

ITF – PLAN DE PRUEBAS

PROYECTO PP

INTRODUCCIÓN

OBJETIVOS

Las pruebas buscan encontrar Incidentes (Defectos) mediante el uso de la aplicación. Estos ocurren cuando el software no hace lo requerido o hace algo que no debe.

El presente documento tiene por finalidad describir las pautas y estrategia a seguir para llevar a cabo las diferentes pruebas inherentes al proyecto, suministrando un marco para la organización, planificación y gestión del proyecto de prueba.

ALCANCE

A continuación se indica qué funcionalidades y procesos se van a probar y cuáles están fuera del alcance.

COMPONENTES Y CARACTERÍSTICAS QUE SERÁN PROBADOS.

- Se probarán las características funcionales asociadas a los procesos detallados en el pliego:
 - Proceso de Mercado y Ventas
 - Procesos de Petición de Servicio y Contratación
 - Procesos de Lectura de Medidores
 - Procesos de Facturación
 - Procesos de Cobro y Gestión de Deuda
 - Procesos de Atención al Cliente y Reclamaciones
 - Procesos de Ordenes de Trabajo
 - Procesos de Irregularidades
 - Control de Gestión

Si bien existen funcionalidades de base de UUU, es necesario implementar todos los procesos tales como se definieron por lo que se desarrollarán GAPS, se implementarán interfaces y generarán reportes para XX (WSHP020, WSHP022, WSHP032, WSHP0RPT).

- Configuraciones realizadas por los equipos funcionales.
- Proceso de conversión de datos



COMPONENTES, CARACTERÍSTICAS Y ACTIVIDADES QUE ESTÁN FUERA DEL ALCANCE.

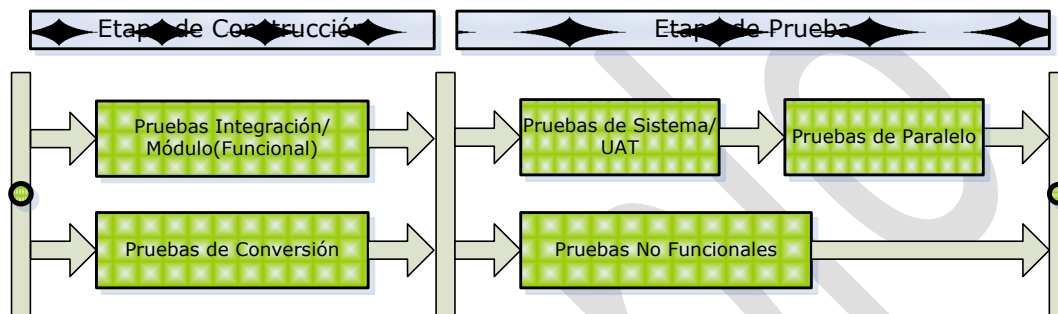
- Pruebas Unitarias: Se asume que todo desarrollador, configurador y equipo de conversión realiza las pruebas unitarias previas a dar por finalizado su trabajo, por lo cual dichas pruebas no forman parte de la estrategia del presente plan. Y en la medida que se registren deberían poder estar a disposición.



ENFOQUE

ESTRATEGIA

Se considera la actividad de pruebas sobre todo el ciclo de vida del proyecto, de forma de obtener así un producto de alta calidad. La ejecución de las pruebas se realizará en la Etapa de Construcción y Prueba. Ver figura.



Las Liberaciones estarán asociadas a las Pruebas Funcionales y los Entregables estarán asociados a las Pruebas de Conversión.

LIBERACIONES DE FUNCIONALIDADES

Se entiende por Liberación a la agrupación lógica de funcionalidades que surgen de los (WSHP20,22,32) que son planificadas para ser incorporadas en conjunto desde el ambiente Master. Su contenido estará detallado en la Nota de Liberación. Ver Figura.

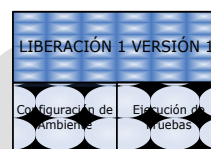
Se estima la realización al menos 5 Liberaciones que serán sometidas a Pruebas a los efectos de poder certificar que la implementación se ajusta a lo especificado. Las mismas estarán incluidas en el Plan de Liberación cuyo objetivo es definir el orden en que los componentes del sistema serán implementados e integrados, de forma de constituir una Liberación-Versión a medida que se vayan



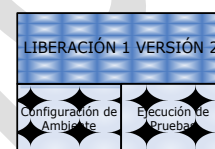
realizando entregas. Dicho plan es necesario a los efectos de planificar el diseño de las pruebas que se van a realizar.

Una Liberación estará conformada por la incorporación de: Gaps, Configuraciones, Interfaces, Reportes y corrección de incidencias. Es posible que en alguna de las Liberaciones se le incluyan Datos de Conversión. La última liberación se corresponderá con la finalización de la etapa de Construcción.

Cada Liberación estará asociada a una o más versiones, siendo una versión una agrupación de funcionalidades y resolución de incidencias que es migrada desde el ambiente Master al ambiente de Testing



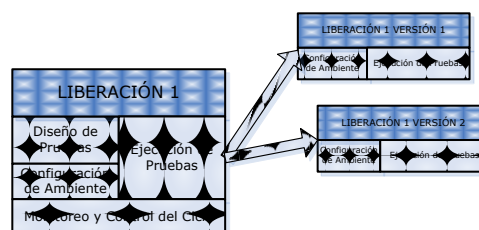
Un ciclo de prueba consiste en un conjunto ordenado de actividades que es realizado por el Equipo de Testing para probar una Liberación-Versión del proyecto en el ambiente de Testing. Ver Figura.



Cada Liberación-Versión será sometida a un ciclo de pruebas, donde el primero está asociado a la ejecución de un conjunto de casos de prueba que se corresponden a las funcionalidades a verificar, y al resto de los ciclos de pruebas se le adicionara la verificación de la resolución de las incidencias reportadas.

En un ciclo de prueba se podrán ejecutar uno, algunos o todos los casos de prueba asociados a las funcionalidades incluidas en una Liberación-Versión y la regresión de aquéllos que se consideren necesarios para validar la corrección del incidente.

Cada Liberación consta de 4 fases bien definidas. Estas son: Configuración del Ambiente de Testing, Diseño y Rediseño de las Pruebas, Ejecución de las Pruebas, Monitoreo y Control del Ciclo



Cada Liberación-Versión consta de 2 fases identificadas por la Configuración del Ambiente y la Ejecución de las Pruebas. Ver Figura.



Fase: Diseño de las Pruebas. Durante esta fase se realiza el diseño de los casos de prueba a partir de la especificación funcional realizada. Se entiende por el proceso de definición de las pruebas (o CPs) que mejor permitan verificar el cumplimiento de los requerimientos. Se informarán los Casos de Prueba diseñados a los usuarios y desarrolladores, de forma de incluir si es necesario casos que estimen pertinentes. Esta actividad consiste en dar instrucciones detalladas acerca de: que funcionalidad se va a probar, cómo llevar a cabo la prueba para alcanzar los objetivos deseados, qué método de prueba, que se comparara para determinar si la funcionalidad pasa o no pasa la prueba. Se explicitarán riesgos, contingencias y el detalle de cómo se va a realizar la prueba. Esta Fase se debe iniciar lo más temprano posible, en el entendido que no es necesario aguardar a una liberación para comenzar. La referencia al Rediseño consiste a la actualización, modificación e inserción de pruebas que se realizan durante el proceso de pruebas.

