



UNIVERSIDAD DON BOSCO
VICERRECTORÍA DE ESTUDIOS DE POSTGRADO

TRABAJO DE GRADUACIÓN

**“GUIA DE CUMPLIMIENTO DE LOGROS PARA LA SEGURIDAD DE LA INFORMACIÓN
EN EL PROCESO DE AUTOMATIZACIÓN DE SOFTWARE. CASO PRÁCTICO: CITI
INTEGRATED SERVICE CENTER”.**

**PARA OPTAR AL GRADO DE
MAESTRÍA EN SEGURIDAD Y GESTIÓN DE RIESGOS INFORMÁTICOS**

ASESOR:

MAGISTER RENÉ ARTURO ANGULO ARRIAZA

PRESENTADO POR:

JESSICA GUADALUPE TORRES CARRANZA

ANTIGUO CUSCATLAN, LA LIBERTAD, EL SALVADOR, CENTROAMÉRICA

AGOSTO 2017

Guía de cumplimiento de logros para la seguridad de la información en el proceso de automatización de Software.

Caso práctico: Citi integrated Service Center.

Jessica G. Torres Carranza
ysik18@hotmail.com

Resumen.— El presente documento, plantea una guía para cumplir con los estándares de Seguridad de la Información, en el proceso de automatización de software que se lleva a cabo en CISC (Citi integrated Service Center) y de cómo los principios de seguridad son puestos en práctica en éstos procedimientos, CISC, es una empresa de servicios tecnológicos en la que su fuerte principal es el desarrollo de software para aplicaciones web dado que sus clientes son bancos bajo una misma plataforma llamada Netbanking Java, el cual es un servicio en línea para que los clientes puedan realizar sus transacciones personales y los empleados de agencias bancarias puedan también hacer gestiones de atención al cliente; el departamento involucrado en ésta actividad es el departamento de Quality Assurance, en éste, se llevan a cabo aplicaciones nuevas o mantenimientos, directamente en la plataforma de Netbanking Java; se presentarán las mejores prácticas en la ejecución de casos de pruebas, para garantizar que los clientes reciban el desarrollo que han solicitado y al mismo tiempo garantice que los procedimientos sean puestos bajo los estándares de seguridad de la información; tomando en cuenta los conceptos: Disponibilidad, Integridad y Confidencialidad.

Se describe cada fase del procedimiento de prueba desde que nace el requerimiento hasta que es entregado a producción.

Términos índice. —Seguridad de la información, casos de prueba, automatización de software, Ciclo de vida de un proyecto.

I. INTRODUCCIÓN.

Debido a la Crisis financiera del 2008 y la gran pérdida de valor en los activos relacionados a las hipotecas "subprime", Citibank obtuvo un rescate económico por el gobierno de los Estados Unidos. El 23 de noviembre del 2008 se invirtieron \$25 mil millones de dólares, adicionales a los \$25 mil millones del apoyo inicial, en la compañía, tomando como garantía activos con valor de \$306 mil millones de dólares. Desde ese momento, Citibank ha pagado los créditos del gobierno completamente, e incluso ha generado una utilidad para el gobierno de los Estados Unidos. El día 30 de junio de 2016, Inversiones Financieras Imperia Cuscatlán S.A., adquirió Banco Citibank de El Salvador, S.A., hoy denominado Banco Cuscatlán de El Salvador, S.A, pero esto ocurre únicamente con la cartera de consumo, La fábrica de software conserva su nombre CITI y está bajo la dirección del equipo CISC por sus siglas en inglés (Citi integrated Service Center).

CITI, Como parte de su transformación, busca convertirse en una empresa competitiva ante sus clientes, por lo que en su estrategia de operación adopta la automatización de casos de prueba, como procedimiento esencial en la fábrica de software.

La visión es ser un equipo integral y proactivo, que brinde servicios de aseguramiento de la calidad sobre los productos y procesos institucionalizados, a través de una revisión de los procedimientos aplicados y exactitud en el desarrollo de las aplicaciones así como

proporcionar el soporte para buscar un cumplimiento completo a las políticas.

La fábrica de software, es un departamento dentro de CISC (Citi integrated Service Center) que cuenta con varias áreas, entre ellas:

*Desarrollo

*QA (Quality assurance)

*Project Managers.

Las tres áreas son lideradas por un Representante de negocios, que es el contacto entre el cliente y todo el equipo CISC (Citi integrated Service Center).

La institución debe de enfatizar la seguridad de la información en cada prueba de software que realiza, por lo que requiere de la sistematización para desempeñar sus proyectos.

Banco Cuscatlán y SISA (Seguros e inversiones) son ahora clientes principales clientes de CISC (Citi integrated Service Center) y la gran labor ha sido durante más de dos años, llevar a cabo las migraciones de cada uno de los diferentes países del conglomerado Banco Cuscatlán, entre los países que se han estado migrando están: Costa Rica, Nicaragua, Honduras, Guatemala, Panamá, El Salvador, todos estos países cuentan con la plataforma NBJ (Netbanking java), para SISA (Seguros e inversiones), únicamente se ha trabajado con El Salvador y se han estado realizando actualizaciones a la aplicación de facturación, reportería y el sistema general de clientes, los cambios y actualizaciones han sido desde rebranding (cambio de logos, disclaimer), cambios de fórmulas y reporterías finales, eliminación de secciones y deshabilitación de opciones por perfiles, esto implica que algunas pantallas serán completamente customizadas.

Los requerimientos son aceptados contractualmente tomando como base los lineamientos de SLA, a través de una figura llamada Business Representative, el cual se

reúne con los cliente y toma la solicitud, dicha solicitud es llevada a comité técnico, donde todas las partes involucradas se reúnen para evaluación del requerimiento, estiman tiempos y se extienden los resultados, luego el Business representative (BR) se reúne nuevamente con el cliente y firman un contrato donde Citi se compromete a entregar lo solicitado en el tiempo estipulado y de no cumplirse con alguna de las disposiciones, los clientes están en derecho de demandar por incumplimiento a la institución, esto conlleva a un impacto realmente legal y de prestigio, por lo que se debe de cumplir con las fechas establecidas en el contrato.

II. Metodología de trabajo.

La metodología que se utilizará en éste trabajo, será a través de reuniones con las personas involucradas, donde se expondrá ciertos puntos en donde se han detectado discrepancias y que es básicamente la problemática actual, investigación sobre mejores prácticas y mejoras continuas implementadas en otras instituciones que se enfoquen al servicio al cliente.

II-A. Planteamiento del problema.

La plataforma NBJ (Netbanking Java) tiene una gran demanda, tanto para los empleados de agencia como para los clientes que utilizan sus servicios de transferencia y solicitudes de servicio en línea; de allí radica la importancia de ésta investigación.

El problema se refleja al entregar las aplicaciones desarrolladas y ejecutadas en fase de pruebas para usuarios expertos llamado UAT (User Assesment test) y las puestas a producción de las aplicaciones desarrolladas, ya que en repetidas ocasiones se han reportado inconsistencias leves y severas, lo que conlleva a inconformidades de los clientes que solicitan los desarrollos en sus aplicaciones y en algunas ocasiones se ha reportado

pérdida de datos en algunas cuentas de clientes activos; Los requerimientos se están atendiendo de manera automatizada y es necesario plantear mejores prácticas al inicio de un proyecto, durante y después de realizar las pruebas automatizadas que son el día a día en el departamento de calidad.

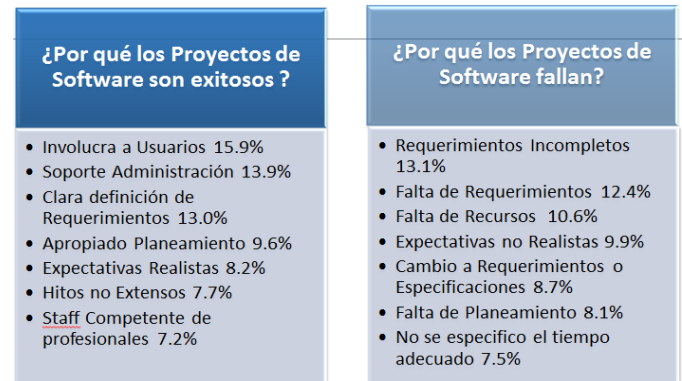
II-B. Hipótesis.

La hipótesis con la que se iniciará la investigación es:

Que el cliente solicita un desarrollo en sus aplicaciones de software que no es calificado por todas las áreas involucradas y durante la ejecución del desarrollo, por lo que durante las pruebas se van realizando cambios que no habían sido contemplados, ni notificados al cliente en un inicio; dicha pérdida de seguimiento es causada por el área de PM (Project Managers), ya que es el área encargada de organizar conferencias, reuniones informativas, enviar reportes de resultados, llevar el ciclo de vida del proyecto calendarizado y documentado correctamente en la herramienta correspondiente; dicha pérdida de seguimiento produce discrepancias entre lo que el cliente solicita y lo que se entrega como producto final, es decir, a causa de ésta falta de involucramiento la integridad de la información inicial es afectada, provocando demandas contractuales además de afectar la imagen de la institución.

The Standish Group ofrece un estudio estadístico con resultados reveladores en base a miles de proyectos relevados a lo largo de varios años basándose en un promedio de 6 grandes instituciones bancarias, tales como Banco Agrícola, Banco de América Central, entre otros. Los resultados son los descritos en la Figura a.

