**Gestión de Pólizas de Seguro**

**Plan de SQA**

**Versión 1.2**

**Historia de revisiones**

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 09/04/2018 | 1.0 | Creación del Plan | Juan Apari, Wilson Díaz, Alex Febre, Ana San Bartolomé, Jhancarlo Silva |
| 23/04/2018 | 1.1 | Gestión, Documentación, Estándares de Documentación y Estándares, prácticas, convenciones y métricas. | Juan Apari, Wilson Díaz, Alex Febre, Ana San Bartolomé, Jhancarlo Silva |
| 26/06/2018 | 1.2 | Estándar de Calidad del Proceso, además de Revisiones y auditorías | Juan Apari, Wilson Díaz, Alex Febre, Ana San Bartolomé, Jhancarlo Silva |

**Índice**

[1. Propósito 3](#_Toc518079273)

[1.1. Situación de la empresa, proyecto. 3](#_Toc518079274)

[1.2. Beneficios de este plan 3](#_Toc518079275)

[2. Gestión 4](#_Toc518079276)

[2.1. Organización 4](#_Toc518079277)

[2.2. Responsables del área SQA 7](#_Toc518079278)

[2.3. Actividades del aseguramiento de la calidad 7](#_Toc518079279)

[3. Documentación 8](#_Toc518079280)

[3.1. Documentación mínima requerida 8](#_Toc518079281)

[4. Estándares de Documentación 9](#_Toc518079282)

[4.1. Estándar de Programación Java (StandartCodigoJava.pdf) 9](#_Toc518079283)

[4.2. Estándar de Base de Datos(estandaresDBA.pdf) 9](#_Toc518079284)

[4.3. Estándar de diccionario de datos 9](#_Toc518079285)

[5. Estándares, prácticas, convenciones y métricas 9](#_Toc518079286)

[5.1. Estándar de Calidad del producto 9](#_Toc518079287)

[5.1.1. Modelo de Calidad del Producto - General 10](#_Toc518079288)

[5.1.2. Modelo de Calidad del Producto – detallado 10](#_Toc518079289)

[5.2. Estándar para Métricas de Calidad 13](#_Toc518079290)

[5.2.1. Lista de Métricas priorizadas 13](#_Toc518079291)

[5.2.2. Definición de Métricas de Calidad 14](#_Toc518079292)

[5.3. Estándar de Calidad del Proceso 19](#_Toc518079293)

[6. Revisiones y auditorías 1](#_Toc518079294)

[6.1. Objetivo 1](#_Toc518079295)

[6.2. Requerimientos mínimos 1](#_Toc518079296)

[6.3. Revisión de requerimientos 1](#_Toc518079297)

[6.4. Revisión de diseño preliminar 2](#_Toc518079298)

[6.5. Revisión de diseño crítico 2](#_Toc518079299)

[6.6. Revisión del Plan de Verificación & Validación 3](#_Toc518079300)

[6.7. Auditoría funcional 4](#_Toc518079301)

[6.8. Auditoría física 4](#_Toc518079302)

[6.9. Revisiones de gestión 4](#_Toc518079303)

[6.10. Revisión del Plan de gestión de configuración 5](#_Toc518079304)

[7. Verificación 5](#_Toc518079305)

[8. Reporte de problemas y acciones correctivas 5](#_Toc518079306)

[9. Herramientas, técnicas y metodologías 6](#_Toc518079307)

## Propósito

### Situación de la empresa, proyecto.

La situación de la empresa BIO SAC con respecto a la gestión de pólizas, presenta los siguientes problemas:

* El control de los pagos pendientes es inexacto, ya que a veces suele omitirse algunas pólizas con pagos por vencer o vencidos.
* El control del vencimiento de las pólizas no es preciso, ya que a medida que pasa el tiempo la cantidad de documentos incrementa.
* La información de las pólizas se escanea y se guardan en pdf, y para realizar las búsquedas es muy difícil si se tiene demasiada información
* Existe un riesgo de pérdida de información, ya que la información de gestión de pólizas se almacena en Excel
* Desconocimiento de la confidencialidad e integridad de la información registradas en las pólizas de seguro.
* El control de la información de los siniestros no es preciso, ya que a medida que pasa el tiempo la cantidad de incidencias reportados incrementa.
* Las ventas de pólizas que realiza el Master Service no tiene un incremento significativo, ya que no conoce qué pólizas podría ofrecer a sus clientes.

### Beneficios de este plan

Se propone desarrollar un sistema de gestión de pólizas que solucione los problemas descritos anteriormente, por lo cual tendrá el siguiente uso:

* Gestionar los pagos de las pólizas.
* Gestionar la información de las pólizas.
* Gestionar la información de los clientes y beneficiarios.
* Gestionar los siniestros reportados por los clientes y/o beneficiarios.
* Sugerir las pólizas de acuerdo a la necesidad del cliente.

## Gestión

### Organización

**Organigrama**

Manager

Project Manager

Developer

Functional Analyst

Quality Analyst

Tester

SQA Manager

Team SQA

|  |  |  |
| --- | --- | --- |
| **Responsables** | **Actividades** | **Impacto en el control de la calidad del software** |
| Manager | * Tomar decisiones de la alta gerencia como responsable de la empresa. | Alto |
| Project Manager | * Liderar el equipo responsable de alcanzar los objetivos del proyecto. * Controlar los recursos asignados al proyecto con el fin de cumplir con los objetivos establecidos. * Gestionar las restricciones del proyecto, como el alcance, cronograma, costo y calidad. | Alto |
| Functional Analyst | * Entender, establecer y formalizar los requerimientos del cliente. * Modelar, clasificar, priorizar y validar los requerimientos del cliente. * Analizar, especificar y verificar los requerimientos del cliente y/o usuarios. * Validar y obtener la aprobación de las definiciones del usuario comprometiendo a los usuarios involucrados. * Definir y generar los casos de prueba funcionales en función de las características del proyecto y del equipo del mismo | Medio |
| Developer | * Comprender el requerimiento de software. * Construir o adaptar clases y módulos de software. * Construir prototipos en fase de alcance técnico y funcional de los proyectos. * Realizar los programas de acuerdo a estándares (documentos funcionales, de arquitectura y comentarios en el código). * Participar en la revisión del diseño y construcción del software. | Medio |
| Quality Analyst | * Dar soporte durante la creación del Plan de Calidad de un Proyecto. * Alinear las actividades programadas del Plan de Calidad a las normas de Calidad ISO 9126. * Ejecutar las actividades diagramadas en el Plan de Calidad, revisando adherencia a los estándares. * Controlar la política de calidad de la empresa. * Realizar las auditorías internas de acuerdo a los Modelos de Calidad existentes. Además de ejecutar los checklists de las áreas de proceso. | Alto |
| Tester | * Diseñar plan de testing. * Definir los casos de prueba de calidad en base a los requisitos funcionales. * Gestionar el armado de ambientes y datos de prueba. * Ejecutar los casos de prueba. * Realizar la documentación de las pruebas e incidentes en la base a los defectos encontrados. | Medio |
| SQA Manager | * Definir políticas y objetivos de software en lo referido a calidad. * Tomar decisiones según lo establecido (políticas o normas) * Gestionar el desarrollo de herramientas para facilitar el proceso de SQA. | Alto |
| Team SQA | * Desarrollar las actividades de SQA planificadas para los proyectos. * Auditar, monitorear, evaluar e informar sobre las actividades de desarrollo. | Alto |

