



Evaluación y mejora para el
desarrollo de Software

ISC. Laura Castillo Salzar

Estructura del Plan de Pruebas de Software

Plan de pruebas

El plan de pruebas de software se elabora para atender los objetivos de calidad en un desarrollo de sistemas, encargándose de definir aspectos como por ejemplo los módulos o funcionalidades sujeto de verificación, tipos de pruebas, entornos, recursos asignados, entre otros aspectos..



Pasos para elaborar un plan de pruebas de software.

➤ 1.- Analizar los requerimientos de desarrollo de software

Lo primero que debe hacerse es entender los requerimientos de usuario que componen la iteración o proyecto, que son el sujeto de la verificación de calidad que se va a realizar.

Analizar toda la información de la ingeniería de requisitos, incluyendo la **matriz de trazabilidad**, especificaciones y diseño funcional, requisitos no funcionales, **casos de uso**, **historias de usuario** (si estás trabajando con **metodologías ágiles**), entre otra documentación.

También es muy importante realizar entrevistas con el equipo encargado de la ingeniería de requisitos para aclarar dudas y ampliar la información que sea necesaria.

Además de las preguntas específicas de cada requisito, es importante hacer preguntas del alcance general, por ejemplo:

¿Es un sistema nuevo o existente?

¿Cuáles funcionalidades existentes están siendo modificadas?

¿Cuáles son los requisitos no funcionales? Entre otras.

➤ 2.- Identificar las funcionalidades nuevas a probar

A partir de la documentación del **análisis de requisitos** y de las entrevistas con el equipo de ingeniería de requisito y desarrollo, se debe identificar e incluir en el plan de pruebas de software la lista de las funcionalidades (Características) totalmente nuevas.

Si estás trabajando con un sistema informático nuevo, no tendrás problemas en discernir, pues todas serán nuevas.

En el caso de desarrollos de software integrados a un sistema existente es necesario revisar con los analistas de negocio y también con los **arquitectos de software** las funcionalidades que forman parte del desarrollo de software, en todas las capas de la arquitectura.

3.- Identificar las funcionalidades de sistemas existentes que deben probarse

Se debe identificar las funcionalidades existentes que estén siendo impactadas por el desarrollo de alguna forma, considerando todos los componentes afectados en todas las capas de la arquitectura de software.

Existen dos situaciones que se puede encontrar al identificar estas funcionalidades:

Funcionalidades modificadas de cara al usuario: Por ejemplo si una funcionalidad está siendo modificada agregando más pantallas o cambios a su flujo de proceso, debe ser incluida en el plan de pruebas de software.

Funcionalidades modificadas en sus componentes internos: Son funcionalidades no modificadas de cara al usuario, manteniendo la misma interfaz gráfica y flujo de procesos, sin embargo, si se modifican componentes internos que comparten con otras funcionalidades del sistema, en las capas de lógica de negocio o acceso a datos. Estas deben incluirse en el plan de pruebas de software para determinar a partir de ellas pruebas de regresión a realizar.

4.- Definir la estrategia de pruebas

Consiste básicamente en seleccionar cuáles son los tipos de pruebas de software que se deben realizar. Es recomendable seguir un marco de referencia para determinar los tipos de prueba, como por ejemplo los tipos de pruebas de software definidos por el ISTQB.

Pruebas funcionales

Se determinan los conjuntos de pruebas a realizar, correspondiente con cada funcionalidad nueva o existente que se esté modificando.

Se tienen distintos tipos de pruebas funcionales, por ejemplo, las pruebas de sistema (o pruebas integradas de sistemas), que se realizan después que el equipo de desarrollo ha integrado los componentes de distintas capas.

Las pruebas funcionales son definidas por el ISTQB como pruebas basadas en especificación. Son diseñadas usando técnicas de diseño de pruebas de caja negra.

