

# Unidad 1

## Fundamentos del Proceso de Prueba

### 1. ¿Qué es probar?

Los sistemas software son una parte integral de la vida, desde las aplicaciones de negocio (por ejemplo, la banca) hasta los productos de consumo (por ejemplo, los automóviles). La mayoría de las personas han tenido una experiencia con algún producto software que no funcionó como se esperaba. Un producto software que no funciona correctamente puede causar muchos problemas, incluyendo pérdida de dinero, tiempo o reputación en el ámbito del negocio, e incluso lesiones o muerte.

**Probar el software es una forma de evaluar la calidad del software y de reducir el riesgo de fallos en un entorno de operaciones o en producción.**

Una percepción errónea común de la prueba es que sólo consiste en ejecutar pruebas, es decir, ejecutar el software y comprobar los resultados. La prueba de software es un proceso que incluye muchas actividades diferentes; la ejecución de la prueba (incluida la comprobación de los resultados) es sólo una de estas actividades. **El proceso de prueba también incluye actividades tales como planificar la prueba, analizar, diseñar e implementar pruebas, informar del avance y de los resultados de la prueba, y evaluar la calidad de un objeto de prueba.**

Cuando la prueba implica la ejecución del componente o sistema que se está probando; esta prueba se denomina **prueba dinámica**.

Otras formas de prueba no implican la ejecución del componente o sistema que se está probando; estas formas de pruebas se denominan **pruebas estáticas**. Por lo tanto, las pruebas también incluyen la revisión de productos de trabajo tales como requisitos, historias de usuario y código fuente.

Otra percepción errónea común de la prueba es que se centra exclusivamente en la verificación de requisitos, historias de usuarios u otras especificaciones. Si bien las pruebas implican la comprobación de si el sistema cumple los requisitos especificados, también implican la **validación, que consiste en comprobar si el sistema es capaz de satisfacer las necesidades de los usuarios y de otros implicados en su(s) entorno(s) de operación.**

#### 1.1 Objetivos Característicos de la Prueba

Para cualquier proyecto dado, los objetivos de prueba pueden incluir:

- Evaluar productos de trabajo tales como requisitos, historias de usuario, diseño y código.
- Verificar el cumplimiento de todos los requisitos especificados.
- Validar si el objeto de prueba está completo y funciona como los usuarios y otros implicados esperan.
- Generar confianza en el nivel de calidad del objeto de prueba.
- Prevenir defectos.
- Encontrar fallos y defectos.

- Proporcionar suficiente información a los implicados para que puedan tomar decisiones informadas<sup>2</sup>, especialmente en relación con el nivel de calidad del objeto de prueba.
- Reducir el nivel de riesgo de calidad inadecuada del software (por ejemplo, fallos que se producen durante la operación que no han sido detectados anteriormente).
- Cumplir con requisitos o normas contractuales, legales o reglamentarias, y/o verificar el cumplimiento de dichos requisitos o normas por parte del objeto de prueba.

**Los objetivos de la prueba pueden variar, dependiendo del contexto del componente o sistema que se está probando, el nivel de prueba y el modelo de ciclo de vida de desarrollo de software.**

Estas diferencias pueden incluir, por ejemplo:

- Durante la prueba de componente, uno de los objetivos puede ser encontrar tantos fallos como sea posible para que los defectos subyacentes se identifiquen y se solucionen de forma temprana. Otro objetivo puede ser aumentar la cobertura de código de las pruebas de componente.
- Durante la prueba de aceptación, uno de los objetivos puede ser confirmar que el sistema funciona como se espera y cumple con los requisitos. Otro objetivo de este proceso de prueba puede ser informar a los implicados sobre el riesgo de liberar<sup>3</sup> el sistema en un momento dado.

## 1.2 Prueba y Depuración

La prueba y la depuración son diferentes.

- La ejecución de pruebas puede mostrar fallos causados por defectos en el software.
- La depuración es la actividad de desarrollo que encuentra, analiza y corrige dichos defectos.
- La prueba de confirmación posterior comprueba si las correcciones han resuelto los defectos.

En algunos casos, los probadores son responsables de la prueba inicial y la prueba de confirmación final, mientras que los desarrolladores realizan la depuración y la prueba de componente asociada. Sin embargo, en el desarrollo Ágil y en algunos otros ciclos de vida, los probadores pueden estar involucrados en la depuración y la prueba de componente.



**La norma ISO (ISO/IEC/IEEE 29119-1) contiene información adicional sobre conceptos de la prueba de software.**

## 2. ¿Por qué es necesario Probar?

La prueba rigurosa de componentes y sistemas, y su documentación asociada, pueden ayudar a reducir el riesgo de que se produzcan fallos durante la operación. Cuando se detectan defectos, y posteriormente se corrigen, esto **contribuye a la calidad de los componentes o sistemas**. Además, la prueba del software también puede ser necesaria para cumplir con requisitos contractuales o legales o estándares específicos de la industria.

### 2.1 Contribuciones de la Prueba al Éxito

A lo largo de la historia de la informática, es bastante común que el software y los sistemas se entreguen operaciones y, debido a la presencia de defectos, posteriormente causen fallos o no satisfagan las necesidades de los implicados.

Sin embargo, la utilización de **técnicas de prueba adecuadas puede reducir la frecuencia de estas entregas problemáticas**, cuando estas técnicas se aplican con el nivel adecuado de experiencia en materia de prueba, en los niveles de prueba adecuados y en los puntos adecuados del ciclo de vida del desarrollo del software.