Fase: Configuración del Ambiente: El objetivo de esta fase es configurar el o los ambientes de pruebas necesarios a los efectos de poder efectuar las tareas de diseño y ejecución de las pruebas. Además, se deben instalar los componentes necesarios y el software a probar en la Liberación-Versión correspondiente a cada ciclo de prueba.

Fase: Ejecución de las Pruebas. Durante esta fase se procede a ejecutar las pruebas previamente diseñadas para el ciclo, contrastando el comportamiento esperado con su comportamiento real, analizando las diferencias y reportando los resultados. En otras palabras lo que se busca es comparar el resultado esperado contra el resultado obtenido. En esta fase también se debe de incluir la preparación de los datos de pruebas para la ejecución y si van a ser armados en forma manual, o por script.

Fase: Monitoreo y Control del Ciclo. Esta fase ocurre en paralelo a las ya mencionadas anteriormente, se efectúa un seguimiento del ciclo, revisan tiempos, alcances, riesgos, calidad, entre otras cosas. Además al inicio se revisa la planificación del ciclo de prueba y se ajusta la misma según las estimaciones y posteriores mediciones realizadas en los ciclos anteriores. Implica también realizar reportes de avance del proyecto, mostrando el avance de las pruebas del ciclo así como la calidad de la Liberación-Versión probada, actualización de lista de riesgos, etc. De esta forma se determinara a partir de los resultados de las pruebas si dicho ciclo cumplió con los objetivos propuestos, si el alcance de los casos de pruebas propuestos fue el correcto y como afectan los incidentes detectados.

Las liberaciones van a ser incrementales que estarán alineadas con el Plan de Liberación según el ejemplo siguiente:



	Liberación 1 Versión 1	Liberación 1 Versión 2	Liberación 1 Versión 3	Liberación 2 Versión 1	Liberación 2 Versión 2
Funcionalidad/Componente	Ciclo	Ciclo	Ciclo	Ciclo	Ciclo
Funcionalidad 1	X	X	X	X	X
Funcionalidad 2	X	X	X	X	X
Funcionalidad 3				X	X
Funcionalidad 4				X	X
Funcionalidad 5				X	X

ENTREGABLES DE CONVERSIÓN

Pruebas de Conversión: Se estableció con el equipo de conversión una estrategia de entregas de datos en forma incremental a medida que el equipo los va convirtiendo a UUU.

El criterio de Entregas es independiente de las Liberaciones, lo que implicara la generación de datos para las pruebas funcionales y en la medida que efectúen Entregas que estén alineadas con las Liberaciones podrán ser utilizadas en las pruebas funcionales.

Se establece una entrega mensual de datos que sigue los siguientes criterios:

- El contenido de la Entrega de Conversión será detallado.
- Se le realizará una Prueba de Humo a cada Entrega, Dicha Prueba de Humo se realizará en el Ambiente TestQA3 a los efectos independizar las Pruebas Funcionales de las Pruebas de Conversión.
- A partir del resultado satisfactorio de la Prueba de Humo se realizarán las Pruebas de Conversión diseñadas a tales efectos.
- En cada nueva entrega se incorporaran nuevos tipos de datos, además de los ya convertidos y la corrección de incidencias.



- Para la primera entrega completa (incluyendo todos los datos convertidos “Abril/11”) se le realizara una Prueba de Humo en primera instancia, luego se realizará un ciclo de certificación de la conversión y a continuación se le va realizar un ciclo de pruebas completo de todas las funcionalidades disponibles a ser probadas.
- Para las Pruebas de Sistema y UAT, se establecen 3 simulacros de Entregas. Dichos simulacros se diferencian de los anteriores en que ellos contienen el total de los datos que serán convertidos por el proyecto (salvo la primera entrega completa).

El detalle con el alcance del Entregable de Conversión, y los criterios de aceptación de cada uno de ellos (por parte del equipo de conversión) está disponible en el SharePoint del proyecto XX en la xxxx. Para la verificación de la conversión se diseñarán un conjunto de casos de prueba que convaliden los datos de la migración. En particular se tomarán caminos tales como:

- Verificación de información utilizando reportes que se puedan comparar como salida de ambos sistemas SXX y UUU.
- Selección de datos que se comparan en ambos Sistemas dirigidas a clientes Top.
- Selección de varios subconjuntos de clientes (o muestras) que cumplan con determinadas características.
- Comparación de archivos que son generados por la interfase hacia otros sistemas, por ejemplo SYU, comparando las interfases de SXX y UUU.

TIPOS DE PRUEBAS

A continuación se definen los tipos de pruebas que se realizarán en el marco del presente proyecto.

Las pruebas podemos agruparlas en Pruebas Funcionales y Pruebas No Funcionales.

Las pruebas funcionales están desarrolladas bajo la perspectiva del usuario, confirmando que el sistema hace lo que los usuarios esperan que haga y lo que las especificaciones indican. Las pruebas no funcionales, buscan mostrar que el sistema se desempeñe (performance) según lo especificado o requerido.

PRUEBAS FUNCIONALES

Pruebas de Humo: Son un conjunto de pruebas aplicadas a cada nueva Liberación-Versión y Entrega, su objetivo es validar que las funcionalidades básicas de la versión se cumplen según lo especificado y para la entrega de datos verificar que la información convertida es la correcta. Estas pruebas buscan grandes inestabilidades o elementos claves faltantes o defectuosos, que hacen imposible realizar las pruebas como fue planificado para el ciclo. Si la Liberación-Versión



no pasa las pruebas de humo, no se comienza la ejecución del ciclo de pruebas y lo mismo sucede con la Entrega.

Pruebas Exploratorias: En este tipo de pruebas se trata de ejecutar las pruebas a medida que se está pensando en ellas, sin insumir tiempo en confeccionar o preparar casos de prueba. En otras palabras, es cualquier prueba en la cual quien prueba controla activamente el diseño de las pruebas mientras las pruebas se ejecutan, y utiliza la información obtenida mientras prueba para diseñar nuevas pruebas o mejorar las actuales.

Prueba de Integración/Modulo (Funcional): El objetivo de las pruebas funcionales de módulo es validar si el comportamiento observado de la aplicación cumple o no con sus especificaciones. La prueba funcional toma el punto de vista del usuario basada en los requerimientos definidos oportunamente. Las funciones son probadas ingresando las entradas y examinando las salidas. Para realizar las pruebas funcionales nos basamos en el documento de requerimientos, que a partir del análisis del mismo se crean los casos de prueba. Técnicas como partición de equivalencia, análisis del valor límite, grafo causa-efecto y conjetura de errores son utilizadas para determinar los casos de prueba.