Pruebas no funcionales

Se define un conjunto de **pruebas no funcionales** para cada requisito de este tipo. Aquí se pueden incluir pruebas sobre el desempeño, tiempo de respuesta, mantenibilidad, Pruebas de seguridad de software, entre otros aspectos, según la **clasificación de requisitos no funcionales** que se tenga para el proyecto.

Pruebas de caja blanca

Según la estructura y arquitectura del software que se esté desarrollando.

Pruebas de regresión

Se definen sobre las funcionalidades modificadas en sus componentes internos.

Tipos de pruebas de software en metodologías ágiles

Si se está trabajando con metodologías ágiles, es recomendable usar como base los 4 cuadrantes del Agile Testing, que define el marco de referencia con todos los tipos de pruebas que debes considerar.

5.- Definir los criterios de inicio, aceptación y suspensión de pruebas

Criterios de aceptación o rechazo:

Para definir los criterios de aceptación o rechazo, es necesario definir el nivel de tolerancia a fallos de calidad. Si la tolerancia a fallos es muy baja puede definirse como criterio de aceptación que el 100% de los casos de prueba estén sin **incidencias**. Lograr este margen en todos los casos de prueba principales y casos borde será muy difícil, y podría comprometer los plazos del proyecto (incrementa los riesgos), pero asegura la calidad del producto.

Por otra parte, puede ser que la intención sea realizar un Soft Launch, o un mínimo producto viable, en ese caso se podría definir como criterio de aceptación el 100% de los casos de prueba principales (considerados clave) y 20% de casos de prueba no principales (casos borde).

Una vez logradas las condiciones, se darán por aceptadas las pruebas y el desarrollo de software.

Criterios de inicio o reanudación:

Definen las condiciones que deben cumplirse para dar inicio o reanudar las pruebas. Por ejemplo, en el caso de inicio la condición podría ser la instalación de los componentes de software en el ambiente y que los casos de pruebas de verificación de ambiente sean exitosos.

Para el caso de la reanudación las condiciones están relacionadas, se determina a partir de cuales criterios de suspensión se presentaron para detener las pruebas. Una vez que estás condiciones ya no existan (sean solventadas) se procede con la reanudación.

Criterios de suspensión:

Las condiciones van a depender de los acuerdos de nivel de servicio (SLAs) internos de la organización y también de los acuerdos establecidos en cada proyecto individual.

Por ejemplo, si se tiene un equipo de pruebas que comparte su esfuerzo entre varios proyectos, se puede definir un criterio de suspensión exigente, un determinado porcentaje de casos fallidos que resulten en incidencias. Si la condición se cumple, se detienen las pruebas y se dedica el personal a otras actividades,

Por otra parte si se tiene un equipo de pruebas con personal dedicado, el criterio de suspensión puede ser poco exigente, por ejemplo solo ocurriendo si se bloquean por incidencia todos los casos de prueba.

6.- Identificar los entornos (ambientes) requeridos

Posteriormente se definen y documentan las características de los entornos de Hardware y Software necesarios para realizar la ejecución de las pruebas de software. Esta información se obtiene a partir del equipo de desarrollo y de los arquitectos de software, quienes pueden suministrar los requisitos mínimos y óptimos para la operación del sistema.

Como mejor práctica, el ambiente de pruebas de software debería ser lo más similar posible al ambiente de producción, sin embargo, no siempre es posible debido a limitaciones de recursos (financieros). En estos casos debe estudiarse cuales son los requisitos que aseguran un mínimo de confiabilidad de estas pruebas respecto al entorno de producción.

Además en esta sección del plan de pruebas, también se definen los requisitos de sistemas operativos, software y herramientas de las estaciones de trabajo de los Testers. Si el alcance del proyecto incluye **pruebas de aplicaciones (Apps) para móviles**, es necesario definir los emuladores y teléfonos inteligentes, con sus respectivos requisitos.

7.- Determinar necesidades de personal y entrenamiento

Debe completarse previamente la estimación del esfuerzo de pruebas a partir del diseño de casos de prueba.