Figura a. Exitos y fracasos en proyectos de software.



II-B. Forma de trabajo.

La investigación, se realizará a partir de resultados obtenidos recientemente en conclusión de reuniones de trabajo, en donde realizaron algunos señalamientos referente a la documentación, utilizada para el seguimiento de los requerimientos, también en las reuniones de trabajo se ha recopilado una serie de ideas y sugerencias para mejorar el proceso de administración de proyectos, tomando siempre en cuenta la seguridad de la información, de manera que garantice la integridad del requerimiento inicial y el valor de que la empresa genera empleando reestructuramiento en sus procesos de gestión de proyectos en software sobre todo en las pruebas automatizadas, todo esto, para lograr la satisfacción final del cliente en cada entrega de las aplicaciones solicitadas y el valor que para la empresa representa llevar todo el proceso de gestión de proyecto y automatización de pruebas bajo lineamientos de seguridad.

III. Marco Conceptual.

Las pruebas de calidad automatizadas son un proceso estructurado y sistemático dirigido a obtener un rendimiento mayor de un proceso, aumentar la calidad de servicio o disminuir el costo de obtención de actividades que ya se desarrollan.

James Harrington afirma que, mejorar un proceso significa cambiarlo para hacerlo más efectivo, eficiente y adaptable.

III.a Definiciones.

Algunas pruebas de software tales como las pruebas de regresión intensivas de bajo nivel pueden ser laboriosas y consumir mucho tiempo para su ejecución si se realizan manualmente. Adicionalmente, una aproximación manual puede no ser efectiva para encontrar ciertos tipos de defectos, mientras que las pruebas automatizadas ofrecen una alternativa que lo permite. Una vez que una prueba ha sido automatizada, ésta puede ejecutarse repetitiva y rápidamente en particular con productos de software que tienen ciclos de mantenimiento largo, ya que incluso cambios relativamente menores en la vida de una aplicación pueden inducir fallos en funcionalidades que anteriormente operaban de manera correcta. Existen dos aproximaciones a las pruebas automatizadas:

Pruebas manejadas por el código: Se prueban las interfaces públicas de las clases, módulos o bibliotecas con una variedad amplia de argumentos de entrada y se valida que los resultados obtenidos sean los esperados.

Pruebas de Interfaz de Usuario: Un marco de pruebas genera un conjunto de eventos de la interfaz de usuario, tales como teclear, hacer click con el ratón e interactuar de otras formas con el software y se observan los cambios resultantes en la interfaz de usuario, validando

que el comportamiento observable del programa sea el correcto.

La elección misma entre automatización y ejecución manual de pruebas, los componentes cuya prueba será automatizada, las herramientas de automatización y otros elementos son críticos en el éxito de las pruebas, y por lo regular deben provenir de una elección conjunta de los equipos de desarrollo, control de calidad y administración. Un ejemplo de mala elección para automatizar, sería escoger componentes cuyas características son inestables o su proceso de desarrollo implica cambios continuos.

El concepto de la calidad, es definido como la satisfacción de los requisitos (explícitos) y necesidades (no necesariamente explícitas) del cliente, ya sea interno o externo de la institución y bajo éste concepto es necesario señalar que no puede haber lugar para demandas contractuales que es la problemática actual y que conlleva a que las pruebas de software que se realizan al final aunque se ejecuten bajo las normas establecidas por el departamento de calidad, es muy probable que los escenarios planteados no sean los que deberían ser probados en base al requerimiento inicial y se invierta tiempo en la elaboración de las pruebas de software automatizadas si en algún momento del ciclo de vida del proyecto el requerimiento inicial ha perdido su integridad esto conlleva a la insatisfacción y pérdida de valores de la institución, las fallas contractuales, es un punto que dá pauta a las demandas realizadas por los clientes actuales de la institución.

A continuación se presenta una lista de algunos elementos asociados a las fallas contractuales y se presentan en la siguiente tabla:

Tabla 1. Detección de elementos de fallas.

Elementos asociados a fallas contractuales relacionados a la calidad
* Fallas en el proceso de revisión de contrato antes de haber sido aceptado por el negocio
* Permitir ausencias de representantes de áreas en comité técnico
* Errores al calcular Sizing en las etapas que componen el SDLC y los SLA
* Errores relacionados con documentos entregables.
* Variación de valores en las bases de datos que se entregan, en algunos casos pérdida de datos

La tabla 1. Fue enviada como resumen por parte del departamento de seguridad de la información, como recopilación de los hallazgos señalados después de reuniones con la alta gerencia, la PMO (Project Manager Officer), Gerencia de desarrollo, Gerencia de Quality Assurance y Departamento legal de la institución para que la alta gerencia tomara con severidad las causas de fallas contractuales.

Se define la Seguridad de la Información como la preservación de su confidencialidad, integridad y su disponibilidad (medidas conocidas por su acrónimo "CIA" en inglés "Confidentiality, Integrity, Availability").

Este es el modelo diseñado para para guiar las políticas de seguridad de la información de una institución, a continuación en la Figura a, se presenta el triángulo CIA.

Figura a. Esquema de seguridad de la información



- Confidencialidad: La información debe de ser resguardada y únicamente manipulada por quienes tengan permiso para hacerlo.
- Integridad: La información no debe de sufrir ninguna modificación a menos que sea solicitada por las personas interesadas.
- Disponibilidad: La información debe de estar disponible siempre que se solicite estarlo.

Los tres conceptos son importantes y deben de ser tomados en cuenta en las pruebas automatizadas de software que se realizan para la plataforma en línea Netbanking Java actualmente.

Los beneficios de poner los conceptos en práctica en las pruebas automatizadas son:

- Minimizar y gestionar los riesgos y detectar los posibles problemas y amenazas a la seguridad.
- Garantizar la adecuada utilización de los recursos y de las aplicaciones del software
- Limitar las pérdidas de información.
- Cumplir con el marco legal y con los requisitos impuestos por los clientes en sus contratos.

III.b Contexto propio del área de Quality Assurance.

La automatización de pruebas en CISC, consiste en el uso de software especial (casi siempre separado del software que se prueba (como es el caso del presente trabajo de tesis) para controlar la ejecución de pruebas y la comparación entre los resultados obtenidos con los resultados esperados. Dicha automatización, permite incluir pruebas repetitivas y necesarias dentro de un proceso formal ya existente o bien adicionar pruebas cuya ejecución manual resultaría difícil.

QA (Quality Assurance) es el área responsable en Centro América de asegurar el cumplimiento a las políticas y estándares Citi de los diferentes productos y servicios elaborados por los Hubs en la región.

QA (Quality Assurance) apoya a la institución en un desarrollo de sistemas y servicios de calidad con alta eficiencia y competitividad para incrementar la satisfacción del cliente interno y externo ya que la empresa trabaja con sus clientes contractualmente de la siguiente forma:

- Al presentarse la solicitud del cliente el BR (Business representative), se reúne con su equipo para tomar notas de lo que el cliente está solicitando, éste a su vez discuten en vivo con su equipo sobre la disponibilidad y asertividad de lo que se está solicitando
- Al tener el requerimiento completo del cliente, se presenta en comité técnico, donde deberían de estar todos los integrantes involucrados en la operación de proyectos.
- El requerimiento se queda en espera de aprobación y al ser aprobado se presenta un Sizing, que es el estimado en días de lo que durará un requerimiento por fases.

- Al tener el dato de parte del negocio, el BR (Business representative) vuelve a reunirse con el cliente y se presenta los resultados de aprobación, observaciones y el estimado en tiempo de la entrega del requerimiento.
- El cliente aprueba o no aprueba la resolución
- Si el cliente aprueba, se firma un contrato en donde el negocio se compromete a entregar el requerimiento en días acordados y el cliente se compromete a no desistir del acuerdo durante el período de tiempo establecido.
- En caso de que una de las dos partes rompa el acuerdo o no cumpla con uno de los puntos establecidos en el contrato, la otra parte está respaldada para demandar en la fiscalía general de la república, demandas que cuestan mucha pérdida tanto, monetaria y sin lugar a dudas la más impactante es la del prestigio.

III.c Uso de herramientas.

Para el presente trabajo de investigación, se hará uso de la herramienta HPE UFT (Unified Functional Testing), el cual, automatiza las pruebas a través de una experiencia de usuario intuitivo y visual que relaciona las pruebas manuales, automatizadas y basadas en marcos en un solo IDE (Integrated Development Environment). Esta solución de largo alcance reduce significativamente el coste y la complejidad del proceso de prueba funcional e impulsa continuamente la calidad.

Durante el ciclo de vida del requerimiento se hace uso de herramientas de soporte tales como:

- **Plan view y PTS:** Ambas son aplicaciones administradoras de requerimientos
- **Clear Quest y Clear Case:** Se realiza acá el control de las tareas, la fabricación de escenarios y versionamiento de los códigos fuentes.
- **Dev Tools:** Se encuentran las diferentes herramientas de desarrollo según la plataforma en la que se va a trabajar, entre ellas están las herramientas para NBJ (Netbanking Java) y SISA (Seguros e inversiones).
- **Service Now:** Se registra y se dá seguimiento a los tickets internos o los elaborados por el negocio, también se atienden acá las solicitudes para los cambios de producción.
- **UFT:** (Unified Functional Testin) herramienta de soporte en el proceso de automatización de casos de prueba, los cuales se hacen a través de la construcción previa de los escenarios.