Cuando los componentes individuales están probados (pruebas unitarias realizadas por el desarrollador) y cuando estos mismos componentes fueron probados funcionalmente y se van a integrar dentro de las pruebas de integración. Estas buscan encontrar fallas en las interacciones entre los distintos componentes a partir de su integración. El aplicativo se construye y prueba en pequeños incrementos, en los cuales resulta más fácil aislar y corregir los errores. Dentro de la estrategia incremental, las pruebas de incrementos se planificaran en función del Plan de Liberación. Al utilizar el enfoque incremental cuando se realizan las pruebas directamente están implícitas las pruebas integrales.

Se podrá hacer mención a las Pruebas de Integración o Pruebas Funcionales indistintamente.

Pruebas de Regresión: Las Pruebas de Regresión tienen por un lado tienen como objetivo verificar que no ocurrió una regresión en la calidad del producto luego de un cambio, asegurando que los cambios no introducen un comportamiento no deseado o errores adicionales y por otro la verificación de la correctitud de la resolución de las incidencias. Implican la re-ejecución de alguna o todas las pruebas realizadas anteriormente y relacionadas, dependiendo del alcance de la verificación.

Pruebas de Permisos: Estas pruebas buscan identificar problemas de permisos de accesibilidad para las distintas actividades y operativas que se realicen según los perfiles y roles que se definieron. Este tipo de pruebas podrá estar incluido en la Pruebas de Integración/Modulo (Funcional).



Pruebas de Sistema y de Aceptación de Usuario: Estas son las pruebas que se realizan sobre el sistema completo. Luego de la Prueba de Integración/Modulo (Funcional), el aplicativo está completamente ensamblado; se han encontrado, corregido los errores, ajustado definiciones, etc. y todo indica que se pueden comenzar a realizar las Pruebas de Sistema y las de Aceptación de Usuario (UAT). La validación de la aplicación se consigue mediante un conjunto de pruebas que demuestran la conformidad con los requisitos detallados en la especificación de requerimientos. Las Pruebas de Aceptación de Usuario (UAT) si bien son ejecutadas por usuarios finales tienen como objetivo lograr la aprobación del aplicativo desde los usuarios.

Pruebas en Paralelo: Las pruebas en paralelo se usan cuando un sistema nuevo sustituye a otro viejo, como es el caso. Durante estas pruebas ambos sistemas funcionan a la vez. Se comparan los resultados de ambos sistemas para conocer si el sistema nuevo funciona adecuadamente.

Pruebas de Conversión: Se realizaran casos de prueba funcionales para testear los resultados de la conversión. Estos casos de prueba se realizaran con el objetivo de verificar los datos, los tipos de los datos y los campos en que fueron importados en UUU.

PRUEBAS NO FUNCIONALES

Pruebas de Desempeño: Realizar las pruebas que miden los tiempos de respuesta y otros requerimientos sensibles al tiempo. La meta de las pruebas de desempeño es verificar y validar que los requerimientos de desempeño han sido alcanzados.

Prueba de Volumen: Esta prueba implica someter al sistema a grandes volúmenes de datos. El propósito es conocer si el sistema puede manejar el volumen de datos especificado.

Prueba de Estrés (esfuerzo): Esta prueba implica someter al sistema a grandes cargas o esfuerzos considerables. Un esfuerzo grande es un pico de volúmenes de datos, normalmente por encima de los límites especificados, en un corto periodo de tiempo. No solo se debe considerar volúmenes de datos sino también de usuarios, etc.

EJECUCIÓN DE PRUEBAS

Funcional: Por cada Liberación-Versión se realizarán los siguientes pasos:

- Prueba de Humo
 - Ejecución y Registro de Casos de Prueba identificados para ese fin.
- Prueba Funcional
 - Ejecución y Registro de Casos de Prueba ejecutados
 - Registro de Incidentes identificados



- Informe de resultados de la Liberación y Ciclos de Prueba

Conversión: Por cada Entrega se realizarán los siguientes pasos:

- Prueba de Humo
 - Ejecución de Casos de Prueba identificados para ese fin.
- Prueba de Conversión
 - Ejecución y Registro de Casos de Prueba ejecutados
 - Registro de Incidentes identificados
- Informe de resultados de la Entrega y Ciclos de Prueba

CUBRIMIENTO DE LAS PRUEBAS

Para asegurar un alto cubrimiento de las pruebas, se ejecutarán los tipos de pruebas antes enunciados.

Tipo de Prueba	Cubrimiento
Pruebas de Humo	A determinar dependiendo de la Liberación durante el ciclo de pruebas, en la Fase de Diseño de Pruebas.
Pruebas Exploratorias	Técnica complementaria al diseño de los casos de prueba. Se ejecutará cuando por algún motivo (por ejemplo tiempo) además de basarse en los casos de prueba ya diseñados, sea necesario incluir casos de prueba que no han sido diseñados aún.
Pruebas de Integración / Módulo (Funcional)	Abarcara la ejecución del conjunto de casos de prueba diseñados de que se identificaron en la Fase de Diseño, para todos los procesos relevados del negocio que se corresponden con los Workshops. Si el tiempo disponible no es suficiente para la ejecución, se procederá en primera instancia a la ejecución de las de alta prioridad, las restantes serán ejecutadas mediante pruebas exploratorias.
Pruebas de Regresión	En primera instancia se verificara la correctitud de la resolución de la incidencia y en segundo término las funcionalidades asociadas a dicha incidencia que pudieron ser impactadas.
Pruebas de Permisos	El 100% de los casos de prueba definidos para esta etapa tienen que ejecutarse exitosamente.
Pruebas de Sistema/UAT	El 100% de los casos de prueba definidos en la fase de diseño tienen que ejecutarse exitosamente.
Pruebas de Paralelo	El 100% de los casos de prueba definidos para esta etapa tienen que ejecutarse exitosamente.
Pruebas de Conversión	El 100% de las pruebas de conversión tienen que ejecutarse exitosamente.



Pruebas No Funcionales	No todas las funcionalidades son críticas en cuanto a las pruebas referidas. La regla 80-20 aplica en estas pruebas. El 80% de los problemas radican en el 20% de las funcionalidades.
------------------------	--

CRITERIOS DE DECISIÓN Y ACEPTACIÓN

PRUEBAS DE INTEGRACIÓN/MODULO (FUNCIONAL)

Los incidentes serán clasificados acorde a los siguientes códigos de severidad:

Severidad	Condiciones necesarias.
A - Grave	<p>Pruebas Funcionales: No se puede probar la funcionalidad, el usuario no puede prácticamente ejecutar ningún caso de prueba. No se puede continuar hasta que el error sea arreglado. El problema es bloqueante en todo sentido.</p> <p>Pruebas de Conversión: Existen inconsistencias graves en los datos que imposibilitan la operativa, no se puede continuar.</p>
B - Serio	<p>Pruebas Funcionales: Comportamiento inconsistente según los requerimientos funcionales. El usuario no puede continuar hasta que el incidente sea arreglado, en el entendido que dicha funcionalidad no podrá ser probada.</p> <p>Pruebas de Conversión: Existen problemas en los datos que no garantizan que la operativa funcione correctamente.</p>
C - Menor	<p>Pruebas Funcionales: El error tiene que ser corregido, pero se puede continuar con la prueba</p> <p>Pruebas de Conversión: Existen temas a resolver en los datos que si bien posibilitan la operativa deben ser resueltos.</p>
D - Sugerencia	Propone sugerencias que mejoran la funcionalidad o usabilidad de la aplicación.



