Bizativiti

Grupo 4

Plan de Verificación y Validación

Versión 1.0

Historia de revisiones

Fecha	Versión	Descripción	Autor
28/08/2013	1.0	Comienzo Documento	Diego Cossio
31/08/2013	1.0	Fin Documento	Diego Cossio
31/08/2013	1.0	Revisión Asistente SQA	Diego Cossio

Contenido

1.Introducción	3
1.1.Propósito	3
1.2.Punto de partida	3
1.3.Alcance	3
1.4.Identificación del proyecto	4
1.5.Estrategia de evolución del Plan	4
2.Requerimientos para verificar	4
3.Estrategia de Verificación	5
3.1.Tipos de pruebas	5
3.1.1.Prueba de integridad de los datos y la base de datos	5
3.1.2.Prueba de Funcionalidad	6
3.1.3.Prueba de Ciclo del Negocio	6
3.1.4.Prueba de Interfase de Usuario	6
3.1.5.Prueba de Performance	6
3.1.6.Prueba de Carga	6
3.1.7.Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos)	6
3.1.8.Prueba de Volumen	6
3.1.9.Prueba de Seguridad y Control de Acceso	6
3.1.10.Prueba de Fallas y Recuperación	6
3.1.11.Prueba de Configuración	6
3.1.12.Prueba de Instalación	7
3.1.13.Prueba de Documentos	7
3.2.Herramientas	7
4.Recursos	7
4.1.Roles	7
4.2.Sistema	8
5.Hitos del proyecto de Verificación	8
6.Entregables	9
6.1.Modelo de Casos de Prueba	9
6.2.Informes de Verificación	9
6.3.Evaluación de la verificación	10
6.4.Informe final de verificación	10
7.Dependencias [opcional]	10
7.1.Dependencia de personal [opcional]	10
7.2.Dependencia de software [opcional]	10
8.Riesgos [opcional]	11
8.1.Planificación [opcional]	11
8.2.Técnico [opcional]	11
8.3.Gestión [opcional]	11
9.Apéndice	11
9.1.Niveles de gravedad de error	11
9.2.Niveles de aceptación para lo elementos verificados	11

1. **Introducción**

1. **Propósito**

Este Plan de Verificación para el proyecto Bizativiti soporta los siguientes objetivos:

- Identificar la información de proyecto existente y los componentes de software que deben ser verificados.
- Enumerar los requerimientos recomendados para verificar.
- Recomendar y describir las estrategias de verificación que serán usadas.
- Identificar los recursos necesarios y proporcionar una estimación de esfuerzo para realizar la verificación.
- Enumerar los entregables del proyecto de verificación.

2. **Punto de partida**

El objetivo de la verificación es detectar defectos y asegurar que estos sean corregidos antes de la liberación del producto de software, así como la comprobación de que el mismo cumple con los requerimientos funcionales y no funcionales.

La verificación se realizará a todos los módulos que componen el software como además su integración.

El proyecto consta básicamente en realizar un conversor que tome como entrada un archivo en formato xml, que cumpla con el estándar XPDL y transformar dicho archivo que cumpla con el estándar BPMN 2.0.

3. **Alcance**

En esta sección se describe los estados de la verificación.

Pruebas unitarias se verifican las funciones de cada módulo son realizadas por el equipo de implementadores, cuando estas pruebas llegan a un estado de aceptación se procede a las pruebas de integración.

Pruebas de integración para verificar que los módulos integrados funcionan en conjunto, se tratará que la realice el equipo de implementadores ya que lo particular de nuestro producto es que es un conversor con lo que carece de interfaz gráfica por módulo, se decidirá con el equipo de verificación como probar estas pruebas.

Pruebas de Funcionales para verificar los requerimientos funcionales del cliente que en este caso es concreto pasar un xml de formato XPDL a BPMN 2.0 estas pruebas serán realizadas por el equipo de verificación en contacto con el equipo de implementadores.

Prueba de aceptación se validará en conjunto con el cliente si es lo que espera.

Es de vital importancia entender bien el estándar XPDL tanto como BPMN 2.0 para que no sea un riesgo que pueda afectar el diseño, desarrollo o implementación de la verificación.

La elección de los casos de prueba van a ir de complejidad mínima e incrementando.

Queda como investigación de herramientas para el automatizado de pruebas

esto último se decidirá con todo el equipo de verificación.

4. **Identificación del proyecto**

Los documentos usados para elaborar el Plan de Verificación son los siguientes:

- Plantilla de Plan de Verificación y Validación - MUM
- Documento especificación de requerimientos de software
- Acta de Reunión de Requerimientos
- Memoria organizacional del Proyecto de Ingeniería de Software:
<http://www.fing.edu.uy/inco/cursos/ingsoft/pis/memoria/index.htm>

5. **Estrategia de evolución del Plan**

El encargado para realizar cambios al Plan de Verificación y Validación será el Responsable de Verificación con acuerdo de su equipo, pudiendo haber cambios que se deleguen a un asistente.