### Responsables del área SQA

|  |  |  |
| --- | --- | --- |
| **Roles** | **Responsabilidades** | **Cantidad** |
| Gestor QA | * Definir procesos y prioridades, reportando problemas. * Observar los cambios que se han realizado para una versión, analizar riesgos. * Desarrollar relaciones sólidas con los de negocio para comprender las necesidades del departamento de control de calidad y las herramientas. * Gestionar las Releases y los entornos, con el objetivo de ir probando los requerimientos que se van desarrollando. | 1 |
| Analista QA | * Detectar problemas en las primeras etapas, antes de empezar a crear. * Utilizar herramientas de analítica de datos para realizar un seguimiento de todos los fallos que se ha encontrado el cliente. * Participar durante todo el ciclo de vida de desarrollo, verificando que se está construyendo bajo lo previsto. * Definir los casos de prueba y requisitos de prueba. | 2 |
| Tester QA | * Definir el plan de pruebas para automatizar los test. * Programar los tests y ejecutarlos. * Colaborar en la creación de Test Unitarios y de Integración. | 1 |

### Actividades del aseguramiento de la calidad

|  |  |  |
| --- | --- | --- |
| **Actividad** | **Semanas** | **Rol** |
| Diseñar el organigrama del área QA | 2 | Gestor QA |
| Diseñar Modelo de Calidad | 3 | Gestor QA |
| Ponderar Características | 1 | Analista QA1 |
| Ponderar Sub características | 1 | Analista QA1 |
| Priorizar las Métricas de Calidad | 1 | Analista QA1 |
| Definir Métricas de Calidad | 1 | Analista QA2 |
| Diseñar Casos de prueba | 2 | Analista QA2 |
| Ejecutar Casos de prueba | 1 | Tester QA |
| Diseño de estándares de documentos | 1 | Analista QA2 |

## Documentación

### Documentación mínima requerida

La documentación mínima es la requerida para asegurar que la implementación logrará satisfacer los requerimientos.

* + 1. **Plan del Proyecto**

En el plan del proyecto contiene la siguiente información:

* Información del proyecto
* Propósito y justificación del proyecto
* Descripción del proyecto y entregables
* Requerimientos de alto nivel
* Objetivos del proyecto y del producto
* Restricciones del proyecto
* Riesgos iniciales de alto nivel
* Cronograma de hitos principales
* Presupuesto estimado
* Lista de Interesados (stakeholders)
* Requisitos de aprobación del proyecto
* Asignación del gerente de proyecto y nivel de autoridad
  + 1. **Documento de Negocio**

En el documento de negocio se describe los cinco procesos que realizan actualmente los trabajadores de la empresa aseguradora para ser incluida en el desarrollo de las soluciones del software. El documento provee una visión global del enfoque de desarrollo propuesto.

Los procesos pertenecientes al negocio son:

* Gestionar Clientes
* Gestionar Pólizas
* Gestionar Pagos
* Sugerir pólizas
* Gestionar Siniestros

## Estándares de Documentación

### Estándar de Programación Java (StandartCodigoJava.pdf)

El objeto del presente documento es el establecimiento de los estándares o convenciones de programación empleados en el desarrollo de software sobre la plataforma Java. Este modelo de programación está basado en los estándares recomendados por Sun Microsystems, que han sido difundidos y aceptados ampliamente por toda la comunidad Java, y que han terminado por consolidarse como un modelo estándar de programación de facto.

Estas normas son muy útiles por muchas razones, entre las que destacan:

* Facilitan el mantenimiento de una aplicación. Dicho mantenimiento constituye el 80% del coste del ciclo de vida de la aplicación.
* Permite que cualquier programador entienda y pueda mantener la aplicación. En muy raras ocasiones una misma aplicación es mantenida por su autor original.
* Los estándares de programación mejoran la legibilidad del código, al mismo tiempo que permiten su comprensión rápida.

### Estándar de Base de Datos(estandaresDBA.pdf)

Uno de los instrumentos que facilitan esta tarea es la adopción de estándares de diseño de bases de datos. El uso de estos estándares tiene innumerables ventajas, entre ellas:

* Asegurar la legibilidad del modelo de datos, inclusive para personas que no están relacionadas con el ambiente informático, en etapas de análisis y diseño
* Facilitar la portabilidad entre motores de bases de datos, plataformas y aplicaciones;
* Facilitar la tarea de los programadores en el desarrollo de los sistemas.

### Estándar de diccionario de datos

El diccionario de datos aporta el registro de los identificadores únicos, representaciones de valor, etiquetas de atributos y multiplicidad de valores de la información.

## Estándares, prácticas, convenciones y métricas

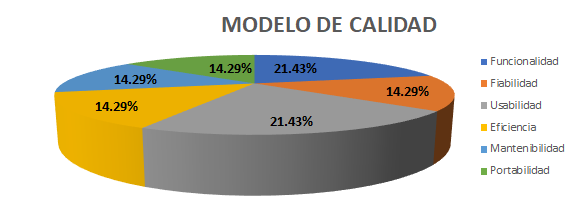
### Estándar de Calidad del producto

ISO 9126

#### Modelo de Calidad del Producto - General

La ISO 9126 es un estándar internacional para la evaluación del Software.

La normativa define seis características de la aplicación, estas seis características son divididas en un número de sub- características, las cuales representan un modelo detallado para la evaluación de cualquier sistema informático.



|  |  |  |
| --- | --- | --- |
| **Característica** | **Peso %** | **Justificación** |
| **Funcionalidad** | 21.43% | En la plataforma web los cálculos matemáticos y la seguridad de la información es una prioridad que se va tomar en cuenta. |
| **Usabilidad** | 21.43% | En la plataforma web, es importante que el usuario pueda adaptarse sin ningún problema en el uso y entendimiento de la plataforma. |
| **Fiabilidad** | 14.29% | La plataforma web debe tener la capacidad de poder recuperarse ante fallos presentados y tener también poder recuperarse se forma automática. |
| **Eficiencia** | 14.29% | El acceso de usuarios simultáneamente a la plataforma web sin tener que sacrificar los recursos del servidor |
| **Mantenibilidad** | 14.29% | Se podrá agregar nuevas funcionalidades a la plataforma web ante nuevos requisitos, sin inconvenientes. |
| **Portabilidad** | 14.29% | La plataforma web debe tener la capacidad de poder ser utilizada en diferentes resoluciones de pantalla y dispositivos sin ningún problema. |