Algunos ejemplos son:

- El hecho de contar con probadores involucrados en la revisión de los requisitos o en el refinamiento de historias de usuario podría resultar en la detección de defectos en estos productos de trabajo. La identificación y eliminación de defectos en los requisitos reduce el riesgo de que se desarrollen funcionalidades incorrectas o que no puedan ser probadas (ausencia de capacidad de ser probado).
- El hecho de que los probadores trabajen en estrecha colaboración con los diseñadores de sistemas mientras se diseña el sistema puede aumentar la comprensión de cada una de las partes sobre el diseño y la forma de probarlo. Esta mayor comprensión puede reducir el riesgo de defectos fundamentales de diseño y permitir la identificación de pruebas en una fase temprana.
- El hecho de que los probadores trabajen en estrecha colaboración con los desarrolladores mientras el código está en desarrollo puede aumentar la comprensión del código por parte de cada una de las partes y la forma de probarlo. Esta mayor comprensión puede reducir el riesgo de defectos dentro del código y de la prueba.
- Hacer que los probadores verifiquen y validen el software antes de liberarlo puede detectar fallos que de otro modo podrían haberse omitido, y apoyar el proceso de eliminación de los defectos que causaron los fallos (es decir, la depuración). Esto aumenta la probabilidad de que el software cumpla con las necesidades de los implicados y satisfaga los requisitos.

Además de estos ejemplos, el logro de objetivos de prueba definidos contribuye al éxito general del desarrollo y mantenimiento del software.

### 2.2 Aseguramiento de la Calidad y Proceso de Prueba

Mientras que, a menudo, las personas utilizan el término aseguramiento de la calidad<sup>6</sup> (o simplemente QA por sus siglas en inglés) para referirse a la prueba, **el aseguramiento de la calidad y la prueba no son lo mismo, pero están relacionados**. Un concepto más amplio, la gestión de la calidad, los une.

**La gestión de la calidad incluye todas las actividades que dirigen y controlan una organización con respecto a la calidad.** Entre otras actividades, la gestión de la calidad incluye tanto el aseguramiento de la calidad como el control de la calidad.

**El aseguramiento de la calidad se centra**, por lo general, **en el cumplimiento de los procesos adecuados, a fin de proporcionar la confianza de que se alcanzarán los niveles de calidad adecuados.** Cuando los procesos se llevan a cabo de forma correcta, los productos de trabajo creados por esos procesos son generalmente de mayor calidad, lo que contribuye a la prevención de defectos. Además, el uso del análisis de la causa raíz para detectar y eliminar las causas de los defectos, junto con la aplicación adecuada de los hallazgos de las reuniones retrospectivas para mejorar los procesos, son importantes para un aseguramiento de la calidad eficaz.

**El control de la calidad implica varias actividades, incluyendo actividades de prueba, que apoyan el logro de niveles apropiados de calidad.** Las actividades de prueba son parte del proceso general de desarrollo y mantenimiento del software.

Dado que el aseguramiento de calidad se ocupa de la correcta ejecución de todo el proceso, el control de calidad promueve la realización de la prueba adecuada.

## 2.3 Errores, Defectos y Fallos

Una persona puede cometer un error (equivocación), que puede llevar a la introducción de un defecto (falta o bug) en el código software o en algún otro producto de trabajo relacionado. Un error que conduce a la introducción de un defecto en un producto de trabajo puede desencadenar un error que conduce a la introducción de un defecto en un producto de trabajo con el que se encuentra relacionado. Por ejemplo, un error en la educación<sup>7</sup> de requisitos puede conducir a un defecto en el requisito, lo que a su vez resulta en un error de programación que conduce a un defecto en el código.

Si se ejecuta un fragmento de código que contiene un defecto, esto puede causar un fallo, pero no necesariamente en todas las circunstancias. Por ejemplo, algunos defectos requieren entradas o precondiciones muy específicas para desencadenar un fallo, que puede ocurrir rara vez o nunca.

Los errores pueden ocurrir por muchas razones, tales como:

- Presión por causa de tiempo.
- Falibilidad humana.
- Participantes en el proyecto sin experiencia o poco cualificados.
- Falta de comunicación entre los participantes en el proyecto, incluida la falta de comunicación con respecto a requisitos y diseño.
- Complejidad del código, diseño, arquitectura, el problema subyacente que se debe resolver, y/o las tecnologías utilizadas.
- Malentendidos acerca de las interfaces intra e intersistemas, especialmente cuando esas interacciones intra e intersistemas son numerosas.
- Tecnologías nuevas y desconocidas.

Además de los fallos causados por defectos en el código, los fallos también pueden ser causados por condiciones medioambientales. Por ejemplo, la radiación, los campos electromagnéticos y la contaminación pueden causar defectos en el firmware o influir en la ejecución del software al cambiar las condiciones del hardware.



No todos los resultados inesperados de la prueba son fallos. Pueden ocurrir falsos positivos debido a errores en la forma en que se ejecutaron las pruebas, o debido a defectos en los datos de prueba, el entorno de prueba, u otro producto de prueba, o por otras razones. La situación inversa también puede ocurrir, donde errores o defectos similares conducen a falsos negativos. Los falsos negativos son pruebas que no detectan defectos que deberían haber detectado; los falsos positivos se informan como defectos, pero en realidad no son defectos.

## 2.4 Defectos, Causas Raíz y Efectos

**Las causas raíz de los defectos son las acciones o condiciones más tempranas que contribuyeron a crear estos defectos.** Se pueden analizar los defectos para identificar sus causas raíz, con el propósito de reducir la ocurrencia de defectos similares en el futuro.

