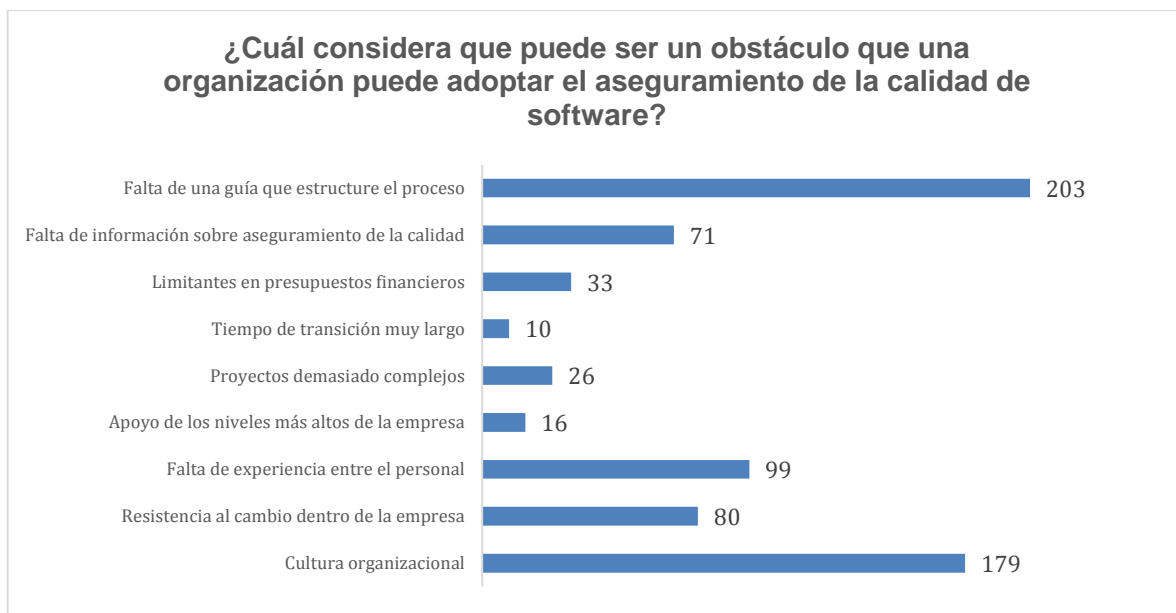


Gráfico 25: Obstáculo para adoptar aseguramiento de la calidad del software dentro de las empresas.

Con el resultado se interpreta que la falta de información es un obstáculo dentro de la implementación del aseguramiento de la calidad de software, ya que la cultura organizacional puede cambiar con una buena fuente de información que ayude a mantener el conocimiento que permita la buena implementación y la seguridad de que se están desarrollando bien los procesos y aplicando bien el aseguramiento.

Una herramienta que ayude en la integración de los marcos de trabajo como *Scrum* (definición de los procesos de aseguramiento de la calidad de manera integrada integración de las herramientas dentro de todo este concepto que beneficie a una buena implementación y definición de aseguramiento de software), beneficiaría en todos los aspectos que hoy son vistos como obstáculos.

4.1.4.4. Documentación de los Resultados Obtenidos en las Pruebas de Desarrollo

De la encuesta se resalta la información relativa en cuanto a si las representadas documentan el resultado obtenido de las pruebas ejecutadas durante la etapa de desarrollo, aspecto que brinda un valor agregado al producto final de la investigación. Para la aclaración de este concepto, se solicitó la información a la población, quienes generosamente emitieron sus criterios según se puede apreciar en la gráfica número 26.

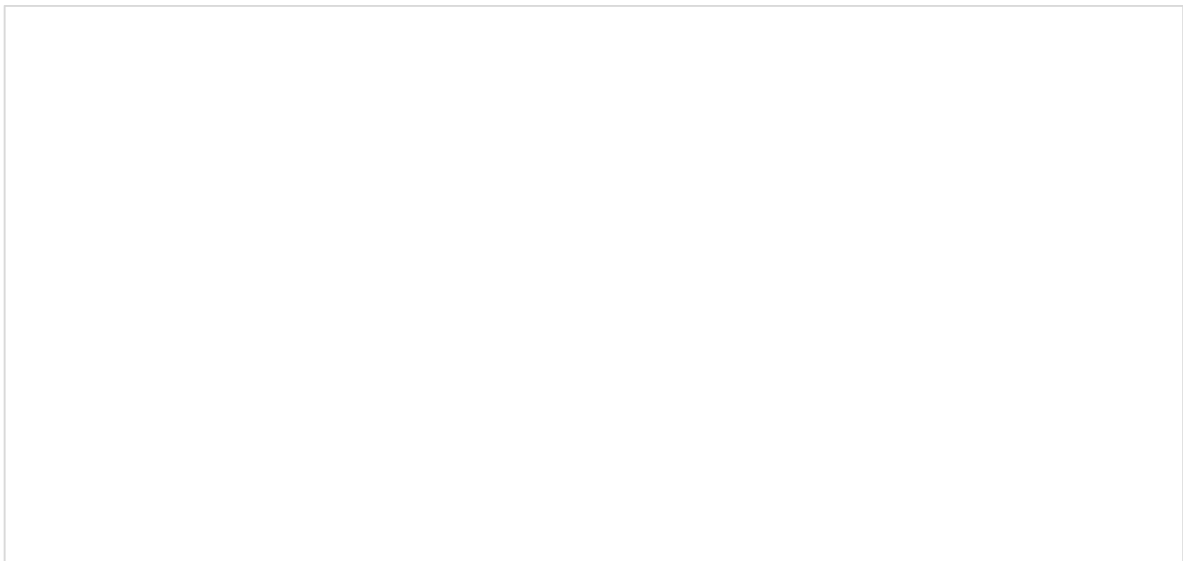


Gráfico 26: Documentación de resultados de pruebas en las empresas.

El 86% de la población encuestada afirma documentar los resultados de las pruebas, aspecto muy importante dentro de un proceso de aseguramiento de la calidad de software, que permiten durante la ejecución de pruebas tomar muestras que afirmen un buen funcionamiento del aplicativo y, por ende, es parte de un buen proceso de aseguramiento.

De ahí se forma la necesidad de seleccionar herramientas que beneficien y potencialicen la documentación de la evidencia conseguida durante la

aplicación del proceso de aseguramiento de la calidad de software, más información ver Marco Contextual, apartado de aseguramiento de la calidad (Análisis de testing y documentación de pruebas).

4.1.4.5. Creación y Planificación de Planes de Prueba

De la encuesta se resalta la información relativa en cuanto a la creación y planificación de los planes de prueba, aspecto que brindar un valor agregado al producto final de la investigación. Para la aclaración de estos conceptos, se solicitó la información a la población, quienes generosamente emitieron sus criterios según se puede apreciar en las gráficas número 27 y gráfica número 28.

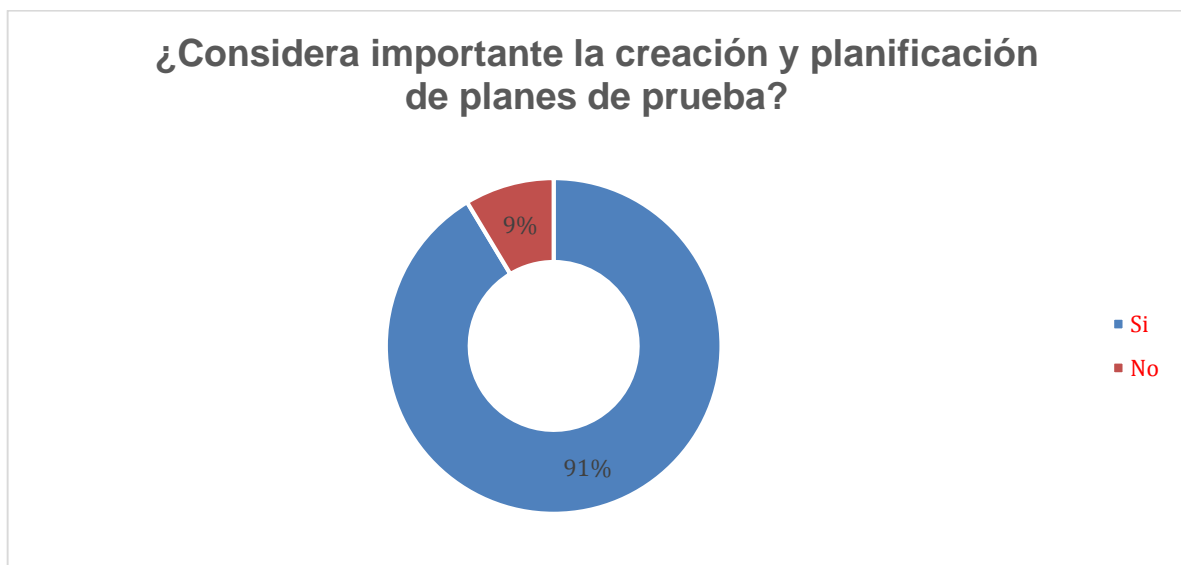


Gráfico 27: Importancia y planificación de planes de prueba.

¿En su representada realizan las diferentes pruebas durante el desarrollo en ambientes definidos para aseguramiento de la calidad?

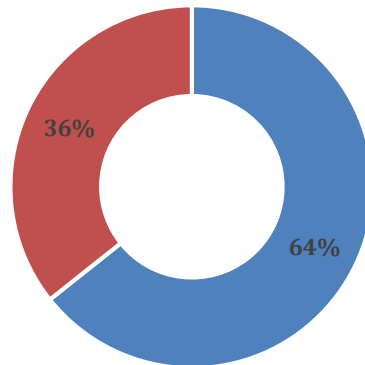


Gráfico 28: Realización de diferentes pruebas durante el desarrollo en ambientes definidos.

La población encuestada afirma casi en su totalidad (91%), que es completamente necesario realizar y planificar los planes de prueba (gráfica 26), los mismos apoyan el valor de ejecutar estos instrumentos y están de acuerdo en la importancia de realizar este proceso. El 9% de los encuestados no apoya la creación y planificación de planes de prueba o lo que permite suponer que no se posee el conocimiento en la aplicación de planes de pruebas en etapas de aseguramiento de software.

Solamente un poco más de la mitad (64%) manifiesta que realizan diferentes pruebas durante el desarrollo en ambientes definidos (gráfica 27), lo que indica que un poco menos de la mitad de la empresa (46%) no planifica bien sus planes de pruebas, ya que no identifican dicha necesidad, a pesar que mencionan que realizan planes de pruebas formales para realizar el aseguramiento software.

La falta de definición de ambientes es un punto de mejora dentro del proceso y por ende se deduce que los planes de pruebas que se crean son deficientes, ya que no identifican las necesidades de un buen proceso de aseguramiento de la calidad de software.

4.1.5. Sobre las Herramientas de Gestión en el Ciclo de Vida del Proyecto

De la encuesta resalta la información relativa en cuanto al uso de herramientas durante la gestión en el ciclo de vida del proyecto, aspectos que brindan un valor agregado al producto final de la investigación. Para la aclaración de estos conceptos, se solicitó la información a la población, quienes generosamente emitieron sus criterios según se puede apreciar en las gráficas número 29 y gráfica número 30.



Gráfico 29: Utilización de herramientas para la gestión de desarrollo.

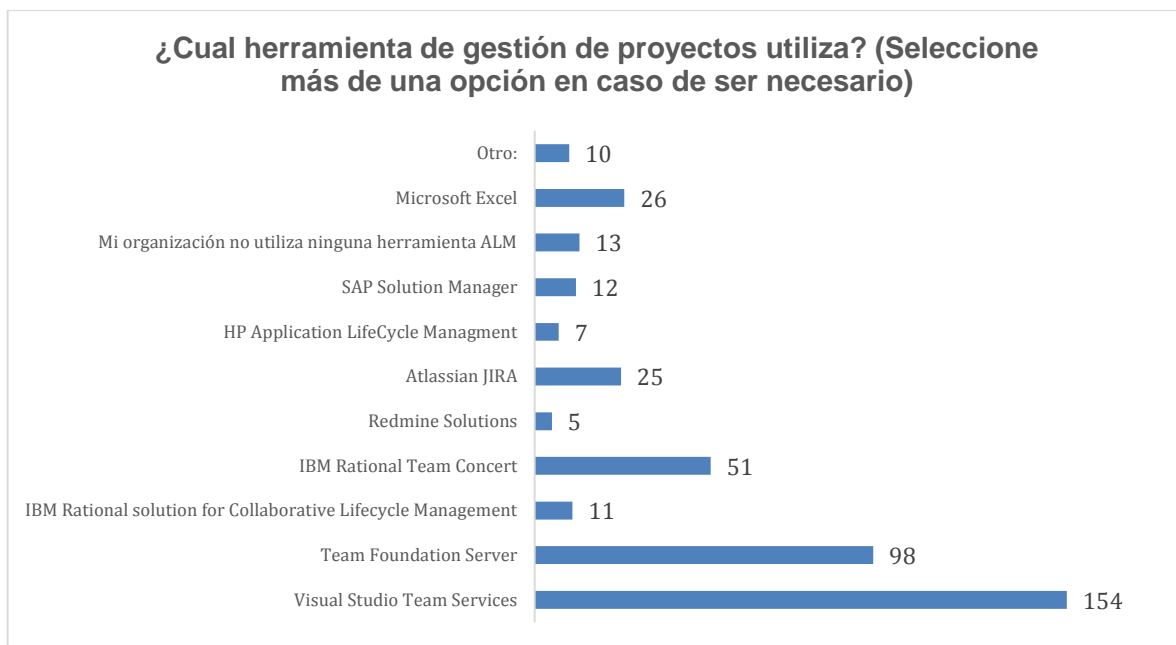


Gráfico 30: Herramientas de gestión de proyectos utilizadas en las empresas.

De acuerdo a la población encuestada la mayoría (85%) menciona hacer uso de herramientas destinada para la gestión de desarrollo (gráfica 29), de los cuáles el que encabeza es la herramienta Microsoft Visual Studio Team Services (60%), Team Foundation Server (39%) y IBM Rational Team Concert (20%).

La importancia del uso de herramientas para la gestión de procesos es evidente, por ende, en la búsqueda de integración se debe revisar factores como la planificación y el seguimiento de tareas y las necesidades para lograr la definición del uso de la herramienta dentro del proceso de aseguramiento y la aplicación de marcos de trabajo como *Scrum*.

4.1.5.1. Herramientas para la Documentación de Pruebas

De la encuesta se resalta la información relativa en cuanto al uso de herramientas para la documentación de pruebas, aspectos que brindan un valor agregado al producto final de la investigación. Para la aclaración de estos

conceptos, se solicitó la información a la población, según se aprecia en las gráficas número 31 y gráfica número 32.

Gráfico 31: ¿Las empresas utilizan herramientas para la documentación de pruebas?

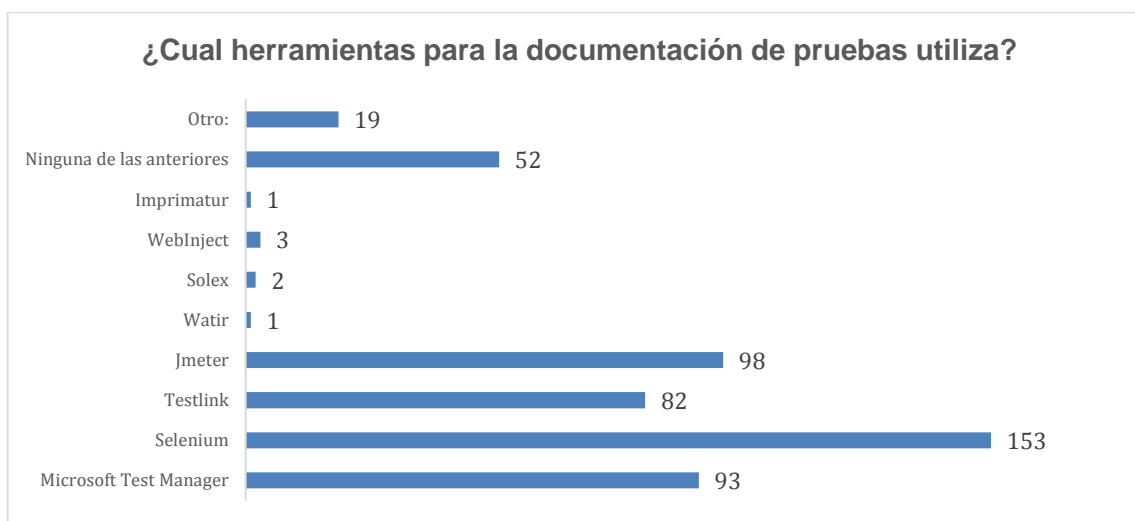


Gráfico 32: Herramientas para la documentación de pruebas utilizadas en las empresas.

El 78% de los encuestados utilizan herramientas para la documentación (gráfica 31), dentro de las mismas figuran herramientas como Selenium (60%), JMeter (39%), Test link (32%), y algunas de pago como Microsoft Test Manager (37%). Esto debido a que las herramientas de pago poseen características extra que complementan mejor la documentación que las gratuitas, además de tener la ventaja que las mismas mantienen una constante evolución y actualización.

Punto importante de integración, ya que como se observa en los resultados de aseguramiento de la calidad de software, la creación de planes de prueba, poder documentar los casos de pruebas y tomar muestras de dicha ejecución, además de la integración de las herramientas de pruebas y las

herramientas de administración de ciclo de vida del desarrollo de software aspecto importante en la toma de resultados donde domina el mercado las herramientas de Microsoft.

La suite de herramientas de Microsoft propone a Visual Studio Team Services como herramienta que integra completamente el aseguramiento con la gestión del desarrollo, permitiendo la planificación de tareas y la documentación de pruebas y toma de evidencias de la ejecución, herramientas que mantiene una buena integración ya que implementa el Marco de trabajo Scrum en su configuración.

4.1.5.2. Beneficios Presentados al utilizar Herramientas para la Gestión de Proyectos

De la encuesta se resalta la información relativa en cuanto a los beneficios aportados que puede adquirir una organización al utilizar herramientas destinadas a la gestión de proyectos, aspectos que vienen a brindar un gran valor agregado al producto final de la investigación.

Para la aclaración de estos conceptos, se solicitó la información a la población, quienes generosamente emitieron sus criterios según se puede apreciar en la gráfica número 33.

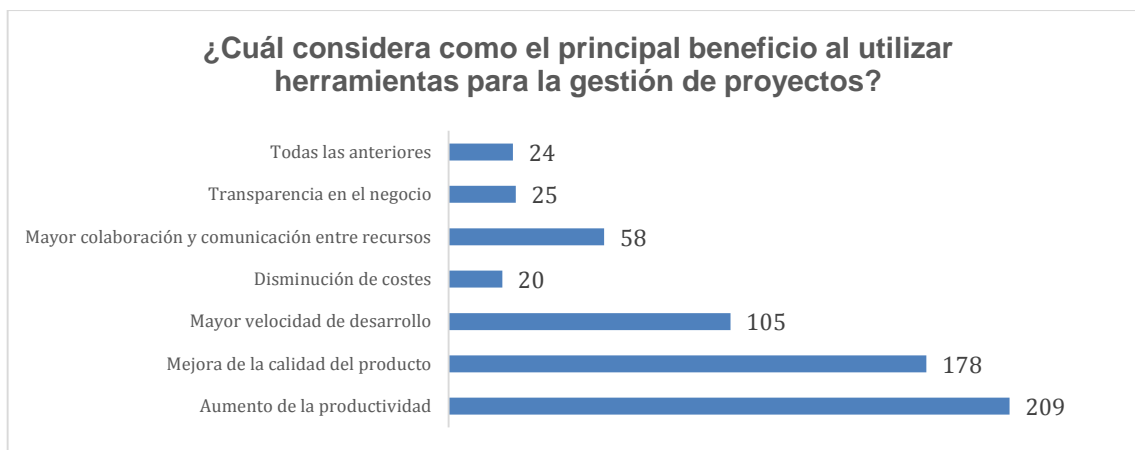


Gráfico 33: Beneficio para utilizar las herramientas para la gestión de proyectos, dentro de las empresas.

Las personas encuestadas consideran que el uso de herramientas para la gestión de proyectos, así como las herramientas para la documentación de las pruebas generan importantes beneficios a la organización entre los que se destaca: aumento de la productividad (82%), mejora de la calidad del producto (70%), mejora fluidez en el desarrollo de los proyectos (41%), así como presenta mejor colaboración y comunicación entre los recursos que rodean el proyecto (23%).

Por lo que se puede identificar la importancia, de que estas herramientas se integren al proceso propuesto por el marco de trabajo “Scrum” y la integración de las herramientas, con la definición de pruebas y los procesos que se deben desarrollar para las mismas dentro del marco de trabajo.

4.1.6. Sobre las Entrevistas

Las entrevistas se realizaron a profesionales en el área informática que han participado activamente en procesos de desarrollo de software con amplia experiencia en el área, con el objetivo de conocer de una manera más amplia los puntos de vista de carácter profesional.

Entre los hallazgos se menciona en la metodología ágil como la más utilizada con el marco de trabajo *Scrum*, inclusive como tendencia para realizar los procesos de desarrollo, como lo menciona Francia Sevilla, Julio Escalante, Carlos Ruiz quienes también dan fortaleza al uso de Scrum el impulso de procesos de aseguramiento de la calidad de software. Además, se puede basar en los resultados de las encuestas, para suponer que Scrum es el marco de trabajo más popularizado en el país en los procesos de desarrollo de software.

El uso del marco de trabajo Scrum, afirman los encuestados, presenta fortalezas como la pronta puesta en producción de los sistemas, como lo indica Francia Sevilla, quien hace referencia a puestas en producción de 1.5 meses para llevar un grupo de necesidad a producción, así como mejorar la comunicación y el trabajo en equipo con el área de negocio en las propuestas de software desarrolladas esto unido a buenas prácticas en temas de aseguramiento de la calidad del software, donde en prácticas como entrega continua benefician a una pronta detección de errores durante el desarrollo, ayudando a tener procesos de desarrollo más estables y con una arista clara en la calidad de software.

En cuanto a la labor diaria indican Francia Sevilla y María Fernanda Salas el cómo las empresas trabajan boletas especiales que son trabajo no planificado que ingresa al área de desarrollo de aplicaciones que son trabajadas bajo una prioridad superior al trabajo planificado además las mismas no pueden ser trabajadas durante un Sprint con todas las condiciones de planificación que el mismo conlleva.

En cuanto a la definición de tiempos David Carballo y Maickol Salamanca menciona la variación de los tiempos estimados para la finalización de requerimientos, donde es importante definir de una mejor manera las tareas a realizar y la dificultad para poder relacionar el pesaje de los requerimientos realizado en la metodología *Scrum* y la definición de tiempos de las tareas a realizar para finalizar la necesidad.

En los procesos de desarrollo y aseguramiento de la calidad la Ing. Francia Sevilla, el señor David Carballo Picado, así como el señor Maickol Salamanca Arias indican que en las empresas actuales aplican procesos de mejora continua, donde ven la retrospectiva como una herramienta que les ayuda a mejorar los procesos e identificar puntos de mejora.

Los señores Maickol Salamanca Arias y Julio Arturo Escalante Arias, mencionan la importancia de manejar ambientes separados de desarrollo y producción, de igual manera el departamento de aseguramiento de calidad (QA) debe estar aislado de los ambientes mencionados, esto para realizar los procesos correspondientes al *testing* (ejecución de pruebas) del producto.

Los profesionales Francia Sevilla, gerente de Aseguramiento de Calidad de Software, desarrolladores como David Carballo Picado y Maickol Salamanca Arias así como expertos en aseguramiento de la calidad de software como Carlos Ruiz y María Fernanda Salas expresan como importante la aplicación de un buen proceso de la calidad de software donde ven como fortaleza del área los programas profesionales impartidos UNED, CENFOTEC, FIDELITAS; donde también indican la importancia de los procesos de certificación que son ofertados

por *ISTQB* (Comité Internacional de Certificaciones de Pruebas de Software, siglas en inglés) el cuál es un ente formador de buenas prácticas para las tendencias de aseguramiento de software, logrando dar la formación a profesionales que cuentan con la certificación, lo que beneficia los procesos estructurados al aplicar la tendencia de aseguramiento integral de desarrollos ágiles.

Un punto que cabe recalcar de los procesos de aseguramiento de la calidad del software en Costa Rica, según lo mencionado por los entrevistados es la fortaleza de contratación de esta disciplina mediante *outsourcing*, lo cual se puede deducir que la falta de experiencia de las empresas los hace buscar la guía en empresas de venta de servicios con experiencia en el área, y que se dedican a la búsqueda de personal certificado en el área para que formen parte de sus equipos, además de buscar la experiencia de dichas empresas para fortalecer y adecuar sus procesos.

En cuanto a herramientas son los entrevistados quienes poseen variedad, por ejemplo, Francia Sevilla Escalante, David Carballo Picado, Carlos Ruiz Campos y María Fernanda Salas consideran que entre las mejores están las herramientas de Microsoft con su ALM de *Visual Studio Team Services*, otro punto de vista el profesional Julio Arturo Escalante consideran como complemento de fortaleza de *Atlassian* con su paquete de herramientas de ciclo de vida de desarrollo de *Jira*, *Bitbucket*, *Confluence*.

Como complemento como lo menciona Carlos Ruiz, sumando herramientas a los procesos de la calidad de software hay herramientas como

Selenium, Appium y Apache JMeter que son herramientas que fortalecen los procesos de aseguramiento de la calidad.

Se determina que los procesos aplicados son integrados en una buena ejecución del desarrollo a lo cual en preguntas generales formuladas en la encuesta, se ven puntos de vista como por ejemplo: Carlos Ruiz menciona la importancia de la transparencia y visibilidad de los procesos de pruebas, donde también María Fernanda Salas da la importancia a los primeros pasos en los procesos de formación e implementación, otro punto de vista es el de Luis Diego Fernández quien observa la importancia de la seguridad y el control de los desarrollos.

De esos criterios y como lo mencionan en las respuestas, la importancia de formar una guía que ayude a las empresas a tener una visión de lo que debe contemplar este proceso y ayude a conformar estos departamentos o bien mantener participación activa en un proceso conformado por una empresa de *outsourcing*, para de esa forma crear el criterio correspondiente de donde quiero llevar el proceso realizado.

Como lo menciona María Fernanda Salas la importancia de formar un proceso que ayude a las empresas a dar los primeros pasos, es una de las fortalezas de contar con una base de conocimiento que integre los procesos necesarios y guíe sobre los cambios necesarios para lograr una buena implementación.

Para dicho planteamiento se estructuraría la integración de una práctica de aseguramiento de la calidad basada en el Syllabus guía de buenas prácticas de *ISTQB* (Comité Internacional de Certificaciones de Pruebas de Software; siglas en inglés), la cual sería integrada al proceso propuesto por Scrum en su ejecución, adecuando procesos modernos de entrega continua y administrando dichas prácticas mediante herramientas de ciclo de vida del desarrollo de software, las cuales se propondrían mediante una configuración con aspectos importantes que beneficien a la integración de los procesos y a una buena ejecución de los mismos.

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Con base en los resultados de la investigación se concluye que, con relación a las metodologías de desarrollo, existe la tendencia en el uso del marco de trabajo ágil, siendo *Scrum* el *Framework* más utilizado por las empresas de desarrollo de software a nivel nacional.
- Queda claro que las empresas de desarrollo mantienen un flujo de trabajo respaldado en una metodología ágil, sin embargo, queda aún camino por recorrer, especialmente en cuanto a la definición teórica y práctica de metodologías, ya que, aunque afirman que su metodología es ágil, muchos no logran interiorizar las diferencias con otros tipos de metodología (por ej. cascada). A pesar de ello, se reconoce en el medio el marco de trabajo “*SCRUM*”, como técnica más popular.
- Se concluye, que la mayor parte de los profesionales encargados de este tipo de labores cuentan con experiencia suficiente (entre 1 y 5 años), y pertenecen a empresas medianas conformadas entre 11 y 50 colaboradores que poseen más de un año de actividad en el mercado nacional.

Muchas son empresas desarrolladoras de software o consultoras del área, que poseen conocimiento en la aplicación de metodologías ágiles e incursionan en procesos de la calidad de software, un hallazgo importante es que a dichos procesos se les detecta puntos de mejora, que son originados por la aplicación de procesos con conocimientos empíricos o profesionales

que no cuentan con la información necesaria para poder definir el proceso y adecuan el mismo a criterios de grupos de profesionales para la gestión.

- Se identifican obstáculos para la incorporación del uso de metodologías ágiles dentro de las empresas desarrolladoras de software nacionales, entre las que se destaca:

1. La falta de experiencia en este campo.
2. Falta de información sobre metodologías ágiles.
3. La cultura organizacional.
4. La resistencia al cambio.

Pero también se identifican ventajas como las siguientes:

1. La mejora del trabajo en equipo.
2. La habilidad en el manejo de cambios.
3. Incremento en la productividad.
4. La mejora en la calidad de software.

El tema del aseguramiento de la calidad de software, es clave, ya que no era común que las empresas lo aplicaran; sin embargo, ahora es más apropiado encontrarlo dentro de las empresas, muchas consideran importante el proceso, a lo cual es importante determinar un mecanismo que ayude a mitigar las deficiencias y de fortalecer las debilidades.

- También se concluye que existen requerimientos o boletas de trabajo que deben ser trabajadas bajo una condición especial, ya que son trabajo que no

entran en las planificaciones de sprint y son atendidos a mayor velocidad, en algunos casos son solicitudes especiales como extracciones de datos en entre otras solicitudes especiales, pero no deben afectar el sprint o la información de seguimiento del mismo.

- En la definición de las labores a realizar se debe nombrar un proceso para el desglose de tareas y la mejor forma de definir los tiempos para desarrollar un trabajo, además se debe definir la técnica de pesaje de necesidades de Scrum para facilitar su entendimiento y uso, además de tener mayor exactitud en los cálculos de velocidad de los equipos de trabajo.
- En relación con la definición de tiempos de trabajo, deben definirse procesos que ayuden a mejorar la exactitud en los cálculos, ya que son modificados durante el proceso, lo cual es importante tener bien definido en procesos ágiles.
- En relación con las preguntas relacionadas con aseguramiento de la calidad de software, se identifica que existe una tendencia de crecimiento de nivel nacional, analizando los resultados muchas empresas ya lo aplican, más si notan una oportunidad de mejora en los procesos, las empresas que actualmente no lo aplican, lo ven como una oportunidad de mejora.
- En cuanto a la importancia de las fases que contemplan un buen proceso de aseguramiento de la calidad según se muestra en la investigación y las entrevistas se identifica a *ISTQB* (Comité Internacional de Certificaciones de Pruebas de Software; siglas en inglés), como la guía de mayor fortaleza

utilizada en tendencias de agilización, la guía contempla los aspectos importantes del proceso como tipos de pruebas, tendencias, ciclos de ejecución, clasificaciones de pruebas y aspectos importantes de pruebas o testing que deben ser tomados en cuenta, etc.

- Con relación a los procesos de aseguramiento de la calidad software, según la investigación realizada en vista a las responsabilidades legales que mantienen las empresas desarrolladoras, se debe tener bien definida la ejecución de los procesos de aseguramiento con el fin que respalden la calidad del software desarrollado y respalde a la empresa para no tener procesos legales en base a la calidad o el mal funcionamiento de un sistema.
- En cuanto a buenas prácticas del proceso de aseguramiento, se debe tomar en cuenta aspectos como:
 - Manejo de ambientes propios para la gestión de aseguramiento de la calidad de software, dónde se realizan procesos de regresión que necesitan tener configurados escenarios de pruebas con scripts de datos establecidos, mismos que ayudan a estos procesos y no pueden ser modificados en etapas de desarrollo, esto permite que los ambientes de QA aporten a la gestión de aseguramiento, así como mantener ambientes más homólogos hacia ambientes productivos.
 - Mantener procesos de entrega continua que ayuden a que los procesos de aseguramiento de la calidad, que se integren al proceso de desarrollo y se puedan realizar evaluaciones tempranas que

permiten beneficiar en la detección de errores y tener productos que evolucionen de manera satisfactoria, además de lograr revisiones tempranas por parte del cliente que ayuden a identificar cambios, mejoras o hasta entendimiento de las historias de usuario solicitadas.

- Definir los planes de pruebas para los procesos de aseguramiento, donde se deben detallar sus características y aspectos de las historias de usuario a evaluar, como se indica en las entrevistas es un artefacto muy importante durante el proceso de pruebas.
- Definir la clasificación de errores para mantener indicadores de medición en el proceso y manejo de éstos dentro de la metodología.
- Tomando en cuenta que la academia costarricense sólo cuenta con dos universidades que tienen la especialidad dentro de la carrera de Ingeniería del Software (análisis que escapa al alcance de esta investigación), pero si se puede concluir que existe la necesidad de fomentar la importancia de la disciplina ya que actualmente se tiene un mercado que tiene la necesidad de mantener estos perfiles con una buena formación profesional, que ayuden a llevar de la mano a las empresas a una buena aplicación de la técnica.
- El CPIC (Colegio de Profesionales en Informática y Computación) reconoce la disciplina de aseguramiento de la calidad como una especialidad que debe ser incluida en los procesos de desarrollo bajo las tendencias *DevOps*, donde se integran a procesos ágiles y los procesos de calidad de software, a lo cual es importante que se tengan bien definidos dichos procesos y el ejecutar de

los profesionales para respaldar la tendencia que genera valor y es apoyada por un ente como lo es el CPIC.

- Las empresas que inician en el proceso, deben mantener una inversión en la certificación y capacitación de los funcionarios que se incluyen en dicha disciplina de estudio, para garantizar la aplicación apropiada del proceso, se debe invertir en herramientas enfocadas en la gestión de la revisión de la calidad de software y que ayude en la administración de las pruebas.
- Las inversiones en herramientas y procesos de capacitación tienen costos altos, que son recuperados al generar software de calidad y las inversiones que se trasladan al inicio de un proceso de desarrollo, que de no ser realizada dicha inversión, se gestionará como gastos (por altas incidencias que impactan su reparación) que se generan al colocar en el mercado software con problemas en su calidad, lo cual se une al cambio de cultura donde se realiza inversiones para la gestión y prevención, de manera que ayuden en la evolución apropiada del software, desarrollado bajo un alto estándar de calidad.
- Gran parte de las empresas utilizan en sus procesos metodologías ágiles, con procesos de aseguramiento de la calidad del software, con pruebas formales y en casos aún empíricas ante la ausencia de una guía apropiada, lo que denota la necesidad de contar con una herramienta de este tipo, que facilite la integración de los procesos de aseguramiento de calidad de software en el trabajo diario.

- Asimismo, se puede determinar el mayor obstáculo para la incorporación correcta del aseguramiento de la calidad del software en empresas nacionales, el consenso establece que es por:
 1. La falta de una guía que estructure el proceso.
 2. La cultura organizacional.
 3. La falta de experiencia del personal.
 4. La resistencia al cambio.
 5. La falta de información sobre el aseguramiento en la calidad de software del lado de los colaboradores.

- Como punto importante, para mostrar la necesidad de crear un artefacto que defina el proceso, es necesario integrar herramientas para la gestión del ciclo de vida de desarrollo de software, así como herramientas para el aseguramiento de la calidad del software. donde destacan las herramientas Microsoft *Visual Studio Team Services*, *Team Foundation Server*, por parte de IBM se tiene *IBM Rational Team Concert* y la suite de herramientas de Atlassian.

- La importancia que tiene el uso de herramientas para la gestión de pruebas, apoyadas especialmente por herramientas de software libre como “Selenium” y herramientas de Licenciamiento de Microsoft como “*Test Manager*”, software que permiten el aumento de la productividad y la mejora en la calidad de los productos entregados.