#### Modelo de Calidad del Producto – detallado

**FUNCIONALIDAD**

|  |  |  |
| --- | --- | --- |
| **Sub característica** | **Peso %** | **Justificación** |
| Precisión | 37.50% | Es importante considerar esta su característica porque el sistema hará cálculos matemáticos como: intereses o similares. |
| Seguridad | 37.50% | La seguridad es un punto importante en aplicaciones web, puesto que almacenará información sensible de los clientes. |
| Cumplimiento de la funcionalidad | 25.00% | Se aplicará estándares de documentación (estándar de programación y estándar de base de datos). |

**USABILIDAD**

|  |  |  |
| --- | --- | --- |
| **Sub característica** | **Peso %** | **Justificación** |
| Entendimiento | 30.00% | Al ser una plataforma web es necesario que sea fácil de entender y los clientes puedan adaptarse al sistema. |
| Aprendizaje | 30.00% | La interfaz de usuario será intuitiva, por lo que el uso de la plataforma web será fácil de aprender. |
| Operabilidad | 20.00% | Es importante que el usuario pueda operar de forma fácil y rápida el sistema sin la necesidad de estar siempre consultando los manuales. |
| Atracción | 20.00% | La plataforma web debe de atraer al usuario a través de ventanas amigables e intuitivas para realizar cualquier tipo de operación. |

**FIABILIDAD**

|  |  |  |
| --- | --- | --- |
| **Sub característica** | **Peso %** | **Justificación** |
| Recuperabilidad | 50.00% | El tiempo de recuperación de la plataforma web ante una operación de usuario no superará los 3 segundos. |
| Tolerancia a errores | 50.00% | La plataforma web debe seguir funcionando correctamente en caso de una falla presentada. |

**EFICIENCIA**

|  |  |  |
| --- | --- | --- |
| **Sub característica** | **Peso %** | **Justificación** |
| Comportamiento de tiempo | 60.00% | Se tendrá un buen performance de la plataforma web cuando ingresen usuarios simultáneamente. |
| Utilización de recursos | 40.00% | El sistema debe ejecutar sus procesos sin sacrificar los recursos del servidor y los del usuario. |

**MANTENIBILIDAD**

|  |  |  |
| --- | --- | --- |
| **Sub característica** | **Peso %** | **Justificación** |
| Capacidad de ser analizado | 28.57% | Permite diagnosticar las deficiencias o causas de fallos en el software, o identificar las partes a modificar. |
| Capacidad de ser cambiado | 28.57% | La plataforma web puede ser modificada sin alterar la funcionalidad del sistema. |
| Capacidad de ser probado | 42.86% | Permite establecer criterios de prueba con la que se pueden llevar a cabo las pruebas para determinar si se cumplen dichos criterios. |

**PORTABILIDAD**

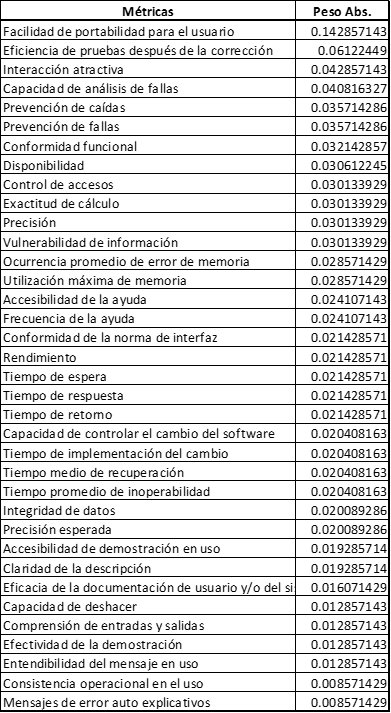
|  |  |  |
| --- | --- | --- |
| **Sub característica** | **Peso %** | **Justificación** |
| Adaptabilidad | 100% | La plataforma web es fácil de ser adaptado en diferentes entornos determinados de hardware o software. |

### Estándar para Métricas de Calidad

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente.

El objetivo principal de la ingeniería del software es producir un producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben utilizar mediciones que evalúen la calidad del análisis y los modelos de desafío, el código fuente, y los casos de prueba que se han creado al aplicar la ingeniería del software. Para lograr esta evaluación de la calidad en tiempo real, el ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad.

### Lista de Métricas priorizadas



### Definición de Métricas de Calidad

|  |  |
| --- | --- |
| **Nombre de la métrica** | Facilidad de portabilidad para el usuario |
| **Código** | M001 |
| **Característica** | Portabilidad |
| **Sub característica** | Adaptabilidad |
| **Propósito** | ¿Puede el usuario o responsable de mantenimiento adaptar fácilmente el software al entorno? |
| **Método de aplicación** | Observar el comportamiento del  usuario o del responsable de mantenimiento  cuando el usuario está tratando de adaptar el  software al ambiente de operación. |
| **Medición fórmula y elementos medibles** | T = Suma de tiempo de operación del usuario  empleado en completar la adaptación del  software al ambiente del usuario, cuando el  usuario intenta instalar o cambiar la configuración. |
| **Tipo de medida** | T=Tiempo |
| **Interpretación del valor medido** | 0 < T  El menor es lo mejor. |
| **Tipo de escala** | Absoluta |

|  |  |
| --- | --- |
| **Nombre de la métrica** | Control de Acceso |
| **Código** | M002 |
| **Característica** | Funcionalidad |
| **Sub característica** | Seguridad |
| **Propósito** | ¿Cuán controlable es el acceso al sistema? |
| **Método de aplicación** | Contar el número de operaciones ilegales  detectadas en comparación con el número de operaciones ilegales permitidas en la  especificación. |
| **Medición fórmula y elementos medibles** | X = A / B  A = Número de tipos de  operaciones ilegales  detectadas.  B = Número de tipos de  operaciones ilegales  permitidas en la  especificación |
| **Tipo de medida** | X = Cantidad /  Cantidad  A = Cantidad  B = Cantidad |
| **Interpretación del valor medido** | 0 < = X < = 1  Lo más cerca de  1,0 es lo mejor |
| **Tipo de escala** | Absoluta |