**Al centrarse en las causas raíz más significativas, el análisis de la causa raíz puede conducir a mejoras en el proceso que previenen la introducción de un número significativo de futuros defectos.**

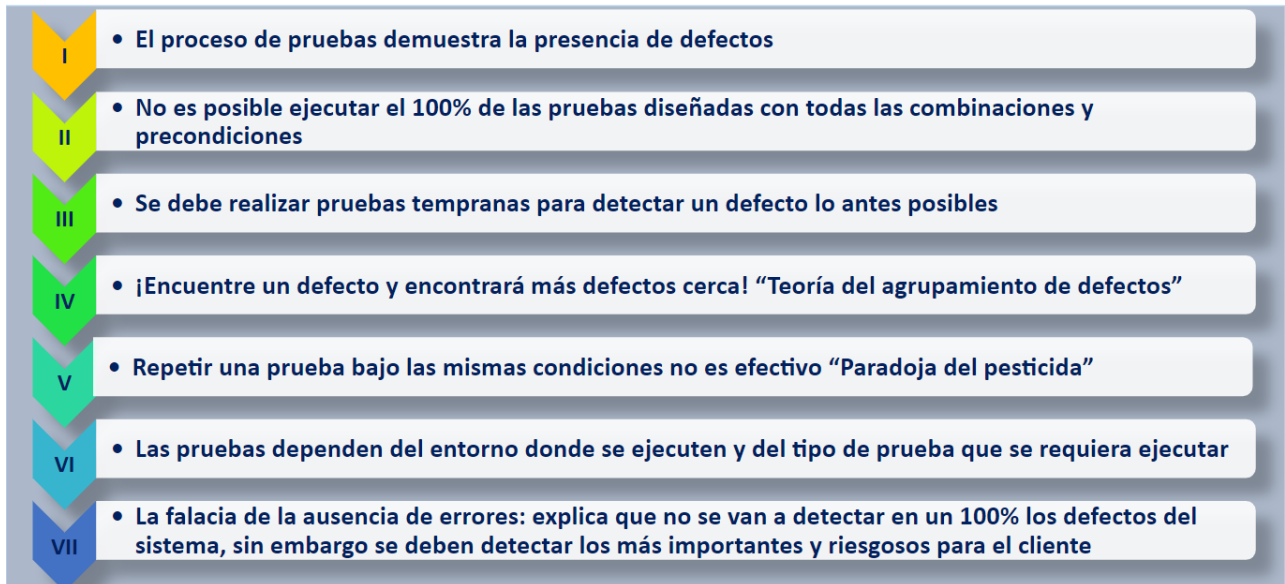
Por ejemplo, suponer que los pagos de intereses incorrectos, debido a una sola línea de código incorrecto, dan lugar a reclamaciones por parte de los clientes. El código defectuoso fue escrito para una historia de

usuario que era ambigua, debido al malentendido del propietario del producto sobre cómo calcular el interés. Si existe un gran porcentaje de defectos en el cálculo de intereses, y estos defectos tienen su origen en malentendidos similares, los propietarios de producto podrían recibir formación en materia de cálculo de intereses para reducir dichos defectos en el futuro.

En este ejemplo, las reclamaciones de los clientes son los efectos. Los pagos de intereses incorrectos son fallos. El cálculo incorrecto en el código es un defecto, y resultó del defecto original, la ambigüedad en la historia de usuario. La causa raíz del defecto original fue la falta de conocimiento por parte del propietario del producto, lo que resultó en que el propietario del producto cometiera una equivocación al escribir la historia de usuario.

### 3. Siete principios de la prueba

## 7 Principios del proceso de prueba



En los últimos 50 años se han sugerido una serie de principios del proceso de prueba que ofrecen directrices generales comunes para toda prueba.

#### 1. La prueba muestra la presencia de defectos, no su ausencia

La prueba puede mostrar la presencia de defectos, pero no puede probar que no hay defectos. La prueba reduce la probabilidad de que queden defectos no descubiertos en el software pero, incluso si no se encuentran defectos, el proceso de prueba no es una demostración de la corrección.

#### 2. La prueba exhaustiva es imposible

No es posible probar todo (todas las combinaciones de entradas y precondiciones) excepto en casos triviales. En lugar de intentar realizar pruebas exhaustivas se deberían utilizar el análisis de riesgos, lastécnicas de prueba y las prioridades para centrar los esfuerzos de prueba.

#### 3. La prueba temprana ahorra tiempo y dinero

Para detectar defectos de forma temprana, las actividades de prueba tanto estáticas como dinámicas deben iniciarse lo antes posible en el ciclo de vida de desarrollo de software. La prueba temprana a veces se denomina *desplazamiento hacia la izquierda*<sup>9</sup>. La prueba temprana en el ciclo de vida de desarrollo de software ayuda a reducir o eliminar cambios costosos (ver sección 3.1).

#### 4. Los defectos se agrupan

En general, un pequeño número de módulos contiene la mayoría de los defectos descubiertos durante la prueba previa al lanzamiento, o es responsable de la mayoría de los fallos operativos. Las agrupaciones de defectos previstas y las agrupaciones de defectos reales observadas en la prueba o producción son una aportación importante a un análisis de riesgos utilizado para centrar el esfuerzo de la prueba (como se menciona en el principio 2).

#### 5. Cuidado con la paradoja del pesticida

Si las mismas pruebas se repiten una y otra vez, eventualmente estas pruebas ya no encontrarán ningún defecto nuevo. Para detectar nuevos defectos, es posible que sea necesario cambiar las

pruebas y los datos de prueba existentes, y es posible que sea necesario redactar nuevas pruebas. (Las pruebas ya no son efectivas para detectar defectos, de la misma manera que los pesticidas ya no son efectivos para matar insectos después de un tiempo). En algunos casos, como la prueba de regresión automatizada, la laparadoja del pesticida tiene un resultado beneficioso, que es el número relativamente bajo de defectos de regresión.

#### **6. La prueba depende del contexto**

La prueba se realiza de manera diferente en diferentes contextos. Por ejemplo, el software de control industrial de seguridad crítica se prueba de forma diferente a una aplicación móvil de comercio electrónico. Como ejemplo adicional, la prueba en un proyecto Ágil se realiza de manera diferente a la prueba en un proyecto que se desarrolla según un ciclo de vida secuencial (ver sección 2.1).