El plan podrá cambiar a medida que pasen las semanas dependiendo de si los documentos que depende el mismo cambian.

Además en cada iteración será ajustado el mismo.

La última versión se encontrará disponible en google docs y además estará consolidada en el repositorio la última entrega.

A continuación se muestra las semanas de entrega :

- Semana 02, Iteración 1 Fase Inicial
- Semana 04, Iteración 2 Fase Inicial

2. **Requerimientos para verificar**

En la lista a continuación se presentan los elementos, casos de uso, requerimientos funcionales y requerimientos no funcionales, que serán verificados.

- Requerimientos funcionales:
 - a. conversor que tome como entrada un archivo en formato xml, que cumpla con el estándar XPDL y transformar dicho archivo que cumpla con el estándar BPMN 2.0
 - b. xml generado en formato BPMN 2.0 que interprete un modelo equivalente al de bizagi
 - c. Auditoría de errores en la conversión de modelos
- Requerimientos no funcionales:
- Casos de Uso: No aplica aún, falta doc Modelo Caso de Uso

3. Estrategia de Verificación

Esta sección es para describir cada una de las pruebas que se implementarán, técnicas para efectuarlas y criterios de aceptación.

1. Tipos de pruebas

1. Prueba de integridad de los datos y la base de datos

1. *Objetivo de la prueba*

Asegurar que los métodos y procesos de acceso a la base de datos funcionan correctamente y sin corromper datos.

2. *Técnica*

Invoque cada método o proceso de acceso a la base de datos con datos válidos y no válidos.

[Inspeccione la base de datos para asegurarse de que se han guardado los datos correctos, que todos los eventos de la base de datos ocurrieron correctamente, o repase los datos devueltos para asegurar que se recuperaron datos correctos por la vía correcta.

3. *Criterio de aceptación*

Todos los métodos y procesos de acceso a la base de datos funcionan como fueron diseñados y sin datos corruptos.

4. *Consideraciones especiales*

La prueba requiere un entorno de administración de DBMS o controladores para ingresar o modificar información directamente en la base de datos.

Los procesos deben ser invocados manualmente.

Se deben usar bases de datos pequeñas para aumentar la facilidad de inspección de los datos para verificar que no sucedan eventos no aceptables.

2. Prueba de Funcionalidad

La prueba de funcionalidad se enfoca en requerimientos para verificar que se corresponden directamente a casos de usos o funciones y reglas del negocio. Los objetivos de estas pruebas son verificar la aceptación de los datos, el proceso, la recuperación y la implementación correcta de las reglas del negocio. Este tipo de prueba se basa en técnicas de caja negra, que consisten en verificar la aplicación y sus procesos interactuando con la aplicación por medio de la interfaz de usuario y analizar los resultados obtenidos.

1. *Objetivo de la prueba*

Asegurar la funcionalidad apropiada del objeto de prueba, incluyendo la navegación, entrada de datos, proceso y recuperación.

2. *Técnica*

Ejecute cada caso de uso, flujo de caso de uso, o función usando datos válidos y no válidos, para verificar lo siguiente:

- Se obtienen los resultados esperados cuando se usan datos válidos.
- Cuando se usan datos no válidos se despliegan los mensajes de error o advertencia apropiados.
- Se aplica apropiadamente cada regla del negocio.

3. *Criterio de aceptación*

Todas las pruebas planificadas se realizaron. Todos los defectos encontrados han sido debidamente identificados.

4. *Consideraciones especiales*

Identificar o describir aquellos elementos o problemas (internos o externos) que impactaron en la implementación y ejecución de las pruebas de funcionalidad.

3. **Prueba de Ciclo del Negocio**

No aplica, no veo el ciclo del negocio que pueda tener un conversor.

4. **Prueba de Interfase de Usuario**

Por el momento no aplica.

5. **Prueba de Performance**

No aplica debido a que no es necesario tiempos de respuestas ya que no es requerimiento.

6. **Prueba de Carga**

No aplica debido a que no hay requerimiento de performance.

7. **Prueba de Esfuerzo (stress, competencia por recursos, bajos recursos)**

No aplica idem anterior.

8. **Prueba de Volumen**

No aplica idem anterior

9. **Prueba de Seguridad y Control de Acceso**

No aplica no es requerimiento.

10. **Prueba de Fallas y Recuperación**

No aplica no es requerimiento

11. **Prueba de Configuración**

La Prueba de Configuración verifica el funcionamiento del software con diferentes configuraciones de software y hardware.

1. *Objetivo de la prueba*

Verificar que el software funcione apropiadamente en las configuraciones requeridas de hardware y software.