- Es importante integrar herramientas que faciliten y conformen la fábrica de software donde se deben de tomar en cuentas aspectos como el manejo de backlog para el registro de necesidades, la creación de sprint backlog para el seguimiento de tareas y aspectos como *dashboard*, poder llevar valores de pesajes en la valoración de esfuerzo, asignación de tareas y la administración de planes de pruebas que permitan la clasificación de pruebas y seguimiento a las ejecuciones, registro de errores y seguimiento a los mismos además de permitir las consultas y llevar un control estadístico de los registro.
- La conjugación de los tres pilares personas, procesos y herramientas para el fortalecimiento de conceptos modernos como lo son técnicas DevOps (Unión de desarrollo y operaciones), que facilitará en las empresas el fortalecimiento de los conocimientos, para la implementación e integración de los procesos aliados a la fábrica de software.

5.2. Recomendaciones

- Basados en los resultados y conclusiones obtenidas se define al marco de trabajo ágil *Scrum* como el más utilizado por las empresas nacionales que incursionan en metodologías ágiles, además se identifica es el más adecuado para el diseño de procesos de aseguramiento de la calidad de software para las empresas nacionales de desarrollo de software.
- El fortalecimiento de una guía que ayude a la aplicación de procesos de marco de trabajo ágil, que venga a fortalecer el conocimiento de los profesionales que participan del proceso y ayude a fortalecer los obstáculos que la incorporación de esta técnica tiene en las empresas de desarrollo nacional que se ha iniciado con la técnica.
- Dentro de la definición del marco de trabajo se debe desarrollar una técnica para la atención de trabajo no planificado, dichos trabajos no debe de afectar los trabajos del sprint, se recomienda el manejo de nuevos sprint que desarrollen todo el trabajo no planificado durante un lapso de tiempo definido, el mismo en paralelo al sprint oficial de la metodología, el manejo de dichos ítems se realizaría con la personalización creando ítems tipo FastTrack en la herramienta de administración de ciclo de vida de las aplicaciones, dónde se deben definir reglas claras para el manejo de dichos ítems y las características del mismo para evitar confusión con el manejo de historias de usuario.

Para el seguimiento de labores se recomienda la creación de ítems de seguimiento para todas las labores que no sean parte de los elementos de

trabajo o labores internas como procesos de capacitación entre otros trabajos a realizar que puedan modificar los tiempos de trabajo en un sprint, esto con el fin de manejar estadísticas de la administración de estos tiempos para toma de decisiones o procesos de retrospectiva en búsqueda de la mejora de la productividad.

- Para la etapa de planeación se debe definir un proceso de desglose de tareas, que permita definir los tiempos a desarrollar para cada tarea dentro de un requerimiento o historia de usuario, para facilitar ser más efectivo en el cálculo, según los hallazgos es importante definir bien el proceso de pesaje de historias de usuario en la etapa de planeación, para facilitar el uso del instrumento utilizado para el pesaje de necesidades utilizado por la metodología Scrum, a lo que se recomienda orientar el pesaje de las historias de usuario a esfuerzo, midiendo factores de complejidad y capacidad de trabajo a desarrollar.
- En cuanto al manejo de reuniones se deben indicar reuniones propias de etapas de planeación para las evaluaciones iniciales de los requerimientos que ayuden con el afinamiento de los insumos de desarrollo, para lograr un mayor entendimiento de la necesidad a trabajar.
- La importancia que las empresas fomenten un proceso con mayor fortaleza donde se logre identificar las necesidades propias del área de desarrollo y de aseguramiento de la calidad de software, especificando los procesos desarrollados por pruebas gestionadas sobre desarrollo de aplicaciones y los procesos paralelos que están incluidos en la técnica de construcción moderna

tipo fabrica donde los procesos los conforman las dos disciplinas completas y a ello se unen los paradigmas DevOps que vienen a fortalecer los principios del área.

- Para lograr el fortalecimiento de las técnicas de aseguramiento de la calidad de software, como se menciona en las conclusiones es importante que se analice los procesos ágiles que se aplican, en las áreas de desarrollo en vista al resultado final que se concluye en la investigación se propone el uso del Marco de trabajo *Scrum* el cual tiene mucha fortaleza a nivel nacional y crece día con día, a lo cual se identifica la necesidad de definir un proceso de aseguramiento que interactúe con el marco de trabajo, y que no deje de lado aspectos importantes que ayudaran a integrar el proceso como lo son las herramientas de ciclo de vida del desarrollo de software y herramientas que dan soporte a los procesos de aseguramiento de la calidad de software.
- Para la conformación de la guía de aseguramiento durante la investigación se identifica la fortaleza de las guías de aseguramiento suministradas por *ISTQB (International Software Testing Qualifications Board)*, (Comité Internacional de Certificaciones de Pruebas de Software; siglas en inglés) las cuales son fortaleza del proceso de aseguramiento que se aplica y serán base de la estructura de los procesos definidos, teniendo la ventaja de contemplar procesos que pueden formarse como perfiles profesionales certificados ya que *ISTQB* cuenta con procesos de certificación para profesionales en el área, que ayuden a mejorar la disciplina y a fomentar las buenas prácticas de

los procesos, para mejorar la calidad del software que se desarrollan a nivel nacional.

- Basado en los resultados de la investigación se fortalece la necesidad de contar con un artefacto de referencia para los procesos, que utilice metodologías ágiles, integren los procesos de calidad de software, consolidando un proceso de mejora continua, que ayude en el fortalecimiento de la estructura actual de los procesos de desarrollo de software y además que identifique el uso adecuado de las herramientas que funcionan como respaldo de los procesos de aseguramiento aplicados ya que documentan la evidencia del proceso y el buen funcionamiento de los software testeados.
- También es importante detallar la planificación de pruebas y aspectos importantes de la conformación de ambientes donde se identifican la necesidad de tener ambientes destinados a desarrollo, aseguramiento de la calidad (QA) y pruebas de aceptación de usuario esto debido a características de los ambientes de QA como podemos mencionar etapas de regresión donde se manejan escenarios de pruebas ya creados y set de datos definidos para las ejecuciones.

Otro aspecto importante es contar con procesos de entrega continua donde se logra identificar errores de manera temprana, debido a la continua ejecución de pruebas y el registro de los errores en cada parte del proceso y las clasificaciones de los mismos para lograr con ello la retroalimentación de los procesos y métricas que nos ayuden a mantener estadísticas de los procesos.

- Durante la investigación se identifica la importancia que la academia adopte dentro de sus planes de desarrollo para profesionales la propuesta de incluir la especialidad dentro de los perfiles profesionales para un mercado que lo exige, donde universidades integran a sus programas una área tan importante como lo es el aseguramiento de la calidad de software creando perfiles profesionales orientados a la disciplina, en vista de la importancia de los procesos de aseguramiento y con ende a la propuesta de evolución que tiene el mercado costarricense.

Actualmente en la educación superior en Costa Rica hay dos Universidades (Universidad Fidélitas y Universidad Estatal a Distancia “UNED”), que imparten la especialidad como orientación de la carrera de Ingeniería del Software, pero tomando en cuenta la importancia que exponen las empresas sobre el auge de dicha disciplina en el mercado costarricense es donde nace la necesidad que exista mayor cantidad de universidades desarrollando la especialidad.

Generar la aplicación de la guía creada en la presente investigación de tesis en proyectos dentro de las empresas nacionales que les ayuden a implementar procesos de aseguramiento de la calidad de software o por ende mejorar los procesos actuales que se aplican en un proceso de madures de los mismos.

- Es importante que las empresas que implementen procesos de aseguramiento tomen en cuenta procesos de certificación y

entrenamientos en el uso de herramientas de ciclo de vida del desarrollo y herramientas de aseguramiento de la calidad de software.

- Las herramientas definidas para la conformación de la guía de aseguramiento es Visual Studio Team Services, la cual es una herramienta que integra desde la administración del ciclo de vida, hasta las herramientas de pruebas como la herramienta Test Plan y que por sus factores de integración y aplicación de plantillas con el marco de trabajo *Scrum* logra integrar lo necesario para estructurar la guía que conforma el proceso.

Además, no permite la modificación de plantillas, base ello en busca de mejorar la integridad con aspectos de la calidad y el proceso *Scrum*, permitiendo la administración de *backlog*, gestión de *sprint*, seguimiento a pesajes y velocidad de equipos, seguimiento de tareas, gestión de dashboard, registro y seguimiento de errores, además de poder crear consulta para el análisis estadístico de la información, entre otras características importantes

- Esta guía estaría compuesta de una estructura de trabajo basada en el marco de trabajo *Scrum*, procesos de aseguramiento de la calidad de software basados en las tendencias propuestas de *ISTQB* (Comité Internacional de Certificaciones de Pruebas de Software; siglas en inglés) y que se integren en la gestión de la herramienta *Visual Studio Team Services* de Microsoft y su herramienta integrada *Test Manager*, que se agregan a dicha tendencia, en análisis al proceso de desarrollo, determinando la necesidad de integrar la

técnica a metodologías de desarrollo, a lo cual en esta tesis se definiría la propuesta de establecer una guía que ayude a las empresas desarrolladoras de software en Costa Rica tener una visión de un proceso integrado y la conformación del mismo.

- El artefacto requiere la implementación de procesos de calidad de software o mejorar los actuales procesos, garantizando el éxito de la aplicación y la integración de marcos de trabajo, con procesos de aseguramiento de la calidad de software y herramientas de aplicación. Sin dejar de lado que el personal asignado debe contar con experiencia para estos procesos.

CAPÍTULO VI PROPUESTA

6.1 Introducción

La metodología Scrum es iterativa, rápida, flexible y eficaz, elaborada con la intención de dar un apoyo al proyecto. Esta metodología garantiza transparencia en la comunicación y genera responsabilidad colectiva y de progreso continuo en el proyecto aplicado.

El marco de Scrum adopta un enfoque en el que los miembros del Equipo conformado por el Product Owner, Scrum Master y Development Team encargados de trabajar en desarrollar y entregar el producto, este equipo recibe el apoyo y colaboración de los Stakeholders conformados por usuarios, clientes y patrocinadores para crear los entregables que proporcionan mayor valor posible para el cliente final.

A nivel de referencia teórica se ha utilizado Scrum body of knowledge (SBOK™ Guide), para el marco de trabajo Scrum, a nivel de implementación han surgido una serie de prácticas propias del negocio que es necesario identificarlas para buscar la estandarización a nivel organizacional.

Para efectos de este esfuerzo, se incorporaron apartados textuales de dicha guía; así mismo se agregaron otros elementos propios de la gestión del desarrollo.

El aseguramiento en la calidad del software pretende mantener un propósito esencial el cual consiste en entregar a la administración una visión correcta del proceso y del producto, a través de acciones sistemáticas que aseguren dicha calidad.

El “proceso” incluye los métodos, actividades y prácticas para el desarrollo o mantenimiento del software, así como los productos involucrados.

El “producto” es el software y todos los aquellos artefactos creados como parte este, incluye procedimientos, planes, especificaciones, definición de procesos y toda aquella documentación orientada a este.

6.2 Propósito

Este documento muestra una guía para el personal que implementa el área de aseguramiento de la calidad, tomando aspectos de procesos definidos en la metodología ágil Scrum y la definición del manejo de herramientas de administración de ciclo de vida y aseguramiento de la calidad de software.

6.3 Objetivo

- Proporcionar una guía para facilitar el diseño e implementación del área de aseguramiento de la calidad de software a empresas de desarrollo a nivel nacional.
- Estructurar los procesos de aseguramiento de software, diseñado para el área de aplicaciones de las empresas nacionales, integrando la metodología ágil Scrum, guías de aseguramiento ISTQB, herramientas de administración de ciclo de vida del desarrollo y herramienta de gestión de calidad de software que se integren en el proceso.

6.4 Alcance

Este documento es aplicable al mantenimiento de sistemas de información y desarrollo de proyectos informáticos, con el fin de propiciar un ambiente de trabajo estándar y práctico, que oriente el trabajo de aseguramiento de la calidad de software.

Definir el proceso de aseguramiento de la calidad de software ejecutado por el área de desarrollo de software en empresas nacionales y generar una guía de aplicación en los procesos de calidad que adecua reportes de seguimiento, además de definir las tareas a realizar durante el aseguramiento, acompañamiento a usuarios y pases a producción.

La documentación de pruebas realiza un detalle de las definiciones importantes del testing indicadas en las guías de aseguramiento de software de ISTQB, proceso aseguramiento apegado a la metodología Scrum, herramientas de gestión de ciclo de vida del desarrollo y herramientas de apoyo a pruebas, además de la definición de acompañamiento a usuarios en el proceso de prueba de aceptación.

6.5 Marco Scrum Implementado con Patrones de Aseguramiento de Software

6.5.1 Definición de Scrum

Scrum es considerado una metodología ágil y flexible que permite gestionar el desarrollo de un software, su principal función es incrementar el retorno de la inversión a la organización. Se dedica a construir la línea de mayor valor para el cliente.

6.5.2 Pilares de Scrum

A. Transparencia

- Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado.
- Todos los participantes deben compartir un lenguaje común para referirse al proceso.
- Aquellos que desempeñan el trabajo y aquellos que aceptan el producto de dicho trabajo deben compartir una definición común de “Hecho”.

B. Inspección

- Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones no deseables.
- No debe ser tan frecuente como para que interfiera en el trabajo.

C. Adaptación

- El proceso o el material que está siendo procesado deben ser ajustados (un proceso se desvía de límites aceptables).
- Ajuste debe ser realizado cuanto antes para minimizar desviaciones mayores.
- Eventos de SCRUM.

6.5.3 Roles de Scrum

A. Stakeholders (Usuarios)

Es un conjunto de personas que no forman parte directamente del proceso en el desarrollo, pero deben ser tomados en cuenta ya que son

personas interesados en el producto final. Estos son los que darán el aval del producto y quienes lo utilizarán constantemente.

B. Product Owner (Dueño del Producto)

Es aquel responsable de informar al equipo de desarrollo la visión del producto a desarrollar. Representa al resto de interesados en el desarrollo del producto. El dueño del producto es quien deberá analizar la pila de épicas y definir la prioridad de las necesidades agregadas a la pila de producto.

Trabajo desarrollado:

- Responsable de gestionar y analizar la Pila de Épicas.
- Responsable de gestionar la Pila de Producto.
- Responsable de maximizar el valor del producto y del trabajo del equipo de desarrollo.
- Define las funcionalidades del producto.
- Decide sobre las fechas y contenidos de los avances.
- Es responsable por la rentabilidad del producto.
- Prioriza funcionalidades de acuerdo con el valor del mercado.
- Ajusta funcionalidades y prioridades en cada iteración si es necesario.
- Acepta o rechaza los resultados del trabajo del equipo.

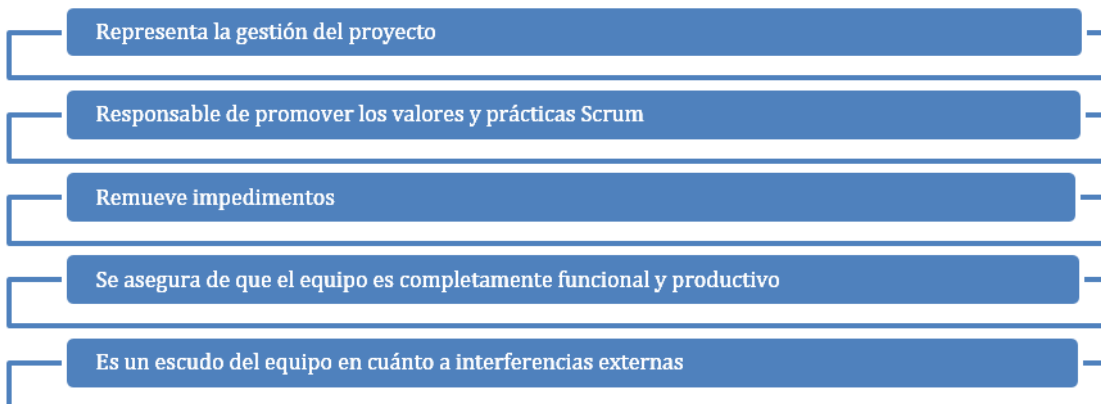
Factores por tener cuenta:



C. Scrum Master (Facilitador)

Es quien lidera el proceso de implementación del producto para que sea ejecutado de manera correcta. Tiene la responsabilidad de entregar el producto y posee habilidades necesarias para realizar el análisis, diseño, desarrollo, pruebas, documentación, implementación.

Entre algunas funciones están:



D. Development Team (Equipo de Desarrollo)

Equipo multidisciplinario encargado del desarrollar la necesidad del dueño del producto.

- Profesionales que desempeñan el trabajo de entregar un incremento de producto “Hecho”, potencialmente utilizable, al final de cada sprint.
- Sólo los miembros del equipo de desarrollo participan en la creación del incremento.

- Los equipos de desarrollo son estructurados y facultados por parte de la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y eficacia general del equipo de desarrollo.
- Son auto-organizados, nadie (ni siquiera el Scrum Master) indica al equipo de desarrollo cómo convertir elementos de la pila de producto en incrementos de funcionalidad potencialmente entregables.
- Los equipos de desarrollo son multidisciplinarios, en este caso están conformados como mínimo por profesionales en desarrollo de software y profesionales en aseguramiento de la calidad de software.
- Scrum no reconoce títulos para los miembros de un equipo de desarrollo.

E. Scrum Team (Equipo Scrum)

- Equipo que se focaliza en construir software de calidad, involucra a Product Owner, Scrum master, Development Team, participan activamente del incremento del producto.

6.5.4 Artefactos del marco de trabajo Scrum y de la Herramienta

VSTS

A. Marco de trabajo Scrum

- a) **Pila del producto:** lista de requisitos de usuario que crece y evoluciona durante el proyecto.

- b) **Incremento:** resultado de cada sprint.
- c) **Pila del sprint:** lista de los trabajos que deben realizarse durante el sprint.
- d) **Sprint:** nombre que recibe cada iteración de desarrollo.
 - Bloque de tiempo de un mes o menos durante el cual se crea un incremento de producto “Hecho”, utilizable y potencialmente entregable.
 - La duración de los sprint es consistente a lo largo del esfuerzo de desarrollo.
 - Cada nuevo sprint comienza inmediatamente después de la finalización del sprint previo.
 - Cada sprint puede ser considerado un proyecto con un objetivo.
 - Se usan para obtener un logro.
 - Tiene una definición de qué va a ser construido, un diseño, y un plan flexible que guiará la construcción, el trabajo y el producto resultante.

B. Visual Studio Team Services

- a) **Backlog Épico:** Pila de necesidades que van a ser desglosadas en Historias de Usuario o FastTrack según sea la necesidad del negocio.
- b) **Épico:** Encapsula las necesidades ingresadas por los stakeholder.
- c) **Backlog Ítem:** Pila de necesidades que van a ser trabajadas en los sprint.

- d) **Product Backlog Ítem (PBI):** Encapsula las necesidades del negocio historias de usuario que van a ser desarrolladas dentro del marco de trabajo.
- e) **FastTrack:** Encapsula las necesidades especiales del negocio, las mismas se subclasifican como extracción de datos, modificación de datos, investigaciones.
- f) **Seguimiento:** Ítem al cual se van a registrar las tareas relacionadas con el seguimiento de labores administrativas.
- g) **Task:** Hace referencia a las tareas que registra el development team relacionada con un product backlog ítem (Historia de usuario, Épica, Bugs, Seguimiento) durante el desarrollo del producto.
- h) **Impediment:** Son registros que permiten gestionar un impedimento asociado a una tarea específica.
- i) **Sprint Backlog:** Pila de necesidades priorizadas para ser trabajadas en un tiempo determinado.
- j) **Burndown Chart:** Es el gráfico en el que se muestra el avance del Sprint.
- k) **Sprint Burn up:** Gráfica donde se muestra la evolución del desarrollo del producto.
- l) **Capacity:** define la capacidad de un colaborador del equipo en un sprint determinado.
- m) **Velocity Chart:** Grafico que muestra la velocidad del equipo a lo largo de cada sprint.
- n) **TaskBoard:** Pizarra de trabajo del development team.

6.5.5 Etapa de planeación

A. Gestión de Requerimientos

El Producto Owner asignado al proyecto debe realizar las siguientes actividades:

- Realizar el análisis de las necesidades tipo épica que son ingresadas por los usuarios.
- Se generar el desglose de historias de usuario ò FastTrack según las necesidades de negocio en la herramienta VSTS.

Los registros de las historias de usuario se deben de realizar mediante el formato que contiene:

- Título: referente a la solicitud.
- Como: En esta sección se coloca quien solicita la historia de usuario
- Quiero: En esta sección se coloca una pequeña descripción de la historia de usuario
- Para: En esta sección se coloca cual es el objetivo de la Historia de Usuario.
- Criterios de Aceptación: se especifica la nueva funcionalidad o el cambio que se requiere.
 - Como buena práctica se puede utilizar la técnica SMART para medir los criterios de aceptación definidos en una historia de usuario ò FastTrack.

Ejemplo de plantilla de historia de usuario:

Historias de usuario: “Nombre de la Historia de usuario”

HU – Nombre de historia de usuario

Como: En esta sección se coloca quien solicita la historia de usuario

Quiero: En esta sección se coloca una pequeña descripción de la historia de usuario

Para: En esta sección se coloca cual es el objetivo del requerimiento

B. Criterios de Aceptación

En los criterios de aceptación se deben colocar todas las validaciones del sistema

- Se debe describir si es una transacción nueva o una mejora, nombre de la transacción en caso de ser nueva y cuál es su ubicación
- Se debe colocar qué rol permite ingresar a la transacción
- Se deben describir los campos de la pantalla:
 - sí son criterios de búsqueda
 - sí son obligatorios
 - cuantos caracteres permite
 - Qué tipo de campo es si es editable, si es combo box
 - Qué formato debe tener el campo si son solo numéricos o solo texto o si son tipo cedula etc.

- Se deben describir las columnas del grid en caso de que tenga la pantalla si son editables, tipo checks hay que validar si puedo seleccionar una opción o varios.
- Se debe describir las validaciones que tengan los botones de la pantalla
- Validar si se tienen mensajes de error

C. Análisis preliminar del backlog

El Product Owner, realiza la priorización de las historias de usuario registradas en el backlog ítems, esta actividad debe ser constante a lo largo de todos los sprint.

Se establece la priorización, donde se han definido las necesidades por sistema a atender durante un periodo según el criterio product owner.

El product owner realiza reuniones con los dueños de otros productos que integren con el desarrollo por trabajar, para definir prioridades y lanzamientos.

En esta etapa se gestionan reuniones de planeación donde participan el product owner y los Scrum Master, esto para revisar y crear ajuste a los ítems por trabajar, además de definir la prioridad de los insumos y fechas de reuniones.

6.5.6 Ceremonias SCRUM

A. Sprint Planning (Planeación de Sprint)

Reunión de arranque 1era parte; conceptualización

Sesión de conceptualización se presentan las Historias de Usuario, FastTrack o Bug donde el Product Owner explica las necesidades priorizadas para que el Scrum Team interprete y aclare dudas referentes a las mismas.

La planeación del sprint contempla:

- Explicación de las necesidades.
- Observaciones y modificaciones relacionadas a las necesidades.
- Visto bueno de entendimiento de las necesidades.
- Pesaje de las historias de usuario, FastTrack o Bug (Effort), mediante la técnica Planning Pòker.
- Compromiso del equipo.

Técnica a utilizar	Periodicidad	Personal requerido	Reglas	Resultado
Reunión	1 vez por sprint al Inicio del sprint.	<u>Responsable:</u> Product Owner	Lectura e interpretación de cada Historia de Usuario ò FastTrack.	Compromiso del sprint por parte del equipo SCRUM: Historias de Usuario o FastTrack a desarrollar.
		<u>Participa:</u> Scrum Team	Equipo Scrum debe aclarar todas las dudas relacionadas a cada necesidad. Las modificaciones de la Historias de Usuario ò	

			<p>FastTrack, surgidas a partir de aclaraciones en esta sección sólo pueden ser incluidas en el momento cuando no afecte el sentido original de la necesidad, en el caso que el cambio amerite un tiempo mayor se habilita gestionar el cambio entendido por el equipo de trabajo en las 4 horas restantes del planning, caso contrario el product owner debe detallar el cambio fuera de esta sesión lo cual ameritaría que no sea contemplado en el sprint.</p> <p>Realizar la votación del esfuerzo por Historia de Usuario ò FastTrack, mediante el uso de Fibonacci, los valores son 1,2,3,5,8,13,21,34, ∞.</p> <p>Definir compromiso del Sprint referente a las necesidades a atender. (Definir prioridad atención del equipo, utilizando el Velocity histórico del equipo).</p>	
--	--	--	--	--

Reunión Arranque 2da Parte: Reunión Técnica

Se desglosan las Historias de Usuario o FastTrack en tareas con su respectiva estimación de tiempo y se procede con la asignación. Este proceso debe ser validado y aprobado por el Scrum Master.

La planeación del sprint contempla:

- Reunión técnica: desglose de tareas y estimación respectiva.

- Definición de meta y compromiso del sprint (necesidades a atender, tamaño del sprint, fechas de compromiso).
- Definir los procesos de entrega continua.
- Resumen de estimación del equipo de desarrollo.
- Comunicación del compromiso adquirido.

Técnica a utilizar	Periodicidad	Personal requerido	Reglas	Resultado
Sesión de trabajo	Posterior a la reunión del Conceptualización.	<u>Responsable:</u> Scrum Master <u>Integrantes:</u> Scrum Team	<ul style="list-style-type: none"> • Definir el CAPACITY del Development Team. • Desglose de tareas por cada Historia de Usuario ò FastTrack. • Definir para cada tarea quién lo hará y cuánto durará. • La granularidad de las tareas se recomienda ser inferior a 8 horas. • Todas las tareas deben estar categorizadas según la disciplina (testing, development) • Manejo mínimo de estándar de tareas. • En la estimación inicial, ninguno de los colaboradores debe tener asignados tareas inferiores al 80 % de su capacidad dentro del Sprint. • Mantener periodicidad constante en la duración del sprint. • Definir reglas del sprint (hora y lugar de la reunión diaria, políticas de control de código, documentación a generar, técnicas de pruebas). • Se debe realizar una validación de la estimación por parte del Scrum Master. 	Gestión del sprint en VSTS. Desglose y estimación de tareas del sprint.

B. Daily Scrum (Scrum Diario)

Control del avance de Sprint.

- Se debe revisar el Burndown.
- Verificar avance grupal e individual según el panel de tareas.
- Cada miembro del equipo responder a las preguntas, (¿Qué hice ayer?, ¿Qué impedimentos tiene?, ¿Qué voy a hacer hoy?).
- Documentar posibles impedimentos o decisiones para apoyar equipo Scrum.

Técnica a utilizar	Periodicidad	Personal requerido	Reglas	Resultado
Reunión	Diario, durante el Sprint.	<u>Responsable:</u> Scrum Team <u>Integrantes:</u> Scrum Team	<ul style="list-style-type: none"> • No superar los 15 minutos. • Es obligatoria. • Apuntar los impedimentos • Verificación del gráfico de burndown del Web Portal. • Validar la actualización del Panel del Web Portal. • Realizar las preguntas diarias. (¿Qué hice ayer?, ¿Qué impedimentos tiene?, ¿Qué voy a hacer hoy?) • El Scrum Master puede delegar las labores de seguimiento de las reuniones diarias al líder técnico asignado. 	Gestión de impedimentos. Seguimiento del equipo.

C. Sprint Review (Revisión del incremento)

Presentación de parte del equipo Scrum al cliente (Producto Owner y representantes de Stakeholder) del incremento logrado con el sprint.

- Reunión con los interesados para mostrarles las nuevas funcionalidades o mejoras.
- Se presenta solo las necesidades terminadas. (Terminados son elementos de trabajo en estado DONE).

- Se presenta trabajo terminado a partir de una demo en el ambiente de pruebas.
- Identificar posibles observaciones referentes al incremento.

Técnica a utilizar	Periodicidad	Personal requerido	Reglas	Resultado
Reunión	Al inicio de pruebas de Aceptación de Usuario.	<u>Responsable:</u> Development Team <u>Integrantes:</u> Scrum Team Stakeholder	<ul style="list-style-type: none"> • Tiene que ser una demostración del Sistema Funcionando en un ambiente de pruebas. • Se presentan las historias de usuario o FastTrack desarrolladas y terminadas dentro del Sprint. • Se presentan las mejoras identificadas durante el proceso de pruebas. 	Entrega de Historias de Usuario desarrollados al Product Owner para inicio de pruebas de usuario. Inicio formal de pruebas de aceptación de usuario.

D. Sprint Retrospective (Retrospectiva del Sprint)

Actividad de mejora continua, para inspeccionar como fue el sprint en cuanto a personas, relaciones, comunicación, proceso y herramientas.

- Se responde a las preguntas: ¿Qué se hizo bien?, ¿Qué se puede mejorar?, ¿Qué tratará de hacer mejor el próximo Sprint?
- Identificar, ordenar y priorizar los elementos más importantes que salieron y las posibles mejoras.
- Crear un plan de acción para implementar las mejoras en el manejo del sprint.
- Como buena práctica se puede hacer uso de la técnica estrella de mar.

Técnica a utilizar	Periodicidad	Personal requerido	Reglas	Resultado
Reunión, start fish	Al Cierre del Sprint	Responsable: <ul style="list-style-type: none"> Scrum Master Otros integrantes: <ul style="list-style-type: none"> Scrum Team Product Owner (Opcional) 	<ul style="list-style-type: none"> Contestar las preguntas de la retrospectiva. Reunión en un lugar no formal. Generar plan de acción, para implementar mejoras. Retroalimentar al equipo de las conclusiones obtenidas. 	<ul style="list-style-type: none"> Apuntes de acuerdos y mejoras a implementar en el próximo sprint.

6.5.7 Lineamientos Artefactos de Registro en VSTS

A. Lineamientos Atención Épicas

Toda necesidad del usuario para un sistema específico que sea solicitada al Área de Ingeniería de Sistemas será ingresada como épicas en el Team Project de VSTS, registro por parte del Product Owner asignado al proyecto, dichas épicas deben ser analizadas en el Backlog de épicas por el Product Owner para su aprobación y realizar el desglose de las necesidades en historias de usuario o FastTrack según sea la necesidad del trabajo a desarrollar.

a) Flujo de trabajo

- Para cada registro de épica se debe documentar los siguientes campos al ingresarse a VSTS:

Título, descripción, criterios de aceptación, prioridad, adjuntos (Opcional), Estado (New).

- La épica es trabajada por el Product Owner según la prioridad en que se ingresó y la necesidad de negocio, se deben completar los siguientes campos:

Estado (Committed), asignado (Se debe al Product Owner o Stakeholder que realice el análisis).

- El análisis de la épica es finalizado, se deben completar los siguientes campos:

Estado (Done)

Nota: Durante todo el flujo de trabajo se pueden crear Link con trabajos relacionados (historias de usuario, FastTrack, Tareas) además se pueden crear discusiones y agregar tag.

b) Reglas

1. Una épica no será una entrada para el equipo de desarrollo.
2. Las épicas serán registradas por el Product owner.

c) Formulario de la épica:

The screenshot shows the 'NEW EPIC' form in Jira. The following fields are highlighted with red boxes:

- Asignado** (Assignee)
- Estado** (State) dropdown menu
- Reason** (Reason) dropdown menu
- Description** (Description) text area
- Related Work** (Related Work) section
- Prioridad** (Priority) dropdown menu
- Criterio de aceptación** (Acceptance Criteria) text area

Other visible fields include: Titulo, Comentarios (0 comments), Add tag, Save & Close, Area (MigraciónSIAC), Iteration (MigraciónSIAC), Start Date, Target Date, Details (Priority), Value area (Business), and Time & Effort.

Ilustración 10: Flujo de Trabajo.

d) Flujo de estados

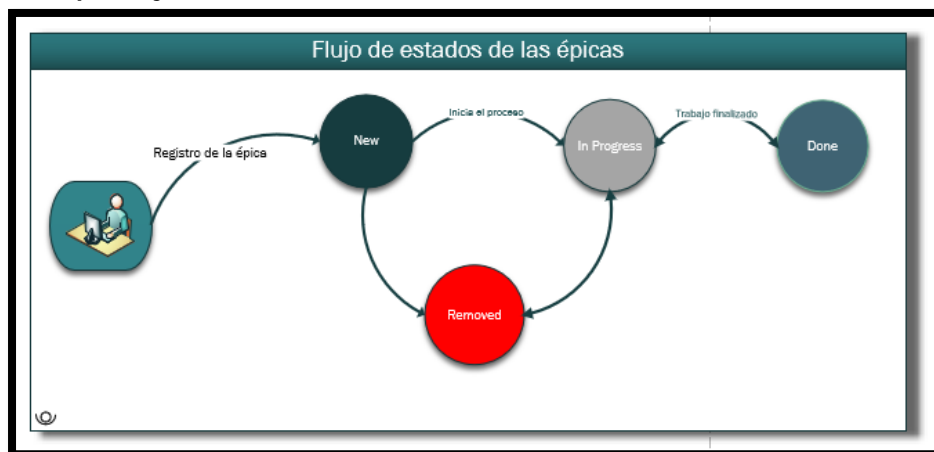


Ilustración 11: Flujo de Estados.

B. Lineamientos atención Historias de Usuario

Toda historia de usuario que sea solicitada al Área de Ingeniería de Sistemas será ingresada en Team Project en VSTS por parte del Product Owner asignado al proyecto, el registro puede provenir del análisis de una

épica o una necesidad registrada de manera directa sin estar enlazada a la épica, por otra parte, puede ser una mejora que este enlazada a un Bug el cual identifica la misma.

Las necesidades trabajadas como historia de usuario deben mantener un nivel mínimo de calidad, las HU serán trabajadas bajo el marco de atención de SCRUM, el cual lo conforman los sprint en un rango de 1 a 4 semanas tomando un máximo de 20 días hábiles.

a) Flujo de trabajo

- Para cada historia de usuario se debe documentar los siguientes campos al ingresarse a VSTS:
- Título, descripción, criterios de aceptación, prioridad, campos definidos por sistema y adjuntos (Opcional), Estado (New).

b) Prioridad de la historia de usuario

- La historia de usuario será priorizada por el product owner, se deben documentar los siguientes campos:

Priority (Prioridad)

a) Trabajo de la historia de usuario

- La historia de usuario es revisada en un sprint planning y aprobada, se deben completar los siguientes campos:

Estado (Approved), Effort, Asignado y seleccionar el sprint.

- La historia de usuario es trabajada en un sprint, se deben completar los siguientes campos:

Estado (Committed)

- La historia de usuario es finalizada en un sprint, se deben completar los siguientes campos:

Estado (Done), se habilita el campo de seguimiento para pruebas y pases a producción, el mismo debe ser gestionado según el seguimiento que está realizando el equipo Scrum a la historia de usuario.

Nota: Durante todo el flujo de trabajo se pueden crear Link con trabajos de desarrollo y con otros trabajos relacionados (Épicas, historias de usuario, FastTrack, Bug, Tareas), además se pueden crear discusiones y agregar tag.

c) Reglas

1. Una historia de usuario solo debe ser trabajada en el sprint planificado, la misma no debe ser atendida en un sprint no planificado.
2. La historia de usuario debe ser aprobada y gestionada por el Product Owner.
3. El trabajo de una historia de usuario finaliza una vez que finalice el sprint.

- b) La historia de usuario debe mantener los criterios mínimos para ser evaluada en un sprint, entendiéndose como criterio mínimo, título, descripción, criterio de aceptación (detallado), prioridad.

d) **Formulario de historia de usuario:**

The image shows a Jira User Story form with several fields highlighted in red boxes. The fields are: Título, Asignado, Estado, Descripción, Aceptance Criteria, and Discussion. The right sidebar contains sections for Details (Priority, Priority Note, Effort, Business Value), Development, and Related work. A 'Seguimiento' (Tracking) section lists 'Pruebas UAT', 'Pruebas Capacitación', 'Pase Producción', and 'Done Producción'. A 'Funcionalidad' field is also present. Two callout boxes on the right explain the tracking fields and the system-defined field.