#### **7. La ausencia de errores es una falacia**

Algunas organizaciones esperan que los probadores puedan realizar todas las pruebas posibles y encontrar todos los defectos posibles, pero los principios 2 y 1, respectivamente, nos dicen que esto es imposible. Además, es una falacia (es decir, una creencia equivocada) esperar que sólo con encontrar y corregir un gran número de defectos se asegure el éxito de un sistema. Por ejemplo, la realización de pruebas exhaustivas de todos los requisitos especificados y la reparación de todos los defectos encontrados podrían dar lugar a un sistema difícil de utilizar, que no satisface las necesidades y expectativas de los usuarios o que es peor en comparación con otros sistemas de la competencia.

## 4. El proceso de prueba

No existe un proceso de prueba de software único y universal, pero existen conjuntos de actividades de prueba comunes sin las cuales es menos probable que la prueba alcance los objetivos establecidos. **Estos conjuntos de actividades de prueba son un proceso de prueba.**

El proceso de prueba de software adecuado y específico en cualquier situación depende de muchos factores. Qué actividades de prueba están involucradas en este proceso de prueba, cómo se implementan estas actividades y cuándo ocurren, pueden ser abordadas en la estrategia de prueba de una organización.

### 4.1 El Proceso de Prueba en Contexto

Los factores de contexto que influyen en el proceso de prueba de una organización incluyen, pero no están limitados a:

- Modelo de ciclo de vida de desarrollo de software y metodologías de proyecto en uso.
- Niveles y tipos de prueba considerados.
- Riesgos de producto y de proyecto.
- Dominio del negocio.
- Restricciones operativas, incluyendo pero no limitadas a:
  - Presupuestos y recursos.
  - Plazos.
  - Complejidad.
  - Requisitos contractuales y normativos.
- Políticas y prácticas de la organización.
- Estándares internos y externos necesarios.

Las siguientes secciones describen los aspectos generales de los procesos de prueba en una organización en los siguientes términos:

- Actividades y tareas de prueba.
- Productos de trabajo de la prueba.
- Trazabilidad entre la base de prueba y los productos de trabajo de la prueba.

Es muy útil si **la base de prueba (para cualquier nivel o tipo de prueba que se esté considerando) tiene definidos criterios de cobertura medibles**. Los criterios de cobertura pueden actuar eficazmente como indicadores clave de desempeño<sup>9</sup> (KPI's<sup>10</sup>, por sus siglas en inglés) para controlar las actividades que permiten demostrar el logro de los objetivos de la prueba de software

Por ejemplo, para una aplicación móvil, la base de prueba puede incluir una lista de requisitos y una lista de dispositivos móviles soportados. Cada requisito es un elemento de la base de prueba. Cada dispositivo soportado es también un elemento de la base de prueba. Es posible que los criterios de cobertura requieran, como mínimo, un caso de prueba para cada elemento de la base de prueba. Una vez ejecutadas, los resultados de estas pruebas indican a los implicados si se cumplen los requisitos especificados y si se observaron fallos en los dispositivos soportados.

La norma ISO (ISO/IEC/IEEE 29119-2) contiene más información sobre los procesos de prueba.



## 4.2 Actividades y Tareas de Prueba

Un proceso de prueba consiste en los siguientes grupos de actividades principales:

1. Planificación de prueba	• Cómo gestionar las actividades de prueba en el proyecto
2. Monitorización y control de la prueba	• Dar seguimiento a la implementación del Plan de Pruebas y ajustarlo si es necesario
3. Análisis de prueba	• Identificar qué probar
4. Diseño de prueba	• Documentar los casos de prueba de alto nivel
5. Implementación de prueba	• Prepararse para iniciar la ejecución de prueba
6. Ejecución de prueba	• ¡Ejecute los casos de prueba, registre los resultados e informe los defectos (si los hay)!
7. Compleción de la prueba	• Informe final sobre las actividades de prueba e identificación de posibles mejoras

Cada grupo de actividades se compone de actividades integrantes, que se describirán en las siguientes subsecciones. Cada actividad dentro de cada grupo de actividades a su vez puede estar compuesta por múltiples tareas individuales, que pueden variar de un proyecto a otro o de un lanzamiento a otro.

Además, aunque muchos de estos grupos de actividades pueden parecer secuenciales, desde un punto de vista lógico, a menudo se implementan de manera iterativa. Por ejemplo, el desarrollo Ágil implica pequeñas iteraciones de diseño, construcción y prueba de software que se realizan de forma continua, con el apoyo de una planificación continua. Por lo tanto, las actividades de prueba también se llevan a cabo de forma iterativa y continua dentro de este enfoque de desarrollo. Incluso en el desarrollo secuencial, la secuencia lógica escalonada de actividades implicará superposición, combinación, concurrencia u omisión, por lo que normalmente es necesario adaptar estas actividades principales dentro del contexto del sistema y del proyecto.

### A. Planificación de la Prueba

La planificación de la prueba implica actividades que definen los objetivos de la prueba y el enfoque para cumplir con los objetivos de la prueba dentro de las restricciones impuestas por el contexto (por ejemplo, la especificación de técnicas y tareas de prueba adecuadas y la formulación de un calendario de pruebas para cumplir con un plazo límite). Los planes de prueba pueden ser revisados en función de la retroalimentación de las actividades de monitorización y control. La planificación de la prueba se explica con más detalle en la sección 5.2.

### B. Monitorización y Control de la Prueba

La monitorización (o seguimiento) de la prueba implica la comparación continua del avance real con respecto al plan de prueba utilizando cualquier métrica de monitorización de la prueba definida en el plan de prueba. El control de la prueba implica tomar las medidas necesarias para cumplir los objetivos del plan de prueba (que puede actualizarse con el tiempo). La monitorización y el control de la prueba se apoyan en la evaluación de los criterios de salida, a los que se hace referencia como definición de hecho<sup>12</sup> en algunos ciclos de vida. Por ejemplo, la evaluación de los criterios de salida para la ejecución de la prueba como parte de un nivel de prueba dado puede incluir:

- Comprobar los resultados y los registros de la prueba en relación con los criterios de cobertura especificados.