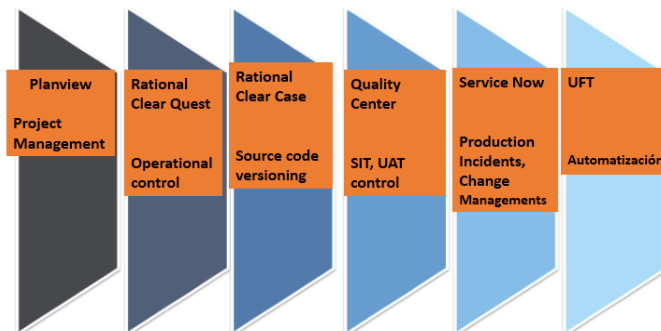
La implementación de actividades específicas dentro del proyecto o servicio para prevenir que ocurran defectos, incidentes o situaciones que afecten la satisfacción del cliente (ésta práctica invoca actividades de Project Managers, testing, análisis automáticos, etc).

La razón de la calidad en todos los procesos que la institución ejecuta, debe de ser justificada bajo la premisa de:

- La calidad es gratuita, lo que significa que la inversión y detección de la calidad, redundan en ahorros de lo que costaría la mala calidad (no hacer nada y esperar que el impacto se produzca).
- La calidad del producto o servicio es altamente dependiente de la calidad de

los procesos usados durante el desarrollo, adquisición o nacimiento del requerimiento y mantenimiento del mismo, el mantenimiento hace alusión a la labor del Project manager que es quien debe de realizar las actualizaciones necesarias al requerimiento notificando paralelamente al cliente y las partes interesadas.

Figura 1. Herramientas en la ejecución de un proyecto



III-d. Calidad en pruebas de software.

La seguridad de la información en pruebas automatizadas de software cubre dos aspectos complementarios de la calidad de acuerdo al tipo de consumo de la institución de estudio es necesario tomar en cuenta los siguientes:

Para Citi, la plataforma Netbanking Java, es una plataforma bancaria en línea que es utilizada por clientes y empleados de agencias y es administrado por un coordinador de canales virtuales, las gestiones de actualizaciones son autorizadas por un comité y se realizan bajo un ambiente de prueba.

Luego de la información de cada elemento involucrado en la automatización, es necesario tocar el tema de la seguridad de información en dicho proceso de automatización.

Cuando un requerimiento es aprobado, es enviado al departamento de Quality assurance (QA), el cual es analizado, se procede a la ejecución de un SIZING y al ser aprobado se asigna un ejecutor del SIT (System test), en la

ejecución de pruebas hay dos grupos de pruebas:

- Pruebas con usuarios clientes.
- Pruebas con usuarios bancarios.

El presente trabajo de investigación mostrará las mejores prácticas para realizar pruebas automatizadas, basándose en la seguridad de la información aplicada a todos los elementos implicados en la automatización, garantizando la efectividad, eficiencia e integridad de las pruebas, La verificación y la planificación adecuada de los procesos establecidos en beneficio del negocio, con el fin de producir el valor a la institución , reducir riesgos y gestionar los recursos adecuadamente.

Se presentarán el flujo del requerimiento desde que es presentado a comité técnico hasta que es puesto a producción.

III-d.1 Resumen de automatización actual

La automatización de pruebas de software en una plataforma bancaria, es un conjunto de pruebas que se realizan de manera programada a los aplicativos que necesitan una actualización

Inicialmente se ha automatizado todas las secciones de la plataforma, utilizando datos reales de la cartera de clientes y luego revirtiendo los cambios realizados en cada prueba asignando a una persona para ésta tarea diariamente.

En el proceso, hay algunas secciones que no podrán ser automatizadas, debido a que no es posible emular el insumo, como por ejemplo los pagos de recibos de agua, luz, pagos de renta, todas estas transacciones son posible realizarlas sin riesgo en tiempo real, pero no

como pruebas ya que los números de recibos deben de ser reales e implicaría una gran variación diaria en los cierres entregados.

Luego de separar los casos efectivos y los fallidos, con el uso de la herramienta UFT (Unified Functional Testin) se van ejecutando caso por caso introduciendo datos reales de clientes, realizando las transacciones entre cuentas propias, cuentas a terceros y cuentas internacionales, dichas pruebas son delicadas y complican mucho la obtención de data real; como inicialmente se mencionó, al final de cada día un equipo de 3 personas realizan las reversiones de las transacciones reportadas por los desarrolladores.

Cada persona involucrada en la inicialización de un proyecto debe de tomar en cuenta los siguientes puntos:

- **Necesita saber:**
 - Escuchar, entrevistar, cuestionar, analizar, facilitar, observar, comunicar (verbal y escrito), organizar, modelar, relacionarse con gente, innovar.
- **Necesita tener:**
 - Experiencia en el dominio del problema y de la solución
 - El rol clave de los requerimientos es mostrar qué se necesita de un sistema
 - Si un producto no es lo que el cliente o los usuarios quieren, entonces la calidad de la construcción es irrelevante.
 - Proveer los requerimientos en un lenguaje que todos comprenden, ya que todos están involucrados, incluyendo los clientes. El primer y básico rol de los requerimientos es por lo tanto: **la comunicación**

III-e. Ciclo de vida de un proyecto

El departamento de calidad se rige por un proceso llamado SDLC (Software Development Life Cycle), que es el flujo que un requerimiento debe de seguir.

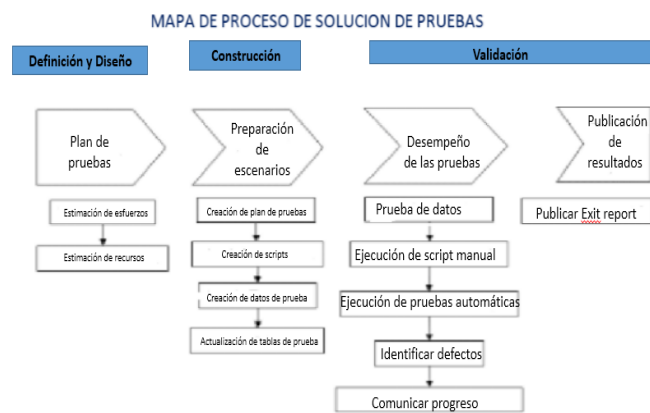
El SDLC está compuesto por fases, las cuales son:

- Iniciación
- Diseño
- Construcción
- Validación
- Implementación y
- Post implementación

Las cuales están descritas en la Figura 3.

En la fase de construcción es donde se realizan las pruebas automatizadas, pero para garantizar que el requerimiento se desarrolle de manera integral los procesos de seguridad deben de iniciarse desde la primera fase.

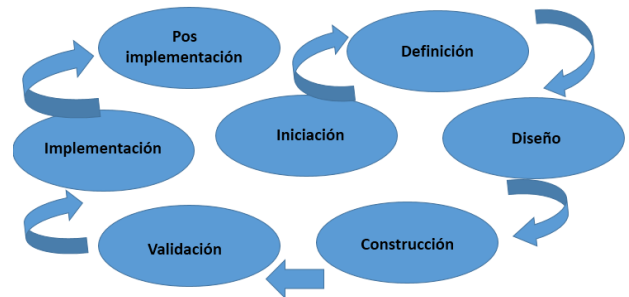
Figura . 2.Mapa de procesos para pruebas de software.



Seguendo el modelo de la figura 2, se muestra la figura 3 a manera de explicación (Las abreviaciones están detalladas en el glosario).

Figura 3. Ciclo de vida de proyectos automatizados y sus roles.

Ciclo de vida del proyecto



Así está el proceso de inicialización de proyecto actualmente en la institución, la cual es el caso de estudio para ésta tesis.

Al iniciar y aceptar un requerimiento en el comité técnico, se hace uso de la herramienta QC Quality Center , descrita anteriormente en la figura 1. Y es así como luce en la herramienta cuando ya queda registrado el proyecto listo para comenzar el SDLC (Software Development Life Cycle).

El ciclo de vida (Figura 2 y 3) de un requerimiento responde a las siguiente fases:

- **Iniciación.**
Involucrados: BA, PM, SQA
Propósito: Definir, documentar, revisar y aprobar el requerimiento.
- **Definición**
Involucrados: BA, PM, SQA
Propósito: Definir, documentar, revisar y aprobar el requerimiento.
- **Diseño.**
Involucrados: Desarrollo
Propósito:
- **Construcción.**
Involucrados: QA
Propósito: Diseñar los casos de prueba, ejecutarlos y documentarlos.

- **Validación.**
Involucrados: Usuarios
Propósito: Probar los escenarios realizados para la fase de UAT, validando su correcto funcionamiento.
- **Implementación.**
Involucrados: Desarrollo
Propósito: Poner en producción la aplicación que se ha desarrollado.
- **Post implementación.**
Involucrados: QA, Desarrollo
Propósito: Tiempo de garantía para reaccionar rápidamente ante cualquier error después de haber puesto a producción un aplicación.

3. Documento sin las aprobaciones mínimas requeridas.
4. Observaciones del Comité.

Además se puede colocar el detalle que el revisor considere pertinente.

