Guía de desarrollo, Anexo 03.00

Proceso de Pruebas

Autor: Oficina de Pruebas

GERENCIA INFORMÁTICA
JOSEFA VALCÁRCEL, 44
28027-MADRID

Índice General

1 INTRODUCCIÓN 3

1.1 OBJETO 3
1.2 ALCANCE 3

2 PROCESO DE PRUEBAS 3

2.1 DIAGRAMA DEL PROCESO 3
2.2 DESCRIPCIÓN DE LAS ACTIVIDADES 4
2.2.1 Análisis de pruebas 4
2.2.2 Especificación de pruebas 5
2.2.3 Ejecución de pruebas 5
2.2.3.1 Diseño de ciclos de pruebas 5
2.2.3.2 Ejecución de pruebas 6
2.2.3.3 Registrar defectos 6
2.2.4 Mantenimiento de pruebas 9

Índice de Ilustraciones

Ilustración 1: Flujo de trabajo del proceso de pruebas 4
Ilustración 2: Ciclo de vida de un defecto 8

1 Introducción

1.1 Objeto

El presente documento establece los pasos que son necesarios para realizar un proceso de pruebas de una aplicación en la DGT. El proceso de pruebas tiene como objetivo demostrar que el software realiza correctamente aquellas tareas para las que fue diseñado.

1.2 Alcance

La normativa que aquí se expone es **de obligado cumplimiento** para todas aquellas personas que vayan a realizar un proceso de pruebas sobre una aplicación en la DGT: La Oficina de Pruebas, Empresas externas de Desarrollo, etc…

2 Proceso de Pruebas

2.1 Diagrama del proceso

El diagrama del proceso de pruebas nos muestra las actividades que componen el flujo de trabajo dentro del proceso de pruebas, indicando para cada una de ellas las entradas y salidas.