- Evaluar el nivel de calidad de los componentes o sistemas en base a los resultados y los registros de prueba.
- Determinar si se necesitan más pruebas (por ejemplo, si las pruebas originalmente destinadas a alcanzar un cierto nivel de cobertura de riesgo de producto no lo alcanzaran, sería necesario redactar y ejecutar pruebas adicionales).

El avance de la prueba con respecto al plan se comunica a los implicados por medio de informes de avance de la prueba, incluyendo las desviaciones con respecto al plan y la información que permita apoyar cualquier decisión de interrumpir la prueba.

La monitorización y el control de la prueba se explican con más detalle en la sección 5.3.

### C. Análisis de la Prueba

Durante el análisis de la prueba, se analiza la base de prueba para identificar las prestaciones<sup>13</sup> que presentan capacidad de ser probadas y definir las condiciones de prueba asociadas. En otras palabras, el análisis de la prueba determina "qué probar" en términos de criterios de cobertura medibles.

El análisis de la prueba incluye las siguientes actividades principales:

- Analizar la base de prueba correspondiente al nivel de prueba considerado, por ejemplo:
  - Especificaciones de requisitos, tales como requisitos de negocio, requisitos funcionales, requisitos de sistema, historias de usuario, épicas, casos de uso o productos de trabajos similares que especifiquen el comportamiento funcional y no funcional deseado del componente o sistema.
  - Información de diseño e implementación, tales como diagramas o documentos de arquitectura del sistema o del software, especificaciones de diseño, flujos de llamada, diagramas de modelado (por ejemplo, diagramas UML o entidad-relación), especificaciones de interfaz o productos de trabajo similares que especifiquen componentes o estructura del sistema.
  - La implementación del componente o sistema en sí, incluyendo código, metadatos de base de datos y consultas, e interfaces.
  - Informes de análisis de riesgos, que pueden considerar aspectos funcionales, no funcionales y estructurales del componente o sistema.
- Evaluar la base de prueba y los elementos de prueba para identificar defectos de distintos tipos, tales como:
  - Ambigüedades.
  - Omisiones.
  - Inconsistencias.
  - Inexactitudes.
  - Contradicciones.
  - Enunciados superfluos.
- Identificar las prestaciones y conjuntos de prestaciones que se van a probar.
- Definir y priorizar las condiciones de prueba para cada prestación basándose en el análisis de la base de prueba, y considerando las características funcionales, no funcionales y estructurales, otros factores de negocio y técnicos, y los niveles de riesgo.
- Captura de la trazabilidad bidireccional entre cada elemento de la base de prueba y las condiciones de prueba asociadas.

La aplicación de técnicas de prueba de caja negra, caja blanca y basadas en la experiencia puede ser útil en el proceso de análisis de la prueba para reducir la probabilidad de omitir condiciones

de prueba importantes y definir condiciones de prueba más precisas y exactas.

En algunos casos, el análisis de la prueba produce condiciones de prueba que deben ser utilizadas como objetivos de prueba en los contratos de prueba. Los contratos de prueba son productos de trabajo característicos en algunos tipos de prueba basadas en la experiencia. Cuando estos objetivos de prueba son trazables a la base de prueba, se puede medir la cobertura alcanzada durante dicha prueba basada en la experiencia.

La identificación de defectos durante el análisis de la prueba es un beneficio potencial importante, especialmente cuando no se utiliza ningún otro proceso de revisión y/o cuando el proceso de prueba está estrechamente vinculado al mismo proceso de revisión. Estas actividades de análisis de prueba no sólo verifican si los requisitos son consistentes, están debidamente expresados y son completos, sino que también validan si los requisitos satisfacen adecuadamente las necesidades de los clientes, usuarios y otros implicados. Por ejemplo, técnicas como el desarrollo dirigido por el comportamiento (BDD por sus siglas en inglés) y el desarrollo dirigido por prueba de aceptación (ATDD por sus siglas en inglés), que involucran la generación de condiciones de prueba y casos de prueba a partir de historias de usuarios y criterios de aceptación antes de la codificación, también verifican, validan y detectan defectos en las historias de usuarios y en los criterios de aceptación.

#### **D. Diseño de la Prueba**

Durante el diseño de la prueba, las condiciones de prueba se transforman en casos de prueba de alto nivel, conjuntos de casos de prueba de alto nivel y otros productos de prueba. Es decir, el análisis de la prueba responde a la pregunta "qué probar" y el diseño de la prueba responde a la pregunta "cómo probar".

El diseño de la prueba incluye las siguientes actividades principales:

- Diseñar y priorizar casos de prueba y conjuntos de casos de prueba.
- Identificar los datos de prueba necesarios para apoyar las condiciones de prueba y los casos de prueba.
- Diseñar el entorno de prueba e identificar la infraestructura y las herramientas necesarias.
- Capturar la trazabilidad bidireccional entre la base de prueba, las condiciones de prueba, los casos de prueba y los procedimientos de prueba

El desarrollo, a partir de condiciones de prueba, de casos de prueba y de conjuntos de casos de prueba durante el diseño de la prueba implica, a menudo, el uso de técnicas de prueba.

Al igual que con el análisis de la prueba, el diseño de la prueba también puede dar lugar a la identificación de tipos similares de defectos en la base de prueba. Al igual que con el análisis de la prueba, la identificación de defectos durante el diseño de la prueba es un beneficio potencial importante.

#### **E. Implementación de la Prueba**

Durante la implementación de la prueba, se crean y/o se completan los productos de prueba necesarios para la ejecución de la prueba, incluyendo la secuenciación de los casos de prueba en procedimientos de prueba. Por lo tanto, el diseño de la prueba responde a la pregunta "cómo probar", mientras que la implementación de la prueba responde a la pregunta "¿está todo preparado para realizar la prueba?"