Figura 5. Registro de requerimiento en comité técnico.

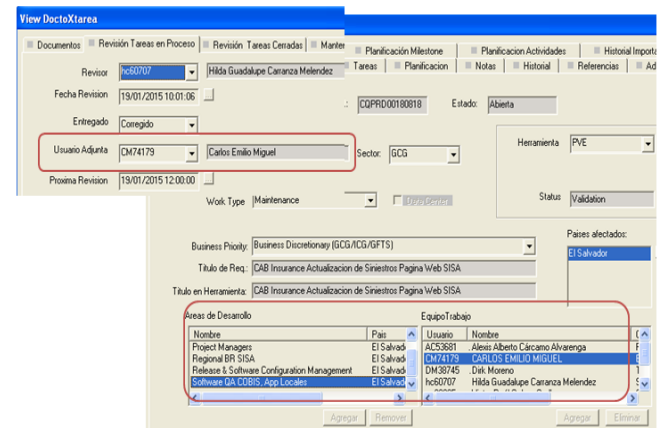
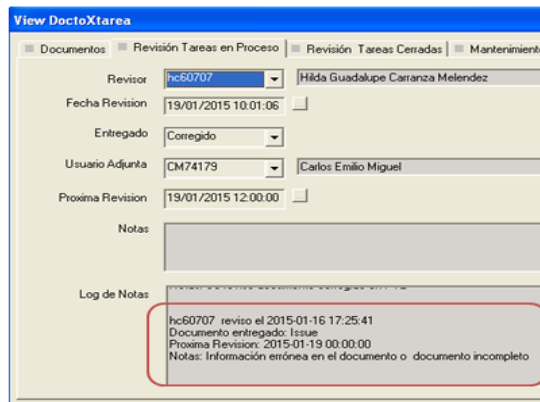


Figura 4. Creación de un requerimiento.



Observamos el user ID Hc60707, el cual queda registrado en la herramienta y es el usuario que tiene la responsabilidad de la correcta ejecución del proceso de aprobación.

Dependiendo del motivo por el cual se ha calificado con Issue, es necesario incluir uno de los siguientes comentarios estandarizados:

1. Documento adjunto con formato no vigente.
2. Información errónea en el documento o documento incompleto.

Debe ser colocado en todos los entregables calificados. Los responsables deben estar incluidos en el Equipo de Trabajo del requerimiento.

IV. Desarrollo de la metodología Investigación

IV-A. Procedimiento propuesto en la revisión inicial del requerimiento.

Actualmente, el SDLC (Software Development Life Cycle) consta de 7 fases, pero debido a la problemática actual se propone incluir una fase de definición, en donde se definen los puntos en función y el estimado en días de proyectos, en éste estimado llamado Sizing, es imperativa la participación de todas las partes involucradas con esto garantizamos la disponibilidad del proyecto, por lo que se propone realizarla en las reuniones de comité técnico que en la actualidad ya están funcionando.

De acuerdo al trabajo de investigación se propone lo siguiente, para la atención y seguimiento de proyectos en donde identificamos el ciclo de vida de un requerimiento.

Tomando como base siempre la figura número 3, podemos observar las diferentes fases que un requerimiento debería de tomar en cuenta para que la ejecución de la fase de construcción vaya manteniendo alineado el requerimiento, no sin antes tomar en consideración alinear estratégicamente los niveles de la compañía. Para la seguridad de la información en las pruebas automatizadas en cada fase, cada documento es importante.

Fase iniciación:

Se procederá a ejecutar la gestión, registro y creación del requerimiento en la herramienta Quality Center, muy importante la captación de información por parte del Business representative porque tomará la solicitud inicial y es quien iniciará el proceso para proporcionando una debida integridad del requerimiento por lo que todos los puntos deben de ser analizados en comité técnico, al final de ésta fase se extiende un documento contractual y la empresa debe de garantizar la seguridad de que lo que se está solicitando se está aceptando y por ende será lo que se va a entregar.

Fase Definición:

- Se realizará el análisis y aprobación del sizing, el cual consiste en la estimación en días de la ejecución del proyecto total .
- Se procederá a la revisión de pares para calificar punto por punto y garantizar la integridad, disponibilidad y

confidencialidad de los puntos a evaluar.

- Se realiza el BRD (Business representative document) y análisis de ambigüedades hasta ser aprobado.
- Se realiza la matriz de trazabilidad (completa a un 25%).

Entrega del Functional requirement.

Fase Diseño:

- Aprobación de functional requirement por parte del negocio.
- Aprobación del requerimiento en comité técnico.
- La matriz de trazabilidad debe de ir completa en un 50%.esto
- permitirá la continuidad de integridad en las pruebas.

Fase de construcción (Pruebas automatizadas)

- Completar la matriz de trazabilidad 100% acción realizada por el tester antes de realizar las pruebas automatizadas.
- Entrega de las pruebas unitarias, que es un archivo realizado por el desarrollador, que contiene las pantallas de las pruebas que se solicitan realizar en el BRD (Business representative document), documento que debe de entregarse con el mayor número de evidencias posibles para proporcionando una debida la correcta confidencialidad de las pruebas entre los desarrolladores y los testers.
- Ejecución de SIT (System interactive Test) ; éste es el punto en el cual el presente trabajo presenta un énfasis relevante, ya que acá se ejecutan las pruebas automatizadas, se asigna un tester que debe de ser experto en UFT

(Unified Functional Testing) y tener conocimientos medios de la plataforma que se va a probar Quality Center se cargan los Scripts de prueba en QC (Quality Center) y los de UAT (User Assesment test).

- Se reportan los defectos encontrados en e SIT (System interactive Test) en caso de existir defectos y se registran en la herramienta QC (Quality Center)

Fase de validación:

- Los usuarios expertos proceden a la ejecución del UAT (User Assesment test).
- Se realizan los registros de los defectos (en caso de existir defectos).
- Envío del email aprobando el UAT (User Assesment test).

Fase de implementación:

- Visto bueno del comité administrativo

El cual lo está conformado por las gerencias de:

- Quality assurance, (QA, Veronica Trejo)
- Development apartment (Departamento de Desarrollo de software, Jhony López)
- PMO (Project Manager officer), Claudia Santiso)
- Se preparan las gestiones del envío del release en la herramienta de QC (Quality Center)
- Se registran defectos en producción (en caso de existir defectos).

Fase post implementación:

- Colocarle al requerimiento en QC (Quality Center) el estatus “garantía”

garantizando la disponibilidad del producto.

- Registrar Defectos (en caso de existir defectos) .

En la fase de *construcción*, en la elaboración de SIT (System interactive Test) la automatización de pruebas se propone realizarlas con la misma herramienta de automatización UFT (Unified Functional Testin), tomando en cuenta los siguientes aspectos generales:

- Los desarrolladores deberán de usar un ambiente de pruebas en un servidor destinado solamente para pruebas de la plataforma, esto no se ha estado realizando y se encontraba la dificultad de que cuando el servidor particionado de producción tenía un mantenimiento o un inconveniente los tester y desarrolladores no podían continuar con su trabajo diario hasta lograr restablecer la contingencia del servidor, con éste cambio se garantiza la disponibilidad del servidor y la continuidad del negocio.
- Los usuarios testers deberán de tener en su rol asignado, tiempo límite para permanecer logueado en el servidor de prueba, la sesión de usuario deberá expirar despues de las 20 horas del día, ésto, para evitar que se realicen pruebas automatizadas sin supervisión o que exista algún error y no se encuentra alguien para dar soporte de inmediato; en ocaciones anteriores, se realizaron transacciones efectivas y se dejaron para ser anuladas al siguiente día por el equipo designado, pero éstas generaron errores y provocaron que las cuentas de los clientes reales se afectara provocando reclamos de unos de los clientes, esto fue escalado generando alertas en el servicio.

IV-A.2 Documentación propuesta

Importante tomar en cuenta los siguientes entregables que no existen en la actualidad, cabe mencionar que deberá existir un programa que contenga las sesiones necesarias para capacitar a los testers, desarrolladores y project manager con el fin de garantizar que los procesos se cumplan de acuerdo a lo propuesto así cumplir con la seguridad de la información en las pruebas automatizadas.

- ✓ **Documento BRD (Business requirement document)**

Propósito

Definir el “Qué y por qué” del requerimiento de Negocios con el fin de garantizar la integridad, la disponibilidad y la confidencialidad de la información.

Incluye la información general del requerimiento, descripción y Objetivos, Justificación, Supuestos, Dependencias, Alcance, requerimientos específicos del usuario y de áreas de cumplimiento, etc.

De este documento se generan:

Matriz de Ambigüedades.

Identificación de ambigüedades al revisar el BRD (Business representative document) con el equipo involucrado para su atención: Development, Test Manager, BR (Business representative), CTI, otro stakeholder necesario.

Matriz de Trazabilidad.

Requerimientos definidos por las diferentes áreas involucradas: Negocio, Development, Testing para preparar sus Test Cases.

RESPONSABILIDADES EN ESTE ENTREGABLE

Local BR (Business representative) : Apoyar en la definición del BRD, Validar la integridad y completitud de la información, Gestionar autorización BRD (Business representative document), Gestionar el análisis de Ambigüedades y Trazabilidad.