Prioridad	Condiciones necesarias.
A - Alta	Se le debe dar curso a la resolución en forma inmediata (lo mas pronto posible)
M – Media	Su resolución es necesaria próximamente.
B – Baja	Su resolución puede ser esperada o diferida

Criterio a aplicar para cuando PASA las pruebas: (En este cuadro solo se considera la Severidad)

Criterio	Descripción
Pasa	Si PASA la prueba posibilita el pasaje a la siguiente Liberación, Liberación-Versión o Entregable, o Tipo de Prueba según corresponda.
No Pasa	Si No PASA la prueba imposibilita el pasaje a la siguiente Liberación, Liberación-Versión o Entregable, o Tipo de Prueba hasta que se resuelva la incidencia.

Tipo de prueba	Descripción
Prueba de Humo	Etapas Construcción: Severidad de incidentes A no están presentes, para los casos de prueba seleccionados, implica que PASA la prueba. Etapas Prueba de Sistema: Severidad de incidentes A, B no están presentes, para los casos de prueba seleccionados implica que PASA la prueba.
Prueba de Integración/ Modular(Funcional)	Etapas Construcción: Severidad de incidentes A, B no están presentes implica que PASA la prueba. Si para una Liberación-Versión de acuerdo al cronograma existen incidentes de severidad en C implica que PASA la prueba, y se realizará la regresión de dichos incidentes en la siguiente Liberación-Versión. Si en la ultima Liberación-Versión (transición entre la etapa de construcción y prueba de sistema) existen incidentes de severidad A, B, y C, se evaluará entre todos los actores dependiendo de las características de los mismos si corresponde pasar a la siguiente etapa de prueba de sistema.
Pruebas Exploratorias	
Pruebas de Sistema y de UAT	Etapas Prueba de Sistema: Severidad de incidentes A, B y C no están presentes implica que PASA la prueba.
Pruebas de Paralelo	Etapas Prueba de Sistema: Severidad de incidentes A, B y C no están presentes implica que PASA la prueba.
Pruebas de Conversión	Etapas Construcción: Severidad de incidentes A, B y C no están presentes implica que PASA la prueba.

PRUEBAS NO FUNCIONALES



Las pruebas No Funcionales se consideran satisfactorias cuando, luego de ser ejecutadas, las respuestas que da el sistema son iguales o mejores que los objetivos marcados.

Se debe lograr medir e identificar parámetros de medición a los efectos de poder comparar y obtener resultados.

Para la finalización de la etapa de construcción ya se debe tener definidos los objetivos, los alcances, los parámetros, el diseño de las pruebas, las herramientas, etc.

DISEÑO, TÉCNICAS Y HERRAMIENTAS

DISEÑO DE CASOS DE PRUEBA

Esta actividad consiste en la mejor definición que permita validar el cumplimiento de los requerimientos y especificaciones definidos. En otras palabras definir el conjunto de unidades de prueba donde se verifique:

- el correcto funcionamiento
- los procesos relacionados se comporten adecuadamente, de forma que cumplan con las especificaciones requeridas previamente

Todo Caso de Prueba estará asociado a un WSHP20, y cada WSHP20 podrá contener un conjunto de casos de prueba.

Contendrán los siguientes atributos: ID. Caso de Prueba, ID. Proceso de Negocio, Objetivo de la Prueba, Prioridad, Registro de Datos, Categoría, Responsable de Diseño, Fecha de Diseño, Fecha Liberación-Versión, Responsable de Ejecución, Fecha de Ejecución, Resultado de la Ejecución, Resultado Esperado, Resultado obtenido, etc. etc.. .

Existirá un instructivo para la creación de Casos de Prueba que se entregara en el Curso y Taller de Testing, donde se indicara paso a paso como se deben de crear y como registrarlos.

La gestión de los Casos de Prueba se hará utilizando una biblioteca de documentos y los Ciclos de Pruebas en una lista, ambos temas con en SharePoint.



TÉCNICAS

Caja Negra. Las pruebas de caja negra se llevan a cabo sobre las diferentes interfaces que tiene la aplicación, obviando el comportamiento interno. Esto implica realizar pruebas de forma que se compruebe que las funcionalidades del sistema son operativas y están acorde con lo solicitado. En otras palabras consiste en verificar que dada la especificación funcional o la descripción de un requerimiento, cumple con las necesidades descritas, es decir que el resultado esperado y es igual al resultado obtenido.

Los casos de prueba de la caja negra pretenden demostrar que:

- Las funciones del producto son operativas
- La entrada se acepta de forma correcta
- Se produce una salida correcta
- La integridad de la información externa se mantiene

Para la confección de los casos de prueba se utilizará la técnica de Caja Negra: Particiones de equivalencia, Control de Borde y Tablas de Decisión.

Partición Equivalente: La idea es que el producto sometido a pruebas se comporte de la misma manera para todos los miembros de la clase. Sus pasos son la identificación de clases de equivalencia y la identificación de los casos de prueba. Esta técnica de prueba trata cada parámetro como un modelo algebraico donde unos datos son equivalentes a otros. Se divide el dominio de entrada de un programa en clases de datos, a partir de las cuales se derivan los casos de prueba. Cada una de estas clases de equivalencia representa a un conjunto de estados válidos o inválidos para las condiciones de entrada.

Para identificar las clases de equivalencia se identifican las condiciones de entradas del programa, es decir, restricciones de formato o contenido de los datos de entrada. A partir de ellas, se identifican las clases de equivalencia que pueden ser de datos válidos y de datos inválidos. Por ejemplo si un parámetro de entrada debe estar comprendido en un cierto rango, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango. Si una entrada requiere un valor concreto, aparecen 3 clases de equivalencia: por debajo, en y por encima del rango; Si una entrada requiere un valor de entre los de un conjunto, aparecen 2 clases de equivalencia: en el conjunto o fuera de él; Si una entrada es booleana, hay 2 clases: si o no...

Análisis de valores límites: Identifica un conjunto de pruebas representativas seleccionando los bordes de una clase de equivalencia. La experiencia nos muestra que un buen número de errores aparecen en torno a los puntos de cambio de esa clase de equivalencia. Usualmente se necesitan 2 valores por frontera, uno justo abajo y otro justo encima, teniendo en cuenta



además del valor frontera propiamente dicho. Con lo cual podríamos decir que esta técnica complementa a la anterior. Algunas reglas que se pueden seguir para identificar estos valores límites son: Si una condición de entrada especifica un rango de valores, se deben generar casos para los extremos del rango y casos no válidos para las situaciones justo más allá de los extremos; Si la condición de entrada especifica un número finito y consecutivo de valores, hay que escribir casos para los números máximo, mínimo, uno más del máximo y uno menos del mínimo de valores. Si la entrada o salida de un programa es un conjunto ordenado, los casos se deben concentrar en el primero y en el último elemento.