La implementación de la prueba incluye las siguientes actividades principales:

- Desarrollar y priorizar procedimientos de prueba y, eventualmente, crear guiones de prueba automatizados.
- Crear juegos de prueba a partir de los procedimientos de prueba y (si los hubiera) guiones de prueba automatizados.

- Organizar los juegos de prueba dentro de un calendario de ejecución de la prueba de forma que se obtenga una ejecución eficiente de los mismos.
- Construir el entorno de prueba (incluyendo, posiblemente, arneses de prueba, virtualización de servicios, simuladores y otros elementos de infraestructura) y verificar que se haya configurado correctamente todo lo necesario.
- Preparar los datos de prueba y asegurarse de que estén correctamente cargados en el entorno de prueba.
- Verificar y actualizar la trazabilidad bidireccional entre la base de prueba, las condiciones de prueba, los casos de prueba, los procedimientos de prueba y los juegos de prueba.

A menudo se combinan las tareas de diseño y ejecución de la prueba.

Es posible que en la prueba exploratoria y otros tipos de prueba basada en la experiencia se desarrolle el diseño y la implementación de la prueba, que pueden ser documentados como parte de la ejecución de la prueba. La prueba exploratoria puede basarse en contratos de prueba (producidos como parte del análisis de la prueba), y la prueba exploratoria se ejecuta inmediatamente, a medida que se diseña e implementa.

### **F. Ejecución de la Prueba**

Durante la ejecución de la prueba, los juegos de prueba se ejecutan de acuerdo al calendario de ejecución de la prueba.

La ejecución de la prueba incluye las siguientes actividades principales:

- Registrar los identificadores y las versiones de los elementos u objetos de prueba, las herramientas de prueba y los productos de prueba.
- Ejecutar pruebas de forma manual o utilizando herramientas de ejecución de pruebas.
- Comparar los resultados reales con los resultados esperados.
- Analizar las anomalías para establecer sus causas probables (por ejemplo, pueden ocurrir fallos debido a defectos en el código, pero también pueden ocurrir falsos positivos [ver sección 1.2.3]).
- Informar sobre los defectos en función de los fallos observados (véase el apartado 5.6).
- Registrar el resultado de la ejecución de la prueba (por ejemplo, pasada, fallada, bloqueada).
- Repetir las actividades de prueba ya sea como resultado de una acción tomada para una anomalía, o como parte de la prueba planificada (por ejemplo, la ejecución de una prueba corregida, una prueba de confirmación y/o una prueba de regresión).
- Verificación y actualización de la trazabilidad bidireccional entre la base de prueba, las condiciones de prueba, los casos de prueba, los procedimientos de prueba y los resultados de la prueba.

### **G. Compleción de la Prueba**

Las actividades de completación de la prueba recopilan datos de las actividades de prueba completadas para consolidar la experiencia, los productos de prueba y cualquier otra información relevante. Las actividades de completación de la prueba ocurren en hitos del proyecto tales como cuando un sistema software es liberado, un proyecto de prueba es completado (o cancelado), cuando finaliza una iteración de un proyecto Ágil (por ejemplo, como parte de una reunión retrospectiva), cuando se completa un nivel de prueba, o cuando se completa la liberación de un mantenimiento.

La compleción de la prueba incluye las siguientes actividades principales:

- Comprobar que todos los informes de defecto estén cerrados, registrando solicitudes de cambio o elementos de la cartera del producto<sup>14</sup> para cualquier defecto que quede sin resolver al final de la ejecución de la prueba. Crear un informe resumen de prueba que se comunicará a los implicados.
- Finalizar, archivar y almacenar (según corresponda) el entorno de prueba, los datos de prueba, la infraestructura de prueba y otros productos de prueba para su posterior reutilización.
- Traspaso de los productos de prueba a los equipos de mantenimiento, a otros equipos del proyecto/u otros implicados que podrían beneficiarse de su uso.
- Analizar las lecciones aprendidas de las actividades de prueba completadas para determinar los cambios necesarios para iteraciones, lanzamientos y proyectos futuros.
- Utilizar la información recopilada para mejorar la madurez del proceso de prueba.

### 4.3 Productos de Trabajo de la prueba

Los productos de trabajo de la prueba se crean como parte del proceso de prueba. Así como hay una variación significativa en la forma en que las organizaciones implementan el proceso de prueba, también hay una variación significativa en los tipos de productos de trabajo creados durante ese proceso, en las formas en que esos productos de trabajo están organizados y gestionados, y en los nombres que se utilizan para esos productos de trabajo.

Este programa de estudio se adhiere al proceso de prueba descrito anteriormente y a los productos de trabajo nombrados en este programa de estudio y en el Glosario del ISTQB. La norma ISO (ISO/IEC/IEEE 29119-3) también puede servir como guía para los productos de trabajo de la prueba.

Muchos de los productos de trabajo de la prueba mostrados en esta sección se pueden capturar y gestionar utilizando herramientas de gestión de pruebas y herramientas de gestión de defectos, los cuales serán revisados en el capítulo 6.

- **Productos de Trabajo de la Planificación de la Prueba**

Los productos de trabajo de la planificación de la prueba, por lo general, incluyen uno o más planes de prueba. El **plan de prueba** incluye información sobre la base de prueba, con la que se relacionarán los demás productos de trabajo de la prueba mediante información de, así como los criterios de salida (o definición de hecho) que se utilizarán durante la monitorización y el control de la prueba.

- **Productos de Trabajo de la Monitorización y Control de la Prueba**

Los productos de trabajo de la monitorización y control de la prueba, normalmente, incluyen varios tipos de informes de prueba, **incluyendo informes del avance de la prueba** (producidos de forma continua y/o regular) e **informes resumen de prueba** (producidos en varios hitos de compleción). Todos los informes de prueba deben proporcionar detalles relevantes para la audiencia sobre el avance de la prueba a la fecha del informe, incluyendo un resumen de los resultados de la ejecución de la prueba una vez que estén disponibles.