Regional BR (Business representative): Revisar BRD para identificar mejoras, correcciones o reutilización de código .

- ✓ **Functional Requirements**

PROPÓSITO

- Definir el “Cómo” se resolverá el requerimiento utilizando un lenguaje de usuario (no técnico) garantizando la integridad de la información.
- Describir los requerimientos funcionales y no-funcionales para cada uno de los requerimientos específicos descritos en el BRD (Business representative document)
- De este documento se generan el diseño técnico.

RESPONSABILIDADES EN ESTE ENTREGABLE

Local BR: Definir y documentar los FR de mantenimiento usando la plantilla establecida para el work-type y segmento, presentar al negocio para obtener su aprobación. Gestionar la autorización del documento para mantenimientos y proyectos

Regional BR: Definir y documentar FR para proyectos y Major Release usando

la plantilla establecida para el work-type y segmento.

Este documento autorizado es mandatorio para proceder al Diseño Técnico

✓ **UAT (User Assesment test) Plan:**

PROPOSITO

Definir la logística a ser utilizada para las pruebas del negocio garantizando la integridad en las pruebas que realizan los usuarios expertos, incluyendo:

- Equipo de trabajo, ambientes, restauración de datos, horarios de pruebas, alcance, etc.
- Definición de Test Scripts y Escenarios de pruebas.
- Definición de calendario y de los procesos a ser ejecutados durante el UAT (User Assesment test)

RESPONSABILIDADES EN ESTE ENTREGABLE

Local BR : definir escenarios y alcances del UAT (User Assesment test), gestionar participación de negocios en UAT(User Assesment test), validar Test Scripts definidos por negocios, dar seguimiento a avances de UAT(User Assesment test), Dar seguimiento a defectos de UAT, filtrar falsos defectos. Definir procesos batch si se requieren cierres, obtener Sign off UAT(User Assesment test), Gestionar con el negocio la aprobación del Change Request (CHG)

Regional BR : Para Major Releases definir escenarios y alcances del UAT (User Assesment test), Monitorear avances de UAT siendo 2do nivel de escalamiento ante

defectos, Aprobar BRM de restauración de datos

✓ **Sanity check**

PROPOSITO

Gestionar la prueba por el negocio de los cambios que han sido promovidos a los ambientes de producción, en la misma fecha de instalación, esto para asegurando la disponibilidad de la aplicación terminada.

RESPONSABILIDADES EN ESTE ENTREGABLE

Local BR: Gestionar la prueba por el negocio de los cambios que han sido promovidos a los ambientes de producción, en la misma fecha de instalación

Email de revisión satisfactoria por parte del negocio que verificó o con defectos para ser atendidos por Development Center.

Post implementacion review.

Propósito

Monitorear durante el período definido de garantía si se presenta cualquier

defecto relacionado con el cambio instalado.

RESPONSABILIDADES EN ESTA ACTIVIDAD

BR (Business representative) local : Registrar los defectos post-implementación encontrados por las áreas de negocio durante la garantía del cambio instalado para que sea atendido por las áreas de proyectos que desarrollaron el cambio.

No existe un entregable para esta actividad. El requerimiento será cerrado luego que el período de garantía ha finalizado.

IV-B. Planteamiento de la metodología

En la fase de construcción se proponen realizar cambios en los procedimientos de salida y entrada:

IV-B.1 Cambios propuestos en procedimientos de entrada para asegurar la integridad y disponibilidad de la información.

Tabla 2. Procedimientos recomendados relacionados a la seguridad de la información

Proc. de entrada	Generales	Comentarios relacionados a la Seguridad de la información		
		Integridad	Disponibilidad	Confidencialidad
1. Debe de asignarse un usuario a cada desarrollador y a cada tester	Esto deberá hacerse para que cualquier cambio o transacción.	Los usuarios no cambiarán sus perfiles sin antes ser autorizados por el jefe de desarrollo	Los usuarios deben de tener sus perfiles configurados y activos en el desarrollo de cada requerimientos	Por ninguna razón los usuarios pueden compartir sus usuarios o contraseñas
2. Crear un código para poder utilizar perfil de maker y checker, en donde un usuario realiza la prueba de transacción y otro usuario la aprueba, ésto creará mayor control y filtro en las transacciones que se realizan (Ver figura 7)	Esto deberá desarrollarse en la herramienta UFT directamente, para generar un mejor control en las transacciones realizadas en la plataforma	Deben de ejecutarse códigos pre fabricados y ser insertados en cada escenario de prueba que requiera realizar transacciones	Los códigos garantizarán que las transacciones se realicen de manera automática y beneficiará en la optimización de tiempo.	Como las transacciones se crean y autorizan de manera automática, no es necesario que otro usuario manipule la información realizada por el usuario maker.

3. Las pruebas automatizadas deberán ser registradas en el repertorio QC, por lo que se debe de solicitar la habilitación de la asociación, de ésta manera quedarán almacenadas y podrán ser ejecutadas sin tener que estar asociando los casos por cada tester.	Esto debe de hacerse para ejecutarlo directamente sin tener que cargar cada script y cada TSUIT en la máquina de los tester	Lo que un usuario registra en la herramienta QC, no debe de poderse actualizar por otro usuario, aunque estén trabajando en el mismo proyecto	Cada tester y desarrollador debe de tener usuario registrado en la herramienta QC, esto beneficiará a que si se reportan defectos, que se le asigne al desarrollador o tester que puede	No existe confidencialidad en lectura de escenarios al registrarlos en la herramienta QC, ya que es necesario que tanto desarrollador como tester vean lo que se está probando,
4. De acuerdo a la figura 6, La matriz de trazabilidad es uno de los entregables más importantes por lo que deberá de sufrir varios cambios con el fin de mejorar la seguridad de la información en las pruebas	En la actualización la matriz de trazabilidad se elabora en la fase de construcción pero para obtener mejor control y resultado en las pruebas automatizadas, es necesario que éste documento se implemente	Esto asegurará el hilo integral de que lo que se esté desarrollando y probando es lo que el cliente ha solicitado	Evidenciará los enlaces habilitados para que las pruebas se realicen de manera eficaz y eficiente.	No existe confidencialidad en lectura de escenarios al registrarlos y evidencia, ya que es necesario que tanto desarrollador como tester vean lo que se está probando, pero ningún otro usuario con rol ajeno a desarrollador o tester pueda ver los registros

	desde la fase de diseño.			
5. Las pruebas deberán de ser realizadas con datos no reales y crear bases para pruebas parametrizadas para evitar la exposición de información de clientes.	Este punto es importante para evitar el riesgo de exponer la información confidencial de los clientes.	Los datos ingresados en el ambiente de prueba, deberá ser aceptados por el sistema para emular la información del cliente	Los datos deben de ser procesados de acuerdo al flujo de proceso correspondiente para cada información y estar disponible en las cuentas correspondientes.	Ningún otro usuario podrá modificar las operaciones realizadas por un usuario en específico, esto deberá de comprobarse en cada prueba

IV-C. aplicación práctica del planteamiento.

IV-C.A cambios de entrada 1,2,3

Se muestra el procedimiento propuesto creando un código en UFT para automatizar los casos de pruebas utilizando usuarios macker – checker.

El caso a probar es realizando una transacción entre cuentas corrientes y como primer paso se realiza una consulta de cheques para verificar si tiene fondos y se procede a la protección de cheques (éste proceso se realizó con cuenta de usuario Macker).

Luego se realiza la autorización con la cuenta de usuario checker y mostrará la autorización exitosa.

La transferencia se finaliza y la prueba arroja sus resultados exitosos.

La ilustración del proceso descrito se presenta a continuación por medio de pantallas en ambiente UFT.

a) Figura muestra los resultados de la prueba en general.

Se realiza una transacción de acuerdo al script de prueba.

HP Unified Functional Testing Informe
Prueba: TSUIT-OPERACIONES MAKER CHECKER

TSUIT-OPERACIONES MAKER CHECKER Resumen de resultados

Prueba: TSUIT-OPERACIONES MAKER CHECKER
Nombre del producto: HP Unified Functional Testing
Versión del producto: 12.51
Nombre del resultado: TempResults
Zona horaria: Hora estándar, América Central
Nombre de host: LACSLVSM1W7L09
Sistema operativo: Windows 7
Ejecución Iniciada: 08/12/2016 - 11:01:15
Ejecución finalizada: 08/12/2016 - 11:15:13
Resultado: Passed

Repetición Nº	Resultados
1	Passed

Estado	Horas
Válido	61
Error	0
Advertencias	0

b) Resumen de resultados en las pruebas automatizadas macker checker.