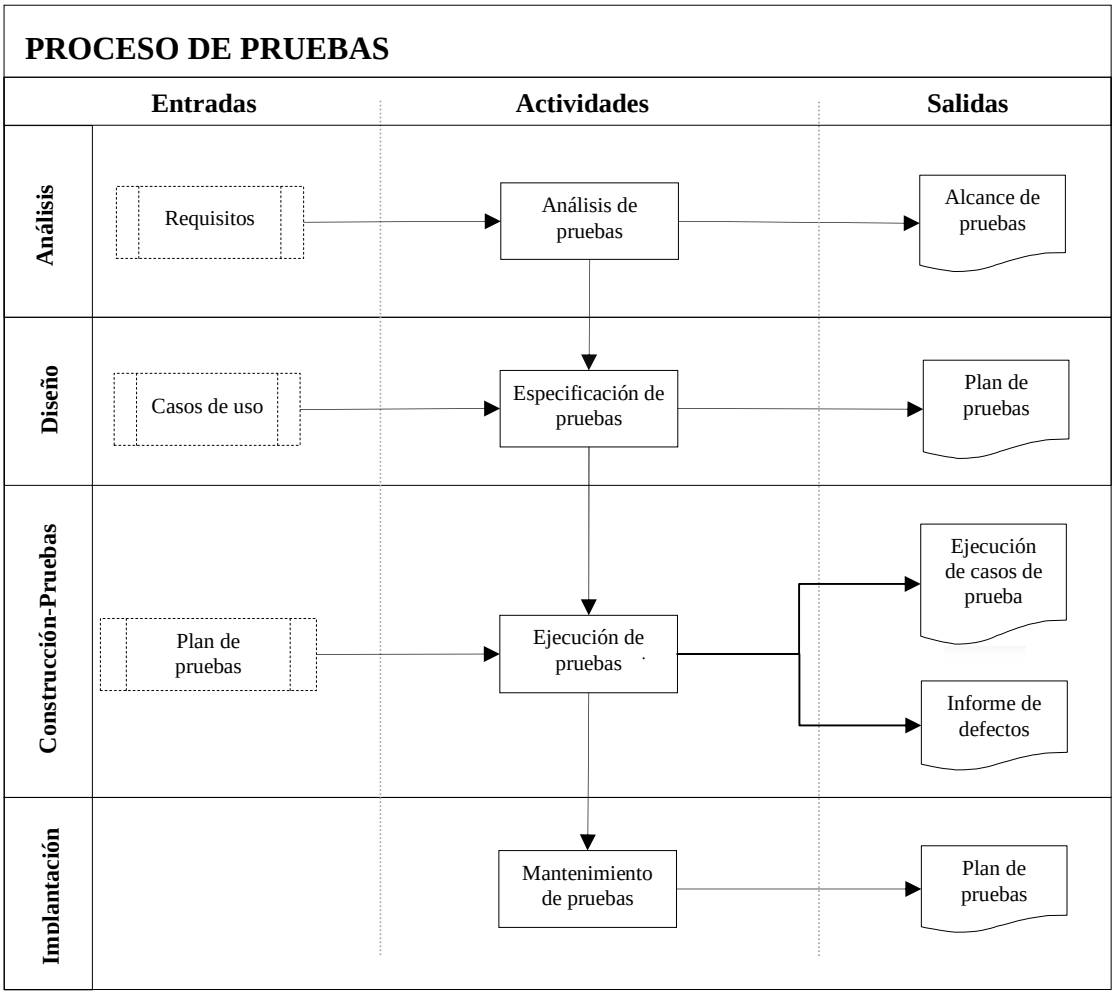

Ilustración 1: Flujo de trabajo del proceso de pruebas

2.2 Descripción de las actividades

2.2.1 Análisis de pruebas

En la actividad de análisis de pruebas es donde se define el alcance de las pruebas que se van a realizar sobre una aplicación, de manera que nos proporcionará las pruebas que aplican y cuáles no. Para poder definir el alcance es necesario analizar los tipos de requisitos dentro del contexto de la aplicación, ya que no todos serán probados con la misma profundidad, y pueden existir requisitos que queden fuera del alcance de las pruebas.

2.2.2 Especificación de pruebas

En la actividad de especificación del plan de pruebas es donde se (re)diseñan los casos de prueba que forman parte del plan de pruebas de la aplicación. Algunos de los entregables para realizar este diseño pueden ser los casos de uso, arquitectura del sistema, diseños, manuales de usuarios, manuales técnicos, etc. La especificación de casos de prueba debe considerar la elaboración de casos positivos y negativos, ya que también se debe validar el comportamiento de la aplicación en situaciones atípicas.

Las tareas necesarias para realizar la especificación del plan de pruebas son las siguientes:

- Crear y/o actualizar casos de prueba.
- Definir para cada caso de prueba un identificador único, una descripción, objetivo, condiciones de ejecución, prioridad, las acciones o pasos y el resultado esperado.
- Identificar los datos de prueba necesarios para la ejecución de los casos de prueba.
- Crear y/o actualizar la cobertura del caso de prueba con el requisito que verifica.

2.2.3 Ejecución de pruebas

En esta fase se realiza la ejecución de los casos de prueba creados en la especificación que permitirán proporcionar información de la calidad del software. También se registran y notifican los defectos encontrados durante la ejecución de las pruebas.

2.2.3.1 Diseño de ciclos de pruebas

En esta fase se diseñan los ciclos de pruebas que se definen como un conjunto de casos de prueba que se ejecutan a la vez y que tienen un objetivo común.

La selección de los casos de prueba para la formación de los ciclos de prueba se realizará mediante la priorización asignada a los requisitos y a los casos de prueba, de manera que serán ejecutados primero los de prioridad más alta y se irá bajando en profundidad.

Otros criterios que pueden ser utilizados para la creación de los ciclos de pruebas son los siguientes:

- Ciclo de pruebas de **regresión**. Son aquellas pruebas que ya se han ejecutado anteriormente y se vuelven a ejecutar en la nueva versión modificada para asegurar el buen funcionamiento

después de modificar el aplicativo. El objetivo de estas pruebas es asegurar que los cambios identificados en la ejecución anterior de la prueba se han corregido, y que los cambios realizados no han introducido nuevos defectos en el aplicativo.