Campos de seguimiento a pruebas y pase a producción una vez finalizado el desarrollo

Campos definidos por sistema

Ilustración 12: Formulario de Historia de Usuario.

e) Flujo de estados:

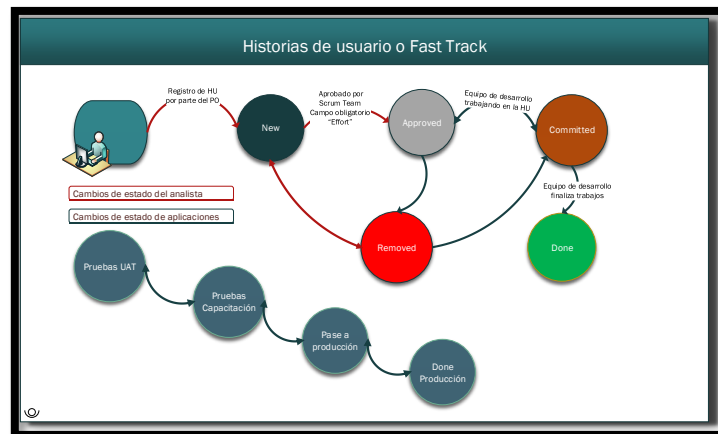


Ilustración 13: Flujo de Estados.

C. Lineamientos atención FastTrack

Los FastTrack que sean solicitados al Área de Ingeniería de Sistemas serán ingresados al Team Project en VSTS por parte del Product Owner asignado al proyecto, el registro puede provenir del análisis de una épica o una necesidad registrada de manera directa sin estar enlazada a la épica, por otra parte, puede ser una necesidad que este enlazado a un Bug la cual identifica la misma.

Los FastTrack serán clasificados de la siguiente manera:

- Extracción de datos.
- Modificación de datos.
- Investigación.

Las necesidades trabajadas como FastTrack deben mantener un nivel mínimo de calidad, los FastTrack serán trabajadas bajo el marco de trabajo SCRUM ò bajo el concepto de sprint no planificado.

a) Flujo de trabajo

- Para cada FastTrack se debe documentar los siguientes campos al ingresarse a VSTS:
- Título, descripción, criterios de aceptación, prioridad, campos definidos por sistema y adjuntos (Opcional), Estado (New).

b) Prioridad de los FastTrack:

- Los FastTrack priorizados por el Scrum Master y el product owner de manera conjunta, se deben documentar los siguientes campos

Priority (Prioridad)
- Los FastTrack priorizados por el Product Owner con los criterios de negocio, se deben documentar los siguientes campos:

c) Trabajo del FastTrack

- Los FastTrack que sean revisados en un sprint planning, se deben completar los siguientes campos:

Estado (Approved), Effort, Asignado y seleccionar el sprint.
- Los FastTrack que sean revisados en un sprint no planificado, se deben completar los siguientes campos:

Estado (Approved), Effort (Opcional), Asignado y seleccionar el sprint.

- Al iniciar los trabajos en el FastTrack, se deben completar los siguientes campos:

Estado (Committed)

- Los FastTrack finalizados, se deben completar los siguientes campos:

Estado (Done), se habilita el campo de seguimiento para de pruebas y pases a producción, el mismo debe ser gestionado según el seguimiento que está realizando el equipo Scrum a la FastTrack.

Nota: Durante todo el flujo de trabajo se pueden crear Link con trabajos de desarrollo y con otros trabajos relacionados (Épicas, historias de usuario, FastTrack, Bug, Tareas), además se pueden crear discusiones y agregar tag.

d) Reglas

1. Los FastTrack se pueden trabajar en sprint planificado o no planificado.
2. La FastTrack debe ser aprobada y gestionados por el Product Owner ò en algunos casos los podrá registrar el equipo y hacer de conocimiento al Scrum Master para gestionar la prioridad de trabajo del mismo.

3. EL trabajo de un FastTrack en un sprint planificado finaliza una vez que finalice el sprint, los trabajo en un sprint no planificado finalizaran una vez que el FastTrack cambie a estado Done.
4. Los FastTrack debe mantener los criterios mínimos para ser evaluada en un sprint planificado o no planificado, entiéndase como criterio mínimo, título, descripción, criterio de aceptación (detallado), prioridad

e) Formulario de FastTrack:

The screenshot shows the 'NEW FAST TRACK' form with the following fields and sections highlighted in red:

- Title** (Título)
- Assignee** (Asignado)
- Status** (State: New)
- Area** (Área)
- Capacity** (Capacitación: Sprint)
- Description** (Description)
- Acceptance Criteria** (Criterio de Aceptación)
- Discussion** (Discussion)
- Custom** (Custom) section containing:
 - Priority (Prioridad)
 - Nota (Note)
 - Effort (Esfuerzo)
- Seguimiento** (Tracking) section containing:
 - Pruebas UAT (UAT Tests)
 - Done Producción (Done Production)
- Funcionalidad** (Functionality) section
- Development** (Development) section
- Related Work** (Related Work) section

Annotations in black boxes provide additional context:

- A box next to the 'Seguimiento' section states: "Campos de seguimiento a pruebas y pase a producción una vez finalizado el desarrollo" (Tracking fields for tests and production once development is finished).
- A box next to the 'Funcionalidad' section states: "Campos definidos por sistema" (Fields defined by the system).

Ilustración 14: Formulario de FastTrack.

f) Flujo de estados:

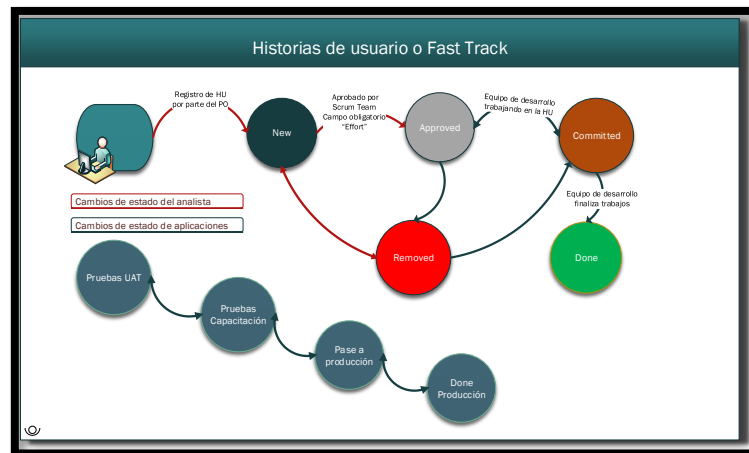


Ilustración 15: Flujo de Estados.

D. Lineamientos para Ítems de Seguimiento

Los Ítems de seguimiento tendrán la labor de encapsular las tareas correspondientes a:

- Reuniones planificadas dentro del marco de trabajo.
- Labores de gestión administrativa.
- Capacitaciones.

De manera inicial en los sprint se crean los Ítems de seguimiento:

- Ceremonias, dicha clasificación mantendrá las ceremonias planificadas en el marco de trabajo.
- Tareas Generales, dicha clasificación contendrá labores de gestión administrativa, capacitaciones, reuniones no planificadas.

a) Flujo de trabajo

- Los mismos son elementos de trabajo que permanecerán en estado committed.
- En el caso de querer cerrarse el objeto de trabajo se cambia a estado Done.

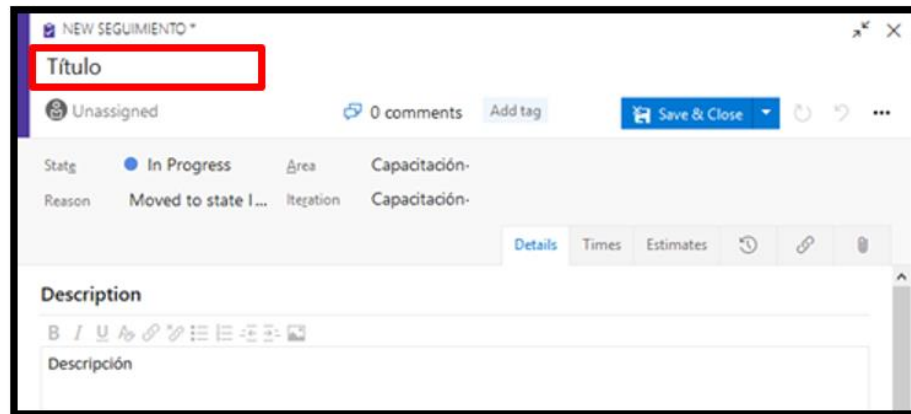


Ilustración 16: Flujo de Trabajo.

b) Reglas

1. Los ítems de seguimiento solo deben contener tareas relacionadas a gestión administrativa, las mismas no deben tener registros de labores que corresponden al trabajo de un ítem de trabajo.

E. Lineamientos atención Bugs

Los Bug`s que sean ingresados al Área de Ingeniería de Sistemas serán ingresados al Team Project en VSTS por parte del Scrum Master o una persona del equipo que sea asignada en el proyecto.

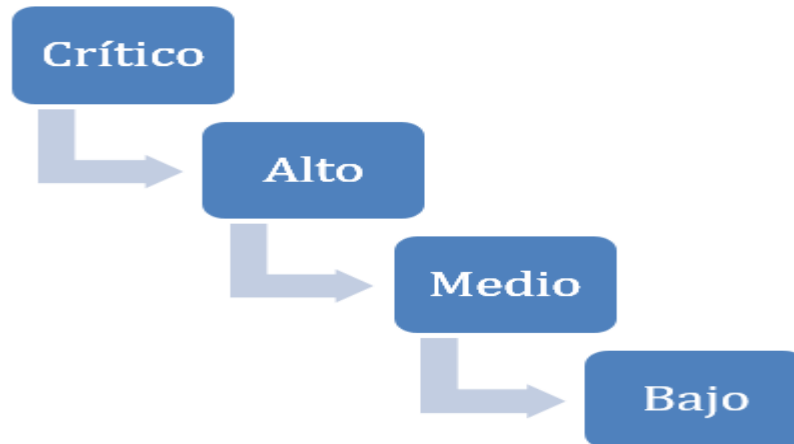
Los Bug`s serán clasificados de la siguiente manera:

- Error de ambiente.
- Error de aplicación.
- Error de datos.
- Error de usuario.

La clasificación del registro de errores se realizará de la siguiente manera:



Los errores serán evaluados en severidad de la siguiente manera:



a) Criterios de severidad

- Errores Críticos:
 - Se clasificarán como errores críticos todo aquello que debe ser solucionado.
 - Son los resultados que provoquen la terminación de uno o más componentes del sistema o del sistema en general.
 - Corrupción de datos.
 - No existe ningún otro método alternativo aceptable para lograr los resultados requeridos.
- Errores Altos
 - Se debe considerar la corrección.
 - Son todos los resultados que provoquen la terminación de uno o más componentes del sistema o del sistema en general.
 - Corrupción de datos.
 - Pero lo que diferencia un error con severidad alta a uno crítico, es que el de severidad alta si existe un método alternativo aceptable para lograr los resultados requeridos.
- Errores Medios
 - Son todos los resultados incorrectos, incompletos o inconsistentes.
 - Es el incumplimiento en funciones secundarias.
- Errores Bajos
 - Se clasifican defectos menores o cosméticos.

- Son todos aquellos resultados en la presentación de datos.
- No adecuación de estándares, estos tienen soluciones aceptables para lograr los resultados requeridos.

b) Flujo de trabajo

- Para cada Bug se debe documentar los siguientes campos al ingresarse a VSTS:
- Título, Repro steps, criterios de aceptación, severidad, Tipo Incidencia, Reportado por, campos definidos por sistema y adjuntos (Opcional), Estado (New).

Los bugs registrados VSTS por pruebas internas y pruebas de aceptación de usuario serán registrados por el equipo de aseguramiento de la calidad.

Los Bugs registrados VSTS por producción, serán ingresados por los usuarios finales mediante el canal que se defina, por ejemplo, mesa de servicio y registrados a VSTS por un encargado del equipo de desarrollo asignado por el Scrum master o el Scrum master.

El flujo de registro, asignación y prioridad de los bugs

- Bugs registrados por producción: Serán priorizados acorde con la prioridad y severidad, un miembro del equipo de desarrollo que registra y asigna la revisión, se deben documentar los siguientes campos:

Estado (Approved), Asignado, Iteration (se asigna el sprint no planificado en el que se va a trabajar).

- Bugs registrados por desarrollo: Serán priorizados acorde con la prioridad y severidad, un miembro del equipo de desarrollo registra y asigna la revisión, se deben documentar los siguientes campos:

Estado (Approved), Asignado, Iteration (se asigna el sprint no planificado en el que se va a trabajar).

- Bugs registrados por pruebas de aceptación: Serán priorizados acorde con la prioridad y severidad en el sprint que está activo, un miembro de aseguramiento revisa el reporte realizado por los usuarios, lo registra en VSTS y asigna la revisión, se deben documentar los siguientes campos:

Estado (Approved), Asignado, Iteration (Sprint activo planificado).

- Bugs registrados por pruebas internas: Serán priorizados acorde con la prioridad y severidad en el sprint que está activo, un miembro de aseguramiento registra el reporte en VSTS y asigna la revisión, se deben documentar los siguientes campos:

Estado (Approved), Asignado, Iteration (Sprint activo planificado).

- Al iniciar los trabajos en el Bug, se deben realizar un análisis y planificación de las tareas a desarrollar, se deben completar los siguientes campos:

Estado (Committed)

- Al finalizar la reparación del Bug, se deben completar los siguientes campos:

Estado (Fixed), en discusión es opcional agregar observaciones de la reparación realizada.

- Si el bug se rechaza, se deben completar los siguientes campos:

Estado (Rejected), en discusión se debe agregar observaciones del rechazo.

- Para los bugs rechazados se debe realizar una validación por el equipo de aseguramiento de la calidad:

- Si el error permanece o el criterio de rechazo no justifica la no atención del bug, se completan los siguientes campos.

Estado (Reopen), en discusión se debe agregar observaciones de la reapertura del bug.

- Si el error reportado se determina que no es un error, se completan los siguientes campos.

Estado (Removed), en discusión se debe agregar una justificación de la anulación.

- Si el error se determina que si es un bug y el criterio de rechazo justifica la no atención, el mismo debe ser coordinado con el Scrum

Master como criterio de aprobación, el mismo es registrado en el reporte de resultados del sprint.

Los Bugs finalizados, se deben completar los siguientes campos:

Estado (Done), se habilita el campo de seguimiento de pruebas y pases a producción, el mismo debe ser gestionado según el seguimiento que está realizando el equipo Scrum al Bug (Esta acción de seguimiento no aplica para pruebas internas).

Nota: Durante todo el flujo de trabajo se pueden crear Link con trabajos de desarrollo y con otros trabajos relacionados (Épicas, historias de usuario, FastTrack, Bug, Tareas), además se pueden crear discusiones y agregar tag.

c) Reglas

1. Los Bugs se pueden trabajar en sprint planificado o no planificado.
2. El Bug debe ser aprobada y gestionados por el Development Team.
3. EL trabajo de un Bug en un sprint planificado finaliza una vez que finalice el sprint, los trabajo en un sprint no planificado finalizaran una vez que el Bug cambie a estado Done.
4. Los Bugs debe mantener los criterios mínimos para ser evaluada en un sprint planificado o no planificado, entiéndase como criterio mínimo, título, descripción del error, criterio de aceptación (detallado), Severidad, Tipo de incidencia, Reportado por.

F. Lineamientos para el Manejo de Tareas

En los sprint planificados las tareas son definidas al inicio del sprint en la segunda fase del sprint planning a partir del desglose de tareas de una historia de usuario, FastTrack o Bug, se debe crear la planificación de las tareas necesarias para desarrollar una necesidad específica, además se debe de tomar en cuenta que para cada sprint se debe mantener una estructura de PBI's tal como se detalla a continuación:

- Historia de usuario, FastTrack o Bug
- Ceremonias
- Tareas Generales

Dentro de cada PBI se detallará las tareas que cada miembro del equipo aporta, se recomienda tener un estándar de tareas por sprint, a continuación, un detalle de un ejemplo de las tareas que podría contener un PBI.

PBI	Detalle de tareas que debe contener
Historia de usuario, FastTrack ò Bug	Dev/QA - Análisis de la HU Dev - Tareas técnicas de desarrollo Dev - Pruebas técnicas locales Dev - Documentación técnica Dev - Aplicar criterios de aceptación Dev - Integración y entregas parciales QA - Gestión y creación de set de datos HU QA - Creación de los casos de prueba HU QA - Ejecución de pruebas y reporte de Bugs HU QA - Pruebas de regresión HU QA - Seguimiento de bugs HU Dev/QA - Instalación y estabilización del ambiente de pruebas HU
Ceremonias	Sprint Planning (por cada participante) Daily meeting (por cada participante) Sprint review (por cada participante) Retrospectiva (por cada participante) Nota: Documentación (en esta tarea debe adjuntarse la documentación por sprint, no debe estimarse y debe estar en done en cada Sprint)
Tareas Generales Sprint-QA	QA - Creación plan de pruebas QA - Creación reporte de resultados del Sprint. QA - Acompañamiento a usuarios QA - Seguimiento de bugs QA - Proceso de Re-testing (Funcionalidad) Dev/QA - Resolución de consultas. Dev/QA - Reuniones de análisis. Dev/QA – Capacitaciones. Dev/QA – Reuniones administrativas. Dev/QA - Instalación y estabilización del ambiente Dev/QA - Gestión de paquetes Dev/QA – Preparación y revisión de insumos

Se recomienda para el manejo documental de cada Sprint, ingresar una tarea en el PBI de ceremonias llamada: “Documentación Sprint”, en esta tarea debe adjuntarse toda la documentación generada dentro del sprint (Minutas de ceremonias (Opcional), Acuerdos, Plan de pruebas, Plan de resultados de pruebas, entre otras). El estado de esta tarea debe estar en “Done” sin tener un valor estimado.

a) Flujo de trabajo

- Al registrar la tarea se debe documentar los siguientes campos al ingresarse a VSTS:

Título, asignado a, Original Estimate (estimado original), Remaining Work (trabajo pendiente), Activity (tipo de actividad), Activity Date (Fecha de actividad), Estado (To Do).

Estado (In Progress)

- En el caso de no completar los trabajos en la tarea y mantener trabajo pendiente, se deben completar los siguientes campos:

Estado (New), Remaining Work (trabajo pendiente), Complete Work (trabajo completado), Tiempo Teletrabajo (parte del tiempo completado realizado en teletrabajo).

- Las tareas finalizadas, se deben completar los siguientes campos:

Estado (Done), Descripción (Opcional), Complete Work (trabajo completado), Tiempo en teletrabajo (parte del tiempo completado realizado en teletrabajo).

Nota: Durante todo el flujo de trabajo se pueden crear Link con trabajos de desarrollo y con otros trabajos relacionados (Épicas, Historias de Usuario, FastTrack, Bug), además se pueden crear discusiones, adjuntos documentos y agregar tag.

b) Reglas

1. Los tiempos se deben manejar en proporción a horas.

Equivalencia	
1	1 hora
0.75	$\frac{3}{4}$ de hora
0.50	$\frac{1}{2}$ hora
0.25	$\frac{1}{4}$ hora

2. Durante el progreso de las tareas las mismas deben ir cambiando de estados y siendo actualizadas de manera continua según el desarrollo del trabajo.
3. No se permiten mantener dos tareas en progreso de manera paralela.
4. El registro de tareas debe estar enlazado a un trabajo asignado sobre una historia de usuario, un ítem tipo FastTrack, un Bug o un ítem de seguimiento, las mismas no deben existir sin esa relación con un ítem principal de relación padre.
5. El campo de descripción en las tareas.

c) Formulario de tareas

NEW TASK* The identity value 'Asignado' for field 'Assigned To' is an unknown identity.

Titulo

Asignado 0 comments Add tag Save & Close

State: To Do Area: Capacitación-1 Reason: New task Iteration: Capacitación-1.18.5

Description

Descripción

Details

Priority: 2

Original Estimate: Trabajo estimado

Remaining Work: Trabajo pendiente

Completed Work: Trabajo completado

Time Teletrabajo: Tiempo en teletrabajo

Development

+ Add link

Development hasn't started on this item.

Related Work

+ Add link

Parent: 98 Cotización y compra de r... Updated 04/09/2018, @ New

Discussion

Add a comment. Use # to link a work item, / to link a pull request, or @ to mention a person.

Time & Effort

Ilustración 19: Formulario de tareas.

d) Flujo de estados:

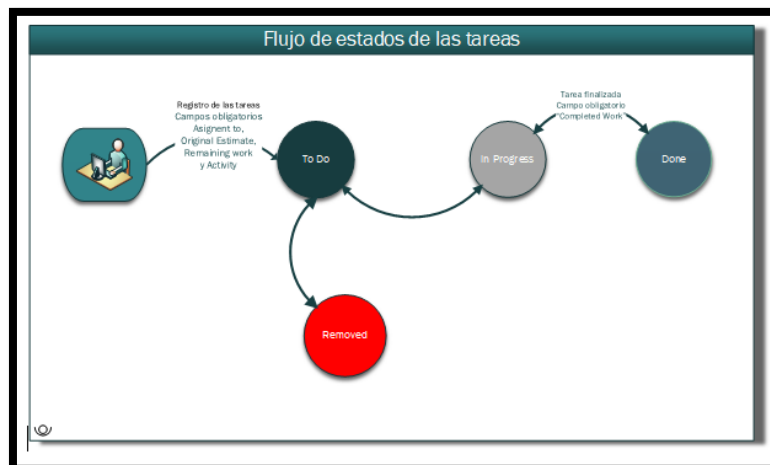


Ilustración 20: Flujo de Estados.

6.5.8 Manejo de Sprint

A. Manejo planificado

Sprint que lo conforma y planifica el marco de trabajo de Scrum, el mismo permite como entrada, elementos de trabajo de tipo:

- Historia de usuario
- FastTrack
- Bug`s

El manejo de las necesidades y todos los ítems trabajados en la herramienta está definido en el apartado de lineamientos, para los sprint planificados se realiza el pesaje de las necesidades, según como se documenta en la sección “8.4 Pesaje de necesidades”.

La definición de la cantidad de puntos de historia que se agregan a un sprint la define la velocidad promedio del equipo según el historial de los sprint anteriores.

A continuación, se muestra un ejemplo de cómo debe quedar una planeación de un sprint posterior a realizar la estimación de tareas.

Order	Work Item Type	Title	State
1	Seguimiento	Tareas Generales	In Progress
2	Seguimiento	Ceremonias	In Progress
3	Product Backlog	Cotización y compra de repuestos	New
4	Product Backlog	Enderezado de marcos de las puertas y soportes de los fa...	New

Ilustración 21: Planeación de Sprint.

Se muestra un ejemplo del detalle de una Historia de usuario:

Title	State	Effort	Original Estimate	Remaining...	Completed...
HU 01 - Mantenimiento de Costos por Canal	Committed	5		40	
QA - Análisis de la HU	To Do		6	6	0
QA - Creación de los casos de pruebas	To Do		6	6	0
QA - Seguimiento de bugs	To Do		4	4	0
Dev - Análisis	To Do		1	1	0
Dev - Diseño	To Do		1	1	0
QA - Ejecución de casos de pruebas y reporte de bugs HU	To Do		6	6	0
Dev - Creación nuevas estructuras y permisos	To Do		2	2	0
Dev - Modificación al menú de conectividad	To Do		2	2	0
Dev - Creación de la forma de mantenimiento	To Do		6	6	0
Dev - Cambios al paquete de auditoría	To Do		3	3	0
Dev - Pruebas internas	To Do		2	2	0
Dev - Documentación interna	To Do		1	1	0

Ilustración 22: Historias de Usuario

B. Manejo No planificado

Para el trabajo no planificado por cada Team Project debe existir un sprint no planificado los mismos deben ser paralelos a los sprint planificados, con el fin de asignar tareas adicionales que realicen los recursos y que no sean propias del sprint planificado, se permite como entrada a:

- FastTrack
- Bug`s (Clasificados como incidentes de producción)
 - El ingreso de incidentes de producción a un sprint planificado o no planificado es una decisión del Product Owner en conjunto con el Scrum Master.

El manejo de las necesidades y todos los ítems trabajados en la herramienta está definido en el apartado de lineamientos.

El desarrollo de dicho sprint debe ser de conocimiento de Producto Owner, el cual mantendrá comunicación con el Scrum Master para la gestión del trabajo asignado al mismo.

C. Granularidad de tareas

Las tareas deben ser estimadas como un máximo de 8 horas, cualquier tarea que se salga de ese número habrá que detallarla en tareas de menor valor.

La planificación de tareas del trabajo en el sprint se realizará en la segunda fase del sprint planning, en el caso de los sprint no planificados la planificación se realizará al elemento de trabajo una vez que sea asignado el mismo.

D. Pesaje de necesidades

Para el pesaje de la historia de usuario, FastTrack ò Bug se tiene que utilizar la sucesión de Fibonacci; en donde se define el nivel de esfuerzo que se requerirá para lograr la completitud.

Los valores referencia están identificados en la siguiente tabla, y deben ser incluidos en el campo “Effort” de cada PBI. Se recomienda bajar la aplicación llamada “Scrum Póker”, para ser usada desde el teléfono al momento de la votación. (para más información de la técnica ver el apartado de métodos de apoyo al marco de trabajo, Técnica Planning Póker).

Peso	Complejidad	Valoración del colaborador (individual)
0	Mínimo	El Compromiso para hacer la HU sería muy simple y no requiere esfuerzo.
1	No significativo	El Compromiso para hacer la necesidad sería para algo puntual por resolver que no tarde más de 3% resolverlo.
2	Ligera	El Compromiso para hacer una necesidad sería para algo puntual que no tarde más de 5% resolverlo.
3	Nivel mínimo de complejidad	Compromiso para hacer la HU, en un 10% del largo del Sprint, asumiendo su aporte al 100 % para actividades exclusivas de su Rol.
5	Nivel bajo de complejidad	Compromiso para hacer la necesidad, en un 20% del largo del Sprint, asumiendo su aporte al 100 % para actividades exclusivas de su rol.
8	Nivel intermedio de complejidad	Compromiso para hacer la necesidad, en el 50% del largo del Sprint, asumiendo su aporte al 100 % para actividades exclusivas de su rol.
13	Complejo	Compromiso para hacer la necesidad, a lo largo de todo el Sprint, asumiendo su aporte al 75 % para actividades exclusivas de su rol.
21	Altamente complejo	Compromiso para hacer solo esta necesidad, a lo largo de todo el Sprint, asumiendo su aporte al 100 % para actividades exclusivas de su rol
34	Muy compleja, requiere análisis para determinar complejidad.	El colaborador valora que por sí solo no podría comprometerse a lo largo del Sprint a terminar la necesidad, pero con la capacidad total del equipo si se podría terminar. Esto representa que se requiere más esfuerzo de un solo colaborador:
∞	Inestimable	Con el detalle actual, no se puede tener certeza de una estimación de esfuerzo aproximada. Se necesita detallar más la necesidad o en su defecto granularlo en otras historias de usuario.

E. Velocidad promedio del equipo

Para la velocidad promedio y estimaciones se debe contemplar el historial de las votaciones de esfuerzo de los sprint anteriores, sacando el promedio histórico de la velocidad del equipo. Si los sprint anteriores han sido atípicos, se debe tomar como referencia de velocidad el último sprint.

La velocidad promedio indicará la cantidad de historias de usuario que el equipo de desarrollo puede comprometerse en el sprint y permitirá estimar el compromiso del equipo por sprint.

A manera de referencia teórica del esfuerzo se plantea la siguiente ecuación para el cálculo del esfuerzo, basada en el análisis histórico de los equipos SCRUM, esta ecuación tiene parámetros que variarán conforme se madure en la implementación de la metodología Scrum.

La ecuación resultante fue $Z=0,746X+0.487Y$, donde:

Valor	Equivalencia	Observaciones
Z	Esfuerzo teórico diario resultante.	Este esfuerzo teórico resultante debe multiplicarse por el tamaño en días hábiles del sprint, para obtener el esfuerzo por sprint.
X	Cantidad de desarrolladores que conforman el equipo	Pueden utilizarse dentro de la fórmula cantidades no enteras, ejem: 0.5, que equivale a un desarrollador a medio tiempo durante el Sprint.
Y	Cantidad de Analistas de QA	Pueden utilizarse dentro de la fórmula cantidades no enteras, ejem: 0.25, que equivale a un tester aun cuarto de tiempo durante el Sprint (2 hrs diarias).
A=0.746	Coeficiente aporte de esfuerzo diario por desarrollador	Equivale al aporte teórico diario en términos de puntos de esfuerzo por cada desarrollador.
B=0.487	Coeficiente aporte de esfuerzo diario por analista de calidad.	Equivale al aporte teórico diario en términos de puntos de esfuerzo por cada QA.

A manera de ejemplo, para el cálculo del esfuerzo, solo se debe incluir las variables: Cantidad de desarrolladores, cantidad de QAs y tamaño del Sprint, tal como se muestra en el ejemplo:

Cantidad Dev(X)	Parámetro A	Cantidad QA(Y)	Parámetro B	Esfuerzo teórico diario(Z)	Tamaño Sprint	Esfuerzo teórico por Sprint
3	0,75	1	0,49	2,74	20	55

F. Manejo capacidad por colaborador

El coordinador debe tener control de la capacidad que tendrá cada colaborador del equipo de desarrollo en los diferentes sprint.

Se recomienda distribuir la capacidad máxima de cada colaborador en los diversos sprint que participa, de manera que su asignación en cada sprint sea productiva y acorde con su disponibilidad.

Para lo anterior debe completar la información del CAPACITY en cada sprint, en la siguiente pantalla:

User	Days off	Activity	Capacity per day
Alexánder Delgado Castillo	2 days	Deployment	8
Alexánder Delgado Castillo	0 days	Deployment	5
Alexánder Delgado Castillo	0 days	Testing	2
Alexánder Delgado Castillo	0 days	Unassigned	0
Alexánder Delgado Castillo	0 days	Deployment	8
Alexánder Delgado Castillo	0 days	Unassigned	0
Alexánder Delgado Castillo	0 days	Unassigned	0
Alexánder Delgado Castillo	0 days	Unassigned	0
Alexánder Delgado Castillo	0 days	Unassigned	0
Team days off	0 days	These days off apply to the whole team.	

Ilustración 23: Capacity.

G. Definición de DONE

La definición de "Terminado" es un entendimiento compartido de completado para el producto.

Los criterios de DONE, serán utilizados para que el Scrum Team identifique cuando una historia de usuario, FastTrack, bug o tarea se da

por “Terminado”. Para esta implementación se toma el Done, todo trabajo que se finaliza durante el sprint.

H. Entrega continua

Bajo la filosofía Scrum, cada sprint debe mostrar un incremento que sea validado por el usuario, debe de propiciarse la agilidad en el manejo del desarrollo y pruebas, para una liberación continua.

Se propone la entrega continua a nivel del equipo SCRUM (desarrolladores y aseguradores), que consiste en que haya revisiones inmediatas a los requisitos que se van finalizando en desarrollo. Esta tendencia a la entrega continua significa que el Scrum Team, se debe organizar para la entrega, instalación y pruebas por cada historia de usuario o FastTrack desarrollados, de manera que la fase de pruebas internas se ejecutaría en paralelo al desarrollo.

Dado que se va a trabajar bajo un marco de despliegue continuo, queda a discreción del equipo poder entregar Historias de Usuario, FastTrack o partes de ellos (siempre y cuando funcionalmente estén completos) para pruebas de aceptación de usuario, con la finalidad de adelantar el proceso de acreditación, si es importante que el acumulado en el incremento del software es la entrega final del sprint.

La integración continua es un concepto que surge con la automatización del proceso de compilación y despliegues a nivel de herramientas de apoyo, el modelo ideal de integración continua permite la

Diseño de pruebas internas	QA	Casos de prueba a aplicar por HU
Instalación y estabilización de HU en ambiente pruebas	QA	HU instalado en ambiente de pruebas.
Pruebas QA de la HU	QA	Script de pruebas ejecutado.
Gestión bugs asociados al HU	QA Construcción	Reporte de bugs.
Aprobación por HU	QA	Aprobación HU
Gestión pase a pruebas usuario	Equipo desarrollador	Pase a ambiente de usuario.
Ejecución de pruebas de aceptación Usuario	Producto Owner o Usuario	Guion de pruebas ejecutadas y bugs asociados.
Gestión Bugs	Equipo desarrollador	Corrección y aprobación de bugs.
Aprobación usuaria de HU.	Usuario	Aprobación del HU.

A continuación, se identifica el uso de los principales recursos conforme avanza el sprint, donde se observa que el personal de desarrollo y QA, van de la mano aportando valor al sprint a lo largo de todas las fases del Sprint.

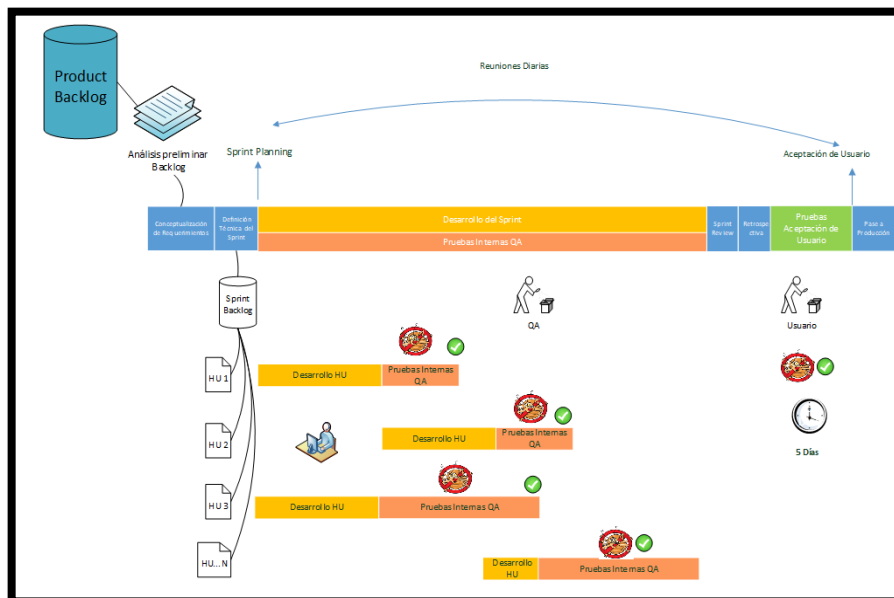


Ilustración 24: Flujo de Reuniones Diarias.

Técnica a utilizar	Periodicidad	Personal requerido	Reglas	Resultado
<p>Aprobación por HU desarrollada.</p>	<p>Durante el sprint por cada HU desarrollada.</p>	<p><u>Responsable:</u> development Team</p> <p>Stakeholder</p> <p><u>Otros Integrantes:</u> Scrum Master Product Owner</p>	<p>La Historia de usuario tiene que ser atómico, de tal manera que se pueda programar, probar e implementar de manera independiente.</p> <p>La historia de usuario debe ser aprobado por el Líder del equipo de desarrollo para el pase a QA.</p> <p>El Scrum Team tiene que estar sincronizado de tal manera que conozcan el orden con que el equipo construirá los HU.</p> <p>Los desarrolladores tienen que realizar el acompañamiento a los colaboradores de QA, para instalación de las Historias de Usuario en los equipos de QA y programación de pases.</p> <p>Dar Prioridad a la corrección de Bugs.</p> <p>Gestionar los Bugs en VSTS.</p> <p>Para el pase a usuario, todo HU tiene que ser aprobado por QA.</p> <p>Aprobación de Usuario: debe dar el visto bueno de la HU posterior a la corrección de todos los Bugs asociados a las Historias de Usuario.</p>	<p>Entrega inmediata de la HU posterior a desarrollo.</p> <p>HU funcionando en los equipos de Acreditación o Pruebas de Usuario.</p> <p>Aprobación de QA</p> <p>Aprobación de Usuario</p>

I. Paralelismo entre QA y Desarrollo

Se presentan las actividades donde los equipos de desarrollo y acreditación deben trabajar en conjunto para sincronizar el esfuerzo, por ende, según se visualiza en la gráfica.