Repetición de prueba 1

Repetición de prueba 1 Resumen:

Repetición Passed

Objeto	Detalles	Resultado	Hora
Resumen de TSUIT-OPERACIONES MAKER CHECKER		Passed	08/12/2016 - 11:01:16

Acción: TSUIT-OPERACIONES MAKER CHECKER

TSUIT-OPERACIONES MAKER CHECKER Resumen de resultados

Acción: TSUIT-OPERACIONES MAKER CHECKER
Ejecución Iniciada: 08/12/2016 - 11:01:16
Ejecución finalizada: 08/12/2016 - 11:15:13
Resultado: Passed

Estado	Horas
Válido	0
Error	0
Advertencias	0

Acción: NBJOP001-Consulta de Cheques [NBJOP001-Consulta de Cheques]

c) Inicia ejecución de script paso a paso consulta de cheques

Acción: NBJOP001-Consulta de Cheques [NBJOP001-Consulta de Cheques]


NBJOP001-Consulta de Cheques [NBJOP001-Consulta de Cheques] Resumen de resultados

Acción: NBJOP001-Consulta de Cheques [NBJOP001-Consulta de Cheques]
Ejecución Iniciada: 08/12/2016 - 11:01:16
Ejecución finalizada: 08/12/2016 - 11:01:44

Resultado: Passed

Estado	Horas
Válido	1
Error	0
Advertencias	0

Nombre de paso: 1.. DETALLE DE CHEQUES Paso Passed

Objeto	Detalles	Resultado	Hora
1.. DETALLE DE CHEQUES		Passed	08/12/2016 - 11:01:44

Acción: NBJOP002-Cheques - Consulta de Cheques [NBJOP002-Cheques - Consulta de Cheques]

d) Validación de chequera, resumen de resultados válidos

Acción: NBJOP003-Compra de Chequeras - Cuenta Corriente [NBJOP003-Compra de Chequeras - Cuenta Corriente]


NBJOP003-Compra de Chequeras - Cuenta Corriente [NBJOP003-Compra de Chequeras - Cuenta Corriente] Resumen de resultados

Acción: NBJOP003-Compra de Chequeras - Cuenta Corriente [NBJOP003-Compra de Chequeras - Cuenta Corriente]
Ejecución Iniciada: 08/12/2016 - 11:02:26
Ejecución finalizada: 08/12/2016 - 11:03:03

Resultado: Passed

Estado	Horas
Válido	2
Error	0
Advertencias	0

Nombre de paso: 1.. COMPRA DE CHEQUERAS3 Paso Passed

Objeto	Detalles	Resultado	Hora
1.. COMPRA DE CHEQUERAS3		Passed	08/12/2016 - 11:03:00

e) Resultados de consultas exitosas

Nombre de paso: 1.1. RESULTADO EXITOSO Paso Passed

Objeto	Detalles	Resultado	Hora
1.1. RESULTADO EXITOSO		Passed	08/12/2016 - 11:03:01

Acción: NBJOP004-Cheques - Conciliaciones de Cheques Protegidos CTE [NBJOP004-Cheques - Conciliaciones de Cheques Protegidos CTE]

NBJOP004-Cheques - Conciliaciones de Cheques Protegidos CTE [NBJOP004-Cheques - Conciliaciones de Cheques Protegidos CTE] Resumen de resultados

Acción: NBJOP004-Cheques - Conciliaciones de Cheques Protegidos CTE [NBJOP004-Cheques - Conciliaciones de Cheques Protegidos CTE]
Ejecución Iniciada: 08/12/2016 - 11:03:03
Ejecución finalizada: 08/12/2016 - 11:03:26

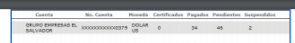
Resultado: Passed

Estado	Horas
Válido	2
Error	0
Advertencias	0

Nombre de paso: 1.. REGISTROS ENCONTRADOS Paso Passed

Objeto	Detalles	Resultado	Hora
1.. REGISTROS ENCONTRADOS		Passed	08/12/2016 - 11:03:26

Nombre de paso: 1.1. REGISTROS ENCONTRADOS Paso Passed

Objeto	Detalles	Resultado	Hora
1.1. REGISTROS ENCONTRADOS		Passed	08/12/2016 - 11:03:26

f) Resultados de validación para cheques protegidos

Acción: NBJOP005-Cheques - Protección de Cheques [NBJOP005-Cheques - Protección de Cheques]

NBJOP005-Cheques - Protección de Cheques [NBJOP005-Cheques - Protección de Cheques] Resumen de resultados

Acción: NBJOP005-Cheques - Protección de Cheques [NBJOP005-Cheques - Protección de Cheques]
Ejecución Iniciada: 08/12/2016 - 11:03:26
Ejecución finalizada: 08/12/2016 - 11:04:59

Resultado: Passed

Estado	Horas
Válido	8
Error	0
Advertencias	0

Nombre de paso: 1.. DETALLE DE CHEQUES Paso Passed

Objeto	Detalles	Resultado	Hora
1.. DETALLE DE CHEQUES		Passed	08/12/2016 - 11:04:03

Nombre de paso: 1.1. DETALLE DE CHEQUES Paso Passed

Objeto	Detalles	Resultado	Hora
1.1. DETALLE DE CHEQUES		Passed	08/12/2016 - 11:04:03

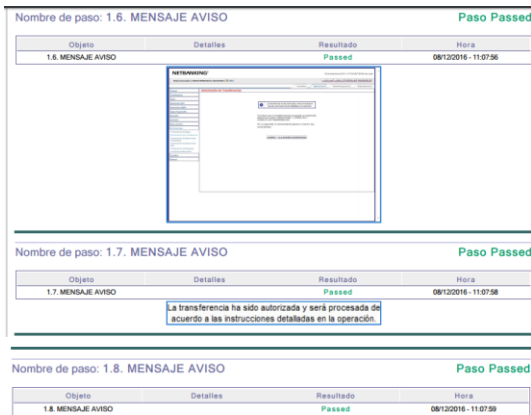
Nombre de paso: 1.2. DETALLE DE CHEQUES Paso Passed

Objeto	Detalles	Resultado	Hora
1.2. DETALLE DE CHEQUES		Passed	08/12/2016 - 11:04:21

Nombre de paso: 1.3. DETALLE DE CHEQUES Paso Passed

Objeto	Detalles	Resultado	Hora
1.3. DETALLE DE CHEQUES		Passed	08/12/2016 - 11:04:21

j) Resultados generales



Luego se debe realizar la misma prueba con cuentas entre cuentas propias de un cliente pero con cuenta de ahorro y cuenta corriente.



IV-C.2 cambio de entrada 4 (Matriz de trazabilidad).

Éste documento, es utilizado para que los usuarios coloquen paso a paso las pruebas

que solicitan que se realicen en la aplicación y contribuye a garantizar la integridad y disponibilidad de la información, es la que conecta la fase inicial hasta la de construcción, contiene una hoja para el departamento de sistemas, que son las personas encargadas del desarrollo de la aplicación y una hoja para que los tester coloquen sus pruebas también, es hablando prácticamente una guía para que los usuarios, sistemas y tester alinien sus pruebas y puedan obtener el resultado solicitado en la fase inicial del requerimiento; éste documento

se presentará completado en comité técnico por lo menos en un 25% (llenar la hoja de usuario) para poder ser presentado a las todo el personal involucrado en las pruebas y generar consultas en dicho comité.

Tabla 2. Elementos que la matriz de trazabilidad

Elementos que aportará la matriz de trazabilidad a la seguridad de la información		
Integridad	Disponibilidad	Confidencialidad
Garantizará la ruta de seguimiento del requerimiento la cual deberá de cotejar los resultados solicitados por el usuario versus los resultados evidenciados por los desarrolladores y testers.	Asegurará que la información esté disponible, teniendo como evidencia las capturas de pantalla de los escenarios probados en cada fase del SDLC.	La matriz de trazabilidad únicamente involucrará las tres áreas interesadas, las cuales son: Usuarios, Sistemas, Tester y solo los usuarios asignados especificados en los BRD de cada proyecto, podrán tener acceso dicha matriz.

Figura 8. Matriz de trazabilidad que debe de ser implementada desde la fase de definición

a) Hoja con datos generales

MATRIZ DE TRAZABILIDAD

Información General del Requerimiento

Item	Descripción
Requerimiento (LLENADO DE MSR)	
WID	PV000000
Nombre PVE / PTS (PV000000 (LLENADO DE MSR)	
Pal(es)	SV
WorkRequest	CGPR0000000
Business Analyst (NOMBRE DE BA 1)	
Business Analyst (NOMBRE DE BA 2)	
Developer (NOMBRE DE DESARROLLADOR)	
Technical Leader (NOMBRE COORDINADOR DE DESARROLLO)	
Tester (NOMBRE DEL TEST ANALIST)	
Test Manager (NOMBRE DE TEST MANAGER)	
Business Analyst (NOMBRE DE BA)	

b) Hoja de usuario

Requerimiento del Usuario (Business Analyst)

Requerimiento del Usuario

Business Analyst (NOMBRE DE BA 1) 0 Ordenar por

Business Analyst (NOMBRE DE BA 2) 0 Adición TSTF03 & QC a

Num. Req. Usuario	Acción	Descripción del Requerimiento	Prioridad Req. Usuario
BRD-0010			Alta
BRD-0020			
BRD-0030			
BRD-0040			
BRD-0050			
BRD-0060			
BRD-0070			
BRD-0080			

Completar la parte REQUERIMIENTOS DE USUARIO

- * TIPO DE REQUERIMIENTO : Debemos de colocar BRD
- * NUMERO DE REQUERIMIENTO DE USUARIO : se colocara el una combinacion con el Numero de solicitud del BRD ejemplo BRD-0010
- * ACCION : Se complementa dependiendo de la accion que se utilizara para cubrir la solicitud en el BRD.
- * DESCRIPCION DEL REQUERIMIENTO : Complementar con una pequeña introduccion por caso solicitado en el BRD
- * PRIORIDAD REQUERIMIENTO USUARIO : Colocar la prioridad del caso según criterio de Negocio

e) Hoja designada para tester (se llenará en la fase de construcción)