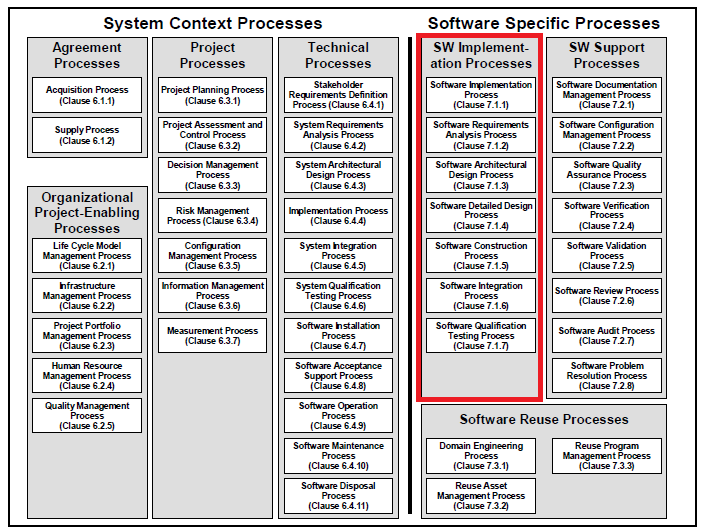
|  |  |
| --- | --- |
| **Nombre de la métrica** | Conformidad funcional |
| **Código** | M003 |
| **Característica** | Funcionalidad |
| **Sub característica** | Cumplimiento de la funcionalidad. |
| **Propósito** | ¿Cuán conforme es la funcionalidad del producto para aplicar regulaciones, normas y convenciones? |
| **Método de aplicación** | Contar el número de ítems requeridos de conformidad que han sido hallados y compararlos con el número de ítems de conformidad requeridos en la especificación. |
| **Medición fórmula y elementos medibles** | X = 1 - A / B  A = Número de ítems con conformidad funcional especificados que no han sido implementados durante la prueba.  B = Número total de ítems de conformidad funcional especificados. |
| **Tipo de medida** | X = Cantidad /Cantidad  A = Cantidad  B = Cantidad |
| **Interpretación del valor medido** | 0 < = X < = 1  Lo más cerca de 1,0 es lo mejor |
| **Tipo de escala** | Absoluta |

|  |  |
| --- | --- |
| **Nombre de la métrica** | Prevención de fallas |
| **Código** | M004 |
| **Característica** | Fiabilidad |
| **Sub característica** | Tolerancia a errores |
| **Propósito** | ¿Cuántos tipos de errores fueron causados bajo el control para evitar fallas críticas y serias? |
| **Método de aplicación** | Contar el número de tipos de errores prevenidos y comparar con el número de tipos de errores a ser considerados. |
| **Medición fórmula y elementos medibles** | X = A / B  A = Número de fallas críticas y serias previstas ocurridas contra casos de prueba de tipos de errores.  B = Número de tipos de errores de casos de prueba ejecutados (falla casi causada) durante la prueba |
| **Tipo de medida** | A = Cantidad  B = Cantidad  X = Cantidad / Cantidad |
| **Interpretación del valor medido** | 0 < = X < = 1  El valor más cercano a 1,0 es lo mejor, cuando el usuario puede evitar más a menudo fallas críticas o serias. |
| **Tipo de escala** | Absoluta |

|  |  |
| --- | --- |
| **Nombre de la métrica** | Rendimiento |
| **Código** | M005 |
| **Característica** | Eficiencia |
| **Sub característica** | Comportamiento en el tiempo |
| **Propósito** | ¿Cuántas tareas pueden ejecutarse  satisfactoriamente en un determinado intervalo  de tiempo? |
| **Método de aplicación** | Calibrar cada tarea de acuerdo a la prioridad preestablecida. Iniciar varias tareas de trabajo. Medir el tiempo que toma completar la operación de las tareas medidas. Mantener un registro de cada intento. |
| **Medición fórmula y elementos medibles** | X = A / T.  A = Número de tareas completadas  T = Intervalo de tiempo de observación. |
| **Tipo de medida** | A = Cantidad  T = Tiempo  X = Cantidad / Tiempo |
| **Interpretación del valor medido** | 0 < X  El mayor valor es  lo mejor. |
| **Tipo de escala** | Ratio |

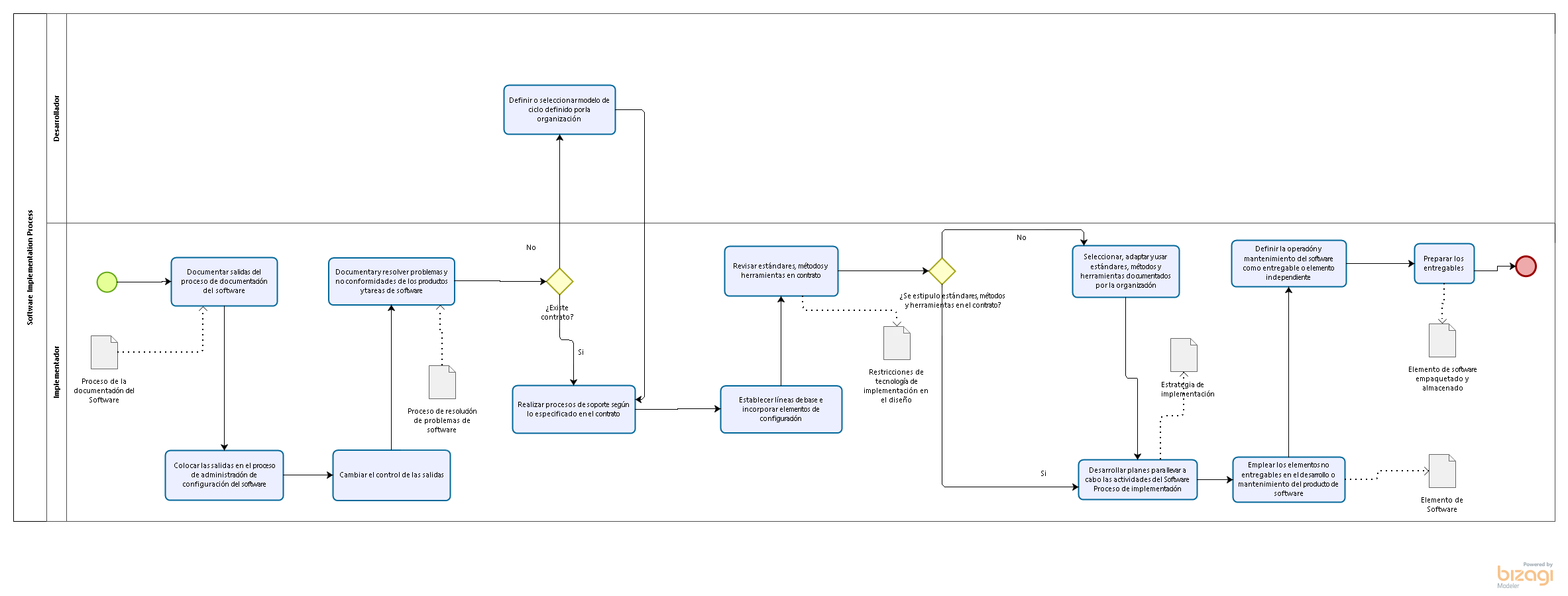
### Estándar de Calidad del Proceso

Se aplicará como estándar de calidad del proceso la **ISO 12207**. De los Grupos de ciclo de vida se trabajará del Software Specific Processes, el SW Implementation Processes:

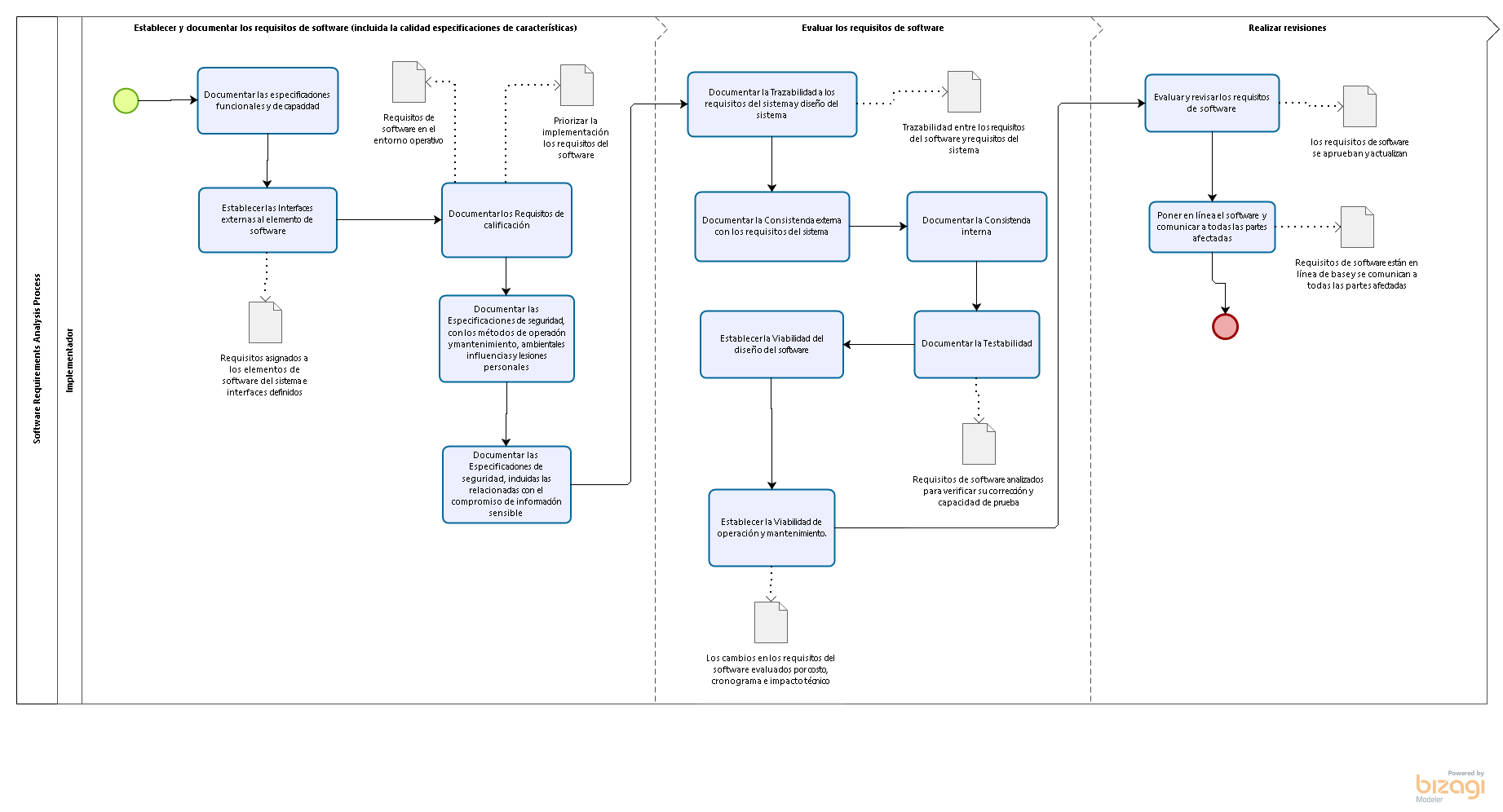


**Grupos de ciclo de vida de proceso**

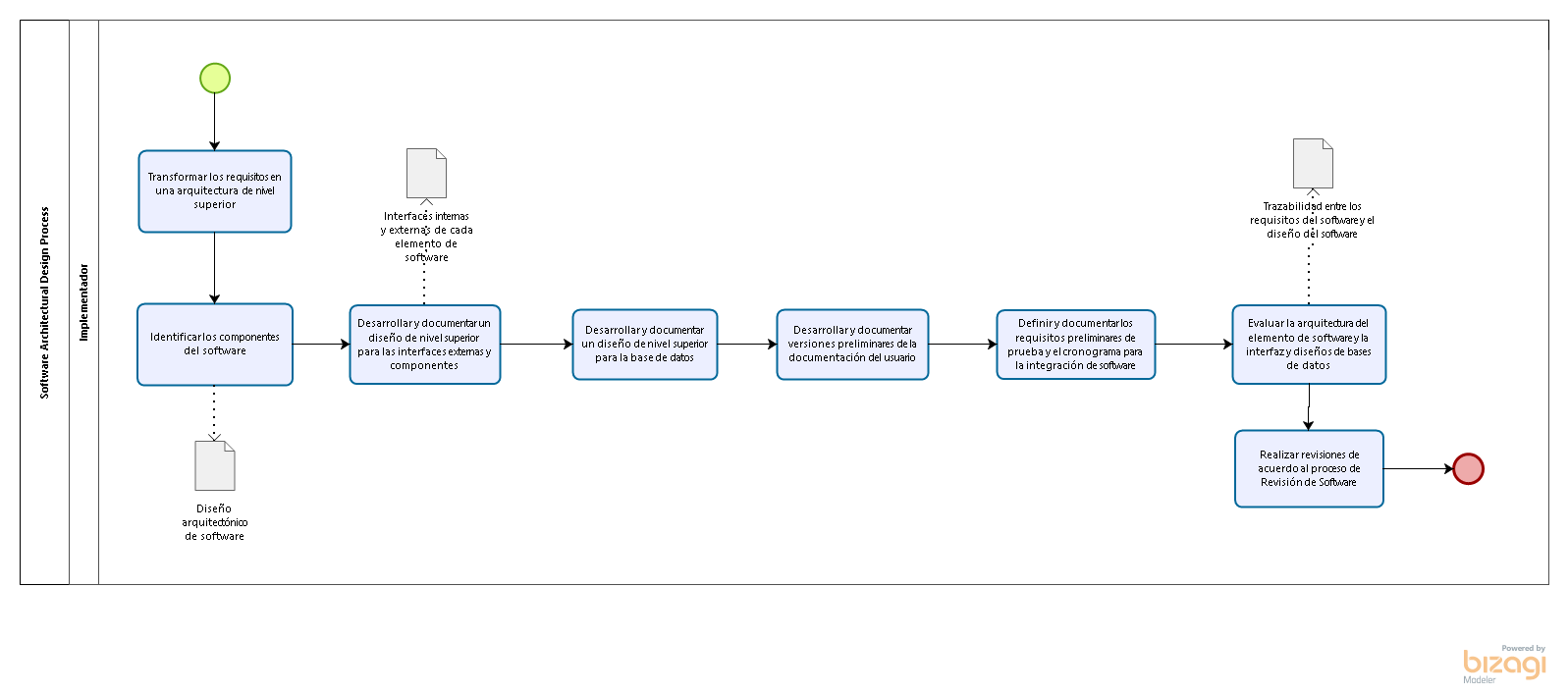
**Clause 7.1.1. Software Implementation Process**