Diagramas de Causa/Efecto y Tablas de Decisión: Identifica un conjunto de diagramas lógicos a partir de la especificación, conteniendo combinaciones de condiciones de entrada. Esto ayuda a la comprensión de dichas combinaciones. Es un enfoque sistemático para seleccionar conjuntos de casos de prueba de alto rendimiento que exploran combinaciones en las condiciones de entrada. Se registran restricciones describiendo combinaciones de causas y efectos posibles, también se crean grafos de causa-efecto y se convierten en tablas de decisiones. Consiste en crear un grafo causa/efecto a partir de las especificaciones, y seleccionar suficientes casos de prueba como para asegurar la cobertura del grafo. Se llama causas a las características de los datos de entrada y efectos a las clases de salidas que puede proporcionar el programa. A partir del grafo causa/efecto se construye una tabla de decisión que refleje las dependencias entre causas y efectos. Lo que se hace entonces es reducir la tabla de decisión y seleccionar sólo un caso de prueba para todas las causas que producen el mismo efecto, o para cada columna de la tabla de decisión.

Se llama causas a las características de los datos de entrada. Los efectos son las clases de salidas que puede proporcionar el programa.

HERRAMIENTAS

Gestión de Casos de Prueba: Se utilizará SharePoint para registrar el diseño, la ejecución y realizar el seguimiento del plan, utilizando bibliotecas de documentos y listas.

Gestión de Incidentes: Se implantará una aplicación de registro de defectos, en adelante MANTIS. Esta herramienta será accesible por todos los integrantes del proyecto, quienes tendrán usuario y contraseña para ingresar a la misma. MANTIS será utilizada hasta la finalización del proyecto.

MANTIS es una aplicación de BugTracking, construida con PHP que utiliza como motor de base de datos a MySQL.



AMBIENTES DE PRUEBAS

Durante la etapa de construcción el equipo de pruebas funcionales tiene asignados cuatro ambientes para la realización de las pruebas divididos en los Tracks respectivos:

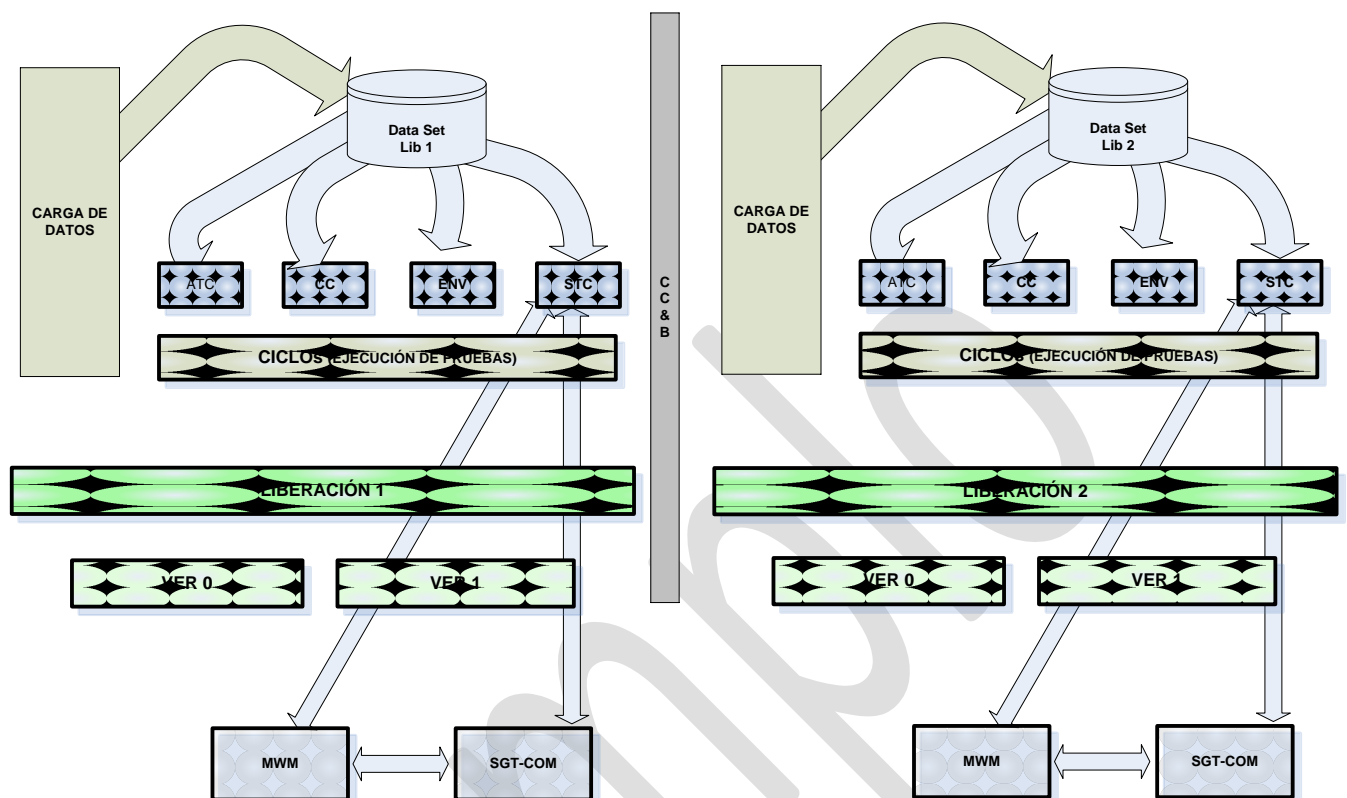
- Test ATC: para la realización de Pruebas de Integración / Modulo (Funcional)
- Test CC: para la realización de Pruebas de Integración / Modulo (Funcional)
- Test STC: para la realización de Pruebas de Integración / Modulo (Funcional)
- Test ENV: para la realización de Pruebas de Integración / Modulo (Funcional) que estén vinculadas al envejecimiento de datos y carga de históricos.

Existirán además dos ambientes adicionales que serán utilizados para la carga de datos después de cada liberación:

- Data Set Lib 1
- Data Set Lib 2

Sobre estos dos ambientes se realizará en primera instancia la copia desde el Master a los Data Set. En particular se configurará utilizando Config Lab entre la base Master y las bases Data Set de cada Liberación. En segundo término por cada Liberación se ingresarán los datos que se deberán utilizar en el Data Set que corresponda. Ver figura:





Y en tercer lugar desde el Data Set se realizara un “Full Copy” hacia los ambientes ATC, CC, ENV y STC, incluyendo datos y programas.

Siempre existirán dos Data Sets, el actual y el anterior. Por ejemplo el Data Set 4 y el Data Set 3.

Antes de realizar un “Full Copy” a un ambiente se realizará un respaldo del mismo. Por ejemplo si es necesario actualizar el ambiente de ATC, se respaldará previamente dicho ambiente y luego se realizara el “Full Copy”. Este procedimiento implicará que los datos que se dispongan desde ese momento en adelante serán los que contenga el Data Set desde el cual se realizó la copia.

Por cada una nueva Liberación sobre uno de los ambientes se realizará una Prueba de Humo.



En particular el ambiente STC estará conectado con MWM y SGT-COM. Y el ambiente ATC estará conectado con SGT, EGEO y SAP.