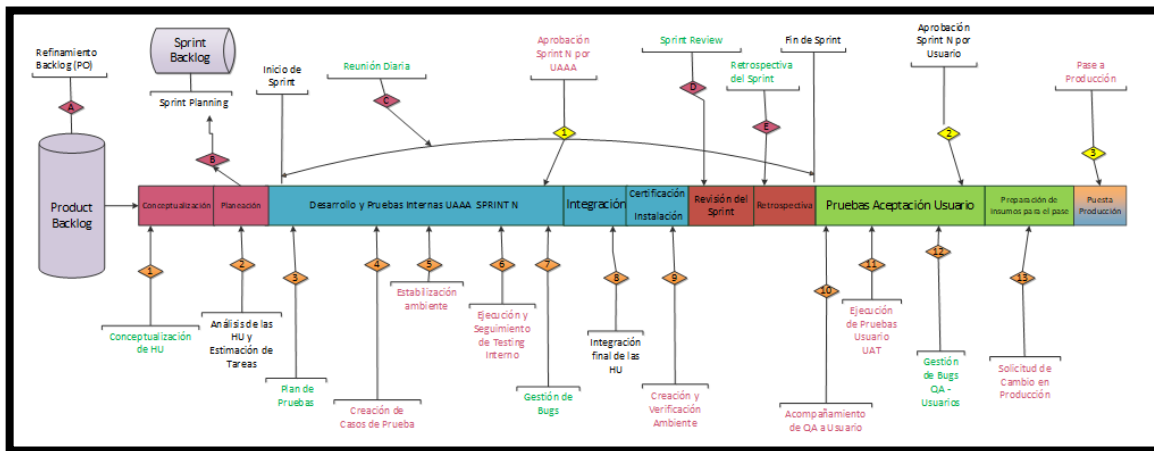


Ilustración 25: Refinamiento de BackLog.

Actividad	Development	QA
(1) Sprint Planning-Conceptualización	Conceptualiza y vota el esfuerzo por HU.	Conceptualiza y vota el esfuerzo por HU.
(2) Reunión técnica-Estimación	Define las tareas de desarrollo y gestión, por parte de ingeniería	Define las tareas de QA y gestión, por parte de ingeniería
(C)Reunión diaria Scrum	Asistir y dar retroalimentación del avance diario.	Asistir y dar retroalimentación del avance diario.
(5) Estabilización ambiente	Proveer los entregables del pase a calidad. Acompañamiento en instalación y resolución de posibles incidentes relacionados.	Revisar los entregables del pase. Instalación y resolución de posibles incidentes relacionados.
(6) Pruebas Internas QA	Corrección de Bugs	Ejecución de pruebas y gestión de bugs.
(D)Sprint Review	Realizar la Demo	Realizar la Demo
(8) Entrega para pruebas a Usuario.	Proveer los entregables del pase a QA.	Revisar los entregables del pase.
(9) Pruebas usuario	Corrección de Bugs	Acompañamiento de pruebas y gestión de bugs.
(14) Pase a producción	Solicitud y gestión el pase a producción.	Solicitud y gestión el pase a producción.

	Acompañamiento en instalación y resolución de posibles incidentes relacionados.	Acompañamiento en instalación y resolución de posibles incidentes relacionados.
(E) Retrospectiva	Participar en la reunión y proponer mejoras.	Participar en la reunión y proponer mejoras.

6.6 Métodos y Técnicas de Apoyo al Marco de Trabajo Ágil

A. Método SMART

Técnica de formulación de objetivos, en el cual cada sigla refiere a una característica a revisar.

- **S** de específico (Según su traducción en inglés).

Los objetivos son más específicos y detallados.

- **M** de medible (Según su traducción en inglés).

El objetivo debe ser medible. Se marcarán los parámetros necesarios para conseguir el objetivo.

- **A** de alcanzable (Según su traducción en inglés).

Búsqueda de evidencias para determinar la posibilidad de ser alcanzable.

- **R** de resultado enfocado - Realista (Según su traducción en inglés).

Los esfuerzos deberán ser medidos, consistentes con los recursos.

Nota: Existe una delgada línea entre no ser realista y tener la creencia de que se puede conseguir. Es necesario e importante buscar las evidencias para convertir el objetivo en realista.

- **T** de temporizado (Según su traducción en inglés).

El objetivo es el tiempo, es decir lo no agendado estará por fuera

Ejemplo de aplicación del método:

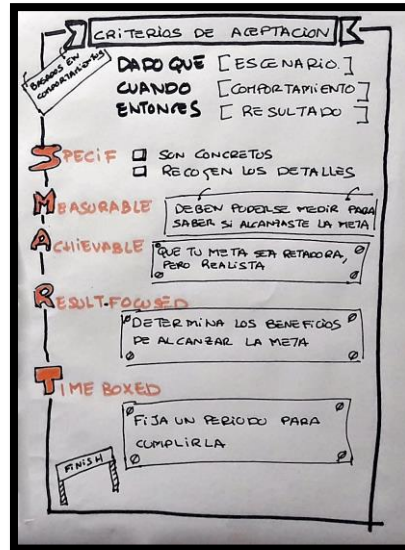


Ilustración 26: S.M.A.R.T.

B. Técnica estrella de mar

La técnica de la estrella de mar permite ver de una manera evolutiva las retrospectivas existentes. Un análisis de esta técnica permite ver la evolución del proyecto y la resolución de problemas.



Ilustración 27: Técnica Estrella de Mar.

Comenzar a hacer: Todas aquellas cosas innovadoras o que nos gustaría poner en marcha, pero entrañan riesgo por su novedad y nuestra inexperiencia.

Más de: Prácticas que aportan valor, son útiles y positivas en nuestro proyecto, será alimentado por [**Comenzar a hacer**] en la siguiente retrospectiva. Es decir que aquellas cosas que estamos utilizando o realizando y que se requiere que se mejore, se deberá practicar y darle más importancia.

Seguir haciendo: Cosas buenas que nos aportan valor y que debemos seguir haciendo tal y como las hacemos actualmente.

Más de: Llegaría a un nivel de maduración y se convertirá en [**Seguir haciendo**]. Es aquello que venimos haciendo bien y debemos seguir haciéndolo, pero no será preocupación mejorarlo.

Menos de: Prácticas que no están ayudando tanto como se esperaba, o que simplemente no son útiles en nuestro escenario actual, tenemos que empezar a abandonar esta línea de trabajo o al menos parte de ella.

Dejar de hacer: Prácticas que no son útiles o no agregan valor. Por lo tanto, hemos decidido eliminarlas completamente. En este punto exponemos aquellas prácticas que a pesar de ser [**Menos de**] podemos eliminarlas.

C. Técnica Planning póker

Es una técnica para calcular una estimación basada en el consenso de equipo, utilizada para estimar esfuerzo.

Durante la reunión de sprint planning cada miembro del development team tendrá un conjunto completo de tarjetas o podrá utilizar una aplicación móvil que cuente con la secuencia de Fibonacci.



Ilustración 28: Técnica Planning Póker.

Sobre el proceso de estimación:

- El Scrum Master realiza la labor de moderador, no realizara votación a menos que participe como parte del equipo de desarrollo.
- Cada persona coloca una tarjeta boca abajo que representa su estimación. Las unidades utilizadas deben ser definidas previamente. La estimación se realiza en puntos de la historia de usuario, FastTrack o Bug. Durante el debate, los números no debe ser mencionados en absoluto.
- Todo el development team muestra sus tarjetas de forma simultánea.
- A las personas con estimaciones altas y bajas se les da un tiempo para ofrecer su justificación para la estimación y la discusión continúa.

- Se repita el proceso de cálculo hasta que se alcance un consenso. De repetirse la votación 3 veces consecutivas sin lograr un consenso del equipo el moderador puede negociar el consenso.
- Se puede utilizar un tiempo específico para que la discusión sea estructurada, se realiza la sugerencia de 1 minuto como valor máximo o a criterio del Scrum Master tomando en cuenta complejidad, el moderador podrá en cualquier punto terminar el tiempo de discusión, es ese momento es cuando se acaba toda discusión y otra ronda de póker se juega.

6.7 Procesos de aseguramiento de calidad en el Software

Los beneficios que se pueden obtener como resultado de aplicar los procesos de aseguramiento de calidad son muchos y variados, algunos que se pueden citar con brevedad son:

1. Se detectan problemas rápidamente, es posible identificar problemas en tempranas etapas del desarrollo de productos de software, ayudando al desarrollador a corregirlos inmediatamente y poder avanzar con más rapidez.
2. Se crean y se siguen estándares de trabajo, Con apoyo del proceso de aseguramiento de calidad, se pueden establecer estándares tan diversos como son los de codificación o de documentación, los cuales apoyan a uniformizar y consolidar el proceso de desarrollo.

5. Se evita incurrir en costos innecesarios, Como un efecto generalizado de algunos de los puntos mencionados con anterioridad, la práctica de procesos de aseguramiento de calidad lleva a las organizaciones a evitar costos no deseados como pueden ser todos aquellos ocasionados por mantenimiento correctivo.
6. Se planea la calidad, Está claro que el concepto de calidad no es algo que se da de una manera automática e impredeciblemente. Es algo que se busca. Por lo mismo, se debe de planear, construir e implantar en el producto.

6.8 Documentación de Pruebas

En el documento se detallan estándares para Software Testing definidos por ISTQB®:

6.8.1 ISTQB

Es una organización de certificación de la calidad del software que opera internacionalmente, suministra el plan de estudios y el glosario que definen las guías para la acreditación y evaluación de los profesionales del testing.



Ilustración 29: ISTQB.

6.8.2 Definición de Aseguramiento de la Calidad

Es el conjunto de actividades planificadas y sistemáticas aplicadas en un sistema de gestión de la calidad para que los requisitos de calidad de un producto o servicio sean satisfechos.

6.8.3 Definición de testing

El testing es una actividad técnica desarrollada para evaluar la calidad del producto y mejorarlo al identificar defectos y problemas.

6.8.4 Principios del testing

Existen siete principios que establecen pautas comunes a todas las pruebas.

Pruebas con presencia de errores

- Las pruebas solamente demuestran la existencia de defectos en el desarrollo de software

Pruebas exhaustivas son imposibles

- No se puede probar todo
- Analizar los riesgos y determinar prioridades para centralizar el esfuerzo

Pruebas tempranas

- Las actividades de pruebas deben realizarse antes de iniciarse la etapa de entregas

Agrupación de defectos

- Las pruebas deben realizarse de manera proporcional
- La mayor parte de los defectos se concentran en el número reducido de módulos del sistema.

Paradoja de Plaguicidas

- La repetición de las mismas pruebas dejarán de encontrarse más defectos

Pruebas dependen del contexto

- La evaluación tomará aspectos importantes al factor influyente de la prueba

Falacia de ausencia de errores

- La detección y corrección de errores es inútil si el sistema contruido no es utilizado o bien no cumple con las expectativas necesarias.

6.8.5 Fundamentos de testing

A. Disparadores de errores

- Las presiones de tiempos.
- La complejidad de los sistemas.
- Los cambios organizacionales.

B. Errores en especificaciones

- El diseño sobre especificación defectuosa.
- La construcción sobre diseño defectuoso.
- Los sistemas defectuosos.

C. Es necesario comenzar con testing tan pronto como comenzamos el proceso de desarrollo.

D. Las pruebas deben continuar hasta estar seguros de que no habrá fallas graves en el sistema, justo al final del proceso de desarrollo.

6.8.6 Fundamentos de Pruebas

A. Los fallos de software pueden llevar a:

- La pérdida de dinero.
- La pérdida de tiempo.
- La pérdida de reputación de la empresa.

- Provocar lesiones o hasta la muerte.
- La contaminación medio ambiente.

Debido a:

- Se realizan pruebas insuficientes o inexistentes.
- Se aplican tipos de pruebas incorrectos.
- Faltante de más escenarios de prueba.
- Se necesita mejorar los escenarios de prueba.

B. Los sistemas grandes y complejos, son imposibles de probar de manera exhaustiva.

C. Los riesgos

Los riesgos son inherentes a todo desarrollo de software, entre mayor es el riesgo, mayor debe ser el nivel y detalle de pruebas. Además, se debe tomar en cuenta que entre más complejo es el sistema, existe una mayor probabilidad de fallo.

D. La calidad

La definición básica es que el sistema cumpla con las HU de sus usuarios, aunque también existen fallos por incumplimiento de HU no funcionales, como la lentitud o la denegación del servicio por causa de muchas peticiones, el tiempo de respuesta puede ser un factor de calidad.

- Las pruebas no remueven defectos por sí solas.

- La prueba es un componente del aseguramiento de la calidad.

E. Decidir cuándo ya es suficiente:

- Definir cuántas pruebas son suficientes y cuándo dejar de probar, sabemos que no podemos probar todo.
- Todo sistema tiene un riesgo que determinará hasta dónde llegar.
- Es necesario priorizar los posibles escenarios de prueba y asegurar que las pruebas más importantes se realizaron.
- Se deben definir criterios de aceptación por parte de los stakeholders, los stakeholders deben indicar el nivel de tolerancia de defectos.

F. Depuración (Debugging)

Proceso en el cual los desarrolladores pasan para identificar la causa de errores o defectos de código y realizar correcciones. Es esencial la depuración antes de las pruebas para elevar el nivel de calidad del componente o sistema.

6.8.7 Enfoque de técnicas

A. Pruebas estáticas

Pruebas en las que no se evalúa el código, hay que recordar que los fallos a menudo comienzan con un error humano, es decir un error en documentación. Para la evaluación de documentos se aplican

técnicas de revisión, que pueden ser eficaces en la prevención de defectos.

Un ejemplo, es la revisión de documentos por una o más personas con el objetivo de encontrar y remover los errores, esta representa la primera forma de pruebas que se llevan a cabo durante el SDLC.

La buena práctica de probar los documentos de especificaciones vía revisiones temprano en el SDLC, ayuda a identificar defectos antes de ser parte del código ejecutable, lo que hace los defectos más baratos y fáciles de remover, el mismo defecto en la fase de codificación tendría un mayor costo debido al re-trabajo, revisar el código contra estándares de desarrollo puede también prevenir defectos a la fase de ejecución.

B. Pruebas dinámicas

Implican la ejecución del software de un componente o el sistema, permite el uso de diferentes técnicas como lo son las pruebas de caja negra y caja blanca.

6.8.8 Técnica de pruebas

A. Técnicas de Cajas Blancas (White-Box Testing)

Técnicas basadas en la estructura, pruebas de software realizadas sobre las funciones internas de un módulo.

EL objetivo principal es verificar que las líneas específicas de código funcionan tal como están definidas.

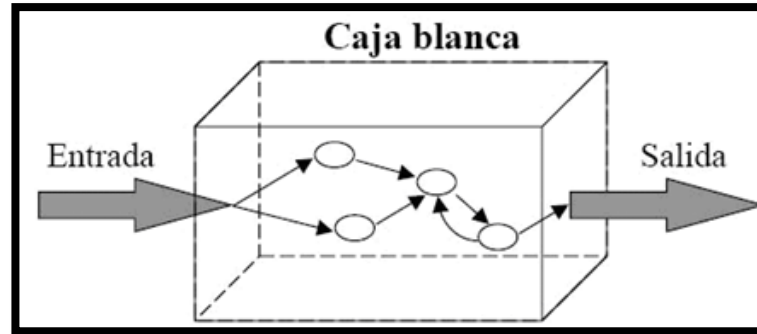


Ilustración 30: Pruebas de Caja Blanca.

Las principales técnicas de diseño de pruebas de caja blanca, se enfocan en asegurar que los elementos particulares de la estructura funcionen correctamente.

a) Pruebas de sentencia y cobertura

Es la evaluación del porcentaje de sentencias ejecutables que han sido practicadas por un juego de caso de pruebas.

Estrategia en la cual el testeo está basado en los caminos internos, estructuras y la implementación del software que estamos testeando.

b) Pruebas de decisión y cobertura

Es un método cuyo objetivo es garantizar que cada una de las posibles ramas de cada punto de decisión se ejecuten y por lo tanto que se ejecute todo el código accesible.

B. Técnicas de caja negra (Black-Box Testing)

Técnicas basadas en especificación, elementos que son estudiados desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su

funcionamiento interno. Estas pruebas verifican que las funciones del sistema retornen los resultados esperados por el cliente según los parámetros de aceptación de las HU.

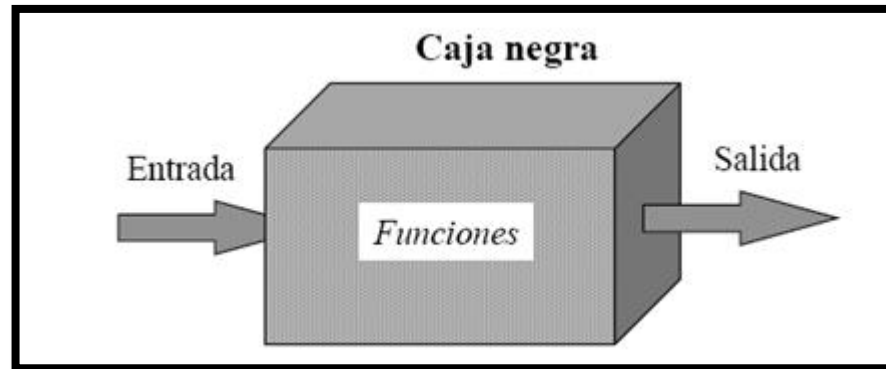


Ilustración 31: Pruebas de Caja Negra.

a) Método de partición de equivalencia

Método de prueba de caja negra que divide el dominio de entrada de un programa en clases de datos, de esta sobresalen los casos de prueba.

Se identifican clases de equivalencia:

- a. Clase válida: genera un valor esperado.
- b. Clase no válida: genera un valor inesperado.

b) Análisis de valores límite

Este método parte del método de clases de equivalencia, la diferencia está en el modo de implementar los casos prueba.

Los casos de prueba son derivados bajo la consideración del espacio de resultados.

La diferencia entre esta técnica y la de particiones de equivalencia, está determinada en la exploración de situaciones

existentes alrededor de los bordes de las particiones de equivalencia.

c) Tablas de decisión

Una decisión es una resolución que se toma entre varias alternativas, es una alternativa seleccionada entre varias que permite alcanzar un estado deseado en respuesta a un problema.

La teoría de decisiones indica que se debe tomar una buena decisión de acuerdo a un determinado problema, se debe establecer primero que se quiere alcanzar para plantearse una serie de alternativas que permitan escoger la más conveniente para solucionar el problema. La decisión es efectiva cuando satisface en la totalidad o al menos en un alto porcentaje, el objetivo o fin deseado y en el momento oportuno en que la decisión debe ser tomada.

La tabla de decisión está integrada por cuatro secciones:



d) Transición de estados

Es una técnica de pruebas de caja negra, en el que las salidas son provocadas por cambios en las condiciones de entrada o

cambios en el "estado" del sistema. En otras palabras, pruebas están diseñadas para ejecutar las transiciones de estado válidas y no válidas.

Derivación de casos de prueba:

Comprender los estados, la transición.

Definir una secuencia de evento que conduce un estado válido o inválido

Se debe señalar de arriba hacia abajo cada estado de transición visitado y recorrido

Repetir los 2 pasos anteriores hasta que todos los estados sean visitados y las transiciones atravesadas

Los casos de prueba con buena cobertura son aquellos cuyos valores de entrada y salida son reales.

C. Pruebas de caja gris (Gray box testing)

Es una combinación de pruebas de caja blanca y pruebas de caja negra, con el objetivo de buscar los defectos debido a una estructura incorrecta o el uso incorrecto de aplicaciones.

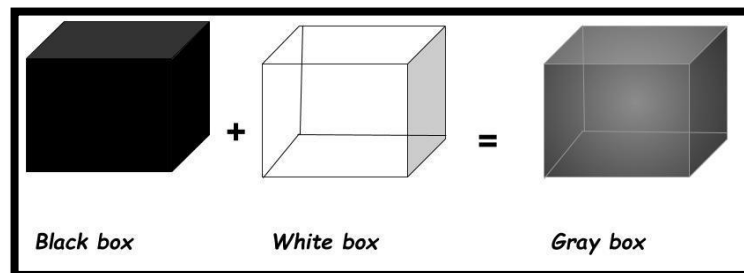


Ilustración 32: Pruebas de Caja Gris.

a) Técnicas basadas en experiencia

Son las técnicas utilizadas cuando los casos de pruebas son especificados en una especificación de HU “no adecuada”, o no hay tiempo para ejecutar una estructura completa de pruebas.

Se utiliza la experiencia de usuarios y testers para determinar las áreas más importantes de un sistema y ejecutarlas para que sean consistentes con la expectativa de uso y los posibles lugares donde suele fallar.

Dado el amplio rango de pruebas desde un enfoque simplista o “ad hoc testing”, prediciendo errores o mediante pruebas exploratorias, con apoyo del conocimiento y la experiencia del tester es mejor para una exploración sistemática contra una especificación.

6.8.9 Técnicas de diseño de pruebas

El diseño de pruebas consta de 3 pasos principales:

A. Identificar condiciones de prueba:

- Ítem o evento de un componente o sistema que pueda ser verificado por uno o más casos de prueba (función, transacción, calidad).
- Características del software que pueden ser chequeados con una prueba o conjunto de pruebas.

B. Identificar casos de prueba:

Conjunto de valores de entrada, ejecución de precondiciones, resultados esperados y ejecución de post-condiciones, desarrollado para un objetivo particular o condición de prueba.

C. Identificar procedimientos de prueba:

- Una secuencia de acciones para la ejecución de una prueba.
- También llamados scripts de pruebas manuales o automáticos.

6.8.10 Estimación de Pruebas**A. Enfoque basado en métricas.**

Datos recolectados de proyectos previos o similares.

Los datos que se pueden incluir:

Nº condiciones de prueba

Nº casos de prueba escritos

Nº casos de prueba ejecutados

Tiempo en el desarrollo de casos de prueba

Tiempo de ejecución de casos de prueba

Nº defectos encontrados

B. El enfoque basado en expertos

Se basa fundamentalmente en que varios expertos, tras crear estimaciones individuales, se reúnen para ponerse de acuerdo en una estimación. (Enfoque Wide Band Delphi).

Los expertos pueden ser:

- Expertos de negocio
- Consultores del proceso de pruebas
- Desarrolladores

- Arquitectos técnicos
- Analistas y diseñadores
- Cualquiera con conocimiento de la aplicación a ser probada

Las vías para usar el enfoque basado en expertos:

- Distribuir la especificación de las HU a los dueños de las tareas para que ellos estimen sus tareas individualmente. Consolidar estimaciones, construir estimación final tomando en cuenta contingencias.
- Distribuir a los expertos conocidos para que vean su punto de vista individual de toda la estimación para luego reunirlos y acordar o debatir la estimación.

C. Variantes de estimación.

Muchas cosas afectan el nivel de esfuerzo requerido para cumplir las HU de pruebas del proyecto.

Estas se pueden dividir en 3 categorías principales:

- **Las características del producto.**
 - Tamaño de la base de pruebas.
 - Complejidad del producto final.
 - La cantidad de HU no funcionales.
 - HU de seguridad según la normativa BS 7799.
 - Cuánta documentación es requerida según las regulaciones.

- Disponibilidad y calidad de las bases de pruebas, según las HU y especificaciones.
- **Las características del proceso de desarrollo.**
 - Escalas de tiempo.
 - Cantidad de presupuesto disponible.
 - Competencias.
 - Herramientas.
- **Las expectativas de las salidas de las pruebas.**
 - La cantidad de errores.
 - La cantidad de casos de prueba a escribir.

6.8.11 Niveles de pruebas



Ilustración 33: Niveles de Pruebas.

A. Pruebas de componente

Conocidas como pruebas de unidad, módulo o programa su principal objetivo es localizar los defectos y comprobar el

funcionamiento de módulos de software, programas, objetivos, clases, etc.

B. Pruebas de integración

Se ocupan de probar las interfaces entre los componentes, las interacciones con distintas partes de un mismo sistema, como el sistema operativo, el sistema de archivos y el hardware, las interfaces entre varios sistemas.

Existen dos niveles de pruebas de integración:

- Pruebas de integración de componentes.
- Pruebas de integración de sistema.

C. Pruebas de sistema

Se refieren al comportamiento del sistema o producto, el alcance de las pruebas dónde debe estar claramente indicado en el plan maestro de pruebas.

D. Pruebas de aceptación

Son a menudo la responsabilidad de los clientes o usuarios de un sistema. El objetivo es crear confianza en el sistema, el objetivo principal no es buscar defectos si no evaluar la buena disposición del sistema para su despliegue y uso, a pesar de no constituir necesariamente el ultimo nivel de pruebas.

E. Pruebas de aceptación de usuario

Verifica el uso del sistema por parte de usuarios comerciales.

F. Pruebas operativas de aceptación

La aceptación del sistema por parte de los administradores del sistema.

Pruebas de restauración

Recuperación de desastres

Gestión de usuarios

Tareas de mantenimiento

Cargas de datos y pruebas de migración

Comprobaciones de vulnerabilidad del sistema

G. Pruebas de Aceptación Contractual y Normativa

Mantiene como base los criterios de aceptación para elaborar un software de desarrollo a la medida. Los criterios de aceptación se presentan en el momento de establecerse el contrato.

H. Pruebas Alfa y Beta

Los desarrolladores de software de mercado, a menudo quiere obtener feedback de clientes potenciales o inexistentes en su mercado antes de poner a disposición un producto de software

- Las pruebas alfa se llevan a cabo en el emplazamiento de la organización de desarrollo.
- Las pruebas beta o de campo, las realizan los clientes en sus propias instalaciones.

6.8.12 Tipos de pruebas

A. Pruebas Manuales

El área de aseguramiento aplicaría validaciones para revisar cada uno de los factores de calidad que conforman cada prueba.

Tipos de pruebas manuales:

B. Pruebas Funcionales

Las pruebas tienen por objetivo probar que los sistemas desarrollados cumplan con las funciones específicas para los cuales han sido creados, se basa en el análisis de los datos de entrada y de salida.

a. Pruebas de Validación

Se validan todas las HU que solicita el usuario para la funcionalidad del sistema, los cuales son agregados en la documentación de cada HU. Responde a la pregunta ¿Estamos construyendo el producto correcto?

b. Pruebas de Funcionalidad

Es verificar que el producto funcione como se espera, son pruebas específicas, concretas y exhaustivas para probar y validar que el software hace lo que se solicita en las historias de usuario.

c. Pruebas de Integridad de Datos

Se validan procesos de almacenamiento de la información, asegurar que los métodos de acceso y procesos funcionan adecuadamente y sin ocasionar corrupción de datos.

d. Pruebas de Interfaz de Usuario

Estas pruebas tienen como objetivo probar la usabilidad de las pantallas, se verifica que la interface cumpla con los estándares solicitados, se verificará que la apariencia del sistema sea amigable para la vista del usuario, además se verificarán alineaciones de formularios, tamaños de controles, posicionamiento de los mismos entre otros aspectos.

- Prueba de estilo (look & feel) y presentación de la información.

e. Pruebas de Verificación

Se realizará una constante evaluación de los bugs reportados para verificar si fueron solucionados los errores.

Responde a la pregunta ¿Estamos construyendo el producto correctamente?

C. Pruebas no Funcionales

Pruebas que hacen referencia a cómo funciona el sistema, toman en cuenta comportamientos externos que tiene el sistema.

a. Pruebas de usabilidad

Métodos de inspección donde expertos usan diferentes técnicas para evaluar una interfaz de usuario sin involucrar a usuarios reales, se aplican planillas de patrones para evaluar características detalladas en la documentación de diseño de interfaz.

b. Pruebas de compatibilidad

El software debe ser compatibles con los exploradores y ambiente de configuración que fueron definidos durante la creación del proyecto.

D. Pruebas de Sistema

- Probar la funcionalidad desde una perspectiva “end-to-end”
 - La prueba de end-to-end es una metodología que se usa para probar si el flujo de una aplicación funciona como se diseñó de principio a fin. El objetivo de realizar pruebas de extremo a extremo es identificar las dependencias del sistema y garantizar que la información correcta se transmita entre los distintos componentes y sistemas.
- Enfocadas en el comportamiento del sistema en un ambiente representativo del ambiente de producción.
- Se prueban HU funcionales y no funcionales.

Pueden incluir especificaciones de HU de sistema y de software, casos de uso, reportes de análisis de riesgos, manuales de sistema, usuario y operación.

E. Pruebas automáticas

Uso de herramientas software para controlar la ejecución de pruebas, la comparación de resultados reales y esperados, el establecimiento de precondiciones de pruebas y otras funciones de control de pruebas,

estas pruebas aplicarán para desarrollos Web que permitan la aplicación de este tipo de pruebas.

Herramienta de captura o reproducción: tipo de herramienta de ejecución de pruebas donde las entradas son grabadas durante las pruebas manuales con el objetivo de generar scripts de pruebas automatizados que puedan ser ejecutados con posterioridad. Estas herramientas son utilizadas a menudo para apoyar pruebas de regresión automatizadas.

a. Pruebas automáticas funcionales

Las pruebas tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, se basa en el análisis de los datos de entrada y de salida.

b. Pruebas automáticas de regresión

Pruebas de un programa previamente probado que ha sufrido modificaciones, para asegurarse que no se han introducido defectos como resultado de los cambios realizados, se realiza cuando el software o su entorno han sido modificados.

c. Pruebas automáticas no funcionales

Pruebas que hacen referencia a cómo funciona el sistema, toman en cuenta comportamientos externos que tiene el sistema.

F. Pruebas de rendimiento

Se aplicarán a desarrollos Web que permita la realización de ese tipo de pruebas, las mismas se realizarán en un ambiente de pruebas ajustado al ambiente de producción y se tomarán como valores de medición los parámetros que se asignen al proyecto, desde la perspectiva de determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo.

a. Pruebas de carga

Permite observar el comportamiento de una aplicación bajo una cantidad de ejecuciones esperada, esta carga puede estar conformada por el número esperado de usuarios concurrentes que utilizará la aplicación y que realizan un número determinado de transacciones durante el tiempo que dura la carga.

b. Pruebas de estrés

Es utilizada normalmente para “botar” la aplicación, se debe de ir incrementando la cantidad de usuarios a la aplicación hasta “botar” el proceso. Este tipo de prueba se realiza para determinar la calidad con que fue realizada la aplicación

G. Pruebas unitarias

Método de comprobar el correcto funcionamiento de una unidad de código.

Algunas características son:

- Se prueba que el código construido funciona para lo que se creó, sin tomar en cuenta integraciones con otros sistemas. Se deben probar todos los componentes de manera separada.
- Se debe verificar que todo el código escrito sea utilizado realizando un testing basado en el código, se debe de tomar en cuenta que una unidad de prueba puede incluir HU, el diseño detallado y el código.
- Se pueden realizar prueba para módulos de base de datos.
- Se aplica para tipos de desarrollo TDD:
 - Primero escribo el caso de prueba
 - Luego construyo el código, pruebo, cambio hasta que pase la prueba.
 - Son ejecutadas por el desarrollador.
- Los defectos encontrados en esta fase no son necesarios documentar, deben ser solucionados para aprobar la prueba.

H. Pruebas exploratorias

Es una técnica que combina la experiencia de los testers con un enfoque estructurado de pruebas donde las especificaciones están incompletas o inadecuadas y hay presiones fuertes de tiempo.

Se utiliza la concurrencia de diseño de pruebas, ejecución de pruebas, registro de pruebas y aprendizaje contenido todo en objetivos de pruebas.

Se maximiza la cantidad de pruebas que se pueden llevar a cabo en un límite de tiempo usando objetivos de prueba para mantener el foco en las áreas más importantes.

a. Pruebas de humo (Smoke testing o ad hoc)

Pruebas Intuitivas que validan los diversos factores de calidad a tomar en cuenta, de una manera rápida y realizando validaciones exploratorias en la aplicación.

I. Pruebas de integración

Identificar errores introducidos por la combinación de programas probados unitariamente, además de la integración de las componentes, involucra los módulos relacionados probando el sistema como conjunto.

- Integrar los componentes previamente probados.
- Mostrar los defectos de las interfaces e interacciones entre componentes o sistemas integrados.
- Se requiere una estrategia de integración antes de probar.
- La prueba puede incluir:
 - Diseño del software o sistema.
 - Diagrama de arquitectura del sistema.
 - Flujos de trabajo.
 - Casos de uso detallados.

a. Estrategias de integración

i. Integración big-bang

- Todos los componentes son integrados al mismo tiempo, resultando de un sistema completo.
- Es difícil aislar los errores encontrados ya que no se presta atención a la verificación de las interfaces entre componentes.
- Introduce riesgos y problemas que pueden ser descubiertos tarde en el proyecto cuando son muy onerosas de corregir.

ReIntegración top-down

Se prueban primero los componentes de alto nivel y luego se avanza hacia los de menor nivel.

ii. Integración bottom up

Se prueban primero los componentes de bajo nivel y luego se avanza hacia los de mayor nivel.

J. Pruebas de Regresión.

Son pruebas reiterativas de un programa ya aprobado, después de haber sido modificado, con el objetivo de detectar defectos surgidos o no descubiertos como consecuencia del cambio; se les denominan pruebas de confirmación ya que se deben de volver a probar los componentes tras haber sido modificados.

K. Pruebas de Estructura o Arquitectura de Software

Pueden realizarse en todos los niveles de prueba, las técnicas estructurales son las idóneas después de las técnicas basadas en la especificación, para ayudar a medir la exhaustividad de las pruebas mediante una evaluación de la cobertura de un tipo de estructura.

La cobertura es la medida en que un set de pruebas cubre un porcentaje de los elementos del software.

L. Pruebas de seguridad y control de acceso

a. Nivel de seguridad de la aplicación

b. Nivel de seguridad del sistema

M. Pruebas de Aceptación de Usuario (UAT)

Pruebas realizadas por el cliente final, con acompañamiento del analista de calidad. Son pruebas funcionales sobre el sistema completo y buscan una cobertura de la especificación de las historias de usuario que cumplan con los criterios de aceptación, estas no se llevan a cabo hasta no haber superado las pruebas internas de aseguramiento

Algunos aspectos para evaluar son:

- Proveer al usuario final un sistema que funciona de acuerdo a sus expectativas.
- La base de las pruebas de aceptación es la especificación de las historias de usuario.
- Se deben evaluar que el sistema está listo para implantar y usar.

- En estas pruebas los representantes del usuario chequean que el sistema cumple sus necesidades de negocio, deben ser realizadas antes de enviar al ambiente de producción.

a. Pruebas de aceptación regulatorias y contractuales

- Se valida que el software cumpla con un contrato establecido.
- Se valida que cumpla con las regulaciones de gobierno, legales y de estándares de seguridad.

b. Pruebas de aceptación operativas

Está listo el sistema para entrar en producción y operarlo.

Las pruebas pueden incluir:

- Detalles de respaldos.
- Procedimientos para recuperación de desastres.
- Entrenamiento a usuarios finales.
- Procedimientos de mantenimiento.
- Carga de datos y tareas de migración.
- Procedimientos de seguridad.

c. Pruebas basadas en el riesgo

Pruebas con el objetivo de reducir el nivel de riesgos de producto e informar a los afectados de su estado, comenzando desde las fases iniciales de un proyecto durante el proceso de planificación, identificar

que riesgos pueden afectar el proyecto. Están enfocadas a riesgos del proyecto y deben ser evaluadas por el product owner, se analiza el listado de funcionalidades a probar, se extrae con base a un análisis de riesgos realizado de manera previa, que tienen en cuenta variables tales como el impacto que podría ocasionar la falla o la probabilidad de falla de una funcionalidad.

6.9 Proceso de Aseguramiento

A continuación, se define un proceso específico basado en la recomendación de ISTQB para lograr gestionar un sprint determinado con el fin de poder realizar un proceso de aseguramiento de calidad ágil dentro de los sistemas de las empresas nacionales productoras de software.

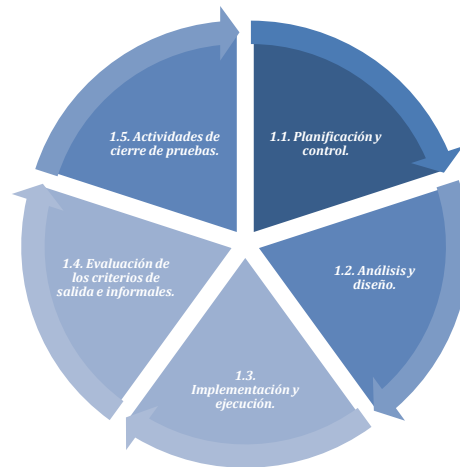


Ilustración 34: Proceso Básico de Aseguramiento.