Los productos de trabajo de monitorización y control de la prueba también deben abordar las inquietudes de la gestión del proyecto, tales como la compleción de tareas, la asignación y el uso de recursos, y el esfuerzo.

- **Productos de Trabajo del Análisis de la Prueba**

**Los productos de trabajo del análisis de la prueba incluyen condiciones de prueba definidas**

y priorizadas, cada una de las cuales es, idealmente, trazable bidireccionalmente a los elementos específicos de la base de prueba que cubre. Para la prueba exploratoria, el análisis de la prueba puede implicar la creación de contratos de prueba. El análisis de la prueba también puede dar lugar al descubrimiento y notificación de defectos en la base de prueba.

- **Productos de Trabajo del Diseño de la Prueba**

**El diseño de la prueba resulta en casos de prueba y conjuntos de casos de prueba para practicar las condiciones de prueba definidas en el análisis de la prueba.** A menudo, es una buena práctica diseñar casos de prueba de alto nivel, sin valores concretos para los datos de entrada y los resultados esperados. Estos casos de prueba de alto nivel son reutilizables a lo largo de múltiples ciclos de prueba con diferentes datos concretos, sin dejar de documentar adecuadamente el alcance del caso de prueba. Lo ideal es que cada caso de prueba se pueda trazar bidireccionalmente hasta la(s) condición(es) de prueba que cubre.

El diseño de la prueba también da lugar al diseño y/o la identificación de los datos de prueba necesarios, el diseño del entorno de prueba y la identificación de la infraestructura y las herramientas, aunque la medida en que se documentan estos resultados varía significativamente.

Las condiciones de prueba definidas en el análisis de la prueba pueden refinarse aún más en el diseño de la prueba.

- **Productos de Trabajo de la Implementación de la Prueba**

Los productos de trabajo de la implementación de la prueba incluyen:

- Procedimientos de prueba y la secuenciación de dichos procedimientos de prueba.
- Juegos de prueba.
- Un calendario de ejecución de pruebas.

Idealmente, una vez finalizada la implementación de la prueba, el logro de los criterios de cobertura establecidos en el plan de prueba puede demostrarse mediante la trazabilidad bidireccional entre los procedimientos de prueba y los elementos específicos de la base de prueba, a través de los casos de prueba y las condiciones de prueba.

En algunos casos, la implementación de la prueba implica la creación de productos de trabajo que utilizan o son utilizados por herramientas, como la virtualización de servicios y los guiones de prueba automatizados.

La implementación de la prueba también puede resultar en la creación y verificación de los datos de la prueba y el entorno de prueba. Los resultados de la verificación de la completitud de la documentación de los datos y/o del entorno pueden variar significativamente.

Los datos de prueba sirven para asignar valores concretos a las entradas y a los resultados esperados de los casos de prueba. Estos valores concretos, junto con instrucciones explícitas sobre el uso de los valores concretos, convierten los casos de prueba de alto nivel en casos de prueba ejecutables de bajo nivel. El mismo caso de prueba de alto nivel puede utilizar diferentes datos de prueba cuando se ejecuta en diferentes versiones del objeto de prueba. Los resultados esperados concretos que se asocian con los datos de prueba concretos se identifican mediante el uso de un oráculo de prueba.

En la prueba exploratoria, se pueden crear algunos productos de trabajo de diseño e implementación de la prueba durante la ejecución de la misma, aunque la medida en que se documenta la prueba exploratoria (y su trazabilidad a elementos específicos de la base de prueba) puede variar significativamente.

Las condiciones de prueba definidas en el análisis de la prueba pueden refinarse aún más en la implementación de la prueba.

- **Productos de Trabajo de la Ejecución de la Prueba**

Los productos de trabajo de la ejecución de la prueba incluyen:

- Documentación sobre el estado de casos de prueba individuales o procedimientos de prueba (por ejemplo, listo para ejecutar, pasado, fallado, bloqueado, omitido de forma deliberada, etc.).
- Informes de defecto
- Documentación sobre el o los elemento(s), objeto(s), herramienta(s) y productos de prueba<sup>16</sup> involucrados en la prueba.

Idealmente, una vez finalizada la ejecución de la prueba, se puede determinar e informar el estado de cada elemento de la base de prueba a través de la trazabilidad bidireccional con el procedimiento o procedimientos de prueba asociados. Por ejemplo, se puede decir qué requisitos han pasado todas las pruebas planificadas, qué requisitos han fallado y/o tienen defectos asociados a ellos, y qué requisitos tienen pruebas planificadas que se encuentran a la espera de ser ejecutadas. Esto permite verificar que se hayan cumplido los criterios de cobertura, y permite informar los resultados de la prueba en términos comprensibles para todos los implicados.

- **Productos de Trabajo de la Compleción de la Prueba**

Los productos de trabajo de la compleción de la prueba incluyen informes de resumen de la prueba, elementos de acción para la mejora de proyectos o iteraciones subsiguientes (por ejemplo, siguiendo una retrospectiva ágil del proyecto), solicitudes de cambio o elementos de la cartera del producto, y productos de prueba finalizados.

#### 4.4 Trazabilidad entre la Base de Prueba y los Productos de Trabajo de la Prueba

Como se mencionó en la sección 4.3, los productos de trabajo de la prueba y los nombres de esos productos de trabajo varían de forma significativa. Independientemente de estas variaciones, para implementar una monitorización y control efectivos de la prueba, es importante establecer y mantener la trazabilidad a lo largo del proceso de prueba entre cada elemento de la base de prueba y los diversos productos de trabajo de la prueba asociados con ese elemento, tal como se describió anteriormente. Además de la evaluación de la cobertura de las pruebas, una buena trazabilidad permite:

- Analizar el impacto de cambios.
- Hacer que la prueba pueda ser auditada.
- Cumplimiento de los criterios de gobernanza de TI.
- Mejorar la comprensión de los informes de avance de la prueba y de los informes resumen de prueba para incluir el estado de los elementos de la base de prueba (por ejemplo, los requisitos que pasaron sus pruebas, los requisitos que fallaron sus pruebas, y los requisitos que tienen pruebas pendientes).
- Relacionar los aspectos técnicos de la prueba con los implicados en términos que éstos puedan entender.
- Aportar información para evaluar la calidad de los productos, la capacidad de los procesos y el avance de los proyectos en relación con los objetivos de negocio.