La solicitud de “Full Copy” sobre los ambientes la realizará el equipo de PMO y por cada uno en forma indistinta. Esto permitirá que en determinado momento se puedan ejecutar pruebas de distintas liberaciones de diferentes Tracks en forma simultánea.

Cabe la posibilidad de que exista un “HotFix” que sea necesario aplicar para poder continuar con la ejecución en determinado ambiente, siendo en primera instancia el Data Set impactado y luego el Track correspondiente. Si bien los Hot Fixes serán críticos serán de aplicación manual y no a través de Config Lab.

Además si es necesario que los datos a ingresar permanezcan con cada Liberación, se habilitará dicho ingreso sobre el Data Set respectivo.

Dentro de una misma Liberación existirán mas de una Versión, y la inicial se corresponderá con la Versión 0.



PLANIFICACIÓN

Para definir un cronograma a detalle de pruebas del proyecto, es necesario conocer el Plan de Liberación, el cual contiene el cronograma en que las funcionalidades serán integradas. Se requiere conocer las fechas en que cada requerimiento es finalizado, en otras palabras conocer que funcionalidades están incluidas en cada nueva Liberación-Versión del producto y cuales son la incidencias resueltas que contiene.

A continuación se muestra un ejemplo de cómo podría ser un plan de integración del producto, donde se tiene pensado construir el producto en forma incremental, liberando versiones intermedias al equipo de pruebas.

Liberación	Versión	Fecha	Identificador de funcionalidades incluidas.
1	1	Fecha 1	1, 2, 4, 6
1	2	Fecha 2	1, 2, 4, 6 + #Def
1	3	Fecha 3	1, 2, 4, 6 + #Def
2	1	Fecha 4	1, 2, 4, 6, 8, 9 + #Def
2	2	Fecha 5	1, 2, 4, 6, 8, 9 + #Def

En la medida que se identifican defectos se irán informando en la ventana de tiempo existente entre cada fecha de Liberación-Versión. La resolución de los mismos deberá realizarse para la siguiente Versión.

Cada nueva Liberación-Versión deberá contener la resolución de los incidentes presentados en el ciclo anterior. Existe la posibilidad de que una Liberación-Versión se rechace completamente lo que implicará una nueva Liberación-Versión.

Las pruebas exploratorias se utilizarán como complementarias de las pruebas funcionales. Es necesario para planificar las pruebas funcionales definir las funcionalidades de alta, media o baja prioridad. Como ejemplo podríamos realizar pruebas exploratorias con funcionalidades de baja prioridad.



Cuando se tenga el documento del Plan de Liberación, se podrán efectuar las acciones antes mencionadas, las cuales permitirán planificar la actividad a detalle.

La citada planificación a la que se llegue, deberá ser revisada al comenzar cada nuevo ciclo de prueba, ya que los supuestos sobre los que se definió la planificación probablemente hayan cambiado. Por ejemplo, al transcurrir el tiempo pueden cambiar las prioridades de las pruebas debido a cambios en las prioridades del negocio o de la confianza adquirida en el producto como resultado de la realización de las pruebas en ciclos anteriores.

También cada ciclo debe planificar las pruebas de regresión. Al obtener una nueva Liberación-Versión donde se corrigieron defectos, se deben ejecutar nuevamente los casos de prueba que encontraron estos defectos y sus relacionados. Como de antemano no se conoce cuales ni cuantos serán estos defectos, al comenzar cada ciclo, deben ser consideradas dichas pruebas de regresión.

Fecha	Actividad



EQUIPO DE PRUEBAS Y RESPONSABILIDADES

Rol y Roles	Responsabilidades
Coordinador de Proceso de Pruebas (CPP)	<p>Definir los objetivos de las pruebas y generar el plan de pruebas maestro.</p> <p>Coordinar el equipo de pruebas para garantizar que las actividades se hagan de acuerdo a lo establecido en el presente plan, cumpliendo con los hitos establecidos.</p> <p>Coordinar y conducir las reuniones de equipo de pruebas.</p> <p>Resolver o escalar las cuestiones planteadas por los integrantes del equipo de pruebas.</p> <p>Escalar los riesgos y problemas al Comité de Dirección del Proyecto.</p> <p>Preparar informes de resumen de las pruebas.</p> <p>Seguimiento de los Incidentes.</p> <p>Asegurar que se consigue el nivel de pruebas apropiado.</p> <p>Generar informes de estado semanales.</p>
Diseñador de Casos de Prueba (DCP)	<p>Realizar el Diseño de los Casos de Prueba.</p> <p>Participar en reuniones coordinadas por el Coordinador de las Pruebas.</p> <p>Documentar y Escalar los temas relacionados al proceso de pruebas.</p> <p>Generar juegos de datos para pruebas, detallar en los casos de prueba que probar y como. (en la configuración de ambiente y ciclo).</p> <p>Ejecutar los Casos de Prueba definidos para cada ciclo de pruebas y registrarlos.</p> <p>Registrar Incidentes.</p>
Tester (TES)	<p>Participar en reuniones coordinadas por el Coordinador de las Pruebas</p> <p>Proveer el estado de las pruebas al Coordinador</p> <p>Generar juegos de datos para pruebas</p> <p>Ejecutar los Casos de Prueba definidos para cada ciclo de pruebas y registrarlos.</p> <p>Registrar Incidentes.</p> <p>Documentar y Escalar los temas relacionados al proceso de pruebas.</p>
Creador de Datos de Prueba (CDP)	<p>Estudiar, desarrollar la forma de generación de juegos de datos (manualmente, con scripts, etc)</p> <p>Generar juegos de datos para pruebas como apoyo de la actividad del Tester</p>



	<p>Registrar Incidentes.</p> <p>Documentar y Escalar los temas relacionados al proceso de pruebas.</p>
Jefe de Proyecto (JDP - Equipo de Desarrollo)	Revisar los Incidentes encontrados y asignar al equipo de diseño encargado de estudiarlos
Diseñador (DIS - Equipo de Desarrollo)	Estudiar los Incidentes asignados y asignar al equipo de desarrollo para resolverlos
Desarrollador (DES - Equipo de Desarrollo)	Resuelve los Incidentes asignados
Diseñador y Desarrollador Pruebas No Funcionales (Equipo)	<p>Desarrollar y ejecutar los scripts para las pruebas no funcionales</p> <p>Registrar Incidentes.</p> <p>Documentar y escalar los temas relacionados al proceso de pruebas.</p> <p>Generar juegos de datos para pruebas</p> <p>Participar en reuniones coordinadas por el Coordinador de las Pruebas</p> <p>Proveer el estado de las pruebas al Coordinador</p>

Se incluyeron además los roles del equipo de desarrollo que interactuará con el equipo de pruebas.

GESTIÓN DE INCIDENTES

Todos los incidentes identificados durante las pruebas deberán ser registrados y gestionados por los distintos grupos que intervienen, desde la identificación hasta su cierre pasando por la resolución.

Más adelante en el presente documento se representará el ciclo de vida, con los estados y transiciones por los que puede pasar un incidente.