6.9.1 Procedimiento de aseguramiento de software

Se detalla un procedimiento adecuado a gestionar el rol que tiene un asegurador de calidad dentro de una metodología ágil Scrum, para realizar una correcta gestión del especialista en calidad del software, durante los procesos:

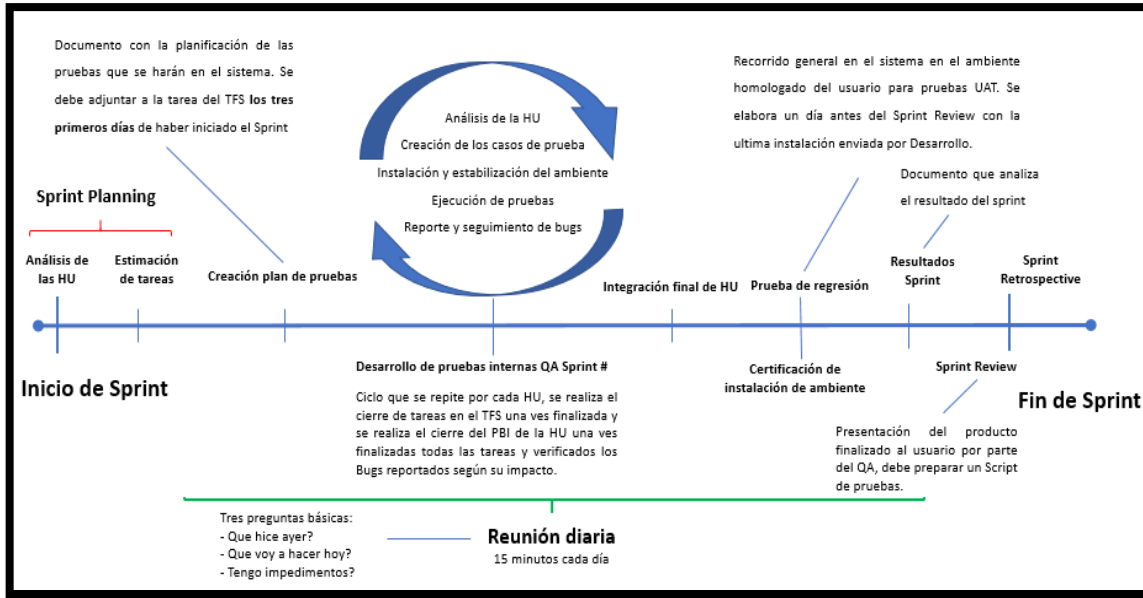


Ilustración 35: Tareas Básicas de un asegurador de calidad dentro de SCRUM.

Las fases a seguir que estructuran la definición del proceso de aseguramiento según SCRUM:

- A. El proceso inicia con la reunión de sprint planning, donde se realiza el análisis de las historias de usuario y estimación de tareas.
 - Primera parte de la reunión, se realiza en un time box de 4 horas como máximo.

El product owner presenta al equipo la lista de requisitos priorizada del producto o proyecto, pone nombre a la meta del sprint y propone las historias de usuario más importantes a desarrollar, con una prioridad definida por el usuario.

El Scrum Team realiza la votación para definir el esfuerzo necesario para lograr el objetivo de cada HU, la votación se realiza mediante el planning poker; técnica para calcular una

estimación basada en el consenso, en su mayoría utilizada para estimar el esfuerzo o el tamaño relativo de las tareas de desarrollo de software.

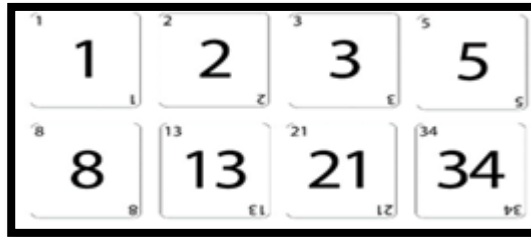


Ilustración 36: Scrum Póker.

- Segunda parte de la reunión, se realiza en un time box de 4 horas como máximo.

El equipo planifica el sprint, elabora la táctica que le permitirá conseguir el mejor resultado posible con el mínimo esfuerzo, esta actividad la realiza el equipo dado que ha adquirido un compromiso y es el responsable de organizar su trabajo.

El trabajo se compone de:

- El Scrum Master asigna la historia de usuario a cada uno de los miembros del Scrum Team.
- El Scrum Team define las tareas necesarias para poder completar cada objetivo, creando la lista de tareas para el desarrollo y aseguramiento de cada la historia de usuario agregada al sprint.
- El Scrum Team realiza una estimación del tiempo necesario para realizar cada tarea.

- Cada miembro del equipo se auto asigna a las tareas que puede realizar.
- B.** Una vez finalizado el sprint planning inicial la etapa de planificación y control donde se genera el plan de pruebas a ejecutar para la revisión de cada historia de usuario durante el sprint.
- C.** El proceso de desarrollo de pruebas internas de aseguramiento de calidad, en esta fase se itera:
- Etapa de análisis y diseño de pruebas.
Se analiza y diseña los casos de prueba para cada HU, se definen y agregan los scripts de datos a los casos de prueba.
 - Etapa de implementación y ejecución de pruebas.
Se ejecutan los casos de prueba desarrollados para cada HU, según se realicen las liberaciones de software por parte de los miembros del equipo encargado del desarrollo.
 - Etapa de evaluación de los criterios de salida e informales.
El reporte de bug se realiza de manera paralela a la etapa de implementación y ejecución, las evaluaciones se realizan con base a la comparación de los criterios de salida y con los resultados esperados.
 - Etapa de validación
Se validan los bugs reparados y se analizan los criterios definidos para los bugs rechazados, además, contempla las pruebas de regresión que se realizan a lo largo de todo el

desarrollo de pruebas y las pruebas de regresión para certificación de ambiente.

- Integración de historias de usuario.

Se realizan pruebas de integración del conjunto de todas las historias de usuario que se trabajaron durante el sprint.

D. Actividad de cierre de pruebas.

- Etapa control y seguimiento del proceso de pruebas.

Se crea la documentación de análisis de resultados del sprint.

- Reunión de sprint review.

Reunión donde el equipo de aseguramiento presenta al usuario las historias de usuario completadas en el sprint, en forma de incremento de producto preparado para ser entregado, haciendo un recorrido por ellos lo más real y cercano posible al objetivo que se pretende cubrir.

- Reunión de sprint retrospective.

Con el objetivo de mejorar de manera continua su productividad y la calidad del producto que está desarrollando, el equipo analiza cómo ha sido su manera de trabajar durante el sprint, el por qué está consiguiendo o no los objetivos a que se comprometió al inicio del sprint y el por qué el incremento de producto que acaba de demostrar al cliente era lo que él esperaba o no:

- Qué cosas han funcionado bien.

- Cuales hay que mejorar.
- Qué cosas quiere probar hacer en el siguiente sprint para mejorar.
- Qué ha aprendido.
- Cuáles son los problemas que podrían impedirle progresar adecuadamente, el facilitador se encargará de ir eliminando los obstáculos identificados que el propio equipo no pueda resolver por sí mismo.

E. Etapa de pruebas de UAT y el acompañamiento por parte del asegurador de calidad.

El acompañamiento a usuarios tiene un time box de 3 a 5 días el cual contempla tiempos reducidos de manera parcial donde se le realiza un acompañamiento y aclaración de dudas al usuario para que él pueda proceder normalmente con sus pruebas UAT, además el asegurador en Visual Studio Team Services realiza la gestión de reporte de bugs realizado por el usuario encargado de la prueba, hasta que se genere aprobación de las historias de usuario entregadas al usuario para la evaluación.

F. La última fase es el pase a producción.

- Preparación de insumos para el pase a producción.
- Solicitud y calendarización del pase a producción.
- Acompañamiento durante el pase a producción, para mitigar problemas que se presente durante el proceso de instalación.

6.9.2 Enfoques y Estrategias de Pruebas

Un enfoque de pruebas es la implementación de una estrategia de pruebas en un proyecto particular, define cómo será implementado el testing. Generando un enfoque de pruebas que refleja el testing de toda una organización, un programa o un proyecto.

El enfoque puede ser:

A. Preventivo

Debe ser desarrollado temprano en el ciclo de vida: en este enfoque el proceso de diseño de pruebas es iniciado tan pronto como sea posible en el ciclo de vida para detener defectos durante la construcción hasta la solución final, trabajo bajo enfoque de metodologías ágiles.

B. Reactivo

Desarrollar antes de iniciar la ejecución de las pruebas, es donde el testing es la última etapa del desarrollo del software y no es iniciada hasta después de que el diseño y codificación hayan sido completados, trabajo bajo un enfoque cascada.

6.9.3 Planificación de Pruebas

La planificación de pruebas es la actividad que define los objetivos de las pruebas y especialización de las actividades de las pruebas.

El control de pruebas es una actividad constante de comparar el proceso real con el plan previsto e informar sobre el estado de las pruebas, incluyendo la existencia de desviaciones con respecto a lo que se había planificado. Implica la adopción de las medidas necesarias para cumplir con la misión y los objetivos

del proyecto. La planificación de pruebas tiene como feedback las actividades de control y seguimiento, en el planteamiento de la estrategia a seguir aplicando los criterios de la guía de aseguramiento de ISTQB® y herramientas de ALM de Microsoft.

El proceso cuenta con estándares de documentación para el plan de pruebas a ejecutar en el sprint, donde se evalúan las historias de usuario que se desarrollen en el mismo, esto con el fin de tener definido el plan de pruebas y datos importantes del ambiente; tomando en cuenta los criterios de aseguramiento para los diferentes escenarios que se pueden presentar.

Aspectos importantes del planeamiento:

- a. Definir el plan de pruebas dentro del TP en la herramienta Visual Studio Team Services para la planeación y ejecución del aseguramiento.
- b. Dentro del plan de pruebas se crean las suites correspondientes a cada historia de usuario o FastTrack, en los cuales se agregan las suites correspondientes a cada tipo de prueba donde alojarán los casos de pruebas correspondientes según la clasificación.
 - Interfaz Gráfica - IG
 - Funcional - FUN
 - Integración - INT
 - Regresión – REG

6.9.4 Control, Gestión y Seguimiento del Proceso de Prueba

El control del proceso de pruebas aplicados mediante los criterios de Scrum, ISTQB® y con el apoyo de las herramientas de ALM, se agregan estándares de documentación de resultados que muestran la evaluación del resultado del sprint y muestra el resultado de la gestión y seguimiento del proceso de pruebas, apegado a la ejecución del sprint y evaluando las historias de usuario desarrolladas, esto con el fin de tener documentos que respalden el resultado de las pruebas.

Otras actividades de cierre del proceso dentro del sprint se generan las reuniones de finalización del Sprint:

- Reunión de sprint review.
- Reunión de retrospective.

Control de pruebas

Usa la información anterior para decidir el curso de acción para asegurar el control de las actividades de pruebas

Las acciones podrían impactar actividades de pruebas u otras actividades del ciclo de vida.

- Actividades de control de pruebas:
 - Tomar decisiones basada en la información del monitoreo del progreso de las pruebas.
 - Re-priorizar pruebas basado en riesgos del proyecto.
 - Cambiar calendario de pruebas debido a la disponibilidad del ambiente de pruebas.

- Definir un nuevo criterio de entrada o criterio de salida esperado.
- Revisar los riesgos.
- Ajustar el alcance de las pruebas.
- Posibles recomendaciones del líder de pruebas, al líder de proyecto:
 - Cambiar alcances.
 - Retrasar reléase.
 - Continuar probando en producción.

6.9.5 Control de estados de la herramienta Visual Studio Team Services

A. Estados de las Épicas.

- NEW: Estado inicial al crear la épica.
- IN PROGRESS: Trabajando sobre la épica.
- DONE: Trabajo de la épica finalizado.
- REMOVED: Épica desestimada.

B. Estados de los Product Backlog Ítem.

- NEW: Estado inicial al crear el PBI.
- APPROVED: Aprobado el PBI.
- COMMITTED: Trabajando sobre el PBI.
- DONE: Trabajo del PBI finalizado.
- REMOVED: PBI desestimado.
- DONE APLICACIONES: Trabajo del PBI finalizado por aplicaciones.
- REJECTED: PBI rechazado.
- EN PRUEBAS UAT: PBI en pruebas de UAT.

C. Estados del registro de tareas en los PBI de las HU, fastrack y bugs.

- TO DO: Estado inicial al crear la tarea.
- IN PROGRESS: Trabajando en la tarea.
- DONE: Tarea finalizada.
- REMOVED: Desestimado.

D. Estados de los Bug

- NEW: Estado inicial al crear un bug.
- APPROVED: Aprobado por el área de aseguramiento de aplicaciones.
- REJECTED: No corresponde a un defecto, duplicado, desestimado.
- COMMITTED: Se está trabajando sobre el defecto.
- FIXED: El defecto esta corregido en una liberación superior.
- DONE: Cierre del bug, error reparado.
- REOPEN: El error no se corrigió, se agrega al historial del bug, una descripción de la reapertura.
- REMOVED: Desestimado.
- ACEPTACION UAT: El bug está siendo revisado por usuarios finales.

E. Estados de los FastTrack.

- NEW: Estado inicial al crear un FastTrack.
- APPROVED: Aprobado para trabajar el FastTrack.

- REJECTED: FastTrack duplicado, desestimado.
- COMMITTED: Se está trabajando en el FastTrack.
- DONE: Cierre del FastTrack.
- REMOVED: FastTrack desestimado.

6.9.6 Definición y Aplicación de Métricas

Una métrica de calidad es una definición operativa que describe un atributo del producto o del proyecto en valor real, indicando la manera en que el proceso de control de calidad medirá el trabajo o el producto. La tolerancia define la variación permisible de las métricas.

Un ejemplo de una métrica de calidad en el proyecto puede ser el tiempo de respuesta de un sistema informático para elaborar un reporte de datos específico.

Clasificación de las métricas de software:

A. Métricas de proceso

Se recopilan durante un largo periodo de tiempo en diversos proyectos

Son caracterizadas por:

- Realizar control de la ejecución durante el proceso.
- Realizar una medición de tiempos de las fases.

B. Métricas de proyecto

Permiten evaluar el estado del proyecto y seguir la pista de los riesgos.

C. Métricas de producto

Se centran en las características del software y no en cómo fue producido, también son productos los artefactos, documentos, modelos, y componentes que conforman el software, se miden cosas como el tamaño, la calidad, la totalidad, la volatilidad, y el esfuerzo.

Para los proyectos desarrollados por ingeniería de aplicaciones en empresas nacionales de desarrollo de software, se utiliza como herramienta oficial para la evaluación de métricas en Visual Studio Team Services.

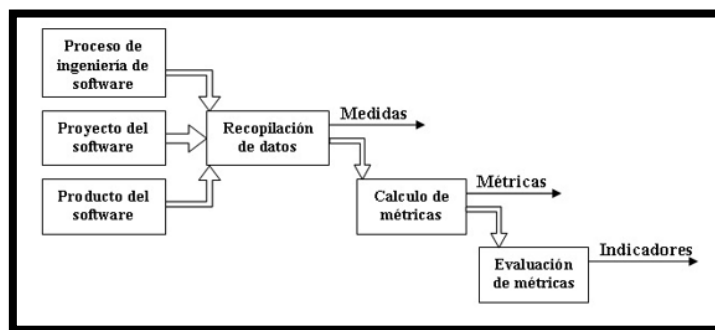


Ilustración 37: Proceso de Recopilación de Métricas de Software.

6.9.7 Análisis y diseño de prueba

Es la actividad en la cual los objetivos de las pruebas generales se transforman en condiciones de pruebas y casos de pruebas tangibles.

Se realiza el análisis correspondiente de las historias de usuario a evaluar dentro del sprint para lograr definir cada uno de los aspectos que serán evaluados dentro del caso de prueba y con ello poder identificar los diferentes escenarios que se pueden evaluar para crear los sets de datos que serán incluidos a cada caso de prueba.

Se debe realizar las pruebas definidas en el plan de pruebas al inicio del sprint, para iniciar la evaluación de cada historia de usuario que se incluyen en el sprint y preparar el diseño de las pruebas; asignando la prioridad para cada caso de prueba, los casos de pruebas desarrollados serán alojadas en el plan de pruebas en Visual Studio Team Services.

Tareas a realizar:

- Se realiza el diseño de los casos de pruebas, según se definieron en la etapa de planeamiento mediante la utilización de la herramienta Microsoft Test Manager.
- Se agregan los scripts de datos, con los valores a evaluar según el escenario que se quiera ejecutar, tomando en cuenta los valores de entrada y criterios de salida definidos.
- Se analiza y priorizan las pruebas, se define la secuencia de ejecución.
- Diseñar la configuración del entorno de pruebas e identificar cualquier infraestructura y herramientas necesarias.

6.9.8 Implementación y Ejecución de Pruebas

Actividad en la que se especifican los procedimientos o guiones de prueba mediante la combinación de casos de prueba en un orden determinado y la inclusión de cualquier otra información necesaria para la ejecución de pruebas, en esta etapa se configura el ambiente de pruebas y se ejecutan las pruebas.

Tareas a realizar:

- Finalizar, implementar y priorizar los casos de pruebas.
- Verificar que el entorno de pruebas ha sido correctamente configurado.

- Etapa que contempla la ejecución de cada uno de los casos de prueba alojados dentro del plan de pruebas definido en Visual Studio Team Services, cada una de las pruebas serán ejecutadas tomando en cuenta las entregas continuas realizadas por los miembros del equipo de desarrollo, las cuales estarán definidas según los criterios de prioridad indicados por el product owner al inicio o durante la ejecución del sprint.
- La etapa de ejecución contempla evaluar cada una de las iteraciones que fueron definidas dentro del caso de pruebas ejecutados en el sprint y con ellos lograr detectar errores que presentan la herramienta de manera temprana.
- Repetir actividades de pruebas para confirmar la corrección de errores y generar pruebas de regresión que nos ayuden a descartar la agregación de nuevos errores.

A. Evaluación de los Criterios de Salida

Es la actividad que evalúa la ejecución de pruebas contra los objetivos definidos. Esta evaluación debe hacerse para cada nivel de pruebas.

Tareas a ejecutar

- Comprobar los requisitos de pruebas con los criterios de salida previstos en la planificación de las pruebas.
- Evaluar si se requieren más pruebas o si deberían modificarse los criterios de salida especificados.

6.9.9 Gestión de Defectos

La gestión de defectos es el proceso de reconocimiento, investigación, y reporte de defectos, además lo conforma aspectos importantes como la clasificación e identificación del impacto.

A. Etapa de Reporte

Contempla el reporte de los diferentes bugs detectados en la aplicación, el registro se realiza mediante la herramienta Microsoft Test Manager.

B. Etapa de Validación

Etapa que valida los bugs reparados y rechazados por el equipo de desarrollo, para el seguimiento se crean consultas que evalúan el estado del bug.

Características de los incidentes o defecto:

- Un incidente es cualquier evento no planeado que ocurre y requiere una investigación.
- Durante las pruebas es donde el resultado actual es diferente del resultado esperado.
- Al investigar el incidente podría convertirse en defecto o también en un cambio a la especificación.
- Exista un proceso mediante el manejo de estados de Visual Studio Team Services, para dar seguimiento a todos los incidentes. Es importante asegurar que a los incidentes se les dé seguimiento desde que se detecta hasta el cierre.
- Los incidentes se pueden generar a través de todo el ciclo de vida de desarrollo de software.

- El registro de incidentes contempla la clasificación y la identificación del impacto.
- El analista de calidad es el responsable del proceso de gestión de incidentes, asegura la gestión o coordinador de incidentes a lo largo del proceso de aseguramiento.

C. Objetivos de la Gestión de Incidentes

- Proveer a los desarrolladores y terceros retroalimentación en el problema de permitir la identificación, aislamiento y corrección. El defecto debe ser corregido, por lo que el reporte debe llevar información concisa del defecto para entender cómo arreglarlo, entre más información se entregue va a ayudar para la identificación del mismo durante la corrección.
- Proveer a los líderes el seguimiento de la calidad del sistema y el progreso de las pruebas.
- Proveer ideas para mejorar el proceso de pruebas según el incidente detectado para que no vuelva a ocurrir.

D. Datos del Informe de Incidentes

- Datos del hallazgo, autor, aprobador, estado.
- Alcance, severidad y prioridad del incidente.
- Referencias a los casos de prueba que encontraron el problema.
- Resultados esperados versus resultado actual.

- Fecha en que el incidente fue descubierto.
- Identificación del Ítem de pruebas y el ambiente.
- Ciclo de vida del sistema o software en el cual el incidente fue observado.
- Sprint del incidente cuando fue observado.
- Descripción del incidente para permitir reproducción y resolución, incluyendo logs, registros de base de datos o pantallazos.
- Grado de impacto en involucrados interesados.
- Severidad del impacto en el sistema.
- Urgencia o prioridad de reparar.
- Estado del incidente (NEW, APPROVED, REJECTED, COMMITTED, FIXED, DONE, REOPEN, REMOVED, ACEPTACION UAT).
- Hallazgos globales, tales como otras áreas que pueden ser afectados por el cambio que soluciona el incidente.
- Historial de cambios

El control de defectos se administra mediante la herramienta de gestión Visual Studio Team Services y como complemento para la detección y reporte Microsoft Test Manager.

E. Reporte de Bugs

- Se crean los bugs mediante la herramienta MTM.
- En el bug se registran screenshots de los errores, un video para evidenciar del bug, criterios de aceptación, quien realiza el reporte y el criterio de severidad y prioridad del bug.
- El bug se crea vinculado al caso de prueba que se ejecutó, el cual proporciona un link al PBI de la HU.

F. Criterios de Severidad

a) Errores Críticos

- Se clasificarán como errores críticos todo aquello que debe ser solucionado.
- Son los resultados que provoquen la terminación de uno o más componentes del sistema o del sistema en general.
- Corrupción de datos.
- No existe ningún otro método alternativo aceptable para lograr los resultados requeridos.

b) Errores Altos

- Se debe considerar la corrección del mismo.
- Son todos los resultados que provoquen la terminación de uno o más componentes del sistema o del sistema en general.

- Corrupción de datos.
- Pero lo que diferencia un error con severidad alta a uno crítico, es que el de severidad alta si existe un método alternativo aceptable para lograr los resultados requeridos.

c) Errores medios

- Son todos los resultados incorrectos, incompletos o inconsistentes.
- Es el incumplimiento en funciones secundarias.

d) Errores Bajos

- Se clasifican defectos menores o cosméticos.
- Son todos aquellos resultados en la presentación de datos.
- No adecuación de estándares, estos tienen soluciones aceptables para lograr los resultados requeridos.

G. Criterios de prioridad de defectos

Según los requisitos del negocio o del cliente, es que se indica la prioridad.

- A. Prioridad 1:** El producto no puede ser entregado si no ha sido resuelto exitosamente y lo más pronto posible.
- B. Prioridad 2:** El producto no puede ser entregado si no ha sido resuelto exitosamente, pero no debe ser tratado inmediatamente.

C. **Prioridad 3:** La resolución del defecto es opcional en función a los recursos, el tiempo y el riesgo.

H. Manejo de Control de Estados del Bug

- **NEW:** Estado inicial al crear un bug.
- **APPROVED:** Aprobado por el área de aseguramiento de aplicaciones.
- **REJECTED:** No corresponde a un defecto, duplicado, desestimado.
- **COMMITTED:** Se está trabajando sobre el defecto.
- **FIXED:** El defecto esta corregido en una liberación superior.
- **DONE:** Cierre del bug, error reparado.
- **REOPEN:** El error no se corrigió, se agrega al historial una descripción de la reapertura.
- **REMOVED:** Desestimado.
- **ACEPTACION UAT:** El bug fue revisado por usuarios.

6.9.10 Actividades de Cierre de Pruebas

Durante la fase de cierre de pruebas de un proceso se recopilan los datos de aquellas actividades de pruebas finalizadas con el objetivo de consolidar la experiencia. Las actividades de cierre de pruebas tienen lugar al finalizar cada

sprint y realizar el pase a las pruebas UAT y a producción, identifica la finalización de un proceso de pruebas en el sprint.

Tareas de cierre de pruebas:

A. Etapa Control y Seguimiento del Proceso de Pruebas

Elaborar un resumen de las pruebas, se crea documentación del resultado de pruebas la cual se elabora al final de cada sprint con el objetivo de mantener comunicación de los resultados en el sprint a las partes interesadas.

B. Reunión de sprint review.

Reunión donde el equipo de aseguramiento presenta al usuario las historias de usuario completadas en el sprint, en forma de incremento de producto preparado para ser entregado, haciendo un recorrido por ellos lo más real y cercano posible al objetivo que se pretende cubrir.

C. Reunión de Sprint Retrospective.

Define algunas funcionalidades:

Aspectos que funcionaron correctamente

Aspectos a mejorar

Aspectos a tomar en cuenta en el siguiente sprint

Aspectos aprendidos

Aspectos que impiden el progreso

D. Etapa de pruebas de UAT y el Acompañamiento por parte del Asegurador de Calidad.

El acompañamiento a usuarios tiene un time box de 3 a 5 días el cual contempla tiempos parciales, además el asegurador realiza la gestión de reporte de bugs realizado por el usuario encargado de la prueba, hasta que se genere aprobación de las historias de usuario entregadas al usuario para la evaluación.

E. La última Fase es el Pase a Producción.

- Preparación de insumos para el pase a producción.
- Solicitud y calendarización del pase a producción.
- Acompañamiento durante el pase a producción, para mitigar problemas que se presente durante el proceso de instalación.

6.9.11 Instalación en Ambiente de QA y Gestión de Incidentes

El proceso de instalación inicia con la revisión de los insumos en la entrega a aseguramiento, la entrega contempla una carpeta con los insumos y las ordenes de proceso las cuales contienen los pasos a seguir durante la instalación de los diferentes componentes que conforman el ambiente a instalar.

Los ambientes serán distribuidos en:

- Desarrollo -> Ambiente de desarrollo
- QA -> Ambiente de aseguramiento de la calidad
- UAT -> ambiente de pruebas de aceptación de usuario
- Producción -> Ambiente productivo

A. Certificación y Blindaje de Ambiente para Pase a Producción.

Se generan pruebas de regresión para validar que la aplicación funciona adecuadamente en integridad de todos sus módulos y con ello certificar que el ambiente homologado con producción funcione correctamente, con el fin de asegurar que el pase a producción se realice sin inconvenientes, este proceso contempla la revisión de insumos para el pase.

6.9.12 Acompañamiento en Proceso de Pruebas UAT

Una vez finalizado el proceso de pruebas y realizada la presentación durante el Sprint Review a usuarios, se realiza el acompañamiento por parte del analista de calidad, el cual tiene una duración de 3 a 5 días tiempos parciales, esto con el fin de solventar dudas que se generen sobre el uso de la aplicación y servir de apoyo al proceso de pruebas de aceptación de usuarios.

6.9.13 Gestión de Incidentes UAT

Se realiza la gestión del bug reportados por los usuarios que realizan las pruebas, esto con el fin de validar que el error se replique y con ello documentar y reportar el mismo con ayuda de la herramienta MTM. El bug se crea mediante pruebas exploratorias las cuales reportan el paso a paso con el que se desarrolló el Bug, se registran screenshot de los errores, criterios de aceptación, además de documentación adjunta de ser necesaria para lograr una mayor explicación del error.

6.10 Herramientas de Apoyo a Pruebas

Las herramientas oficiales para la gestión de aseguramiento de software y toda la respectiva gestión del proceso de pruebas será Microsoft Visual Studio

Team Services como herramienta gestión y control en el ALM y Microsoft Test Manager como complemento para los procesos de aseguramiento, producto de software que soporta una o más actividades de pruebas, tales como planificación y control, especificación construcción inicial de archivos y datos, ejecución de pruebas y análisis de pruebas, hace el proceso de pruebas más efectivo y eficiente.

Además, para la ejecución de pruebas de Integridad de datos se utilizaría una herramienta que permita la conexión con base de datos, por ejemplo, SQL Server Management Studio, Toad de Oracle , PL/SQL.

Para realizar estabilización de ambiente se utilizarían herramientas como WcfStrom y WebServiceStudio, las cuales se aplicarían para probar servicios web, entre otras herramientas que pueden ser utilizadas en los procesos de calidad de software.

6.10.1 Microsoft Visual Studio Team Services (VSTS)

Herramienta de ALM que permite realizar la gestión y control administrativa de los proyectos informáticos.

6.10.2 Microsoft Test Manager (MTM)

Complemento de herramientas de ALM para la gestión de aseguramiento de software, permite la creación, ejecución y análisis de resultados de los casos de prueba.

6.10.3 PL/SQL

Consola para realizar consultas a bases de datos Oracle.

6.10.4 Toad

Herramienta gráfica para realizar administración de base de datos Oracle.

6.10.5 SQL Server Managment Studio

Herramienta gráfica para realizar administración de bases de datos Microsoft.

6.10.6 WcfStorm

Herramienta capaz de probar WCF y Web Services. Permite a los usuarios crear casos de prueba funcional aplicando casos de prueba que se pueden guardar en un proyecto.

6.10.7 WebServiceStudio

Herramienta que permite gestionar llamados a Web Services, con el fin de realizar pruebas a los mismos.

6.10.8 Guía de uso Microsoft Visual Studio Team Services

A. Ingreso a VSTS

Desde Internet Explorer digite el URL, una vez ingresado al URL debe seleccionar el team project.

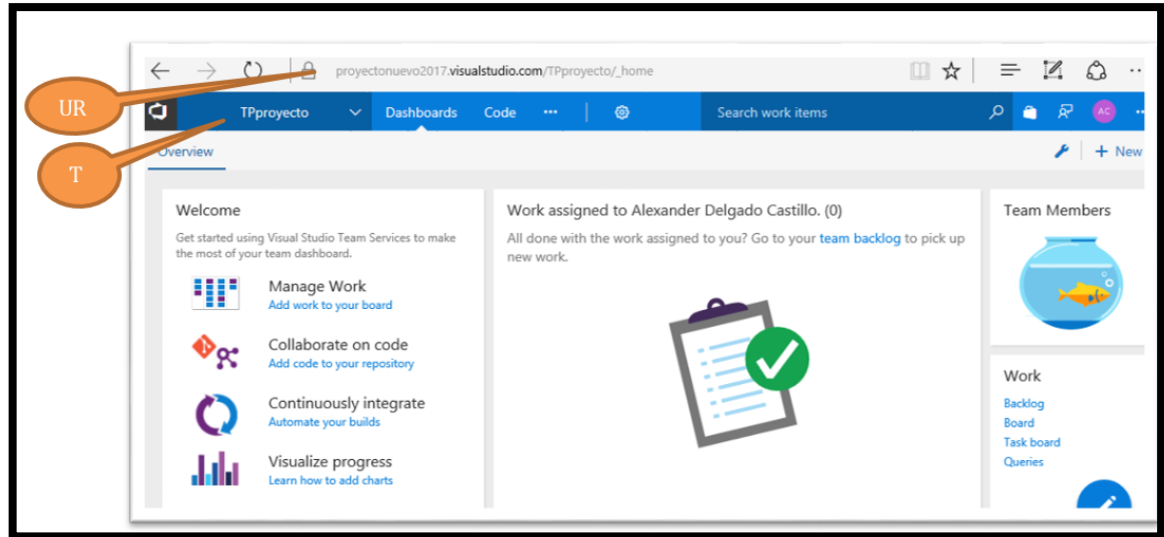


Ilustración 38: Ingreso a VSTS.

B. Panel de Épicas

Desde el panel de épicas, se crean y administran las épicas.

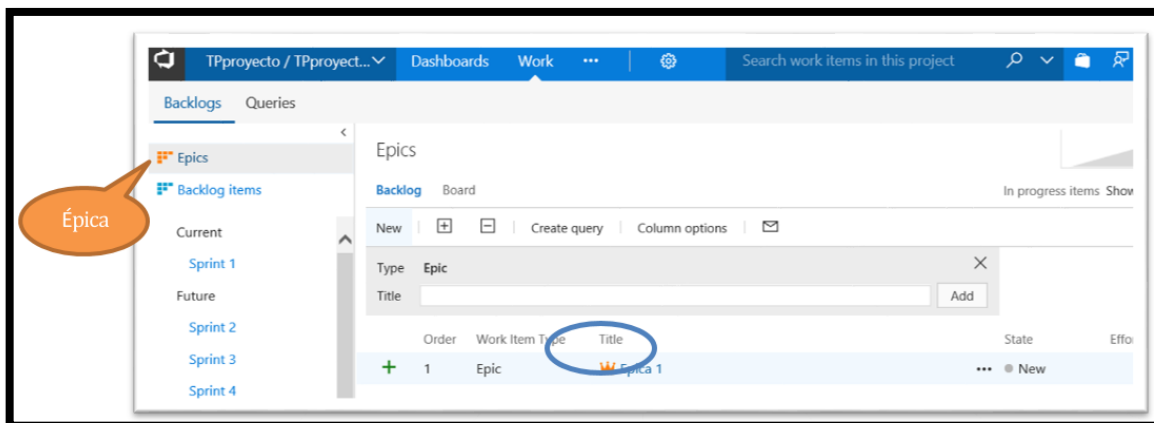


Ilustración 39: Panel de Épicas.

C. Panel de Backlog ítems

El panel de backlog ítems contiene los PBI a trabajar durante el desarrollo del proyecto.

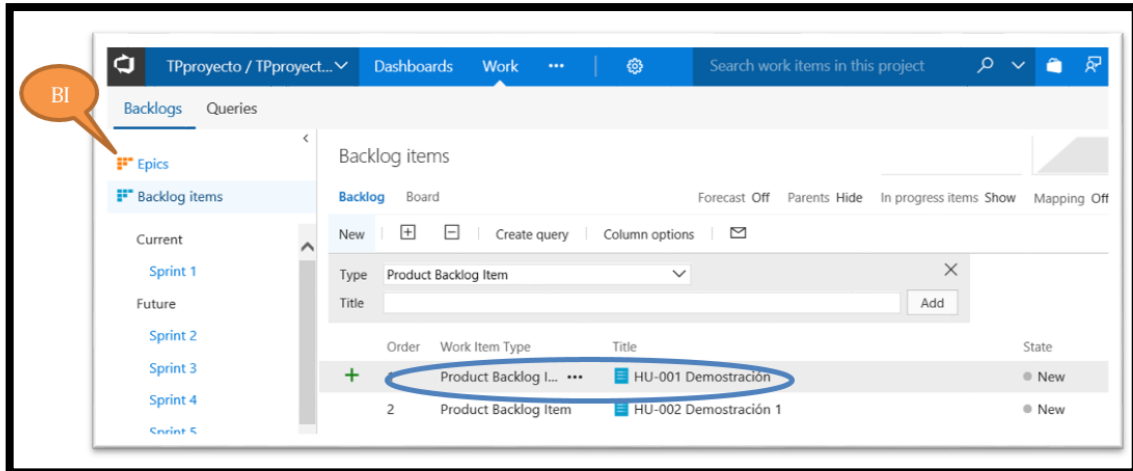


Ilustración 40: Panel de BackLog Ítems.

D. Trabajo en sprint

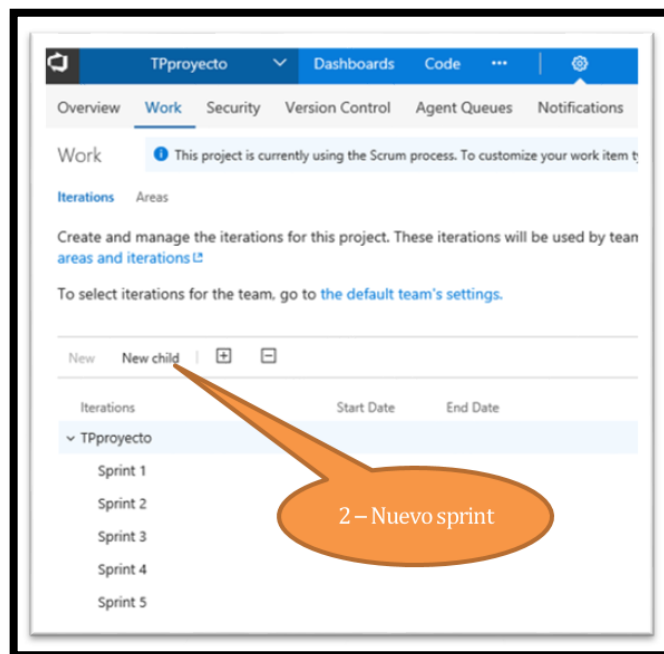


Ilustración 41: Crear Nuevo Sprint.