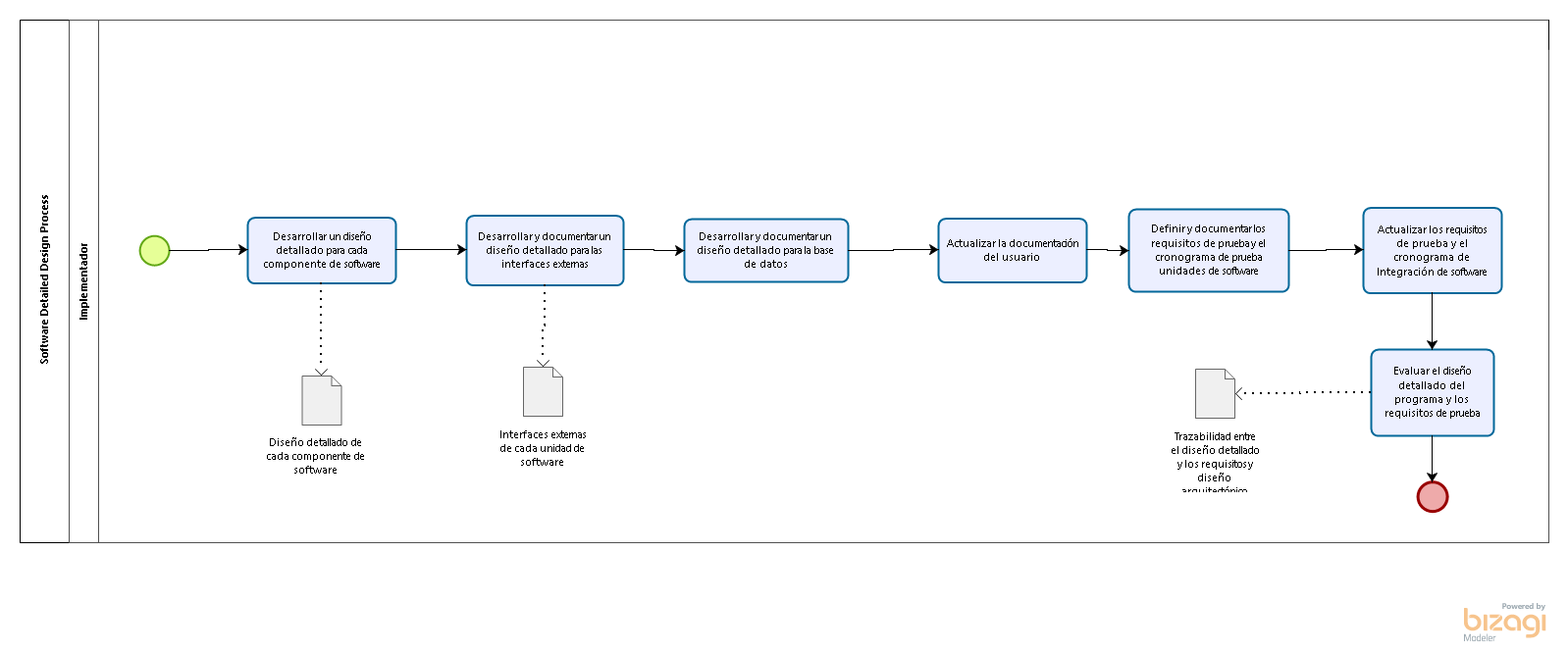
**Clause 7.1.2. Software Requirements Analysis Process**



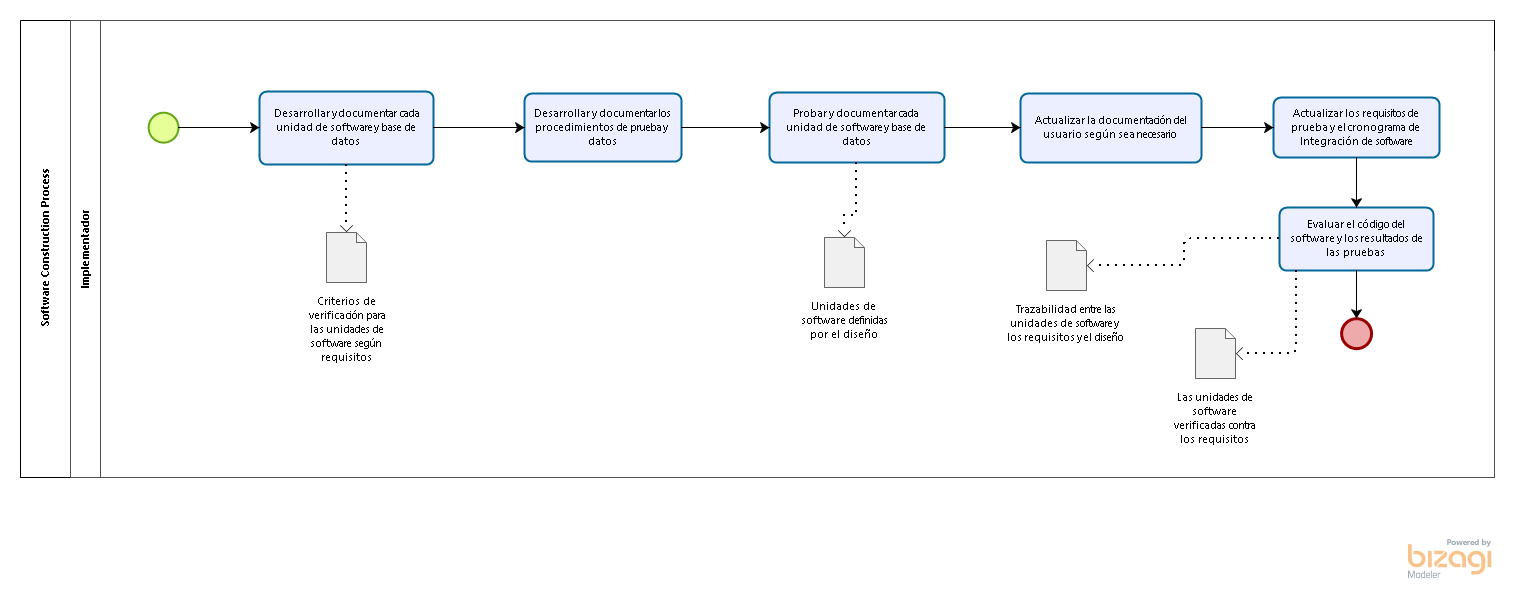
**Clause 7.1.3. Software Architectural Design Process**