2. *Técnica*

Usar las pruebas de Funcionalidad.

- Abrir y cerrar varias sesiones de software que no son objeto de prueba, como parte de la prueba o antes de comenzar la prueba.
- Ejecutar operaciones seleccionadas para simular la interacción del actor con el software objeto de prueba y con el software que no es objeto de prueba.
- Repetir los procedimientos anteriores minimizando la memoria convencional disponible en la máquina cliente.

3. *Criterio de aceptación*

Por cada combinación de software objeto de prueba y software que no es objeto de prueba, todas las operaciones son completadas exitosamente sin fallas.

4. *Consideraciones especiales*

Todo el software que no es objeto de prueba que es necesario y debe estar accesible.

¿Qué aplicaciones se usan normalmente?

¿Qué información se maneja en las aplicaciones que se usan normalmente, y que tamaño de información?

Los sistemas, red, servidores de red, bases de datos, etc., deben ser documentados como parte de esta prueba.

12. Prueba de Instalación

Como parte de prueba de instalación para la aplicación de escritorio se realizarán pruebas automatizadas de funcionalidad.

13. Prueba de Documentos

Tarea equipo de SQA.

2. Herramientas

Lista de herramientas para el proyecto:

- GitHub el cual se usa como repositorio posee también herramienta para gestión de incidentes.

Es de interés del equipo de verificación buscar una herramienta para la automatización de las pruebas ya que sería de gran ayuda para el testeo del producto considerando lo particular de las pruebas.

4. Recursos

En esta sección se presentan los recursos recomendados para el proyecto **Bizatviti**, sus principales responsabilidades y su conocimiento o habilidades.

1. Roles

En la tabla a continuación se muestra la composición de personal para el proyecto **Bizatviti** en el área Verificación del Software.

Rol	Cantidad mínima de recursos recomendada	Responsabilidades
Responsable de verificación	1	Identifica, prioriza e implementa los casos de prueba. <ul style="list-style-type: none">• Genera el Plan de Verificación.• Genera el Modelo de Prueba.• Evalúa el esfuerzo necesario para verificar.• Proporciona la dirección técnica.• Adquiere los recursos apropiados.• Proporciona informes sobre la verificación.
Asistente de verificación	3	<ul style="list-style-type: none">• Ejecuta las pruebas• Registra los resultados de las pruebas.• Recuperar el software de errores.• Documenta los pedidos de cambio.

2. Sistema

En la siguiente tabla se establecen los recursos de sistema necesarios para realizar la verificación.

Recurso	Nombre/Tipo
Servidor de base de datos	MySql
PC Cliente para pruebas	Windows y Linux
Repositorio de pruebas	GitHub

5. Hitos del proyecto de Verificación

La verificación del Bizativiti debe incorporar actividades de prueba para cada verificación identificada en las secciones anteriores. Se deben identificar los hitos del proyecto de verificación separados para comunicar los logros de estado de proyecto.

Actividad que determina el hito	Esfuerzo	Fecha de comienzo	Fecha de finalización
Fase 1 - Iteración I		19/08/13	01/09/13
Investigación	9 hs		
Planificar la verificación	15 hs		
Fase 1 - Iteración II		02/09/13	15/09/13
Elaborar Casos de Prueba	16 hs		
Ajuste y Control de Verificación	10 hs		
Ejecutar la verificación	11 hs		
Evaluar la verificación	11 hs		
Fase 2 - Iteración I		16/09/13	29/09/13
Elaborar Casos de Prueba	16-20 hs		
Ajuste y Control de Verificación	9 hs		
Ejecutar la verificación	8 hs		
Evaluar la verificación	8 hs		
Fase 2 - Iteración II		30/09/13	13/10/13
Elaborar Casos de Prueba	16 hs		
Ajuste y Control de Verificación	8 hs		
Ejecutar la verificación	8 hs		
Evaluar la verificación	8 hs		
Fase 3 - Iteración I		14/10/13	27/10/13
Elaborar Casos de Prueba	15 hs		
Ajuste y Control de Verificación	15 hs		
Ejecutar la verificación	15 hs		
Evaluar la verificación	15 hs		
Fase 3 - Iteración II		28/10/13	10/11/13
Elaborar Casos de Prueba	15 hs		
Ajuste y Control de	9 hs		

Verificación			
Ejecutar la verificación	15 hs		
Evaluar la verificación	15 hs		
Fase 4 - Iteración I		11/11/13	24/11/13
Ajuste y Control de Verificación	4 hs		
Ejecutar la verificación	15 hs		
Evaluar la verificación	15 hs		

6. Entregables

1. Modelo de Casos de Prueba

Documento	Modelo de Casos de Prueba
Creado por	El Responsable de verificación, Diego Cossio.
Para quien	Es la guía para realizar las pruebas del sistema y lo usarán los Asistentes de verificación y el Responsable de verificación cuando se ejecuten las pruebas del sistema.
Fecha de liberación	Será liberado el 15/09/13.