The image shows a 'New iteration' dialog box with the following fields and values:

- Iteration name:** Sprint 8
- Start date:** 12/20/2017
- End date:** 12/21/2017
- Location:** TPproyecto

Buttons at the bottom: Save and close, Cancel.

Callout labels:

- Nombre (points to Iteration name)
- Fecha (points to Start date)
- Fecha fin (points to End date)
- Guardar (points to Save and close button)

Ilustración 42: Crear Sprint.

a) Administración del Sprint

Seleccione el sprint para visualizar los ítems a trabajar durante el sprint, también puede visualizar el trabajo del equipo y el colaborador con las horas de trabajo que tiene asignadas en el sprint.

Desde el panel puede realizar la administración, permite crear nuevos PBI, crear tareas dentro de los PBI, asignar trabajos, cambiar estados, etc.

Para agregar ítems al sprint debe arrastrar los mismos desde el panel de backlog ítems

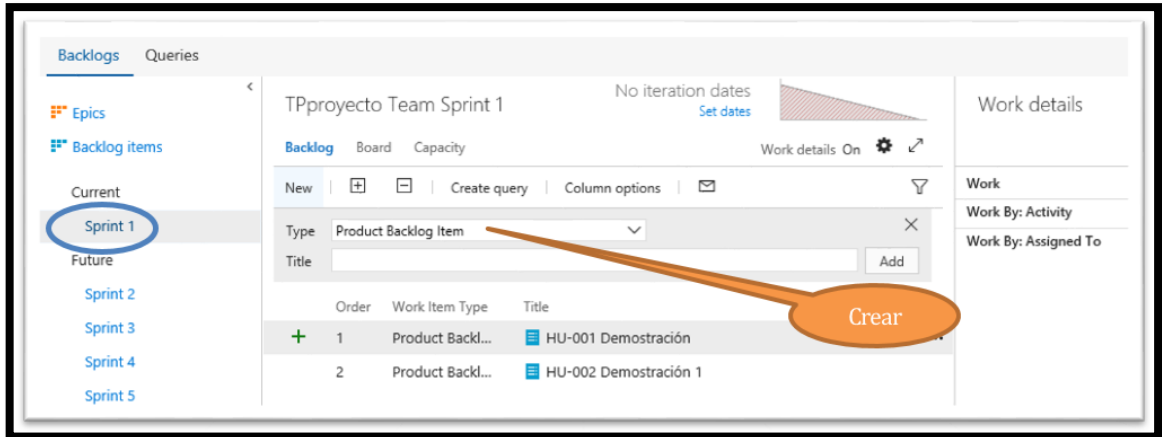


Ilustración 43: Administrador de un sprint.

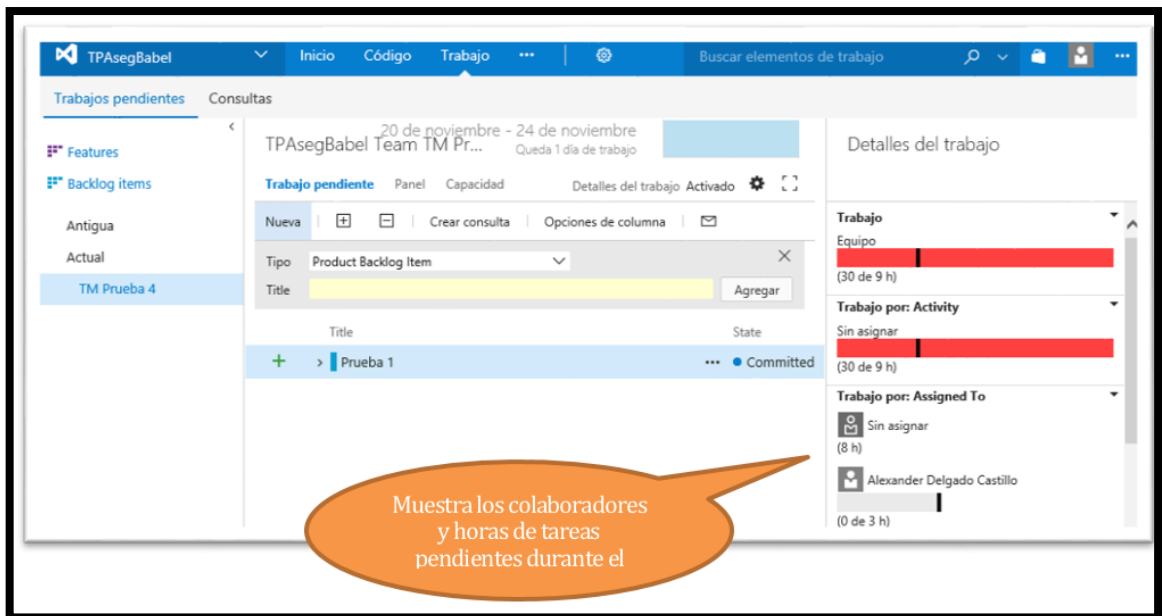


Ilustración 44: Administrador de un Sprint. 2.

E. Visualizaciones de trabajo y configuración

La herramienta cuenta con dos opciones para visualizar los ítems a trabajar:

a) **Cejilla de trabajo pendiente:**

Muestra la lista de Ítems asociados al sprint o área de trabajo.

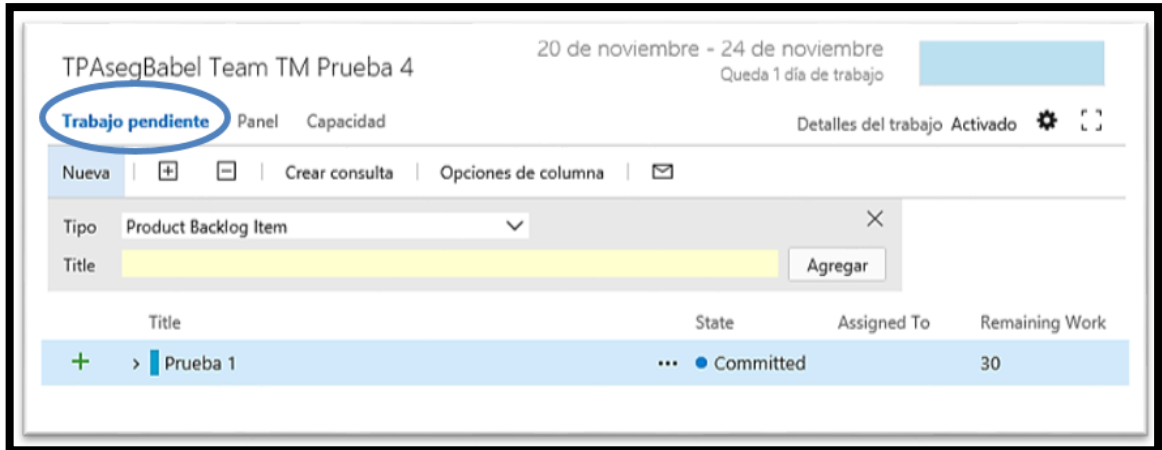


Ilustración 45: Visualizar Trabajos Pendientes.

b) Desde la cejilla del panel:

Muestra el tablero del sprint seleccionado, donde podemos agrupar las tareas por backlog y permite el trabajo de manera dinámica de las diferentes tareas.

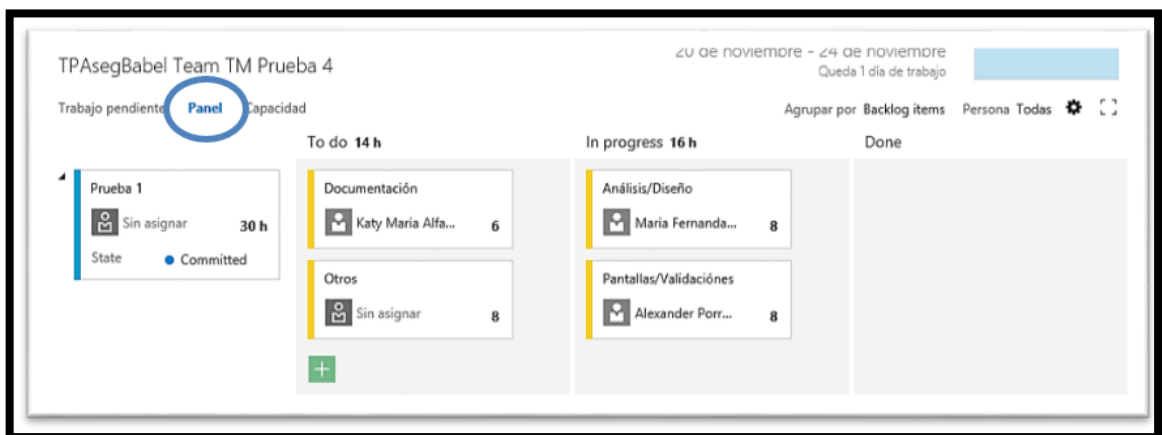


Ilustración 46: Visualizador Trabajos Pendientes. Panel.

c) Desde la cejilla de Capacidad:

La cejilla de capacidad permite configurar el trabajo de cada miembro del equipo en el sprint.

TPAsegBabel Team TM Prueba 4 20 de noviembre - 24 de noviembre
Queda 1 día de trabajo

Trabajo pendiente Panel **Capacidad** Detalles del trabajo Activado ⚙️ 🗄️

🏠 ↶ | + 🔍 📄

Usuario	Días libres	Activity	Capacidad por día
Alexander Delgado Castillo	Días: 0	▼	3
Maria Fernanda Salas Loria	Días: 0	▼	3
Saira Rebeca Martinez San...	Días: 0	▼	3
Días libres del equipo	Días: 0	Estos días libres se aplican al grupo entero.	

Ilustración 47: Visualizador de Trabajos Pendientes. Capacidad

F. Grafica de esfuerzo.

Muestra el avance del proyecto dando seguimiento a la evolución de esfuerzo, evaluando la línea de tendencia.

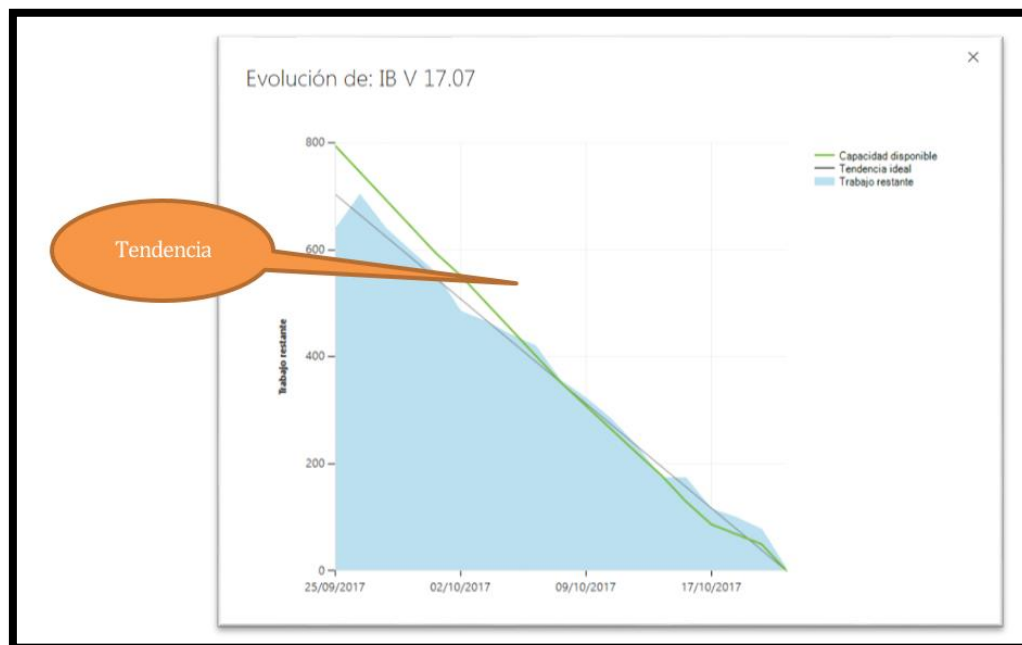


Ilustración 48: Gráficas de Esfuerzo.

G. Consultas listas plana.

The screenshot shows the 'Consulta PBI' interface. At the top, it says 'Query returned no results.' Below that, there are tabs for 'Results', 'Editor', and 'Charts'. The main area is titled 'Flat list of work items' and includes a 'Query across projects' checkbox. Underneath, there are 'Filters for top level work items' with two clauses: 'Work Item Type = Product Backlog Item' and 'State = Accepted'. An orange callout bubble labeled 'Filtro' points to the filter section. Below the filters, there are icons for 'Save query', 'Refresh', 'Share', and 'Copy query URL'. At the bottom, a table header is visible with columns: 'ID', 'Work Item...', 'Title', 'Assigned To', and 'State'. An orange callout bubble labeled 'Resultado' points to the table area.

Ilustración 49: Consulta de Listas Planas.

H. Consultas Graficadas

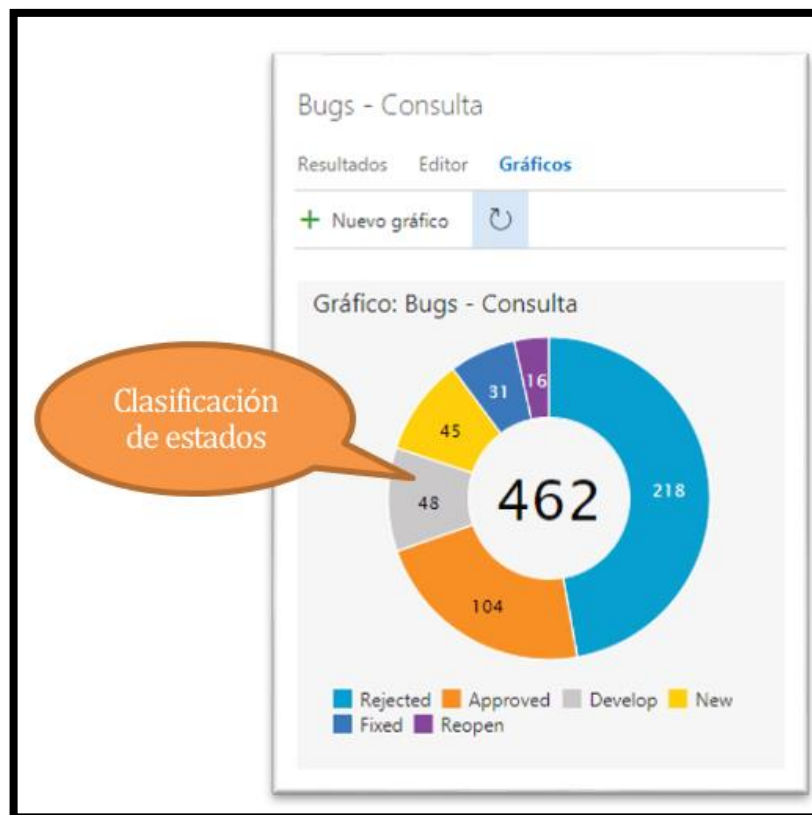


Ilustración 50: Consultas Graficadas.

I. Consultas listas vínculo directo.

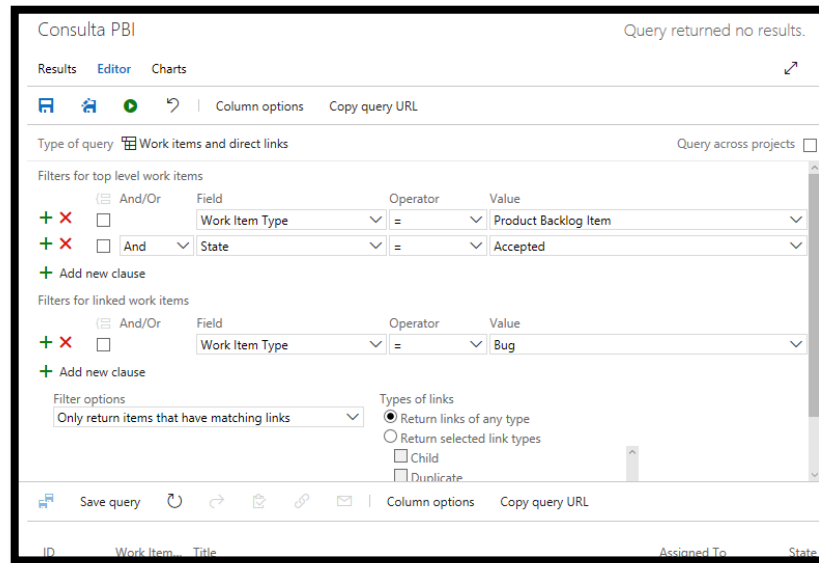


Ilustración 51: Consulta listas de vínculos directos.

J. Consultas tipo árbol.

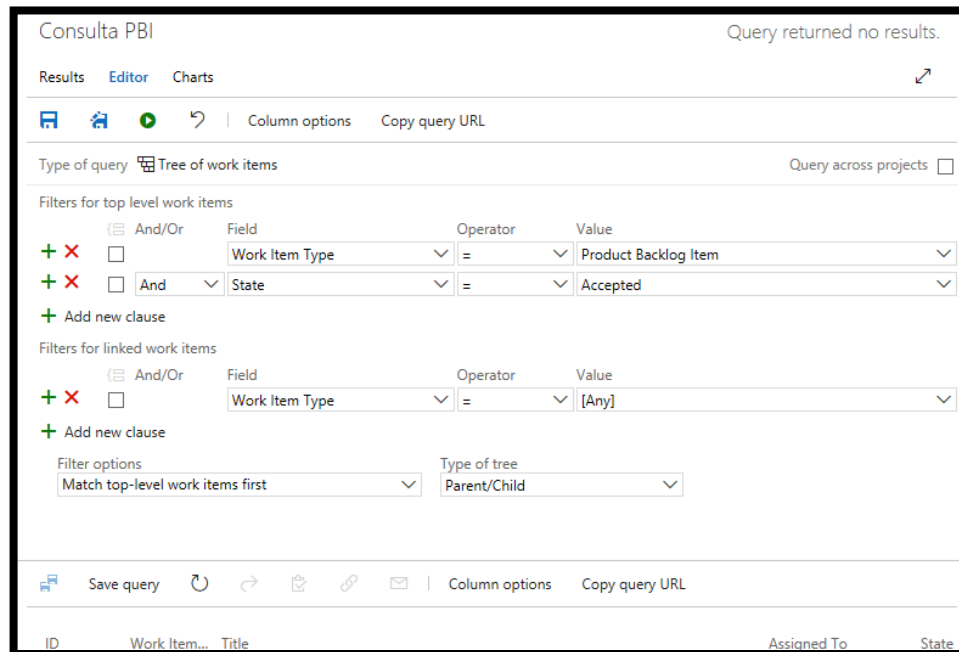


Ilustración 52: Consultas de Tipo Árbol.

6.11 Guía de uso Microsoft Test Manager

6.11.1 Configuración de la herramienta

- a. Al inicio se agrega el URL que aloja el VSTS a conectar.



Ilustración 53: Configuración de la Herramienta.

- b. Se selecciona al team project a trabajar

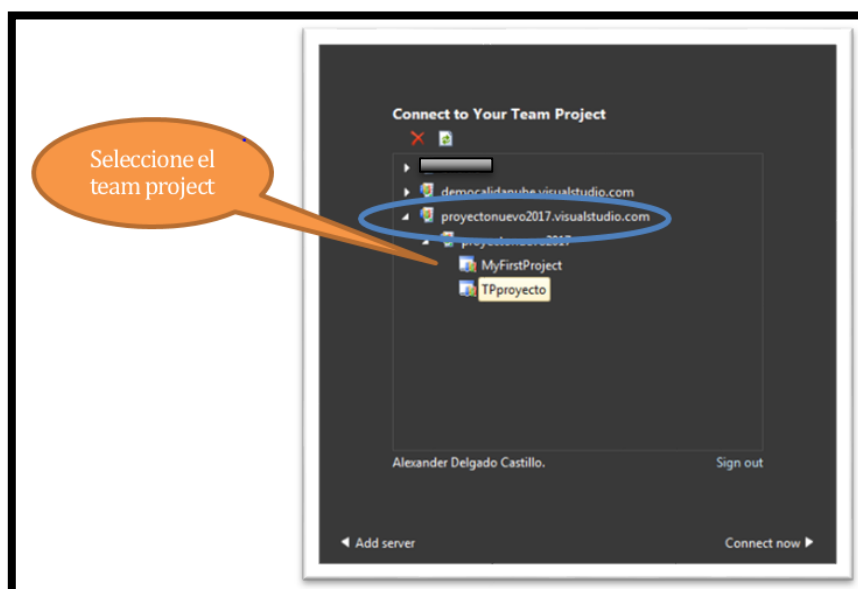


Ilustración 54: Seleccionar Team Project

- c. Se crea o se selecciona el plan de pruebas ligado al sprint a trabajar.

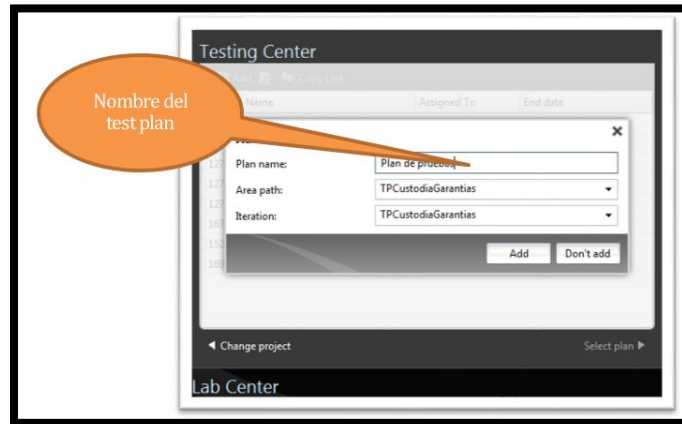


Ilustración 55: Seleccionar Plan de Pruebas ligado al sprint.

6.11.2 Planeamiento y diseño de casos de pruebas.

- a. El planeamiento de pruebas se desarrolla en la cejilla de Plan.

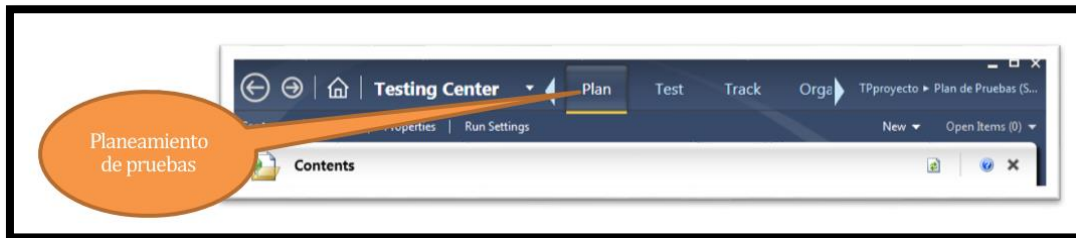


Ilustración 56: Planeamiento de pruebas. Plan.

- b. Configuración del plan de pruebas.

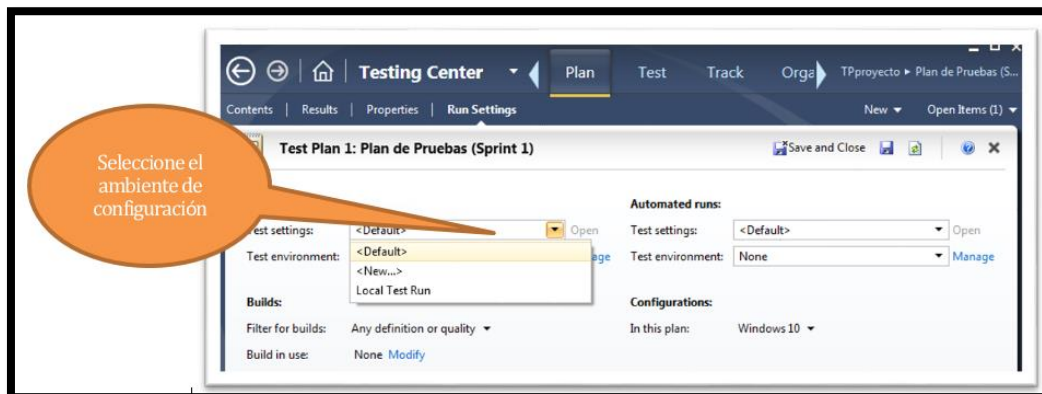


Ilustración 57: Configuración Plan de Pruebas.

- c. Crear una suite correspondiente a la historia de usuario.

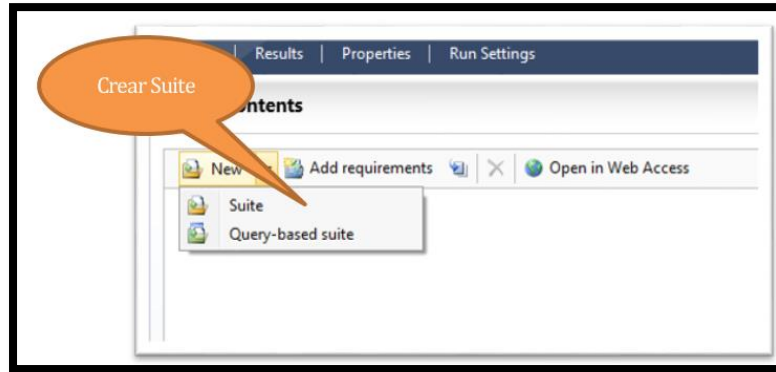


Ilustración 58: Crear Suite.

- Se crean dentro de cada HU las suites para clasificar los tipos de pruebas
 - Interfaz gráfica – IG
 - Pruebas Funcionales – FUN
 - Pruebas de integración – INT
 - Pruebas de regresión – REG

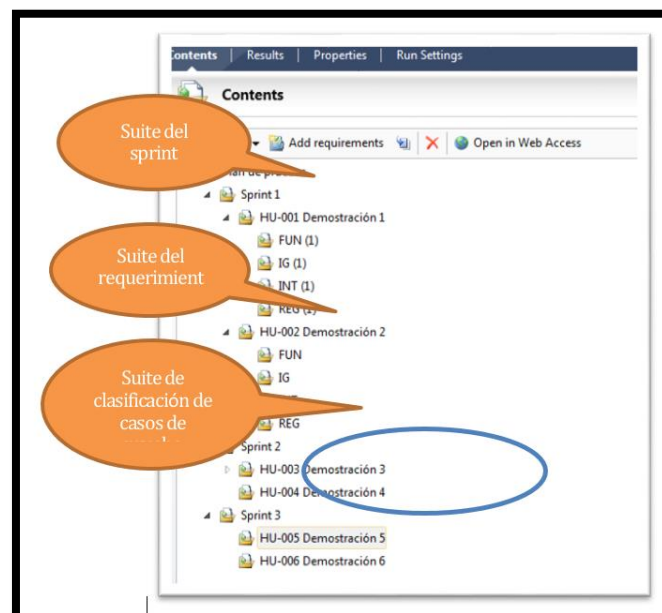


Ilustración 59: Creación de Suites en Historias de Usuario.

- d. Se crean los casos de pruebas para las historias de usuario en cada suite según la clasificación del tipo de prueba.

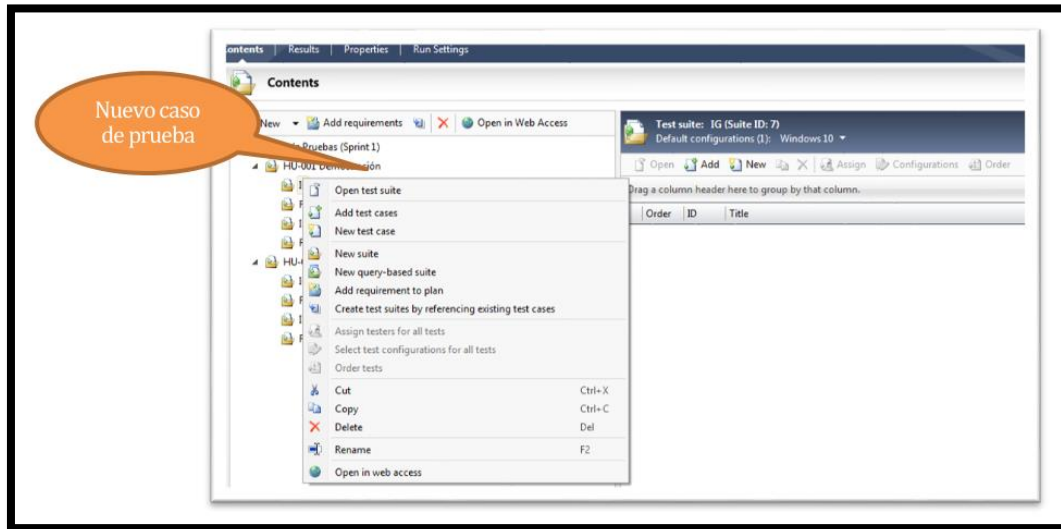


Ilustración 60: Creación de casos de prueba para Historias de Usuario.

- Crear el caso de prueba y agregar parámetros al set de datos.

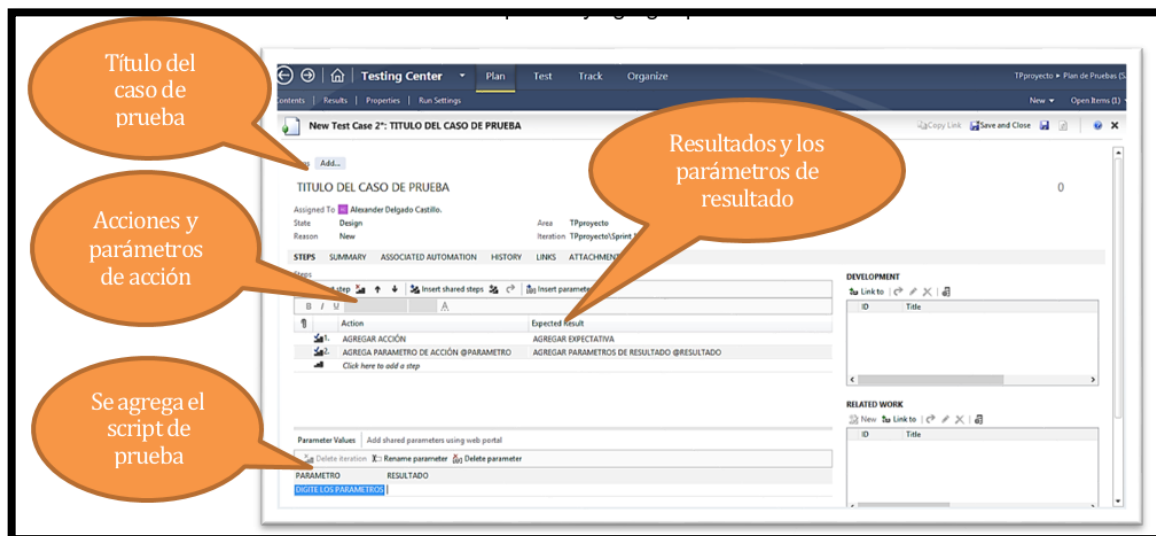


Ilustración 61: Crear Caso de Pruebas

- Agregar documentos para respaldar las pruebas o documentación que ayude a la revisión.

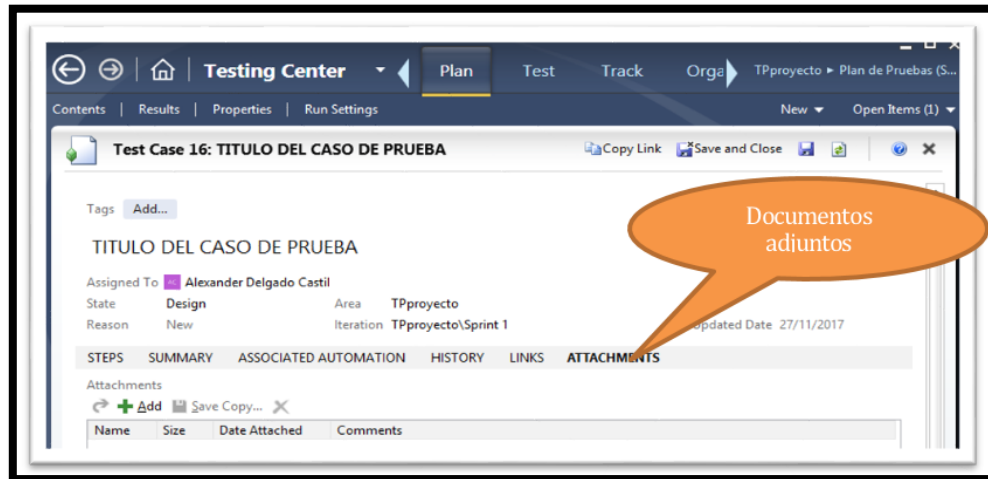


Ilustración 62: Agregar Documentos para respaldar pruebas.

- e. Se cambia el estado de Desing a Ready una vez que el caso de prueba esté listo para ejecución.

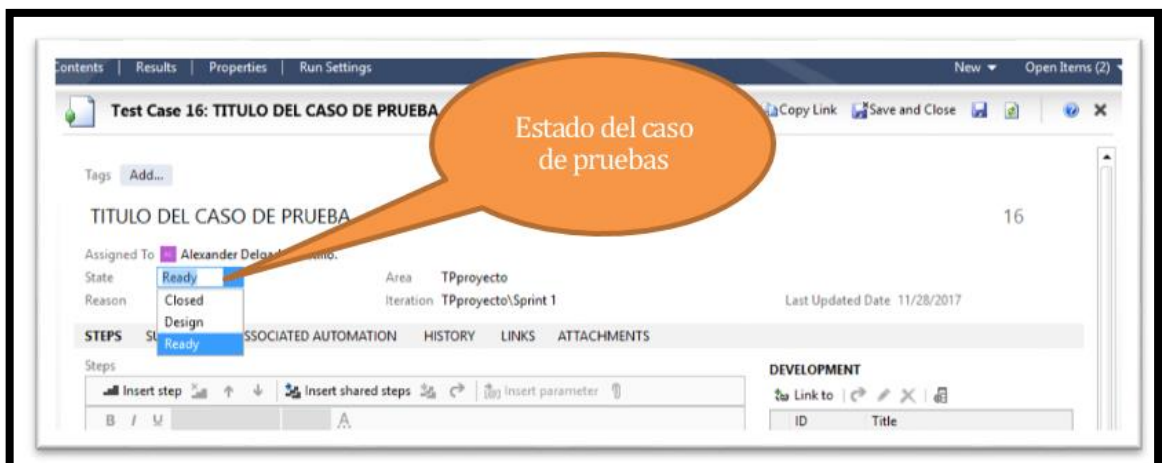


Ilustración 63: Estado Casos de Pruebas.

- a) **ID:** número de identificación del caso de prueba, la herramienta lo asigna automáticamente.

- b) **Título:** El título del caso de prueba indica específicamente lo que se va a probar.
 - c) **Descripción:** Descripción de lo que se va a realizar, se indica el propósito de la prueba.
 - d) **Precondiciones:** Son especificaciones necesarias que se deben cumplir para la ejecución de la prueba.
 - e) **Pasos:** recorrido paso a paso de la prueba que se va a realizar, contiene los parámetros de entrada.
 - f) **Resultados Esperados:** Resultado de todos los pasos definidos en el caso de prueba, contiene los parámetros de resultado.
 - g) **Historia de Usuario:** se agrega un link para enlazar la HU a evaluar.
 - h) **Sprint:** Numero de sprint a evaluar, se relaciona con el caso de prueba.
 - i) **Estatus:** Estado del caso de prueba.
 - j) **Archivos adjuntos:** imágenes o archivos que ayude a ejecutar el caso de prueba.
- **Consejos para la realización de casos de prueba**
 - Los casos de prueba tienen que ser escritos de manera detallada.
 - Si el cliente pide cambios en las especificaciones o las historias de usuario, hay que actualizar los casos de prueba.
 - Una vez listo el caso de prueba revíselo para asegurarse que contiene todo lo necesario para realizar la prueba.