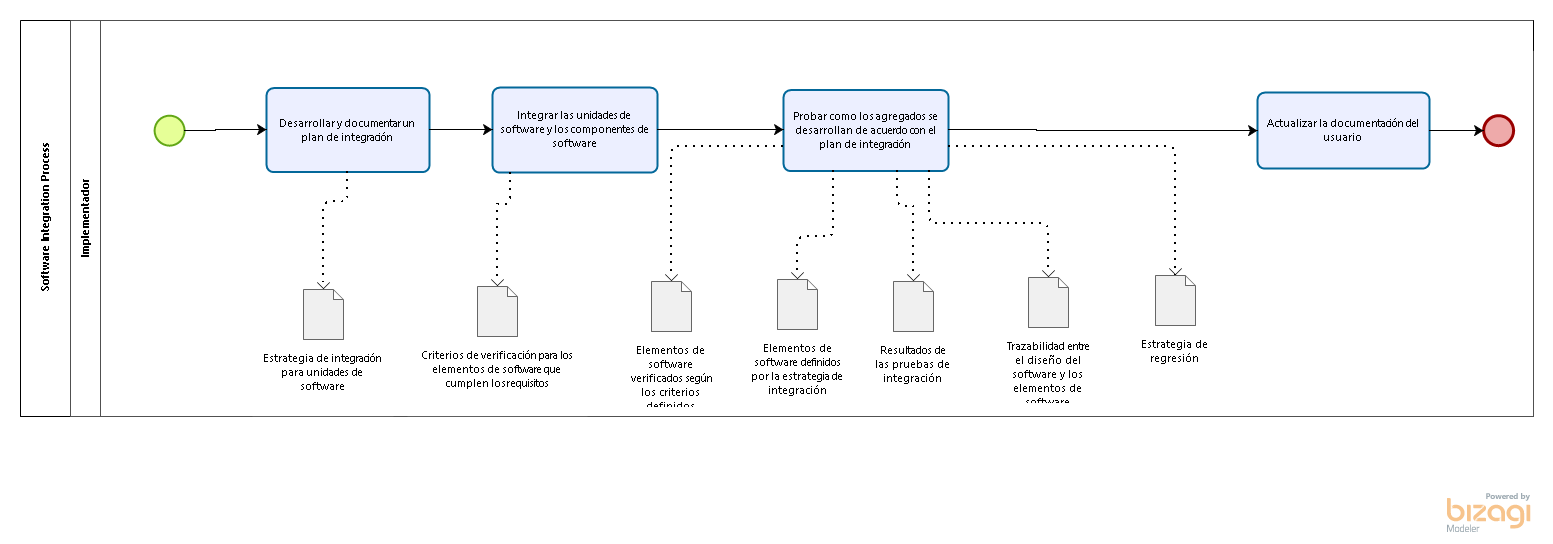
**Clause 7.1.4. Software Detailed Design Process**



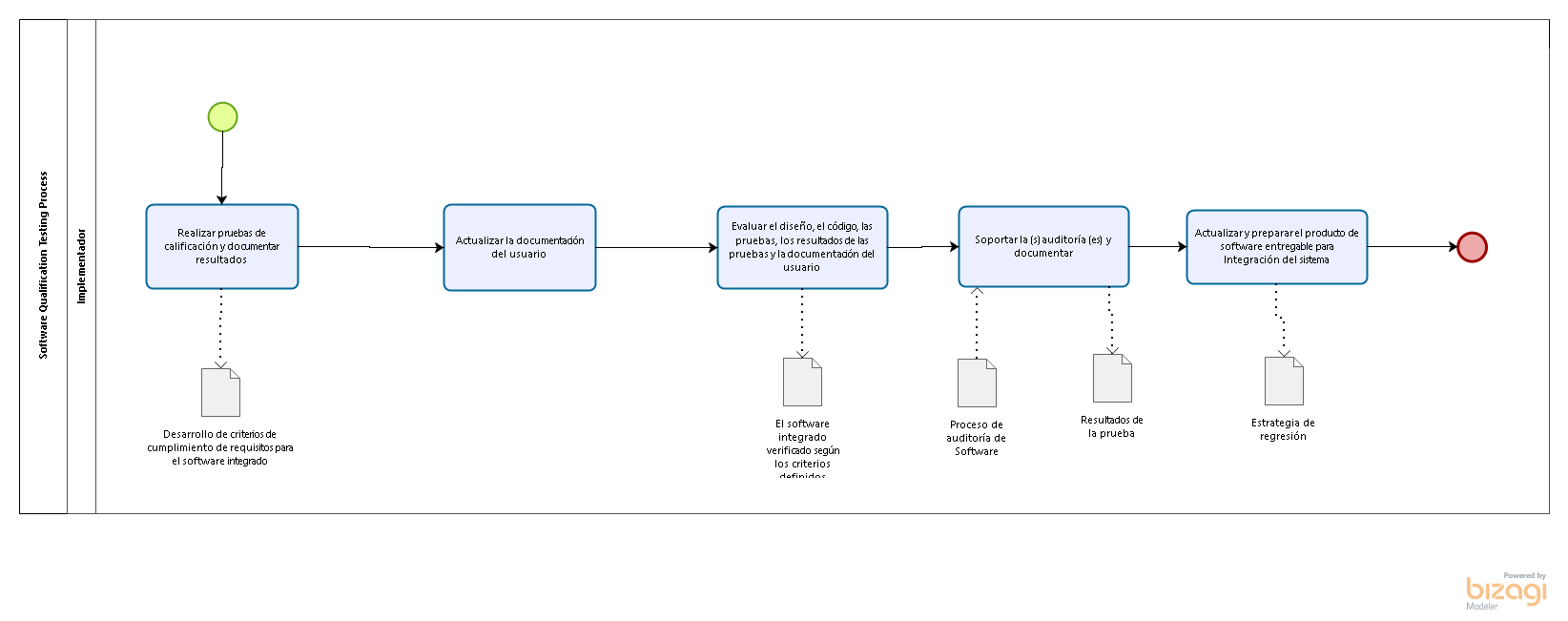
**Clause 7.1.5. Software Construction Process**



**Clause 7.1.6. Software Integration Process**



**Clause 7.1.7. Software Qualification Testing Process**



## Revisiones y auditorías

### Objetivo

Conocer la eficiencia y el rendimiento del sistema de gestión de pólizas, identificar los riesgos y presentar las oportunidades de mejora

### Requerimientos mínimos

* Distribución de los equipos por área
* Revisión de servidores y terminales que existen en el área.
* Instalaciones y conexiones físicas.
* Definir las características generales de los sistemas que serán auditados
* Las medidas de seguridad física existentes en el área
* El Cumplimiento de las funciones del personal informático y usuarios de los sistemas.
* Los programas de Capacitación para el personal del área y usuarios de los sistemas.
* Protección de las bases de datos.
* Datos Confidenciales y accesos a las mismas.
* Protección de las Copias de Seguridad y la restauración de la información.