- Ciclo de pruebas de **humo**. Su objetivo es validar que la operativa básica de la versión se comportan según lo especificado. Las pruebas de humo son menos exigentes que las pruebas de regresión, de manera que si la versión no pasa este ciclo de pruebas, no se comienza con la ejecución de las pruebas planificadas de esta versión.
- Ciclo de pruebas de **confirmación**. Su objetivo es comprobar que los defectos encontrados en la versión anterior han sido reparados en la versión actual.

2.2.3.2 Ejecución de pruebas

En esta fase se ejecutarán los casos de prueba dentro de los ciclos de prueba definidos en la fase anterior y respetando la priorización otorgada a los mismos.
Cada ciclo de pruebas debe estar asociado a una revisión de la aplicación y debe tener asociado un juego de datos.
Durante el transcurso de la ejecución de pruebas se actualizará el estado de los casos de prueba de manera que se permite conocer en todo momento el grado de avance de las pruebas sobre el sistema.

2.2.3.3 Registrar defectos

Cualquier error encontrado durante la ejecución de pruebas será registrado y documentado por la oficina de pruebas, con el fin de tener la suficiente información como para poder reproducir el error y realizar un análisis del mismo para su resolución.

Un defecto contendrá la siguiente información:

- **Título**. En este campo se debe especificar un pequeño resumen del defecto encontrado.
- **Descripción**. Este campo debe contener una información muy detallada del defecto encontrado.
- **Severidad**. Nivel de impacto que supone el defecto encontrado. Existen tres niveles de severidad:

    - Crítico. Son defectos que impiden la operativa del sistema total o parcialmente, imposibilitando su utilización, producen una inconsistencia o pérdida de datos y dificultan el trabajo sin impedir la explotación de la funcionalidad.
    - Medio. Son defectos que producen una perdida menor de funcionalidad o un problema al cual se le puede dar una solución alternativa.
    - Trivial. Son defectos cosméticos como puede ser una falta de ortografía o un texto desalineado.
- **Detectado en Release**. Versión etiquetada en la que se detecta el defecto.
- **Detectado en ciclo**. Ciclo-Revisión etiquetada en la que se detecta el defecto.
- **Entorno de Ejecución**. Entorno de ejecución en el cual fue detectado el defecto.
- **Reparado en Release**. Versión etiquetada en la que se soluciona el defecto.
- **Reparado en ciclo**. Ciclo- Revisión etiquetada en la que se soluciona el defecto.
- **Estado**. Estado en el que se encuentra el defecto.
- **Comentarios**. En este campo se registra todos los comentarios que se consideren oportunos por cualquier usuario que esté involucrado en la resolución del defecto.
- **Evidencia**. Siempre que sea posible se asociará al defecto un snapshot del mismo que permita al equipo de desarrollo entender con mayor claridad el defecto encontrado.

Existe un ciclo de vida del defecto desde su creación hasta su cierre, en el cual se ven implicados tres perfiles distintos, Jefe de Proyecto DGT, Desarrollo y Oficina de Pruebas. A continuación se muestra el ciclo de vida de un defecto:

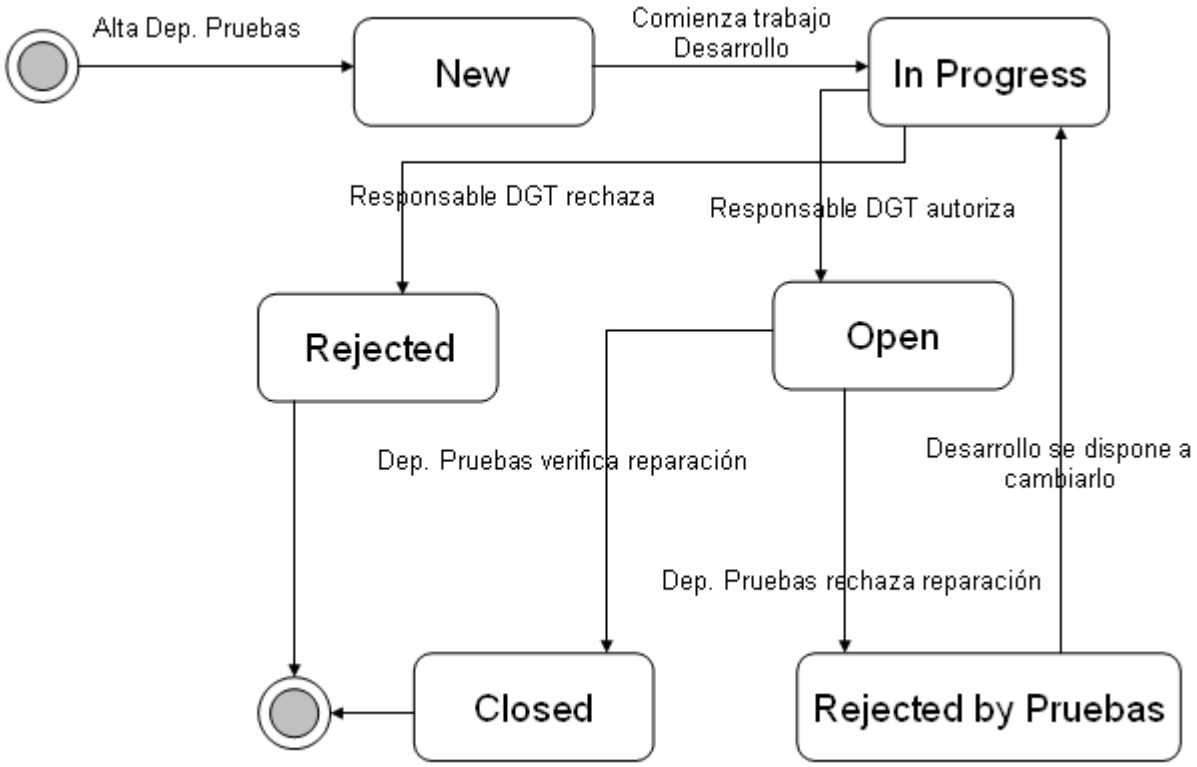

Ilustración 2: Ciclo de vida de un defecto

Los estados de un defecto son los siguientes:

- **New**. Es el estado que tiene un defecto en el momento de su creación por la Oficina de Pruebas.
- **In Progress**. El equipo de desarrollo se ha puesto a trabajar en el defecto.
- **Open**. Defecto aprobado por el Jefe de Proyecto DGT para su resolución.
- **Rejected**. Defecto rechazado por el Jefe de Proyecto DGT, ya que no es un defecto sino que la aplicación debe funcionar de esta manera.
- **Rejected by Pruebas**. La oficina de Pruebas verifica que el defecto no ha sido solucionado por lo que lo devuelve al equipo de desarrollo para su nueva resolución.
- **Closed**. La oficina de Pruebas verifica que el defecto se ha solucionado y lo cierra.

Un defecto estará siempre enlazado a un caso de prueba del plan de pruebas de manera que se permita conocer, mediante la cobertura entre casos de prueba y requisitos, la operativa que está fallando para una determinada revisión de la aplicación. Si el defecto en ejecución no tiene asociado un caso de prueba será necesario enlazarle uno o varios, que pueden ser de los ya existentes en el plan de pruebas o puede que sea necesario la creación de uno/s nuevo/s debido a que el defecto en ejecución detectado no se pueda cubrir con ninguno de los casos de prueba definidos.

2.2.4 Mantenimiento de pruebas

El mantenimiento del plan de pruebas será necesario ante cualquiera de los siguientes cambios:

- Cambio evolutivo. Son las acciones a realizar por la incorporación de nuevos requisitos, modificaciones o eliminaciones de los existentes con el fin de incorporar las nuevas necesidades en el plan de pruebas.
- Cambio correctivo. Son las acciones a realizar ante un funcionamiento incorrecto o incompleto de los casos de prueba. Durante la fase de ejecución de pruebas se suelen ejecutar pruebas explícitas que no están diseñadas por lo que se hace necesario actualizar la especificación de pruebas con los nuevos casos de prueba.
- Cambio perfectivo. Son las acciones llevadas a cabo para mejorar la calidad de los casos de prueba en cualquiera de sus aspectos: descripción, objetivo, datos, pasos, etc.