- Agregue los resultados esperados y en el caso necesario agregue variables a los resultados para validar escenarios distintos.
 - Al hacer referencia a documentos o imágenes, de debe adjuntar estos para completar la revisión durante la ejecución.
 - Siempre revisar que cada caso de prueba, fallido o exitoso tenga su debida validación.
- f. Consultar el resultado del Plan de Pruebas, seleccionar la cejilla “Test Plan” y la cejilla “Results”

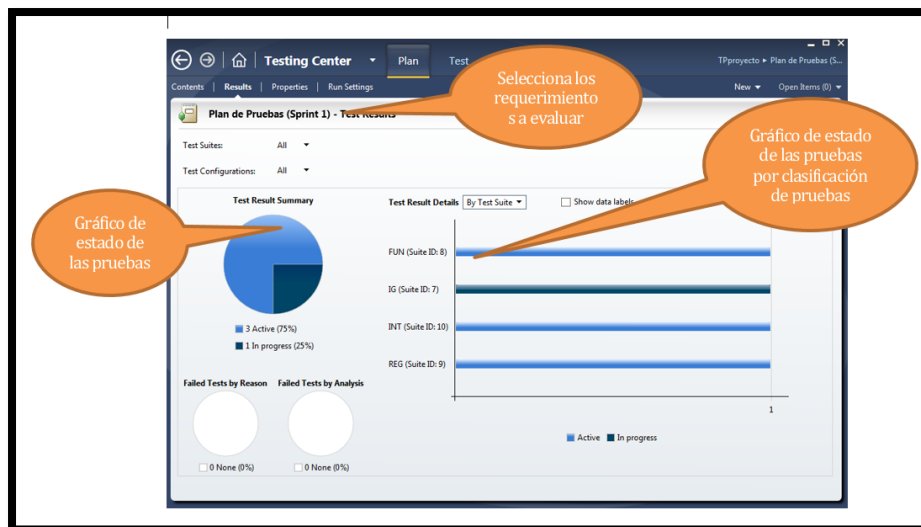


Ilustración 64: Consulta Plan de Pruebas.

6.11.3 La ejecución de casos de prueba.

- a. La ejecución de pruebas se desarrolla en la cejilla de “Test” -> “Run Tests”.

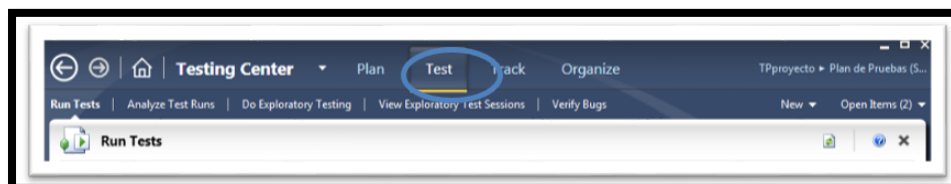


Ilustración 65: Ejecución Casos de Prueba. Test.

- b. Seleccione la suite, según la historia de usuario y la clasificación de la prueba a evaluar.

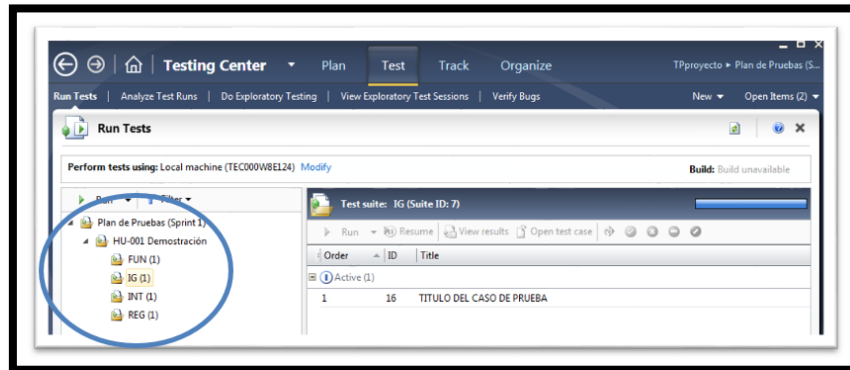


Ilustración 66: Seleccionar Suites según Historia de Usuario .

- c. Seleccione el caso de prueba y presione el botón “Run” para ejecutar.

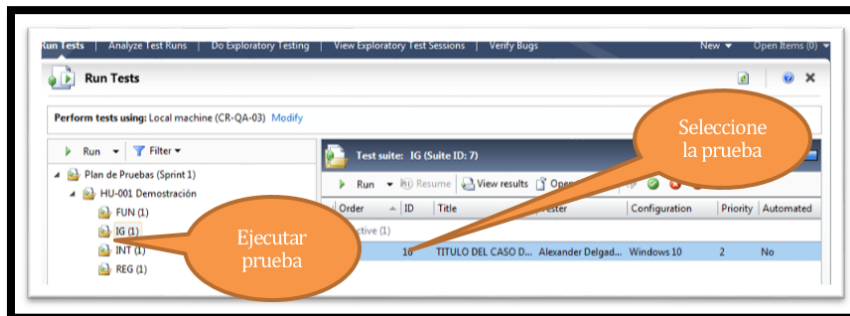


Ilustración 67: Casos de Prueba. Run.

d. Selección de la Iteración a evaluar.

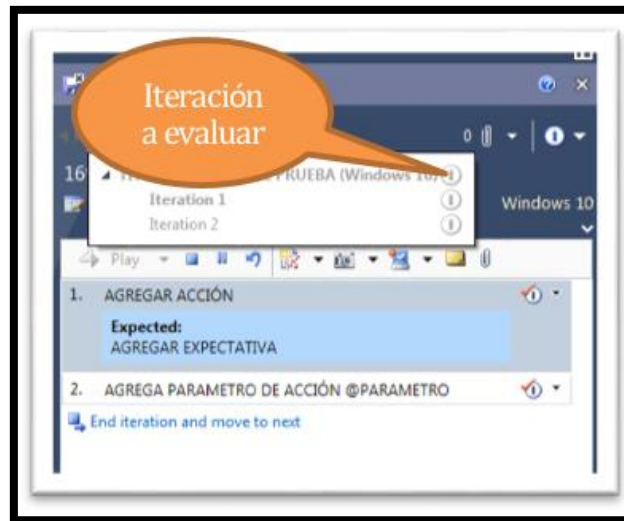


Ilustración 68: Selección de la Iteración a Evaluar.

e. Presione la opción “Start Test” para iniciar la ejecución.

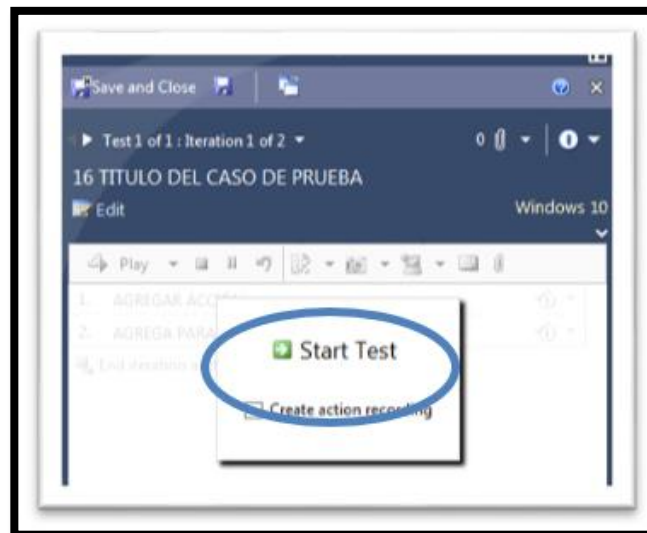


Ilustración 69: Start Test.

6.11.4 Ejecutar casos de prueba

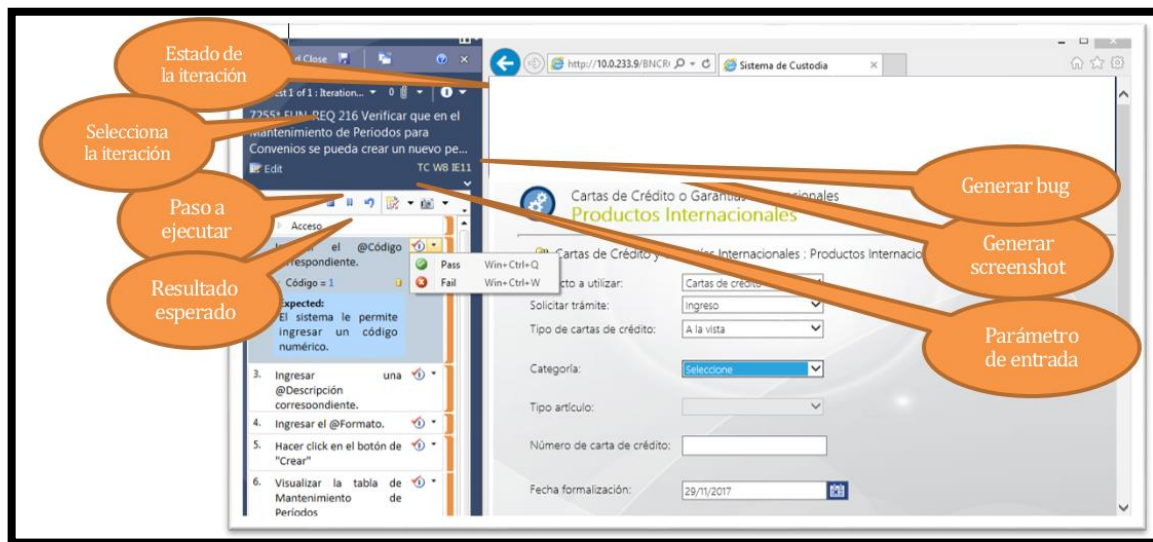


Ilustración 70: Ejecución de Casos de Prueba.

- a. Inicie la navegación del sistema acorde con los pasos y parámetros, realizando evaluación de las expectativas de cada iteración a evaluar.

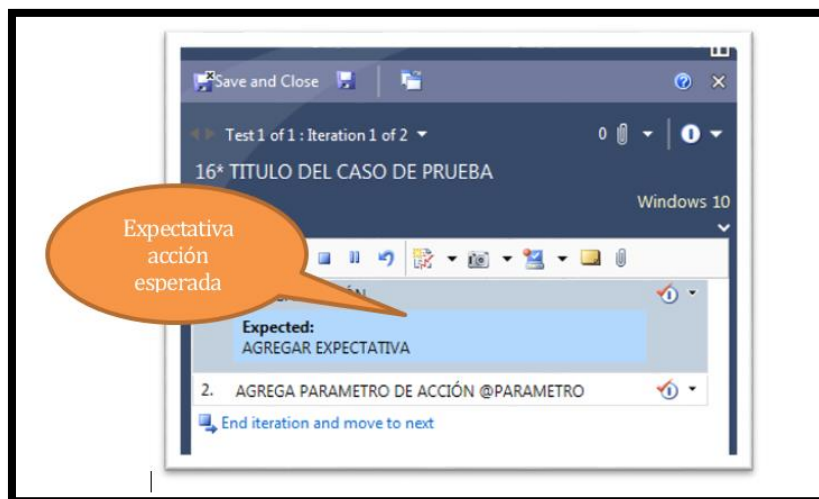


Ilustración 71: Evaluación de las Expectativas.

b. Seleccione el resultado de la prueba para cada paso de la iteración.

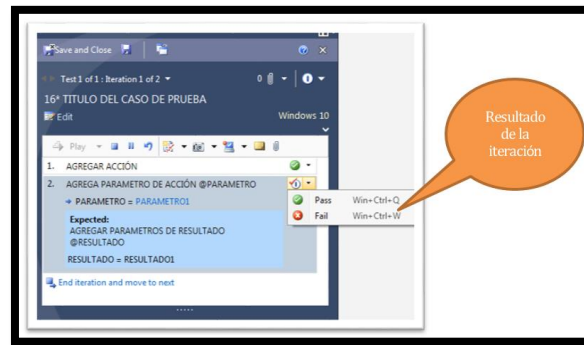


Ilustración 72: Resultados de la prueba de la iteración.

c. Continuar con la siguiente iteración.

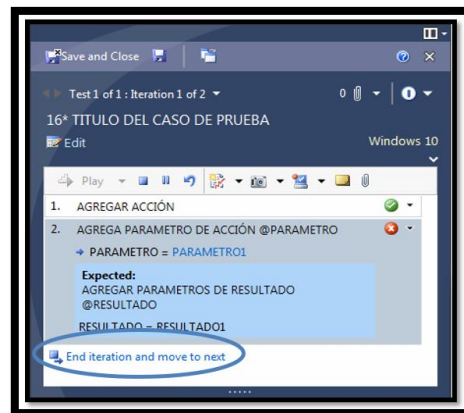


Ilustración 73: Continuar con la Siguiente Iteración.

d. Guardar el resultado de la prueba

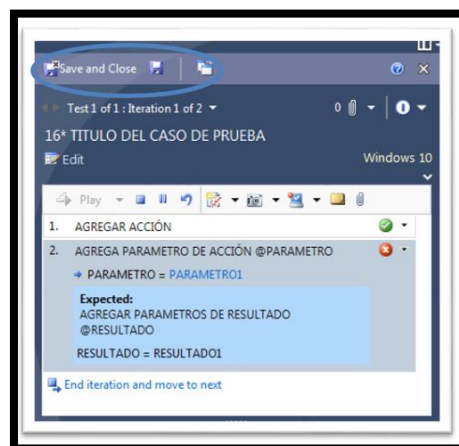


Ilustración 74: Guardar Resultados de la Prueba.

e. Resultado de una prueba específica

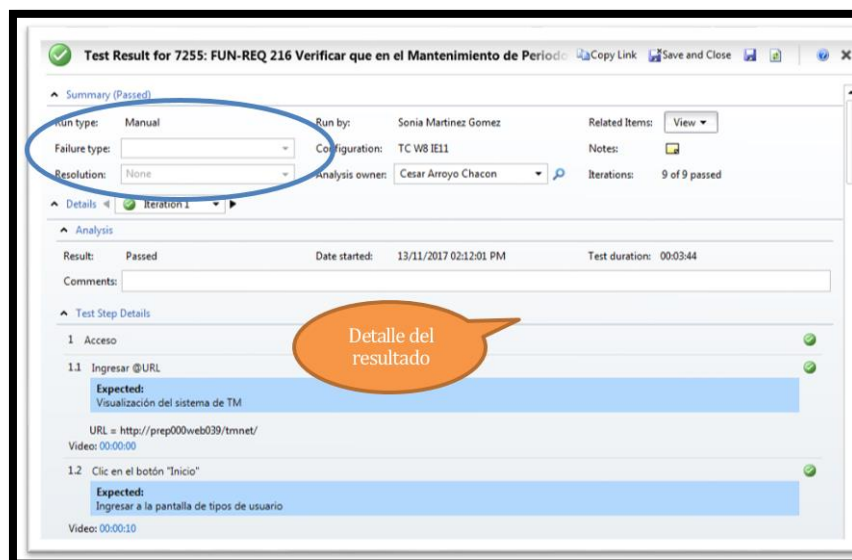


Ilustración 75: Resultado de una Prueba Específica.

6.11.5 Ejecución de pruebas exploratorias

a. Ejecutar pruebas exploratorias

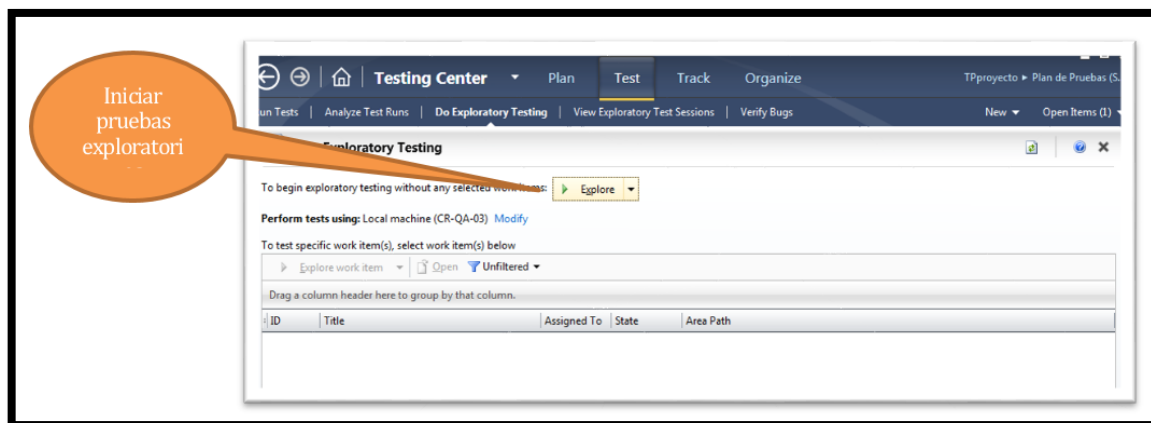


Ilustración 76: Ejecutar Pruebas Exploratorias.

6.11.6 Gestión de Dashboard y métricas de evaluación de procesos de desarrollo y calidad de software.

a. Gráficos auto-generados por recomendación de VSTS.

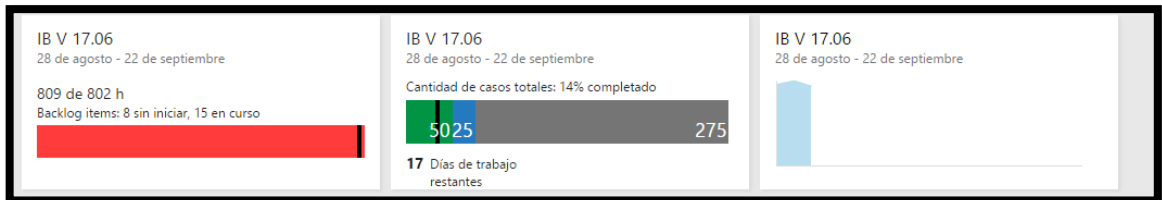


Ilustración 77: Gráficos Auto-Generados por recomendación de VSTS.

Uso de Widget

1. Capacidad de sprint
2. Evolución de sprint
3. Introducción al sprint

- b. Métricas del proceso y aseguramiento de la calidad de software:
- Análisis de bugs según las clasificaciones de los incidentes.
 - Estado de los Bug agregados al Sprint.
 - Cantidad de Bug asignados a un colaborador
 - Cantidad de Bug y el estado correspondiente asignado a un colaborador.
 - Cantidad de Bug asignados a un colaborador y la categoría específica.
 - Cantidad de Bug reportados por un área específica y agrupados por clasificación.

- Cantidad de historias de usuario evaluadas en un sprint específico.
- Cantidad de Historias de usuario asignadas a un colaborador.
- Cantidad de historias por estado, para evaluar el seguimiento a las mismas.
- Cruce de tareas y las historias de usuario, FastTrack o bug's.
- Cantidad de tareas registradas a una historia de usuario y la sumatoria del tiempos estimados y tiempos finalizados, análisis de desfase de ejecución con planeación.
- Cantidad de Esfuerzo trabajado en un sprint específico.
- Cantidad de casos de prueba desarrollados a una historia de usuario.
- Total, de tareas del sprint y cantidad asignada a los diferentes colaboradores
- Total, de trabajo pendiente por Actividad
- Total, de casos de prueba por sprint y clasificación de las mismas.

c. Ejemplo de Dashboard.



Ilustración 78: Ejemplo de Dashboard.

Referencia Bibliográfica

Referencias

Albaladejo, X. (01 de 01 de 01). *¿Qué es SCRUM? Proyectos Agiles.org*.
Obtenido de Proyectos Agiles.org : <https://proyectosagiles.org/>

Aparicio, A. (01 de enero de 2012). *Universidad nacional abierta y a distancia*.
Recuperado el 24 de junio de 2018, de Universidad nacional abierta y a distancia: <http://datateca.unad.edu.co/contenidos/301404/301404.pdf>

Apiumhub. (s.f.). <https://apiumhub.com/>. Obtenido de <https://apiumhub.com/es/tech-blog-barcelona/transformacion-agil-pasos-estadisticas/>

Atlassian. (SF de 2019). Atlassian. Recuperado el 07 de Enero de 2019, de Atlassian: líder en el Cuadrante Mágico 2018 de Gartner para herramientas de planificación ágil empresarial: https://es.atlassian.com/gartner?_ga=2.241521030.581869224.1546904566-718806983.1546904566

Barrios, W., Godoy, M. G., Fernández, M. I., Mariño, S., Ferreira, F. M., & Zarrabeitia, C. T. (01 de octubre de 2011). *Universidad Nacional de La Plata*. Recuperado el 19 de junio de 2018, de SCRUM: Experiencia de Aplicación en una Empresa de Desarrollo de Software del NEA.: http://sedici.unlp.edu.ar/bitstream/handle/10915/18745/Documento_completo.pdf7sequence=1

Bravent It Consulting Company. (01 de enero de 2013). *POR QUÉ IMPLANTAR ALM PARA GESTIONAR EL CICLO DE VIDA DE TUS PROYECTOS*. Obtenido de BENEFICIOS DE ALM. Recuperado el 19 de junio de 2018, de <https://www.bravent.net/por-que-implantar-alm-para-gestionar-el-ciclo-de-vida-de-tus-proyectos>

- J., Letelier, P., & Penadés, M. (01 de enero de 2003). *Metodologías Ágiles en el Desarrollo de Software*. Recuperado el 26 de junio de 2018, de Metodologías Ágiles en el Desarrollo de Software: http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf
- Castro, F. F. (26 de abril de 2013). *Ministerio De Justicia*. Obtenido de Asamblea Legislativa De la República de Costa Rica: <https://www.ilo.org/dyn/natlex/docs/ELECTRONIC/95503/112471/F-2082388954/CRI95503.pdf>
- Copyright 2018 Psicología y Mente. (30 de mayo de 2017). <https://psicologiaymente.net>. Obtenido de <https://psicologiaymente.net/miscelanea/tipos-de-investigacion>
- Cruz Ramírez, J. (01 de enero de 2015). *HISTORÍA DE LA CALIDAD*. Recuperado el 01 de agosto de 2018, de HISTORÍA DE LA CALIDAD: http://docencia.fca.unam.mx/~lvaldes/cal_pdf/cal5.pdf
- Del Rincón, A. J., & Latorre, A. (01 de enero de 1992). Investigación educativa. *Metodologías de investigación educativa*. Barcelona, España. Recuperado el 20 de junio de 2018
- Díaz Agudelo, J. E., & Olaya Figueroa, J. F. (01 de enero de 2010). *Guía metodológica para la gestión de proyectos de software basados en metodologías ágiles, utilizando ambientes de desarrollo colaborativo*. (P. U. SISTEMAS, Ed.) Recuperado el 26 de junio de 2018, de Guía metodológica para la gestión de proyectos de software basados en metodologías ágiles, utilizando ambientes de desarrollo colaborativo: <http://webcache.googleusercontent.com/search?q=cache:UbVeKLRJJKQJ:pegasus.javeriana.edu.co/~CIS0910IS05/Memoria/Memoria%2520de%2520Trabajo%2520de%2520Grado%2520%5BConstruColectiva%5D.doc+&cd=2&hl=es&ct=clnk&gl=cr>
- Eduardo, P. P. (01 de 01 de 2015). *SISTEMA DE INFORMACIÓN WEB SERVICE DESK PARA LA EMPRESA ALMACENES POPULARES S.R.L. BASADO*

EN WEB SEMÁNTICA. Obtenido de UNIVERSIDAD PRIVADA ANTENOR ORREGO - REPOSITORIO UPAO: http://repositorio.upao.edu.pe/bitstream/upaorep/2899/1/RE_ING.SIST_MAURO.PAZ_INFORMACION.WEB_DATOS.PDF

Eyssautier de la Mora, M. (01 de enero de 2002). Metodología de la Investigación Desarrollo de la inteligencia. *Metodología de la Investigación Desarrollo de la inteligencia*. México, México: Cengage Learning. Recuperado el 10 de Setiembre de 2018, de Desarrollo de la inteligencia.

Foundation Level Syllabus, recuperado el 12 de enero del 2019, de he International Software Testing Qualifications Board® (ISTQB®). Consultado en: <https://www.istqb.org/downloads/syllabi/foundation-level-syllabus.html>

Garzás, J. (05 de Julio de 2012).). *¿Cuáles son las mejores herramientas de gestión del desarrollo software? ¿Atlassian, Microsoft, IBM, HP?* Recuperado el 20 de junio de 2018, de). *¿Cuáles son las mejores herramientas de gestión del desarrollo software? ¿Atlassian, Microsoft, IBM, HP?* <http://www.javiergarzas.com/2012/07/herramientas-alm.html>

Garzas, J. (10 de enero de 2013). El Humilde Programador. Recuperado el 14 de noviembre de 2018, de <https://www.javiergarzas.com/2013/01/dijkstra-programador-humilde.html>

González, H. (11 de Julio de 2012). Recuperado el 20 de junio de 2018. *Gestión y Calidad*. Obtenido de HERRAMIENTAS PARA LA MEJORA CONTINUA: <https://calidadgestion.wordpress.com/2012/07/11/herramientas-para-la-mejora-continua/>

Hastqb (El comité miembro de ISTQB® encargado de la región Hispanoamericana) consultado el 12 de enero del 2019, de Hastqb. Consultado en: <http://hastqb.org>

- ISTQB (International Software Testing Qualifications Board) consultado el 12 de enero de 2019, de ISTQB. Consultado en: <https://www.istqb.org/>
- James Sánchez, Freelancer. (30 de abril de 2015). Consultado el 12 de enero de 2019. *Community, FreeLancer*. Obtenido de Proceso del Desarrollo de Software ¿Qué metodología utilizar?: <https://www.freelancer.com/community/articles/proceso-del-desarrollo-software>
- Letelier, P., & Penadés, M. (15 de diciembre de 2005). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Recuperado el 13 de mayo de 2018, de Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP): <http://www.cyta.com.ar/ta0502/v5n2a1.htm>
- Mera-Paz, J. A. (19 de Julio de 2016). *Análisis del proceso de pruebas de calidad de software*. Obtenido de Análisis del proceso de pruebas de calidad de software: <http://dx.doi.org/10.16925/in.v12i20.1482>
- Millan, Y. (18 de abril de 2013). *Software*. Recuperado el 12 de marzo de 2018, de Software: <https://www.importancia.org/software.php>
- Muñoz, C. (01 de enero de 2011). *Cómo elaborar y asesorar una investigación de tesis. Cómo elaborar y asesorar una investigación de tesis*. México, México: Pearson Educación. Recuperado el 26 de junio de 2018
- Pérez, C. M. (01 de enero de 2015). *UNA METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE EN UNA COMPAÑÍA FINANCIERA*. Recuperado el 17 de Julio de 2018, de UNA METODOLOGÍA ÁGIL PARA EL DESARROLLO DE SOFTWARE EN UNA COMPAÑÍA FINANCIERA: <https://repository.unimilitar.edu.co/bitstream/handle/10654/13235/Proyecto%20investigacion%20-%20Claudia%20Perez.pdf?sequence=1>
- Riquelme Leiva, M. (01 de junio de 2015). *Las 5 Fuerzas de Porter*. Recuperado el 13 de noviembre de 2018, de Clave para el Éxito de la Empresa: <http://www.5fuerzasdeporter.com/>

- Rodríguez F, C. E., Clavijo, A. M., Chacón Torres, J. A., & Valero Pinzón, B. A. (25 de febrero de 2016). Sistematización de Datos. *Sistematización de Datos*. Bogotá, Colombia. Recuperado el 13 de noviembre de 2018, de <https://webcache.googleusercontent.com/search?q=cache:mOd6fx4dPAUJ:https:// analisisoft.files.wordpress.com/2016/02/trabajo-investigativo-02-anc3a1lisis-de-sistemas.docx+&cd=4&hl=es&ct=clnk&gl=cr>
- Sampieri, R. H. (s.f.). Metodología de la Investigación. En R. H. Sampieri. MC Graw Hill Education.
- Sanchez Freelancer, J. (30 de abril de 2015). *Community, freelancer*. Recuperado el 03 de Julio de 2018, de Proceso del Desarrollo de Software ¿Qué metodología utilizar? <https://www.freelancer.com/community/articles/proceso-del-desarrollo-software>
- Sancho, M. C. (15 de DICIEMBRE de 2013). Recuperado el 08 julio de 2018 de *TECNOLOGÍA, EL FINANCIERO*. Obtenido de Servicios de software en Costa Rica representan el 1,3% de la producción local, <https://www.elfinancierocr.com/tecnologia/servicios-de-software-en-costarica-representan-el-13-de-la-produccion-local/KRWAF4ETORHSPCJIIBMWNHDEI/story/>
- Scrum Manager. (S.F.). *Scrum Manager BoK*. Recuperado el 28 de junio de 2018, de Modelo original de Scrum para desarrollo de Software: http://www.scrummanager.net/bok/index.php?title=Modelo_original_de_Scrum_para_desarrollo_de_software
- SINAES. (01 de 01 de 2010). *SINAES*. Obtenido de Sistema de Aseguramiento de la Calidad de la Educación Superior de Costa Rica durante el 2010: https://www.sinaes.ac.cr/documentos/Logros_y_Retos%20Calidad_del_SINAES_2010.pdf

- Stake, R. E. (01 de enero de 1998). Investigación con estudio de casos (Roc Filella, trad.). *Investigación con estudio de casos (Roc Filella, trad.)*. Morata, España.
- The Agile Alliance. (01 de enero de 2014). *The Twelve Principles of Agile*. Recuperado el 04 de Julio de 2018, de The Twelve Principles of Agile: <http://www.agilealliance.org/the-alliance/the-agile-manifesto/the-twelve-principles-of-agile-software>.
- Ticona Yanqui, F. E. (30 de abril de 2014). Metodología Scrum para el desarrollo de software y gestión de proyectos en las pequeñas y medianas empresas de la ciudad de Juliaca. *Revista científica Investigación Andina*, 13, 88-95. Obtenido el 04 de Julio del 2018 <https://revistas.uancv.edu.pe/index.php/RCIA/article/view/142/122>
- Tomaselli, G. P., Acuña, C., Estayno, M., & Lenkovich, C. (01 de enero de 2013). *Conaiisi*. Recuperado el 18 de Julio de 2018, de SCRUM: Una revisión de la literatura: <http://www.conaiisi.unsl.edu.ar/2013/199-502-1-DR.pdf>
- Woods, P. (01 de enero de 1987). La escuela por dentro. *La etnografía en la investigación educativa*. Barcelona, España: Paidós.
- CAPACITA-TECH © 2019. El concepto de desarrollo de software, consultado en url: <http://www.capacita-tech.com/Desarrollo.html>
- Ecured, Desarrollo de software, consultado en el sitio web, https://www.ecured.cu/Desarrollo_de_software

Anexos

Encuestas

Las encuestas se realizaron con la herramienta de “**googleforms**”, y se compartió con el siguiente link:

https://docs.google.com/forms/d/e/1FAIpQLSeh9yWVUXg5LeIrljZYFrnUr7MBLR_PS93pGeWTvE79-91yYuQ/viewform?usp=sf_link

Entre las empresas a considerar dentro de las encuestas están las siguientes:

Empresa	Contacto
Gruponaby	info@gruponaby.com
FacturacionElectrónica	info@facturacionelectronicacr.com
Ekomercio	pacevedo@ekomercio.com.mx
MSASOFT	msalamaca@msasoft.com
AVD TEC Internacional S.A.	mmorales@avdinternacional.com
Factura Tributaria.com	raymundo@actualidadtributaria.net
GTI Costa Rica	montalvo@gticr.com
Sisnet soluciones informáticas	oramirez@sisnet.co.cr
Grupo Diverscan S.A.	mercadeo@grupodiverscan.com
Almamater de Centroamérica	karen@edi.co.cr
Factura Profesional	info@facturaprofesional.com
Edicomunicaciones de México S.A. de C.V.	info_cam@edicomgroup.com
Signature South Consulting Costa Rica S.A.	info@cr.southconsulting.com
NG Technology	mercadeo@ng.cr;soporte@ng.cr
NEOTECNOLOGIAS S.A.	info@neotecnologias.com
CR ADVANCED CONSULTING S.A.	msandoval@crconsulting.co.cr
CEWTEC SEGURIDAD TI	pbolanos@cewtec.com
TECAPRO	luis.gomez@tecapro.com
INTEL	jbrenes@utn.ac.cr
DXC	jose-venustiano.ticante.hernandez@dxc.com

Empresa	Contacto
HoneyWell HWL	Luis.Gonzalez3@Honeywell.com
Capgemini (Oracle)	bernardo.vargas@capgemini.com
RICOH	adriana.fernandez@ricoh-la.com
Cyberfuel	cmoreno@cyberfuel.com
TECNOLOGÍA PERSONALIZADA Y SOLUCIONES TPS SA	arnaldo@blue-loop.com
IT REGIONAL SOFT SRL	dimoi@regionalssoft.com
ACD SYSTEM	info@acdsystemcr.com
Bimbo México S.A.	miguel.garcia@grupobimbo.com

Encuesta presentada

Aseguramiento de la calidad de software, Tesis de Licenciatura - UTN

La siguiente encuesta pretende realizar un análisis de las diversas metodologías, herramientas y formas de asegurar la calidad del software de los productos realizados en el mercado laboral.

1. Datos básicos de las representadas

Se pretende conocer el entorno laboral de las representadas de los encuestados.

¿Alguna vez ha participado en una encuesta sobre metodologías de desarrollo? *

Si
 Primer vez
 No recuerdo

Ilustración 1: Preguntas Encuesta. Datos Básicos de las Empresas. 1

¿Cuántos años en el mercado tiene la empresa para la cual trabaja? *

Menos de 1 año
 Entre 1 año y 5 años
 Mayor a 5 años

¿Qué tan grande es la empresa de software a la que perteneces? Deberá incluir el personal relacionado con temas sobre desarrollo de software. *

Menor a 10 colaboradores
 Entre 11 y 50 colaboradores
 Entre 51 y 100 colaboradores
 Mayor a 100 colaboradores

Ilustración 2: Preguntas Encuesta. Datos Básicos de las Empresas. 2-3

¿Cuál de los siguientes roles describe mejor tu actual posición en la organización? *

- Administrador de proyectos
- Desarrollador/Programador
- Jefe de desarrollo
- Consultor
- Arquitecto de Software
- QA/Pruebas/Tester
- IT Staff/Soporte
- Otra...

Ilustración 3: Preguntas Encuesta. Datos Básicos de las Empresas. 4

¿Cuántos años de experiencia posee en el puesto indicado anteriormente? *

- Menor a 1 año
- Entre 1 año y 5 años
- Mayor a 5 años

¿Tienes conocimiento en las metodologías ágiles? *

- Sí
- No

Ilustración 4: Preguntas Encuesta. Datos Básicos de las Empresas. 5-6

¿Cuál de las siguientes situaciones describe mejor tu nivel actual de conocimiento/participación en metodologías ágiles? *

- Líder de equipo de desarrollo que usa metodologías ágiles
- Formo parte de un equipo de desarrollo que hace uso de metodologías ágiles
- Estoy considerando introducir metodologías ágiles en mi equipo actual
- He formado parte de equipos de trabajo que hacen uso de metodologías ágiles
- He escuchado acerca de las metodologías ágiles, conozco muy poco al respecto
- Ninguna de las anteriores

Ilustración 5: Preguntas Encuesta. Datos Básicos de las Empresas. 7

¿Por cuánto tiempo has hecho uso personalmente de las metodologías ágiles sin importar si es en el orden personal o laboral? *

- Nunca
- 6 Meses
- Hasta 1 año
- Entre 1 y 2 años
- Entre 2 y 5 años
- Más de 5 años

Ilustración 6: Preguntas Encuesta. Datos Básicos de las Empresas. 8

2. Metodologías del Proceso

En esta sección se determinará si las empresas de los encuestados utilizan metodologías para la mejora en los procesos, así como identificar cuál es la metodología más utilizada y la más popular en el entorno nacional.