De forma de hacer más comprensible dicho grafo (ciclo de vida), presentamos el siguiente escenario que con mayor frecuencia deberá ser utilizado:

- El Tester a partir de la ejecución de un caso de prueba identifica un incidente donde el resultado esperado es diferente al resultado obtenido.
- El Tester registra el incidente en la herramienta de Gestión de Incidentes (MANTIS), realizando una breve descripción y luego ingresa el detalle lo más completo posible, facilitando datos de pruebas, pantallas, pasos seguidos y completando el resto de la información necesaria del incidente.
- El estado en que queda el incidente es “Nuevo”.
- Finaliza la registración del incidente.
- El grupo de Testing toma ese incidente verificando la pertinencia y la elegibilidad del mismo y lo envía a “Abierto”. Existe la posibilidad de cambiar el atributo “Equipo Asignado” en el incidente (por ejemplo ATC, CC, STC, CONF, DES, INT, REP).
- El grupo de Jefes de Proyectos toma ese incidente lo verifica, da su conformidad y lo envía a “En Análisis”. Existe la posibilidad de cambiar el atributo “Equipo Asignado” en el incidente (por ejemplo ATC, CC, STC, CONF, DES, INT, REP) de forma de asignar el equipo de trabajo.
- El grupo de Analistas toma ese incidente lo analiza y investiga la naturaleza del mismo, su resolución y lo envía a “En Corrección”
- El grupo de “En Corrección” toma ese incidente, lo resuelve y lo envía a “Validación”
- El Tester toma ese incidente, aguarda a que sea liberada la resolución del mismo y realiza la regresión donde al verificar que el comportamiento es el correcto pasa al estado de “Cerrado”.

En cada cambio de estado del incidente se debe registrar la información pertinente.

Como escenarios diferentes podemos tener como ejemplo:

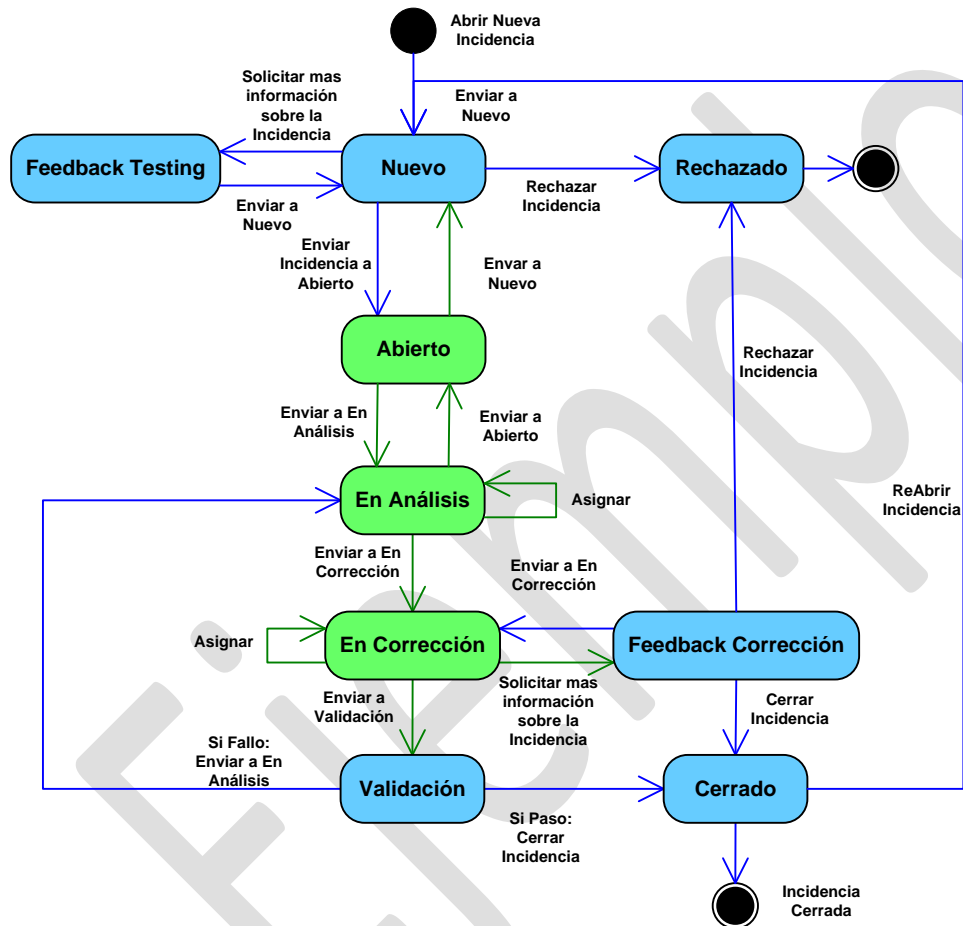
- a. El incidente no corresponde y se envía a “Rechazado”
- b. Es necesario asignar a una persona la resolución del incidente
- c. Se solicita la ampliación de la información sobre la incidencia

Existen un conjunto mayor de situaciones que están consideradas en el Ciclo de Vida del Incidente.

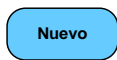


CICLO DE VIDA DE LOS INCIDENTES

Se detalla a continuación el ciclo de vida del incidente:



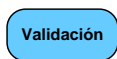
Donde:



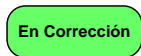
Esta figura corresponde a un Estado



Este conector corresponde a una Acción



El color azul corresponde a una actividad del equipo de pruebas



El color verde corresponde a una actividad del equipo de desarrollo



Corresponde al ingreso de un nuevo incidente



Corresponde a la finalización del ciclo

DESCRIPCIÓN DE LOS ESTADOS

Estado	Descripción
Nuevo	Existe un incidente que se ha identificado y que será revisado por el equipo de CPP, verificando la pertinencia y la elegibilidad del mismo.
Abierto	El equipo de JDP continuará verificando su conformidad.
En Análisis	El equipo de DES (los Analistas) continuaran identificando y analizando la naturaleza del mismo y gestionando su resolución.
En Corrección	El equipo de DES (los Desarrolladores) resolverá el defecto.
Validación	El equipo de TES realizará la regresión del incidente verificando si efectivamente fue resuelto.
Feedback Testing	Es necesaria la ampliación y aclaración por parte de TES de la información que se dispone del incidente.
Feedback Corrección	Es necesaria la ampliación y aclaración por parte de TES de la información que se dispone del incidente.
Rechazado	Implica que el incidente no corresponde.



Cerrado	Es cuando el incidente es resuelto y deja de existir como tal.
---------	--

TRANSICIONES DE ESTADOS

Equipo de Pruebas (Transición Color Azul)	Equipo de Desarrollo (Transición Color Verde)
Inicio → Nuevo	Abierto → En Análisis
Nuevo → Abierto	Abierto → Nuevo
Nuevo → Feedback Testing	En Análisis → Abierto
Nuevo → Rechazado	En Análisis → En Análisis
Validación → Cerrado	En Análisis → En Corrección
Validación → En Análisis	En Corrección → Validación
Feedback Corrección → En Análisis	En Corrección → Feedback Corrección
Feedback Corrección → Cerrado	En Corrección → En Corrección
Feedback Corrección → Rechazado	
Feedback Testing → Nuevo	
Cerrado → Nuevo	

En dichas transiciones se muestra los posibles caminos a partir del posicionamiento en un estado en particular.