Algunas herramientas de gestión de la prueba proporcionan modelos de productos de trabajo de la prueba que coinciden con parte o la totalidad de los productos de trabajo de la prueba descritos en esta sección. Algunas organizaciones construyen sus propios sistemas de gestión para organizar los productos de trabajo y proporcionar la trazabilidad de la información que necesitan.

## 5. La psicología del proceso de prueba

El desarrollo de software, incluyendo la prueba de software, involucra a seres humanos. Por lo tanto, la psicología humana tiene efectos importantes en la prueba de software.

### 5.1 Psicología Humana y el Proceso de Prueba

La identificación de defectos durante una prueba estática, como una revisión de requisitos o una sesión de refinamiento de historias de usuario, o la identificación de fallos durante la ejecución de una prueba dinámica, puede ser percibida como una crítica al producto y a su autor. **Un elemento de la psicología humana llamado sesgo de confirmación puede dificultar la aceptación de información que esté en desacuerdo con las creencias actuales.** Por ejemplo, dado que los desarrolladores esperan que su código sea correcto, tienen un sesgo de confirmación que hace difícil aceptar que el código sea incorrecto. Además del sesgo de confirmación, otros sesgos cognitivos pueden dificultar que las personas entiendan o acepten la información producida por la prueba. Asimismo, es un rasgo humano común culpar al portador de malas noticias, y la información producida por la prueba, a menudo, contiene malas noticias.

Como resultado de estos factores psicológicos, algunas personas pueden percibir al proceso de prueba como una actividad destructiva, a pesar de que contribuye en gran medida al avance del proyecto y a la calidad del producto. Para tratar de reducir estas percepciones, la información sobre defectos y fallos debe ser comunicada de manera constructiva. De esta manera, se pueden reducir las tensiones entre los probadores y los analistas, los propietarios de producto, los diseñadores y los desarrolladores. Esto se aplica tanto a las pruebas estáticas como a las dinámicas.

Los probadores y jefes de prueba necesitan tener buenas competencias interpersonales para poder comunicarse eficazmente sobre defectos, fallos, resultados de la prueba, avance de la prueba y riesgos, y para construir relaciones positivas con las personas que forman parte de su entorno de trabajo. Las formas de comunicarse de manera adecuada incluyen los siguientes ejemplos:

- Comenzar con colaboración en lugar de batallas. Recordar a todos que el objetivo común es lograr sistemas de mejor calidad.
- Enfatizar los beneficios de la prueba. Por ejemplo, para los autores, la información sobre los defectos puede ayudarles a mejorar sus productos de trabajo y sus competencias. Para la organización, los defectos detectados y corregidos durante la prueba ahorrarán tiempo y dinero y reducirán el riesgo general para la calidad de producto.
- Comunicar los resultados de las pruebas y otros hallazgos de manera neutral y centrada en los hechos sin criticar a la persona que creó el elemento defectuoso. Redactar informes de defecto objetivos y fundamentados en hechos y revisar los hallazgos.
- Tratar de entender cómo se siente la otra persona y las razones por las que puede reaccionar negativamente a la información.
- Confirmar que el interlocutor ha entendido lo que se ha dicho y viceversa.

Los objetivos característicos de la prueba se discutieron anteriormente (ver sección 1.1). Definir claramente el conjunto correcto de objetivos de prueba tiene importantes implicaciones psicológicas. La mayoría de las personas tienden a alinear sus planes y comportamientos con los objetivos establecidos por el equipo, la dirección y otros implicados. También es importante que los probadores se adhieran a estos objetivos con un mínimo sesgo personal.

### 5.2 Formas de Pensar del Probador y del Desarrollador



**Los desarrolladores y los probadores, a menudo, piensan de forma diferente.** El objetivo principal del desarrollo es diseñar y construir un producto. Como se expuso anteriormente, los objetivos de la prueba incluyen la verificación y validación del producto, la detección de defectos antes de liberarlo, y así sucesivamente. Se trata de diferentes conjuntos de objetivos que requieren diferentes mentalidades, o formas de pensar. La combinación de estas mentalidades ayuda a lograr un mayor nivel de calidad del producto.

Una mentalidad refleja las suposiciones de un individuo y los métodos preferidos para la toma de decisiones y la resolución de problemas. La mentalidad de un probador debe incluir curiosidad, pesimismo profesional, sentido crítico, atención al detalle y motivación para una comunicación y relaciones buenas y positivas. La mentalidad de un probador tiende a ampliarse y madurar a medida que adquiere experiencia.

La mentalidad de un desarrollador puede incluir algunos de los elementos de la mentalidad de un probador, pero los desarrolladores de éxito, a menudo, están más interesados en el diseño y la construcción de soluciones que en la contemplación de lo que podría estar mal en esas soluciones. Además, el sesgo de confirmación hace difícil encontrar errores en su propio trabajo.

Con la mentalidad correcta, los desarrolladores pueden probar su propio código. Los diferentes modelos de ciclo de vida de desarrollo de software, a menudo, tienen diferentes maneras de organizar a los probadores y las actividades de prueba. El hecho de que algunas de las actividades de prueba sean realizadas por probadores independientes aumenta la eficacia de la detección de defectos, lo cual es particularmente importante para sistemas grandes, complejos o de seguridad crítica. **Los probadores independientes aportan una perspectiva diferente a la de los autores de los productos de trabajo (es decir, analistas de negocio, propietarios de producto, diseñadores y programadores), ya que tienen diferentes sesgos cognitivos de los autores.**