Matriz de Casos de Prueba (Usuario)

Matriz de Casos de Prueba

Nombre Proyecto: [] Nombre Requerimiento: [] Fecha Actualización: 13/11/2016

Nombre Proyecto	Nombre Requerimiento	Fecha Actualización
		13/11/2016

COMPLETAR HOJA Matriz de Casos de Prueba

se debera de completar con los casos de prueba del usuario llenando todos los campos pero dando prioridad a los campos (Requeridos)

La columna Nombre de Caso de Prueba (Requerido) Sera la que se importara a la hoja MSR-RTM

c) Hoja de sistemas (desarrollo)

Requerimiento de Sistema (Developer)

Requerimiento de Sistema

Developer (NOMBRE DE DESARROLLADOR) 0

Technical Leader (NOMBRE COORDINADOR DE D O) 0

Num. Req. Sistema	Nombre del Sistema	Descripción Requerimiento de Sistema	Prioridad Req. Sistema
RSI-0010			Baja
RSI-0020			
RSI-0030			
RSI-0040			
RSI-0050			
RSI-0060			
RSI-0070			
RSI-0080			

Completar la parte REQUERIMIENTOS DE SISTEMA

- * NUMERO DE REQUERIMIENTO DE SISTEMA : Complementar combinando RSI mas el numero de caso del BRD que se esta cubriendo con el desarrollo
- * NOMBRE DEL SISTEMA : Completar con el nombre del sistema que se utilizara para solucionar el caso planteado en el BRD
- * DESCRIPCION DEL REQUERIMIENTO DEL SISTEMA : Complementar con una pequeña explicacion por caso donde se plantea la solución al requerimiento
- * PRIORIDAD REQUERIMIENTO SISTEMA : Colocar la prioridad del caso según criterio del Desarrollador

TSTQC (Tester QA)

Script de Pruebas para Quality Center

COMPLETAR HOJA Script de Pruebas para Quality Center

Esta hoja se completara por parte del Equipo de QA, y seran los casos y script de pruebas que servirán para las pruebas SIT, deben de estar enumeradas de tal forma que sea identificable la trazabilidad contra los requerimientos del usuario, sistema, y contra los casos de prueba del UAT.

Se debera de llenar todos los campos, dandole enfasis a la columna de Test Name que es la que se importa a la hoja MSR-RTM

d) Primera actualización después de ser presentada en comité técnico en la fase de diseño

Actualizar MSR-RTM II parte

COMPLETAR la hoja MSR-RTM

Para completar esta Hoja despues de haber completado las hojas de Matriz de Casos de Prueba (UAT) y TSTQC (SIT) debemos de ubicarnos en la hoja Req_Usuario_Sistema y darle click al boton Adición TSTF03 & QC, eso nos actualizara los datos; por lo que ahora nos podremos colocar en la columna Caso de Prueba UAT (Asociado) y Caso de Prueba QC (Asociado) donde ahora nos desplegara los casos que completamos anteriormente y Asociar casos de prueba a cada una de las solicitudes requerimiento.

Asegurandonos de cubrir todos los requerimientos planteados por el negocio

ACTUALIZAR MSR-RTM I parte

ACTUALIZAR HOJA MSR - RTM

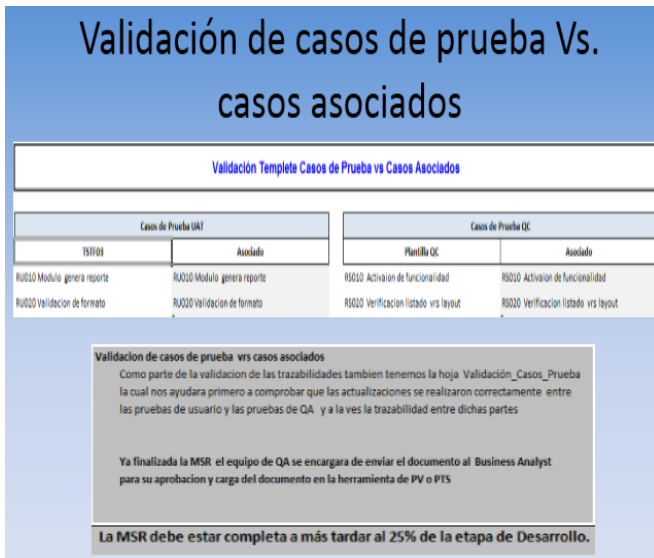
* Despues de haber completado la hoja de req_Usuario_Sistema nos ubicamos en la hoja MSR-RTM le damos un Click al boton de **Actualizar MSR-RTM** entonces nos mostrara los datos competados en la hoja anterior y podremos observar la TRAZABILIDAD existente entre los casos de Usuario y los de Sistema.

* Aclaración importante : todos los casos de Usuario deberan de esta cubiertos al menos con un caso de sistema

* Despues de esta revision la MSR esta lista para ser subida a la herramienta y podria ingresar al comité



f) Se validan los resultados de los casos de prueba entre los solicitados y lo ejecutado por los testers.



IV-D. Cambios propuestos en procedimientos de salida para garantizar la integridad y disponibilidad de la información en las pruebas.

Procedimientos de salida	Comentarios generales	Comentarios relacionados a la Seguridad de la información		
		Integridad	Disponibilidad	Confidencialidad
Los usuarios tester, no deben de tener rol de administrador	Para garantizar la seguridad de la información	Garantizará que desarrollo realice,	No aplica disponibilidad en este punto por la	Implica el bloqueo o inhabilitación de las ventanas de

	ón y restringir la administración de la plataforma a NBJ.	no sufran alteraciones accidentales por tester.	restricción en el rol de tester	administración dentro de la plataforma Netbanking
Las pruebas automatizadas deberán programarse después de las pruebas manuales pero se registrarán siempre en QC.	Esto garantizará la optimización de tiempo en la ejecución de pruebas automatizadas, contando solamente con pruebas efectivas.	Los escenarios registrados quedarán registrados en QC sin ser alterados.	Para asegurar que los escenarios automatizados serán los efectivos y no fallarán al momento de correr el TSUIT en UFT.	No aplica confidencialidad en este punto.
Implementar documentación nueva a presentar para generar un mejor control en las pruebas automatizadas.	Son los documentos mencionados anteriormente: BRD, Funcional requirements, UAT plan, Sanity check.	Ayudará a asegurar el hilo de lo solicitado por el cliente con el resultado final.	La documentación estará siempre disponible para los desarrolladores y tester en la herramienta Plan view o PTS.	solo podrá ser descargada por testers y desarrolladores.
Elaborar un flujo de procedimiento para el tratamiento de los defectos, pues en la actualidad no existe tal procedimiento (ver figura 9)	Permitirá un mejor seguimiento a los defectos reportados en las diferentes fases	Seguimiento de los defectos para registrar los avances de resolución.	los registros estarán siempre disponibles en la herramienta QC.	No aplica confidencialidad en este punto.

Figura 9. Procedimiento propuesto para los defectos en detectados en las pruebas automatizadas

Flujo de Defectos Quality Center					
Rol	Descripción	Flujo	Campos que Completa/Valida	Notificación via Email QC - SIT	Notificación via Email QC - UAT
Tester/Usuario	Crea defecto	New	Aplicación, Severidad, Categoría, Subcategoría, Comentario		
Tester/Usuario	Coloca en estado Open	Open	Asignado a	Analista, Coordinador	Analista, Coordinador
Analista	Coloca en estado Review	Review	Severidad, Categoría, Subcategoría, Comentario	Tester, Coordinador	Usuario Funcional, BR, Coordinador
Analista	Verificar que este bien documentado	OK			
Analista	Si se tienen observaciones se coloca en estado Pending.	Pending	Comentario	Tester, Coordinador	Usuario Funcional, BR, Coordinador
Analista	Una vez solventado el defecto se coloca en Fixed	Fixed	Comentario	Tester, Coordinador	Usuario Funcional, BR, Coordinador
Analista	Aplicar el cambio en el ambiente y colocar en Test Ready.	Test Ready	Comentario	Tester, Coordinador	Usuario Funcional, BR, Coordinador
Analista	Colocar en estado Re-test	Re-Test	Comentario	Tester, Coordinador	Usuario Funcional, BR, Coordinador
Tester/Usuario	Realizar nuevamente la prueba y verifica si esta ok.	OK			
Tester/Usuario	Si el issue persiste coloca en estado Re-open	Re-open	Comentario	Analista, Coordinador	BR, Coordinador
Test Lead	Si el issue fue superado coloca en estado Closed.	Closed	Comentario	Tester, Coordinador	BR, Coordinador

En la figura 9 describe el proceso que deberá realizarse para la resolución de errores encontrados a nivel de construcción (hallazgos detectados por testers de QA o UAT, por ésta razón en el cuadro dice Tester o usuario).

En el que inicialmente se detecta un error en las pruebas se reporta en la herramienta Clear Quest colocándole el status Open asignándole un responsable en éste caso deberá ser el programador (Analista) de la aplicación, el cual lo cambiará a status Review.

Si es un error válido lo pasará a status Pending y quedará en responsabilidad de desarrollo.