2. Informes de Verificación

Documento	Se genera un documento Informe de Verificación Unitaria por cada prueba unitaria que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación unitaria.

Documento	Se genera un documento Informe Consolidación por cada consolidación que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de consolidación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada consolidación.

Documento	Se genera un documento Informe de Verificación de Integración por cada prueba de integración que se realice al sistema.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación de integración.

Documento	Se genera un documento Informe de Verificación de Sistema por cada prueba de sistema que se realice.
Creado por	Las personas que ejecutan las pruebas.
Para quien	Es el retorno para los implementadores de la tarea de verificación, que detalla los errores encontrados para que puedan ser corregidos.
Fecha de liberación	Será liberado luego de cada verificación de sistema.

3. Evaluación de la verificación

Documento	Se genera un documento Evaluación de la verificación por cada prueba que se realice al sistema. Este documento contiene las fallas encontradas en el sistema, la cobertura de la verificación realizada y el estado del sistema.
Creado por	El Responsable de verificación, que toma como fuente de su trabajo los Informes de verificación.
Para quien	Es el resumen de la tarea de verificación y es el retorno para todo el equipo de trabajo del estado del sistema.
Fecha de liberación	Será liberado luego de cada verificación, unitaria, de integración y de sistema.

4. Informe final de verificación

Documento	El documento Informe final de verificación es el resumen de la verificación final del sistema antes de que sea liberado al entorno del usuario.
Creado por	El Responsable de verificación, que toma como fuente de su trabajo los Informes de verificación.
Para quien	Indica el estado del sistema.
Fecha de liberación	Será liberado luego de la verificación final del sistema.

7. Dependencias

1. Dependencia de personal [opcional]

Para la construcción de pruebas del prototipo bastará con 3 personas el Responsable de Verificación y 2 Asistentes de Verificación.

Luego para la construcción del proyecto se necesitará de todo el equipo de verificación.

En las próximas semanas se definirá si será necesario la incorporación de más personas para el equipo de verificación debido que la necesidad de tener un producto funcionando al final de la semana 14 es requerimiento.

2. Dependencia de software

El tiempo para la verificación del producto depende fuertemente de la construcción del software con lo que si se retrasa la construcción habrá menos tiempo para verificar.

8. Riesgos

1. Planificación

Riesgos relativos a la planificación:

- Falla en la planificación
- Retraso en la liberación de software para testear

2. Técnico

Se considera que un riesgo puede ser la búsqueda de una herramienta para el automatizado de las pruebas que encajen perfectamente con las características de las pruebas.

3. Gestión

La sobreestimación de tiempo en las tareas es una forma de mitigar los riesgos.

9. Apéndice

1. Niveles de gravedad de error

En muchas actividades del proceso de verificación se deben clasificar los errores según su nivel de gravedad. Se asigna un nivel de gravedad a los errores para poder capturar de alguna manera su impacto en el sistema. Además para poder evaluar la verificación y el sistema.

A continuación se da una sugerencia de cuatro niveles diferentes de gravedad de error:

- **Catastrófico:** un error cuya presencia impide el uso del sistema.
- **Crítico:** un error cuya presencia causa la pérdida de una funcionalidad crítica del sistema. Si no se corrige el sistema no satisfará las necesidades del cliente.
- **Marginal:** un error que causa un daño menor, produciendo pérdida de efectividad, pérdida de disponibilidad o degradación de una funcionalidad que no se realiza fácilmente de otra manera.
- **Menor:** un error que no causa perjuicio al sistema, pero que requiere mantenimiento o reparación. No causa pérdida de funcionalidades que no se puedan realizar de otra manera.

2. Niveles de aceptación para lo elementos verificados

[Se debe establecer un nivel de aceptación para los elementos verificados para poder establecer el estado en el que se encuentra el proyecto.

En esta sección defina niveles de aceptación y los criterios de pertenencia a cada nivel.

Como ejemplo de niveles de aceptación:

- **No aprobado:** el elemento verificado tiene errores catastróficos (uno o varios) que impiden su uso o tiene errores críticos (uno o varios) que hacen que el elemento verificado no sea confiable. El usuario no puede depender de él para realizar el trabajo.
- **Aprobado con Observaciones:** el elemento verificado no tiene errores catastróficos, ni errores críticos, pero tiene errores marginales (uno o varios) que hacen que el elemento de software se degrade en algunas situaciones.
- **Aprobado:** el elemento verificado no tiene errores o tiene errores menores que no afectan el normal funcionamiento del elemento.]