Si aún no se cuenta con la estimación, se puede comenzar por definir los tipos de perfiles de habilidades y conocimientos en Software Testing que se necesitan.

Para ello se puede buscar la respuesta a las siguientes preguntas:

¿Qué conocimientos de procesos de negocio se necesitan?

¿Qué sistemas se están probando y quienes tienen experiencia en su funcionamiento?

¿Se necesitan conocimientos específicos en pruebas de requisitos no funcionales? Por ejemplo para pruebas de desempeño o estrés.

¿Cual herramientas de gestión de calidad de software se va a utilizar?

¿Se necesitan conocimientos en herramientas técnicas como Lenguajes de programación o herramientas de pruebas de webservices?

¿Se necesitan conocimientos en herramientas de pruebas automatizadas?

Requisitos de entrenamiento

Por ejemplo:

Transferencia de conocimientos en el sistema y su operación con el área de negocio.

Formación en metodologías de pruebas de software.

Entrenamiento en tipos de pruebas especializados (desempeño, estrés, arquitectura, caja blanca).

Formación en automatización de pruebas de software.

8.- Establecer la metodología y procedimientos de prueba

La metodología de pruebas de software dependerá de la que se esté utilizando para la gestión del proyecto.

Si se está utilizando una metodología predictiva, las pruebas de software comenzaran con la estimación del esfuerzo de pruebas, diseño y luego la ejecución de las pruebas, Si se están utilizando metodologías ágiles de desarrollo de software, debe considerar las diferencias de las pruebas ágiles de software respecto al enfoque predictivo, por lo que la metodología debe estar alineada con el manifiesto ágil.

Luego de seleccionar la metodología de referencia, se documentan los procedimientos para diseño y ejecución, siguiendo el orden de los pasos definidos, flujos de procesos, condiciones para tomar decisiones, y demás aspectos.

9.- Elaborar la planificación de las pruebas

La planificación de las pruebas abarca:

Matriz de responsabilidades

Esta se define con perfiles genéricos o inclusive con el equipo de trabajo si ya se conoce cuál es el que será asignado. Las tareas del plan de pruebas deben estar alineadas con las habilidades y conocimientos de cada persona.

Cronograma

Elaborado a partir de la estimación de las actividades de Software Testing realizada por el equipo. Para elaborar un cronograma real, es importante definir actividades críticas como por ejemplo los tiempos de instalación de versiones en los entornos de pruebas, pruebas de validación de ambientes antes de comenzar a hacer las pruebas y las iteraciones por incidencias, que es el tiempo invertido en volver a probar los casos de prueba fallidos.

Premisas

Son las condiciones que deben cumplirse para que el cronograma sea realizable, estas se determinan a partir de la documentación de entornos y de los requisitos de personal. Por ejemplo disponibilidad de ciertos entornos, disponibilidad de personal con algún conocimiento técnico específico, la metodología que se va a utilizar, entre otros.

10.- Identificar los riesgos y definir planes de respuesta

Para el Software Testing, los riesgos por lo general están vinculados con factores como:

Posibles dificultades en la disponibilidad de entornos.

Pruebas que dependen de factores externos al proyecto y la organización.

Disponibilidad de personal con conocimientos especializados en alguna herramienta, o en la funcionalidad específica que se está desarrollando.

Dependencias con otros proyectos.

Posibilidad que alguna premisa no se cumpla.

Para identificar los riesgos es necesario enumerar cada una de estas dependencias y por medio de mesas de trabajo y tormentas de ideas pensar en las posibilidades de que algo salga mal (u oportunidades para que salga bien).

Luego de la identificación, es necesario también definir planes de respuesta, los cuales deben ser específicos para cada situación particular y riesgo.

¿Dudas o Preguntas?

<https://coggle.it/diagram/XXp0A0d5XRpWv-R9/t/estructura-de-plan-de-pruebas>

El ISTQB (International Software Testing Qualifications Board) es una organización de certificación de la calidad del software que opera internacionalmente.