Al encontrar la solución del error, lo pasará a status fixed

Test Ready mientras se realiza el cambio a ambiente SIT o producción, según sea el caso.

yo lo dejará en manos de Calidad nuevamente cambiando su status a Re test

Calidad tomará el caso cambiando a status OK y si es error persiste lo colocará en status Reopen, sino lo colocará en status Close finalizando el proceso de seguimiento al error.

Tabla 3.

Elementos que aportará el flujo de resolución de defectos a la seguridad de la calidad y seguridad de la información		
Integridad	Disponibilidad	Confidencialidad
Seguimiento de los defectos para registrar los avances de resolución colocando sus observaciones y cierres	los registros estarán siempre disponible en la herramienta QC	No aplica confidencialidad en éste punto ya que los defectos pueden ser vistos hasta por el Project manager, que es quien lleva el seguimiento del requerimiento promoviendo la resolución de los defectos que aún no están resueltos, reportando y actualizando en las herramientas de Plan View y PTS

IV-E. Otras propuestas a documentación existente a tomar en cuenta:

Actualmente existe un entregable llamado *Plan de prueba* (Test Plan), en el cual también hay que realizarle una modificación relacionada a la seguridad de la información debe de incluir un aspecto importante a tomar en cuenta, en la plataforma NBJ (Netbanking Java) y cualquier otra que utilice esquemas de seguridad roles/usuarios.

Deberá incluir una sección llamada *Testing Strategy* (estrategia de pruebas), deberá incorporarse una tabla destinada para identificar detalladamente las lista de los Usuarios, Oficinas y Roles, con los cuales se van a llevar a cabo las pruebas de UAT (User Assesment test). El rol responsable.

IV-E.1 Usuarios para ambientes UAT

Tabla 4. Ejemplo de información requerida

<i>SOEID / Usuarios</i>	<i>Oficina</i>	<i>Roles</i>

IV-E.2 Plan de mejora continua

La administración de la calidad total, requiere de un proceso constante, que se llamará mejoramiento continuo, donde la perfección nunca se logra, pero siempre se busca.

Figura 10. Seguimiento propuesto para una mejora



A manera de ejemplo a continuación se presenta un esquema, para proponer una mejora, que puede ser modificado de acuerdo a la necesidad de cada institución y ser presentada en reuniones gerenciales periódicas.

Figura 11. Ejemplo de plantilla para proponer una mejora.

Identificación	Fecha	Propuesta por	Origen de la propuesta	Proceso asociado	Descripción de Mejora	Resultado de valoración	Aprobado Por	Justificación
1	01/06/2016	Aida de Aveller	Análisis de indicadores	Ingeniería	Informar al equipo de desarrollo la calificación por parte de equipo SIT que se otorga al proceso de desarrollo y pruebas unitarias realizadas.	Aprobado	Veronica Triep	

Responsable	Personal requerido	Actividades	Fecha Estimada de Finalización	Entregables	Estado	Comentarios
Aroldo Aveller, Melvin Hernandez	Judith Cañas, Hector Villalobos, Jaime Ayala, Orlando Baulista, Freddy López, Dikil Moreno, Sandra Araya	Definir evaluación y contenido de informe Enviar informe a equipo de desarrollo	14/05/2016	Informe de calificación de desarrollo y pruebas unitarias realizadas	Retrasada	Pendiente reunión con Lucy para revisión de FIS

Fecha cierre	% Avance Periodo anterior	% Avance Real
	50%	90%

Aportando a la seguridad de la información los siguientes elementos:

Tabla 5. Elementos de seguridad de la información.

Integridad	Disponibilidad	Confidencialidad
Dejar evidencia de la propuesta realizada, la toma de atención y seguimiento; relegando la aprobación o rechazo de la misma	No aplicará la disponibilidad en éste caso, al menos que la propuesta se refiera a un tema de mejoras en servidores y temas relacionados	No aplica la confidencialidad, debido a que éste documento debe de dejarse anclado en un sharepoint para que los jefes de área puedan disponer del mismo y proponer mejoras a sus gerentes de área.

Conclusiones:

- Para garantizar la Seguridad de la información, es necesario reforzar puntos en la documentación ya existente e implementar nuevos controles.
- Se deberán crear bases con datos no reales para hacer pruebas en los diferentes módulos de la plataforma.
- También es importante reforzar puntos de los procesos de entrada y salida actuales.
- Los procesos actuales referentes al manejo de usuarios son muy vulnerables y no poseen metodología supervisada para su administración.
- En su totalidad no se estaban aplicando procesos de Seguridad de la información en cada fase del SDLC (Software Development Life Cycle).
- Es necesaria una capacitación continua con los empleados para reforzar los procesos que garantizarán la seguridad de la información.
- Las bases con las que se inicia un requerimiento actualmente, no son suficientes para garantizar la integridad y la disponibilidad de la información, por lo que el hilo de la información inicial se iba perdiendo y como consecuencia se entregaba un resultado no solicitado.

Glosario:

- **IDE:** Integrated Development Environment (**IDE**), es una aplicación informática que proporciona

servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

- **QC:** Quality Center, herramienta para registrar casos de prueba y sus resultados.
- **SIZIN:** Dimensionamiento en días de la ejecución de una etapa del requerimiento.
- **SIT:** System interactive Test, es la fase o etapa del requerimiento donde se realizan las pruebas automatizadas.
- **UAT:** User assesment test, es la etapa del requerimiento donde los usuarios expertos realizan las
- Pruebas en las aplicaciones desarrolladas.
- **SDLC:** Solution Delivery Life cycle (Ciclo de vida para la entre de un Proyecto).
- **NBJ:** Abreviación de Netbanking Java (Plataforma bancaria en línea).
- **BRD:** Business requirement document, es el documento con el que se recibe el requerimiento en la fase de contrucción.
- **PRD:** abreviación de Producción, que es, poner en funcionamiento en tiempo real el software solicitado en la plataforma correspondiente.
- **Tester:** Rol de un usuario de que realiza las pruebas en fase SIT.
- **Script:** Se refiere a cada código individual que se realiza paso a paso en los diferentes escenarios de pruebas.
- **Tsuit:** Es el conjunto de scripts que se ejecutan de manera consecutiva

para optimizar tiempos en la ejecución de fases de pruebas SIT.

- **SLA:** Service Level Agreement. Acuerdo a nivel de servicio.
- **Comité Técnico:** Reunión de trabajo en el que participan los diferentes departamentos involucrados en un requerimiento, los cuales son: Calidad (QA), Project Manager (PMO), Desarrollo, BR y BA
- **BR:** Business representative, persona que interviene entre el negocio y la institución.
- **BA:** Business Analyst, persona que tiene la habilidad de analizar la rentabilidad y pontencialidad de un negocio.
- **HPE UFT :** Unified Functional Testing, lenguaje de programación para pruebas automatizadas.
- **Test Plan:** Plan de pruebas que se presenta antes de ejecutar los casos de pruebas automatizadas.
- **CISC:** Nombre de la institución en el que se ha basado el presente trabajo. (Citi integrated Service Center)
- **SISA:** Seguros e inversiones
- **CHG:** Change Request, documento para solicitar un cambio.

Anexo

En el escrito se menciona la importancia del sizing en la primer fase del SDLC (Software Development Life Cycle), a continuación se detalla un poco como debe de ser un formato

Definición de Sizing

Es la parte en donde se define, el tiempo, los recursos, h

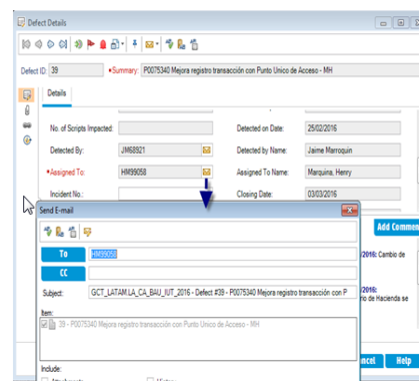
erramientas y los porcentajes y observaciones sobre cada proyecto que se le realizaran pruebas de parte de control de calidad y el cual es muy importante ya que nos proporciona un estimado financiero o términos en que se gestionara un proyecto u tarea.

Figura 11. Formato de Sizing

OBSERVACIONES - RIESGOS - ASSUMPTIONS													
#	TIPO	DESCRIPCIÓN	RESPONSABLE (Quien lo rep.)	ESTATUS									
1				PENDIENTE									
2				PENDIENTE									
3				PENDIENTE									
4				PENDIENTE									
5				PENDIENTE									
6				PENDIENTE									
7				PENDIENTE									

A continuación se muestra el paso para la notificación del Defecto al Analista y al Coordinador de desarrollo.

NOTIFICACION DE DEFECTO



El siguiente paso despues de la generacion del DEFECTO es la NOTIFICACION utilizando la herramienta de QC se le mandara un correo al Analista y al Coordinador de Desarrollo seleccionando los sold y dando un clic al (Sobre) en la aplicacion

Bibliografía:

- ❖ *Reporte de evaluación y análisis de riesgos v.2.1, Verónica Elizabeth Trejo de Hernández 2015.*
- ❖ *Diagnóstico de seguridad de la información en automatización de plataforma bancaria Netbanking Java Junio 2015*
- ❖ *Enciclopedia de la Seguridad de la información, segunda edición, 2014.*
- ❖ *Normas CMMI 2016*