REGISTRO DE INCIDENTES

El registro de incidentes se debe realizar completando la siguiente información:

- a. Detalle escueto y representativo
- b. Detalle general incluyendo datos de prueba, copia de pantallas y información relevante a los efectos de que al realizar la comprobación no sea necesario ampliar la misma, vinculo al documento de Diseño y Ejecución del Caso de Prueba.
- c. Contendrá los siguientes atributos: Usuario, Asignado a quién, Categoría, Reproducibilidad, Severidad, Prioridad, Encontrado en Liberación, Resumen, Descripción, Información Adicional, Caso de Prueba asociado, Funcionalidad asociada (20), Tipo de Prueba, Equipo Asignado, etc. etc.
- d. Dependiendo del estado en el que este se le solicitara mas información.

Existirá un instructivo para el Registro de Incidentes que se entregara en el Curso y Taller de Testing, donde se indicara paso a paso como se debe registrar.

MÉTRICAS

La información correspondiente a este punto se extrae del documento “Resumen Metodología de Etapa de Construcción”

ESTIMACIONES

- Para cada funcionalidad (WKSHP20), se evaluará la complejidad de la misma en Alta, Media, Baja y se definirán estimadores de horas en base a la misma
- Se realizará una planificación Macro (mensual) de la cantidad de diseños a producir (tomando como unidad el 20) y cantidad de ejecuciones pruebas (de 20s) previstos.



PLANIFICACIÓN

- Planificación en Project. Se realizará a nivel macro y con la siguiente agrupación:
 1. Release
 - 1.1. Diseño de Casos (la apertura será por doc 20)
 - 1.2. Configuración del Ambiente
 - 1.3. Ejecución Pruebas

La asignación detallada (al equipo) para el diseño y la ejecución de casos se realizarán utilizando listas de SharePoint. Se utilizarán **2 listas**

- una para asignar los diseños llamada **“Casos de Prueba”**. En la misma los coordinadores de Testing asignan el proceso al cual hay que hacerle los casos de prueba a un integrante del equipo de testing. Se identifica el proceso (doc 20) y si está restringido a la funcionalidad de la liberación se indica en el campo “Restringido a”. Esto aplica en los casos que habrá liberación parcial de la información del proceso en una liberación. El registro inicial (que crean los coordinadores de testing se identifica por el nombre y CP01. En el mismo se indica el tiempo estimado para la creación de todos los casos de prueba asociados a dicho proceso, así como la fecha fin planificada. El diseñador agregará tantos registros como casos de prueba identifique copiando la info del registro inicial y agregándole como identificación CP02, CP03, CP..... Los datos de planificación se copian del registro inicial.
- Otra para planificar la ejecución de casos de prueba. Esta lista asigna la ejecución de cada caso de prueba a cada tester (para cada liberación/versión) y permite hacer el seguimiento de la misma. El tester actualizará el estado de ejecución de la misma. Cada ejecución está vinculado al caso de prueba asociado donde se registra el detalle del resultado de la ejecución (Hoja: Ejecución/liberación/versión).

En la planilla de Capacidades se realiza el seguimiento del diseño de los casos de prueba por liberación. La misma tiene la planificación (que debe coincidir con el Project), y semanalmente los Coordinadores de Prueba registrarán el % de avance en el diseño de cada caso. En base a dicha información se calcularán las métricas de avance definidas para el diseño de casos.



SEGUIMIENTO

Seguimiento: Realizaremos el seguimiento del avance sobre el equipo en términos de:

- Cobertura de 20's – por tipo de prueba:
 - cant de 20s/liberación a los que se diseño casos de prueba/total de 20s, .--> Fuente: Lista Casos de Prueba: % fin (fin: si todos los casos asociados están terminados), % en diseño (si tiene al menos un caso en diseño), % en espera (si hay casos en espera y el resto están finalizados), % sin comenzar (todos sus casos están sin comenzar) – Por liberación y total
 - cant de 20s que fueron testeados (se ejecutaron todos los casos de prueba planificados – fuente: ciclos de ejecución)- al menos 1 vez y sin importar el resultado / total de 20s Total (sale de la lista de casos – se cuentan los 20/liberación) y por release
- Valor ganado: Se comparará el valor planificado sobre el obtenido.

Realizaremos el seguimiento del producto en términos de:

- Porcentaje de Procesos de Negocio (20) **Pasa (modular/integración, sistema)**
 - # Proceso de Negocio Sin Defectos / Total de Proceso de Negocio al que se le ejecutaron Casos de Prueba
- #Defectos (abiertos y total) por Proceso de Negocio (20). Se podrá visualizar totales o abierto por severidad
- Cobertura de Proceso de Negocio o Total
 - Casos de Prueba Ejecutados / Total Casos de Prueba Definidos - se diferencia en: modular/integración, sistema
 - Casos de Prueba Pasa / Total Casos de Prueba Definidos - se diferencia en: modular/integración, sistema



CRITERIOS DE SUSPENSIÓN Y REANUDACIÓN DE LAS PRUEBAS

SUSPENSIÓN DE LAS PRUEBAS

El equipo de pruebas puede suspender parcial o totalmente actividades de prueba de una Liberación-Versión o Entregable si se produce alguno de los sucesos siguientes:

- La ejecución de las pruebas no pasa las pruebas de humo definidas.
- El entorno de pruebas no es lo suficientemente estable como para confiar en los resultados.
- El entorno de pruebas es muy diferente del entorno de producción previsto y no se puede confiar en los resultados.
- Si se identifica un incidente que bloquea la secuencia de pasos de un caso de prueba, se continuará trabajando en el caso de prueba siguiente a no ser que el usuario que prueba considere que los resultados de la prueba se verán influenciados por el defecto ya identificado. En tal caso el usuario que realiza las pruebas deberá asignar prioridad “B - Serio” a dicho incidente y detendrá la ejecución del caso de prueba.
- La cantidad de incidentes encontrados es importante, por lo que se acuerda esperar a una nueva Liberación-Versión o Entregable.

REANUDACIÓN DE LAS PRUEBAS

El equipo de pruebas puede reanudar las pruebas si:

- Se resuelve el problema que supuso la suspensión de las pruebas. Y se incrementa el número de ciclos propuestos y se pasa a la siguiente Liberación-Versión.
- Los equipos de desarrollo y de pruebas acuerdan que no es necesario corregir el incidente inmediatamente y que podría solucionarse durante la siguiente Liberación-Versión.



ENTREGABLES

Entregable	Responsabilidad
Plan de Pruebas	PMO
Diseño de las Pruebas	Diseñador
Ejecución de las Pruebas	Tester
Registro de Incidentes	Equipo de Pruebas y Equipo de Desarrollo

REQUISITOS PREVIOS PARA LAS PRUEBAS

Tarea	Depende	Responsable	Fin
Capacitación en Proceso de Pruebas			
Herramienta para Gestión de Incidentes instalada y configurada			
Herramienta para la Gestión de los Casos de Prueba instalada y configurada.			
Capacitación en Registro de Incidentes			
Capacitación en Diseño y Registro de Casos de Prueba			
Ambientes creados y Configurados			