¿Actualmente utilizan alguna metodología para definir el flujo del trabajo? *

- Sí
- No

Ilustración 7: Preguntas Encuesta. Metodologías del Proceso. 1

¿Dentro de la organización a la que pertenece utilizan alguna metodología Ágil? *

- Sí
- No

¿Cuántas localidades distintas están haciendo uso de metodologías ágiles en la empresa? *

- 1
- 2
- 3
- Más de 3
- Desconozco

Ilustración 8: Preguntas Encuesta. Metodologías del Proceso. 2-3

¿Cuál metodología utilizan? (Seleccione más de una opción en caso de ser necesario) *

Programación extrema (XP)

Metodología Cascada

Metodología Espiral

Metodologías de Crystal

Métodos de desarrollo de sistemas dinámicos (DSDM)

Desarrollo de Software adaptivo (ASD)

SERUM

RUP (Rational Unified Process)

SCRUM

Mi organización no utiliza ninguna metodología

Ilustración 9: Preguntas Encuesta. Metodologías del Proceso. 4

¿Los funcionarios manejan roles definidos para el desarrollo de las actividades? *

Sí

No

¿Tienen algún método para mejorar y desarrollar el proceso actual, bajo un concepto de mejora continua? *

Sí

No

¿Como administran el trabajo pendiente durante el desarrollo?

Texto de respuesta larga

Ilustración 10: Preguntas Encuesta. Metodologías del Proceso. 5-7

¿Actualmente el equipo ve como positiva la metodología aplicada? *

Sí

No

¿Actualmente realizan planificación de las tareas? *

Sí

No

Ilustración 11: Preguntas Encuesta. Metodologías del Proceso. 8-9

¿Cuál considera que puede ser un obstáculo que una organización puede adoptar para el uso de metodologías ágiles? (Seleccione más de una opción si lo considera necesario) *

Cultura organizacional

Resistencia al cambio dentro de la empresa

Falta de Experiencia con metodologías ágiles entre el personal

Apoyo de los niveles más altos de la empresa

Proyectos demasiado complejos

Falta de colaboración por parte de los clientes

Desconfianza en los resultados de las metodologías ágiles

Tiempo de transición muy largo

Limitantes en presupuestos financieros

Falta de información sobre las metodologías ágiles

Ilustración 12: Preguntas Encuesta. Metodologías del Proceso. 10

¿Dentro de su organización que porcentaje en proyectos desarrollados con metodologías ágiles han sido exitosos? *

- Menos al 10%
- Entre 11 y 30%
- Entre 31 y 60%
- Entre 61% y 90%
- Entre 91 y 99%
- El 100%
- Desconozco el dato
- No se utilizan metodologías ágiles.

Ilustración 13: Preguntas Encuesta. Metodologías del Proceso. 11

En su opinión personal ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles? (Seleccione más de una opción si lo considera necesario) *

- Mejorar la habilidad en el manejo de cambios
- Incrementar productividad
- Mejorar la calidad del software
- Mejorar la relación entre TI y Negocios
- Reducir el riesgo del Negocio
- Mejorar o aumentar la disciplina del desarrollo
- Mejorar el proceso de mantenimiento y cambios de software
- Mejorar el equipo de trabajo

Ilustración 14: Preguntas Encuesta. Metodologías del Proceso. 12

3. Aseguramiento de la calidad del software

En esta sección se definirán si las representadas de los encuestadores realizan casos de prueba, pruebas formales al desarrollo así como si cuentan con aseguramiento en la calidad del software.

¿En la empresa para la que labora realiza aseguramiento en la calidad del software? *

- Sí
- No

¿Considera que debe existir una mejora dentro de los procesos de aseguramiento de calidad de software que actualmente realiza su representada? *

- Sí
- No

Ilustración 15: Preguntas Encuesta. Aseguramiento en la Calidad del Software. 1-2

¿Considera que debe existir una guía que facilite la integración de los procesos de aseguramiento de calidad de software a la metodologías actuales? *

- Sí
- No

Ilustración 16: Preguntas Encuesta. Aseguramiento en la Calidad del Software. 3

¿Cuál considera que puede ser un obstáculo que una organización puede adoptar el aseguramiento de la calidad de software? (Seleccione más de una opción si lo considera necesario)

- Cultura organizacional
- Resistencia al cambio dentro de la empresa
- Falta de experiencia entre el personal
- Apoyo de los niveles más altos de la empresa
- Proyectos demasiado complejos
- Tiempo de transición muy largo
- Limitantes en presupuestos financieros
- Falta de información sobre aseguramiento de la calidad
- Falta de una guía que estructure el proceso

Ilustración 17: Preguntas Encuesta. Aseguramiento en la Calidad del Software. 4

¿En su representada realizan pruebas formales durante el desarrollo? *

- Sí
- No

¿Considera importante la creación y planificación de planes de prueba? *

- Sí
- No

Ilustración 18: Preguntas Encuesta. Aseguramiento en la Calidad del Software. 5-6

¿En su representada realizan casos de pruebas para documentar las pruebas?

- Sí
- No

¿En su representada documentan el resultado de las pruebas? *

- Sí
- No

¿En su representada realizan las diferentes pruebas durante el desarrollo en ambientes definidos para aseguramiento de la calidad? *

- Sí
- No

Ilustración 19: Preguntas Encuesta. Aseguramiento en la Calidad del Software. 7-9

4. Herramientas para la administración de Ciclo de Vida de las aplicaciones y aseguramiento de la calidad

En esta sección se pretende determinar las herramientas utilizadas en las representadas de los encuestados, para la administración en el ciclo de vida del proyecto.

¿Utilizan una herramienta para la gestión del desarrollo? *

- Sí
- No

Ilustración 20: Preguntas Encuesta. Herramientas Utilizadas. 1

¿Cual herramienta de gestión de proyectos utiliza? (Seleccione más de una opción en caso de ser necesario) *

Visual Studio Team Services

Team Foundation Server

IBM Rational solution for Collaborative Lifecycle Management

IBM Rational Team Concert

Redmine Solutions

Atlassian JIRA

HP Application LifeCycle Managment

SAP Solution Manager

Mi organización no utiliza ninguna herramienta ALM

Otro: _____

Ilustración 21: Preguntas Encuesta. Herramientas Utilizadas. 2

¿Utilizan herramientas para la documentación de pruebas? *

Sí

No

¿Cual herramientas para la documentación de pruebas utiliza? *

Microsoft Test Manager

Selenium

Testlink

Jmeter

Watir

Solex

WebInject

Imprimatur

Otro: _____

Ilustración 22: Preguntas Encuesta. Herramientas Utilizadas. 3-4

¿Cuál considera como el principal beneficio al utilizar herramientas para la gestión de proyectos? *

- Aumento de la productividad
- Mejora de la calidad del producto
- Mayor velocidad de desarrollo
- Disminución de costes
- Mayor colaboración y comunicación entre recursos
- Transparencia en el negocio
- Todas las anteriores

Gracias!!!

Ilustración 23: Preguntas Encuesta. Herramientas Utilizadas. 5

Entrevista

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se busca conocer. El proceso de

encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Questionario para entrevistas:

Información del funcionario entrevistado:

Empresa	
Área	
Entrevistado	
Puesto de desempeña	
Fecha y hora	

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?
2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?

6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?
8. ¿Cuál es la regularidad en que realizan pases a producción?
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?
14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?

18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
23. ¿En su representada realizan la documentación del resultado de las pruebas?
24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?
25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?
26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?
28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?
29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?

30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?
32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?
33. ¿Cuál sería su opinión en cuánto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?
34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?

Resultados de las entrevistas

Entrevista 1: David Carballo Picado

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se busca conocer. El proceso de encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Cuestionario para entrevistas:

Información del funcionario entrevistado:

Empresa	Mobilize.net SA
Área	Desarrollo de software
Entrevistado	David Carballo Picado
Puesto de desempeño	Desarrollador de Software
Fecha y hora	05 de enero 2019 14:10

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?
No
2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
Más de 10 años
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?
8 años de experiencia.

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
Sí, posee 5 años de experiencia
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?
Sí, utilizamos SCRUM

6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
Si, QA, Desarrollo, líder tecno, PO y tienen los roles que aplica SCRUM
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?
Si
8. ¿Cuál es la regularidad en que realizan pases a producción?
Al finalizar un reléase del proyecto.
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
Sí, posee beneficios como reuniones de retrospectiva ya que las mismas proponen mejoras dentro de los procesos.
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
SCRUM y VSTS
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
Con la metodología SCRUM
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
Si, porque ayuda a mejorar el proceso de desarrollo del mismo.
13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?
No, debido a que las tareas ya están registradas y asignadas por un líder técnico, existen dificultades para definir los tiempos para realizar un requerimiento, dificultad en la definición de tareas.
14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
Yo considero que están la cultura organizacional, la falta de Experiencia con metodologías ágiles y desde luego la desconfianza en los resultados de las metodologías ágiles.
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?
La mejora en el avance del desarrollo de los productos

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
Si

17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?
Si
18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
Falta de conocimiento de los resultados obtenidos y el miedo a la inversión económica que amerita el mismo.
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
Si
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
No se realizan planificación de pruebas.
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
Sí, algunas de las pruebas que realizan son pruebas de caja negra y testing.
22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
Sí, se trabaja con la herramienta Microsoft Test Manager
23. ¿En su representada realizan la documentación del resultado de las pruebas?
Si
24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?
No, las pruebas se ejecutan en los ambientes de desarrollo
25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?
Considero que entre las mejores están HASTQB (Latinoamérica) e ISO
26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?
En lo personal, yo me inclino por los programas de certificación de ISTQB.

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?
Si, en la empresa se utiliza Visual Studio Team Services
28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?
Creo que sería la gestión de cambios y el control de la labor diaria.

29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?
Se utiliza, Microsoft Test Manager.
30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?
Se puede mencionar que es poder documentar correctamente y la organización de las pruebas.

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?
Sí, es bastante importante, porque ayuda a facilitar la inclusión a las empresas en las prácticas de aseguramiento de la calidad del software y que puedan tener un proceso completo que se acople a la organización.
32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?
La disminución de incidentes al entregar el software terminado al cliente.
33. ¿Cuál sería su opinión en cuánto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?
Para tener claridad de todos los actores en el proceso el rol que van a desempeñar.
34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?
Todas las áreas que interactúen con el proceso de desarrollo del software como lo que son las áreas de planeación.

Entrevista 2: Ingeniera Francia Sevilla Escalante

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se busca conocer. El proceso de encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Cuestionario para entrevistas:

Información del funcionario entrevistado:

Empresa	Novatesting
Área	Aseguramiento de Calidad
Entrevistado	Ingeniera Francia Sevilla Escalante
Puesto de desempeña	Gerente de Aseguramiento de calidad
Fecha y hora	13/01/2019 11:34 pm

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?

Si

2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
21 años
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?
6 años

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
Sí, tengo 6 años
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?
Si, utilizo SCRUM y KAMBAN
6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
Si, SCRUM MASTER, PRODUCT OWNER, SCRUM TEAM
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?

Si

8. ¿Cuál es la regularidad en que realizan pases a producción?
Mes y Medio
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
Sí, los beneficios son la retrospectiva constante.
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
Basados en priorización, comunicación y trabajo en equipo
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
Al día a día se lleva un control de los incidentes con el fin de que los Scrum Master puedan tener visibilidad y resolver impedimentos
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
Sí, porque la organización, estructura, comunicación y trabajo en equipo
13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?

Si

14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
La cultura organizacional, la resistencia al cambio dentro de la empresa, el apoyo de los niveles más altos de la empresa, la falta de colaboración por parte de los clientes, la desconfianza en los resultados de las metodologías ágiles, además de la mayor problemática para que un proceso sea más ágil está localizado en la etapa final con los usuarios que no comprenden la importancia de desarrollar el SW más rápido y generan atrasos a los miembros de los equipos. Además, las metodologías ágiles no define un proceso para las solicitudes especiales que ingresan al área de ingeniería del software y que son trabajadas sin planificación dentro de un ciclo de trabajo.
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?
Comprender la gran necesidad de la demanda actual de los clientes, donde todos los usuarios buscan que las cosas sean más rápidas

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
Si
17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?
Si
18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
Esto ya no es solo un puesto o un rol, esto es una necesidad que el mercado de la informática está creciendo, hacer cada día las cosas mejor y con eficiencia, con el fin de entregar un producto de mejor calidad pensada completamente en los clientes.
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
Si
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
Si
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
Si, se realizan pruebas manuales y automatizados.
22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
Si, la herramienta utilizada es Test Manager de Microsoft

23. ¿En su representada realizan la documentación del resultado de las pruebas?
Si
24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?
Si, en ambientes de QA
25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?
ISTQB Alemania
26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?
ISTQB Fundamentals

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?
Si, se utiliza la herramienta Azure Devops.
28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?
Si, se utiliza la herramienta para centralizar las necesidades por medio de HU, que posterior serán trabajadas y aseguradas.
29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?
Microsoft Test manager
30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?
Tener el trabajo estructurado, ordenado, centralizar las necesidades, realizar pruebas de regresión.

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?
Claro que si ya que el trabajo planificado lleva una construcción correcta.
32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?
La inseguridad en los procesos, los respaldos documentales son de los aportes.

33. ¿Cuál sería su opinión en cuánto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?

Esto es primordial para garantizar que el software creado es software asegurado.

34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?

Clientes, Análisis de requerimientos, aseguramiento de calidad y usuarios final, el análisis de requerimiento es un aspecto muy importante ya que es el insumo base para la gestión de calidad.

Entrevista 3: Julio Arturo Escalante Abarca

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se busca conocer. El proceso de encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Cuestionario para entrevistas:

Información del funcionario entrevistado:

Empresa	Novacomp
Área	Ingeniería de sistemas
Entrevistado	Julio Arturo Escalante Abarca
Puesto de desempeño	Desarrollador
Fecha y hora	03 Enero 2018

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?
No
2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
Más de 10 años.
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?
6 años de experiencia

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
Sí, tengo 3 años.
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?
Sí, se utiliza Scrum (Lo utilizan los clientes donde ha estado asignado).
6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
Sí, los roles son: QA, Desarrollo, líder tecno, PO, clientes y tienen todos los roles de SCRUM.
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?
No
8. ¿Cuál es la regularidad en que realizan pases a producción?
Una vez al mes en el actual cliente, en algunos casos nulos.
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
No
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
Lo administra SCRUM
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
Se gestiona mediante SCRUM.
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
Sí, por el control que se logra llevar.

13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?
Si
14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
La cultura organizacional, la falta de experiencia con metodologías ágiles entre el personal y desde luego el miedo que tiene el personal ante cambios nuevos.
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?
La mejora en el control del proceso es el punto más importante, lograr observar el avance.

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
Si
17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?
Si
18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
EL tiempo y los recursos económicos, además de agregar la fase de QA aumenta en un tiempo adicional que son tiempos no contemplados.
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
Si
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
Si
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
Si, algunas son pruebas de caja negra, pero no se aplican pruebas de carga
22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
Si, la herramienta es Test Manager
23. ¿En su representada realizan la documentación del resultado de las pruebas?
Si
24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?

No

25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?
Novacom y Grupo Babel que están respaldados bajo los estándares propuestos por ISTQB.
26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?
Programas aplicados por UNED, CENFOTEC, FIDELITAS, además de programas como lo son los impartidos por CENFOTEC que son programas de certificación ISTQB, los cuales incluyen un curso de preparación.

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?
Si, la herramienta utilizada es Team Foundation Server, bitbucket, también utilizamos herramientas como Jira, Confluence
28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?
Si, una ventaja es mayor control de versiones cambios
29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?
Test Manager de Microsoft
30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?
Control de la ejecución de las pruebas, el ligar las pruebas con el desarrollo ya a nivel automatizado.

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?
Si es importante, en el caso de lograr que las empresas lo implementen, es importante que las guías lleven una capacitación o una inducción de personal experto.
32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?
Radica en que se tenga un ambiente de pruebas aislado, llevando un control de los cambios que se realizan, el poder detectar que cambios colaterales puede tener un cambio dentro de la misma herramienta que se está desarrollando.

33. ¿Cuál sería su opinión en cuánto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?

La importancia es llevar el orden y poder mejorar el control sobre los errores.

34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?

Implementaría un departamento de gestión de la configuración que controle el trabajo de Calidad de software para que ayude a definir el área de pruebas manejando el control de cambios.

Entrevista 4: Carlos Ruiz Campos

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se busca conocer. El proceso de encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Questionario para entrevistas:

Información del funcionario entrevistado:

Empresa	Grupo Babel
Área	Aseguramiento del Software
Entrevistado	Carlos Ruiz Campos
Puesto de desempeña	Analista de Calidad
Fecha y hora	14 enero, 2019 8:04am

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?
Si
2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
16 años
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?
2 años

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
Sí, tengo 2 años
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?
Si, utilizo SCRUM
6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
Si, Product Owner, SCRUM Master, SCRUM Team
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?
Si
8. ¿Cuál es la regularidad en que realizan pases a producción?
Aproximadamente cada mes
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
No
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
Administrando las tareas con la herramienta Visual Studio Team Services
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
De igual forma ingresándolos en el VSTS, y exponiéndolos en la reunión diaria al SCRUM Master
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
Sí, porque nos ha permitido trabajar de manera eficiente, ordenada, y ser más auto didactas y auto gestionables dentro de la organización.

13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?
Si
14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
Creemos que es la resistencia al cambio dentro de la empresa, el tema del enfoque cultural dentro de la empresa, la inexperiencia con metodologías ágiles entre el personal, el interés y apoyo de los niveles más altos de la empresa, el tiempo de transición muy largo y desde luego falta de información sobre las metodologías ágiles, el manejo del trabajo no planificado.
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?
La principal personalmente es lanzar mejores productos/servicios al mercado ya que permite que todas las personas participen en el desarrollo de un producto y se sientan cómodas trabajando, a gusto con lo que hacen y con cómo lo hacen, así como con una interacción constante y fluida.

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
Si
17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?
Si
18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
Una de las principales es la resistencia al cambio, algunas personas llevan mucho tiempo trabajando de cierta manera, y adaptar una metodología ágil provoca miedo y desconfianza al inicio del proceso, abonado también al poco conocimiento de marco de trabajo una vez iniciado.
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
Si
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
Si
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
Si, totalmente

22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
Si, la herramienta es Visual Studio Team Services
23. ¿En su representada realizan la documentación del resultado de las pruebas?
Si
24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?
Si
25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?
ISTQB
26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?
ISTQB Foundation Level y la carrera de Licenciatura con Énfasis en la Calidad del Software de la Universidad Fidelitas.

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?
Si, la herramienta es Visual Studio Team Services
28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?
El manejo de versiones, ramas y sectores de código para tener una integración continua en el proceso de construcción del software.
29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?
Microsoft Test Manager, Visual Studio, Selenium, Appium, Apache JMeter
30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?
Ayudan al seguimiento, administración y gestion de las pruebas, además de almacenar evidencia de los procesos de certificación de los sistemas.

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?

La transparencia de la creación y manejo de las pruebas dentro del proceso, el orden y seguridad que esto implica dentro del equipo de construcción. También la visibilidad que expone el tener las pruebas en una herramienta de gestión ante todo el equipo involucrado, y el guardado de evidencias luego de que el proceso haya concluido.

32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?

Ayudaría sin dudas en el proceso de inicio, de integración y de inyección de la organización a la metodología, creo personalmente que este paso es el que más complicaciones puede tener ya que las empresas temen cambiar totalmente su marco de trabajo en algunos casos, y esto debilita la idea de que las metodologías ágiles se integren dentro del proceso de una empresa como tal.

33. ¿Cuál sería su opinión en cuanto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?

Principalmente para no perder el enfoque en el camino, algunas veces el asegurador contempla más objetivos de los cuales debería y eso hace que el proceso del asegurador se vuelva más complejo sin razón.

34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?

Áreas como la planeación de los proyectos, desarrollo y el área de producción. Donde estos deben contemplar el nuevo proceso de aseguramiento y el tener claro cómo funciona dentro del marco.

Entrevista 5: Maickol Salamanca Arias

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se conocer. El proceso de encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Questionario para entrevistas:

Información del funcionario entrevistado:

Empresa	Empresa Consorcio Sysco Eprom
Área	Ingeniería de sistemas
Entrevistado	Maickol Salamanca Arias
Puesto de desempeña	Develop
Fecha y hora	15 Diciembre 2018

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?
No
2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
Más de 30 años
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?
8 años de Experiencia

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
Sí, tengo 6 años.
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?
Sí, utilizamos SCRUM
6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
Sí, algunos roles son: Asegurador, Desarrollador, líder técnico, Project Manager y tienen todos los roles de SCRUM.
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?
Si
8. ¿Cuál es la regularidad en que realizan pases a producción?
Cada 90 días.
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
Sí, algunos beneficios son: Reuniones de seguimiento y retrospectiva.
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
Lo administra Scrum y la herramienta ALM (VSTS)
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
Se gestiona mediante SCRUM
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
Sí, porque permite llevar un buen control. Pero existe la dificultad de poder relacionar el pesaje de las necesidades con los tiempos definidos para las actividades.

13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?
Si
14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
Siempre existe el cambio de cultura a nivel de empresa, la falta de experiencia con las metodologías ágiles esto lleva a que la gente tenga miedo al cambio.
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?
La mejora en el control del proceso es el punto más importante, lograr observar el avance.

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
Si
17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?
Si
18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
EL tiempo y los recursos económicos, además de agregar la fase de QA aumenta en un tiempo adicional que son tiempos no contemplados.
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
Si
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
Si
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
Pruebas de caja negra, pero sin aplicar pruebas de carga
22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
Si, desde luego que utilizamos Microsoft Test Manager.
23. ¿En su representada realizan la documentación del resultado de las pruebas?
Si
24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?

No

25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?
Novacom y Grupo Babel que están respaldados bajo los estándares propuestos por ISTQB
26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?
Programas aplicados por UNED, CENFOTEC, FIDELITAS, además de programas como lo son los impartidos por CENFOTEC que son programas de certificación ISTQB, los cuales incluyen un curso de preparación

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?
Si, la herramienta utilizada es Team Foundation Server y bitbucket
28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?
Si, y la ventaja es el control de versiones
29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?
Microsoft Test Manager
30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?
Control de la ejecución de las pruebas, el ligar las pruebas con el desarrollo ya a nivel automatizado.

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?
Si es importante, en el caso de lograr que las empresas lo implementen, es importante que las guías lleven una capacitación o una inducción de personal experto.
32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?
Radica en que se tenga un ambiente de pruebas aislado, llevando un control de los cambios que se realizan, el poder detectar que cambios colaterales puede tener un cambio dentro de la misma herramienta que se está desarrollando.

33. ¿Cuál sería su opinión en cuánto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?

La importancia es llevar el orden y poder mejorar el control sobre los errores.

34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?

Implementaría un departamento de gestión de la configuración que controle el trabajo de Calidad de software para que ayude a definir el área de pruebas manejando el control de cambios.

Entrevista 6: María Fernanda Salas

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se busca conocer. El proceso de encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Cuestionario para entrevistas:

Información del funcionario entrevistado:

Empresa	Grupo Babel
Área	Aseguramiento de calidad
Entrevistado	María Fernanda Salas
Puesto de desempeño	Analista de calidad
Fecha y hora	10 Enero 2019 15:30

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?

No

2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
17 años
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?
6 años

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
Si poseo 4 años
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?
Si, utilizamos SCRUM
6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
Si, los roles utilizados son: SCRUM master, Product Owner, team
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?
Si
8. ¿Cuál es la regularidad en que realizan pases a producción?
6 Meses
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
No
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
Por medio de asignaciones de cargas de trabajo equitativas entre los miembros del equipo.
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
Buscando soluciones de implementación pronta para que no afecten los tiempos estimados en los proyectos.
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
Sí, porque se detectan mejor los puntos de mejoras a los proyectos
13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?

Si

14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
Resistencia al cambio, manejar solicitudes ordenes de trabajo que no sean planificadas que sean vista con urgencia sobre el trabajo planificado, falta de colaboración por parte de los miembros del equipo y por parte de los clientes.
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?
Para realizar entregas continuas. Y poder atender solicitudes de cambios aun cuando ya no se esté en etapas tempranas del proyecto.

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
Si
17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?
Si
18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
Poco conocimiento del proceso por parte de los colaboradores acerca del proceso. No tener definidos las metas claras de cómo se vaya manejar la implementación en los proyectos.
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
Si
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
Si
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
Funcionales, no funcionales, integración, interfaz gráfica, carga y estrés
22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
Si, se utiliza Microsoft Test Manager
23. ¿En su representada realizan la documentación del resultado de las pruebas?
Si

24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?
Si, se realizan en QA
25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?
ISTQB
26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?
Cursos de certificación de CENFOTEC

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?
Si, se utiliza la herramienta Azure DevOps
28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?
Para el mejor manejo de recursos y costos. Cumplimiento y control de tiempos establecidos. Para mantener los controles de calidad actualizados.
29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?
Microsoft Test manager, Test plan
30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?
Mayor control de incidencias en los sistemas

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?
Si ya que muchas empresas no tienen la noción de cómo dar el primer pasó a conformar el proceso de aseguramiento. Y de esta forma no quedarían por fuera lineamientos importantes por falta de conocimiento. Y se podrían implementar muchas de las mejores prácticas de las metodologías.
32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?
El seguimiento a las metodologías agiles para encontrar puntos de mejora ya que en ciertos grupos de trabajo se pasan por alto algunos aspectos.

33. ¿Cuál sería su opinión en cuanto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?

Es importante ya que es una forma de organizar y controlar la calidad de los proyectos en curso. Mantener una adecuada gestión de incidencias y tener mayor conocimiento su estado actual.

34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?

Planificación de proyectos, equipos de desarrollo, gestión de la configuración

Entrevista 7: Anónimo

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se busca conocer. El proceso de encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Questionario para entrevistas:

Información del funcionario entrevistado:

Empresa	Anónima
Área	Ingeniería en Sistemas
Entrevistado	Anónimo
Puesto de desempeña	Desarrollador
Fecha y hora	08 Enero 2019 09:00

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?
No
2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
Más de 5 años
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?
4 años de Experiencia

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
Si, poseo 2 años
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?
Si, SCRUM
6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
Si, los roles son: Calidad y Desarrollo.
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?
Si
8. ¿Cuál es la regularidad en que realizan pases a producción?
Una vez completado lo requerido en el documento CR (Cambio de Requerimiento) o bien al finalizar el DRN (Definición Requerimiento de Negocio) en muchas ocasiones se entrega al completar algún escenario o historias de usuario, según lo requiera el cliente.
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
No
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
Lo administra el líder del proyecto junto con el equipo SCRUM
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
Se analizan en conjunto con el equipo de trabajo, y se toman las decisiones necesarias para avanzar de la mejor manera.
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
Si, posee mayor orden

13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?
Si
14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
Considero que están, cambio en cultura, resistencia al cambio, utilizado en proyectos muy complejos no es viable, desconfianza en los resultados, tiempo de transición muy largos, limita el presupuesto, falta de información sobre las metodologías ágiles, definición de los requerimientos y especialistas que definan los mismos
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?
La principal razón definitivamente es el control y orden que brinda con lo cual se logrará determinar a corto y mediano plazo avances en el proceso.

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
Si
17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?
Si
18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
Tiempo, recursos a implementar en el proyecto, una excelente revisión en la calidad.
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
Si
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
No
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
No
22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
Si, Microsoft Test Manager
23. ¿En su representada realizan la documentación del resultado de las pruebas?
Si

24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?
No
25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?
Desconozco cuál puede ser la organización más importante en estos momentos
26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?
He escuchado que la UNED posee programas especializados en esta rama, por ahí también escuché sobre la Universidad Fidelitas.

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?
Si, GITHUB, aunque he escuchado que JIRA es una herramienta muy fuerte que puede brindar muy buen rendimiento
28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?
Se lleva un mejor control en las versiones de cambios.
29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?
Microsoft Test Manager
30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?
Mejor control de la realización de las pruebas efectuadas, también se puede tener enlazado las pruebas con el desarrollo y monitorear este, o bien realizar cambios pequeños en una y llevar el control de lo efectuado.

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?
Bastante, ya que ayudaría muchos al personal en las empresas a poder tener un estándar en la implementación de las metodologías en los proyectos, claro está que deberá estar amarrado a una capacitación adecuada para que el personal abarque todas las dudas que pueda surgir.
32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?

Es importante contar con un adecuado ambiente de pruebas, ojalá separado del departamento de desarrollo, dónde al igual que el departamento de desarrollo lleve un control de cambios efectuados, esto para poder monitorear un posible impacto y no tener que devolverse mucho.

33. ¿Cuál sería su opinión en cuánto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?

Como bien lo dice la palabra, esto mantendrá estar seguros del producto que se estará entregando, con ello el orden y control sobre los posibles errores encontrados durante su desarrollo serán mermados día con día, con lo cual se logrará realizar a lo largo del tiempo proyectos con mejor calidad, basados claro está en las experiencias y lecciones aprendidas durante los desarrollos anteriores.

34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?

Un buen departamento de calidad de software que permita medir y determinar a tiempo alguna área a mejorar, y en caso de encontrar errores que puedan perjudicar fuertemente la calidad del producto a entregar.

Entrevista 8: Luis Diego Fernández

Propuesta de guía para el aseguramiento del proceso de calidad de software para las empresas costarricenses que desarrollan software para consumo nacional.

Artefacto para entrevistas.

Objetivo: Conocer la opinión de personas involucradas en el proceso de desarrollo de software en relación con la función que desempeña en temas de metodología de desarrollo, procesos de aseguramiento y el uso de herramientas para administrar el ciclo de desarrollo.

Mecánica: Entrevista personal o mediante el envío de un correo con el formulario a funcionarios acordes con el perfil de opinión que se busca conocer. El proceso de encuesta se realizará mediante un sitio web que será habilitado para que las personas puedan acceder y completar los formularios.

Durante la entrevista se llenarán el documento con las opiniones resultado de la entrevista.

Perfil del funcionario a entrevistar:

- Empresas con más de 5 años en el mercado de desarrollo de software.
- Perfil de personas a entrevistar:
 - Desarrolladores
 - Aseguradores de calidad
 - Administradores de proyecto
 - Líderes técnicos
 - Gerentes del área de desarrollo de aplicaciones
- Áreas de interés: Procesos de desarrollo de software y aseguramiento de la calidad de software.

Questionario para entrevistas:

Información del funcionario entrevistado:

Empresa	UTN
Área	Dirección de TI
Entrevistado	Luis Diego Fernández
Puesto de desempeña	Dirección de TI y Soporte
Fecha y hora	15/01/19

1. ¿Ha estado familiarizado con una entrevista sobre metodologías de desarrollo?
Si
2. ¿Cuántos años en el mercado tiene la empresa para la cual trabaja actualmente?
10 años
3. ¿Cuántos años de experiencia posee en el área que desempeña actualmente?
15 años

Apartado de metodología del proceso

4. ¿Posee algún grado de conocimiento en las metodologías ágiles? Si su respuesta es sí, indicar por cuánto tiempo ha hecho uso de las metodologías ágiles sin importar si es en el orden personal o laboral.
Si, 2 años
5. ¿En la empresa para la que labora actualmente, utilizan alguna metodología Ágil? ¿Cuál?
Si, SCRUM
6. ¿Dentro de su empresa, los funcionarios manejan roles definidos para el desarrollo de las actividades? En caso de ser sí, cuáles serían esos roles
Si, algunos roles son: SCRUM master, desarrolladores.
7. ¿En su representada, durante el desarrollo del proyecto realizan entregas continuas del mismo?
Si
8. ¿Cuál es la regularidad en que realizan pases a producción?
Una vez cada tres días cuando hay desarrollo
9. ¿En su empresa, existe algún método para mejorar el proceso actual, bajo un concepto de mejora continua? Si la respuesta es afirmativa podría mencionar algunos beneficios.
No
10. ¿En su representada, como administran el trabajo pendiente durante el desarrollo del proyecto?
Dirección y planificación de trabajo hacia otras áreas
11. ¿Cómo controlan los impedimentos que pueden surgir durante el desarrollo del proyecto?
Se tercerizan desarrollos hacia los sistemas secundarios
12. ¿Cree usted que el equipo de trabajo vea positivamente la metodología aplicada? ¿Por qué?
Sí, porque permite entregas a tiempo de los productos
13. ¿En la empresa para la que labora realizan desglose y registro de tareas para planificar el trabajo?
Si

14. ¿Cuál considera usted, que puede ser un obstáculo que presente una organización al adoptar el uso de metodologías ágiles?
Presupuesto limitado, no hay apoyo de los altos niveles de la empresa, resistencia al cambio y cultura organizacional.
15. En su opinión ¿Cuál cree que es la principal razón para que una empresa adopte las metodologías ágiles?
Las metodologías ágiles permiten la simplificación y desarrollo de productos directamente derivados de los requerimientos expuestos en las historias de usuario en un tiempo determinado.

Apartado de aseguramiento de la calidad del Software

16. ¿En su representada, mantienen prácticas de aseguramiento de la calidad de software?
Si
17. ¿Actualmente durante el desarrollo del proyecto, poseen perfiles de profesionales con enfocados en el aseguramiento de la calidad de software?
Si
18. ¿Según su criterio, cuál considera que es una dificultad que puede presentar cualquier empresa al intentar incorporar procesos de aseguramiento de la calidad de software?
Falta de personal
19. ¿En la empresa para la cual trabaja, realizan pruebas formales durante el desarrollo de software?
Si
20. ¿Planifican las pruebas mediante un debido plan de pruebas, en la empresa para la que trabaja?
Si
21. ¿Usted tiene algún conocimiento del tipo de pruebas realizan?
Pruebas de caja negra
22. ¿En su representada, realizan documentación de los casos de pruebas? En caso de ser afirmativa la respuesta cuál es la herramienta utilizada
Si, las herramientas son Machotes creados por la dirección.
23. ¿En su representada realizan la documentación del resultado de las pruebas?
Si
24. ¿En su representada, las pruebas de aseguramiento de calidad del software, se realizan en ambientes designados exclusivamente para QA?
Si

25. ¿A criterio personal, cuál consideraría que es la organización más importante en temas de certificación de pruebas de software?

ISO

26. ¿Cuál cree que es el programa más utilizado a nivel nacional para aprender sobre pruebas de calidad de software?

SoapUI

Apartado de herramientas

27. ¿En su representada, utilizan alguna herramienta para la gestión del desarrollo de software?

Si, la herramienta utilizada es BaseCamp

28. En su opinión ¿Cuál es la principal ventaja al utilizar herramientas para la gestión del desarrollo de software?

Si, la ventaja es compartir archivos.

29. ¿Qué herramientas utilizan para el desarrollo de pruebas de software en su representada?

SoapUI

30. En su opinión ¿Cuál es la principal ventaja de utilizar una herramienta para la gestión de pruebas?

Automatización de las pruebas

Preguntas generales

31. ¿Considera usted que crear una guía que implemente las metodologías ágiles, así como el aseguramiento de la calidad de software y herramientas de ALM y testing, esto ayudaría a un mejor entendimiento de la integración de los procesos en el desarrollo de proyectos?

En nuestro caso al ser un ente público debemos incorporar una metodología basado en las Guías de la Contraloría general de la república, por lo tanto, es de suma importancia tenerla, usarla y actualizarla constantemente

32. ¿Cuáles son los factores más importantes que han detectado al implementar como mejora el proceso de aseguramiento de la calidad del software?

Control y seguridad de los desarrollos, diseño de los requerimientos, detalle de los mismos

33. ¿Cuál sería su opinión en cuánto al, porqué considera importante definir el proceso de aseguramiento de la calidad del software?

Generar software de calidad que venga a dar satisfacción al cliente final

34. ¿Qué áreas contemplaría en el proceso de cambio, al implementar el aseguramiento de la calidad?

Todas