### Revisión de requerimientos

|  |  |
| --- | --- |
| **Revisión** | **Control** |
| Distribución de los equipos por área |  |
| Revisión de servidores y terminales que existen en el área. |  |
| Instalaciones y conexiones físicas. |  |
| Definir las características generales de los sistemas que serán auditados |  |
| Las medidas de seguridad física existentes en el área |  |
| El Cumplimiento de las funciones del personal informático y usuarios de los sistemas. |  |
| Los programas de Capacitación para el personal del área y usuarios de los sistemas. |  |
| Protección de las bases de datos. |  |
| Datos Confidenciales y accesos a las mismas. |  |
| Protección de las Copias de Seguridad y la restauración de la información. |  |

### Revisión de diseño preliminar

|  |  |
| --- | --- |
| **Revisión** | **Control** |
| Definir el objetivo del sistema a desarrollar |  |
| Definir el equipo de trabajo |  |
| Definir el alcance del sistema |  |
| Definir tipos de usuario y su jerarquía |  |
| Construcción de Diagramas de casos de uso |  |
| Construcción de Diagramas de Clases |  |
| Construcción de Diagramas de Secuencia |  |

### Revisión de diseño crítico

|  |  |
| --- | --- |
| **Revisión** | **Control** |
| Modelo de dominio |  |
| Descripción de la arquitectura |  |
| Revisar cada producto |  |
| Revisar el ajuste al proceso |  |
| Realizar Revisión Técnica Formal (RTF) |  |
| Asegurar que las desviaciones son documentadas. |  |

### Revisión del Plan de Verificación & Validación

|  |  |  |
| --- | --- | --- |
| **Etapa** | **Revisión** | **Control** |
| MODELADO DE NEGOCIO | Especificación del modelo de negocio |  |
| Plan de riesgo |  |
| REQUISITOS | Catálogo de requisitos |  |
| Especificaciones de caso de uso |  |
| Modelo de Casos de Uso |  |
| Modelo de caso de prueba |  |
| Métricas de la etapa |  |
| ANÁLISIS | Descripción de la arquitectura |  |
| Realización de casos de uso |  |
| Modelo de análisis |  |
| Prototipo GUI |  |
| Métricas de análisis |  |
| DISEÑO | Modelo de diseño |  |
| Modelo de datos |  |
| Especificación de casos de prueba |  |
| Métricas de diseño |  |
| CODIFICACIÓN | Modelo de codificación |  |
| Métricas de codificación |  |
| PRUEBAS | Lista de errores |  |
| Métricas de pruebas |  |
| LIBERACIÓN | Manual de usuario |  |
| Manual de instalación |  |
| Aval de liberación |  |
| Adicciones y/o correcciones a nueva versión |  |
| CIERRE | Valoración final del proyecto |  |

### Auditoría funcional

Realizar una evaluación integral al Sistema de Gestión de Pólizas, con la finalidad de verificar su desarrollo y detectar deficiencias.

### Auditoría física

Realizar la revisión y evaluación sobre la seguridad física a nivel de infraestructura, equipos, sistemas de información y el entorno donde será instalado el Sistema de Gestión de pólizas, para esto se establecerán políticas, procedimientos y prácticas para evitar interrupciones prolongadas del servicio de procesamiento de información por algún tipo de contingencia.

### Revisiones de gestión

|  |  |
| --- | --- |
| **Actividad** | **Semana cuando se realiza** |
| Elaboración del Plan de Calidad | 2, 3 y 4 |
| Evaluar y ajustar el plan de SQA | 6, 8 y 11 |
| Evaluar la calidad de los entregables | 1,2,3,4,5,6,7,8,9,10,11,12 |
| Revisar el ajuste al proceso | 3,4,5,6,7,8,9,10,11 y 12 |
| Revisión Técnica Formal (RTF) | 5,6,7,8,10,11 y 12 |
| Realizar el informe final de calidad | 13 |

### Revisión del Plan de gestión de configuración

|  |  |
| --- | --- |
| **Actividad** | **Semana cuando se realiza** |
| Especificación del sistema | 2 |
| Plan del proyecto software | 3 |
| a) Especificación de requerimientos del software  b) Prototipo ejecutable o en papel | 3 |
| Manual de usuario preliminar | 4 |
| Especificación de diseño  a) Diseño preliminar  b) Diseño detallado | 4 |
| Listados del código fuente | 5 |
| a) Planificación y procedimiento de prueba  b) Casos de prueba y resultados registrados | 6 y 7 |
| Manuales de operación y de instalación | 6 |
| Programas ejecutables | 8 |
| Manual de usuario | 8 |
| Documentos de mantenimiento  a) Informes de problemas del software  b) Peticiones de mantenimiento  c) Órdenes de cambios de ingeniería | 9 y 10 |
| Estándares y procedimientos de ingeniería del software | 11 |

## Verificación

La verificación se hará conforme a lo expresado en el documento: Plan de Verificación y Validación.

## Reporte de problemas y acciones correctivas

Ante la detección de un error en la documentación se informará a él/los responsable/s de dicho documento para que lo corrijan. Esto se hará mediante el uso del grupo del proyecto. En caso de que el error se repita reiteradas veces se tratará el tema en alguna reunión de grupo, ya sea ordinaria o, en caso que la situación lo amerite, en una reunión extraordinaria.

## Herramientas, técnicas y metodologías

**Se utilizará la técnica de Check List.**

En esta actividad se revisan los productos que se definieron como claves para verificar en el Plan de calidad. Se debe verificar que no queden correcciones sin resolver en los informes de revisión previos, si se encuentra alguna no resuelta, debe ser incluida en la siguiente revisión. Se revisan los productos contra los estándares, utilizando la checklist definida para el producto. Se debe identificar, documentar y seguir la pista a las desviaciones encontradas y verificar que se hayan realizado las correcciones. Como salida se obtiene el Informe de revisión de SQA, este informe debe ser distribuido a los responsables del producto y se debe asegurar de que son conscientes de desviaciones o discrepancias encontradas

|  |
| --- |
| **HERRAMIENTAS DE SOFTWARE** |
| Testlink  Sistema de gestión de pruebas de software basado en la web, es de código abierto (Open Source) y dispone de una amplia comunidad de foristas y voluntarios que publican guías e información sobre cómo utilizarlo. |
| Zephyr / Jira  Software de gestión de ciclo de pruebas de software, disponible como versión empresarial y como un Add-On de Atalassian Jira. |
| Hewlett Packard Quality Center (HP QC)  Software de gestión de calidad suministrado por la división de software de HP, que ofrece diversas capacidades para el aseguramiento de calidad, gestión de requerimientos, gestión de software testing, entre otros. |
| **TÉCNICAS Y METODOLOGÍAS** |
| * Definir proyectos de pruebas * Definir los usuarios que accederán al sistema. * Crear casos de prueba y su información * Organizar los casos de pruebas en “conjuntos de pruebas” * Asignar palabras claves (keywords) a los casos de pruebas. * Crear planes de pruebas (Test Plan) y enlazarlos casos de pruebas * Ejecutar los casos de prueba y registrar resultados. * Visualizar los resultados de las pruebas |