

Resumen documento ISTQB

El Probador Certificado Nivel Básico en Pruebas de Software

La cualificación de Nivel Básico está dirigida a cualquier persona involucrada en las pruebas de software, esto incluye todos los roles dentro de las pruebas de software.

Carrera Profesional para Probadores

El esquema ISTQB® brinda apoyo a los profesionales de pruebas en todas las etapas de sus carreras, ofreciendo amplitud y profundidad de conocimiento. Las personas que realizaron el nivel básico ISTQB pueden realizar los niveles avanzado y experto.

Resultados de Negocio

Un probador certificado de nivel básico puede...

1. Comprender qué son las pruebas y por qué son beneficiosas
2. Comprender los conceptos fundamentales de las pruebas de software
3. Identificar el enfoque de la prueba y las actividades a implementar dependiendo del contexto de la prueba
4. Evaluar y mejorar la calidad de la documentación
5. Aumentar la eficacia y eficiencia de las pruebas
6. Alinear el proceso de prueba con el ciclo de vida de desarrollo de software
7. Comprender los principios de gestión de pruebas
8. Escribir y comunicar informes de defectos claros y comprensibles
9. Comprender los factores que influyen en las prioridades y esfuerzos relacionados con las pruebas
10. Trabajar como parte de un equipo multifuncional
11. Conocer los riesgos y beneficios relacionados con la automatización de pruebas
12. Identificar las habilidades esenciales requeridas para las pruebas
13. Comprender el impacto del riesgo en las pruebas
14. Informar eficazmente sobre el progreso y la calidad de la prueba

Objetivos de Aprendizaje y Nivel Cognitivo de Conocimiento Evaluables

Los niveles específicos de los objetivos de aprendizaje se muestran al comienzo de cada capítulo y se clasifican de la siguiente manera:

- K1: Recordar
- K2: Comprender
- K3: Aplicar

Fundamentos de la Prueba

¿Qué es probar?

Probar es el proceso de evaluar la calidad y descubrir defectos en un producto, para minimizar los posibles fallos a la hora del funcionamiento

Objetivos de la Prueba

- Los objetivos característicos/típicos de la prueba son:
- Evaluar productos de trabajo como requisitos, historias de usuario, diseños y código
- Desencadenar fallas y encontrar defectos
- Garantizar la cobertura requerida de un objeto de prueba
- Reducir el nivel de riesgo de una calidad de software inadecuada
- Verificar si se han cumplido los requisitos especificados
- Verificar que un objeto de prueba cumple con los requisitos contractuales, legales y reglamentarios
- Proporcionar información a las partes interesadas para permitirles tomar decisiones informadas
- Generar confianza en la calidad del objeto de prueba
- Validar si el objeto de prueba está completo y funciona según lo esperado por las partes interesadas

Pruebas y Depuración

Las pruebas y la depuración son actividades separadas. Las pruebas pueden desencadenar fallas causadas por defectos en el software o pueden encontrar directamente defectos en el objeto de prueba, cuando la prueba dinámica desencadena una falla, la depuración se encarga de encontrar la causa, analizar y eliminarla

Proceso de depuración típico:

- Reproducción de una falla
- Diagnóstico (encontrar la causa raíz)
- Corregir la causa

¿Por Qué es Necesario Probar?

Porque esto contribuye al éxito del equipo de trabajo y a la empresa como tal, detectar defectos, analizarlos y solucionarlos es un aporte optimo al proyecto en cuestión

Pruebas y Aseguramiento de Calidad (QA)

Si bien estos términos están relacionados, no tiene el mismo significado ya que las pruebas son una forma de control de calidad y QA es un enfoque preventivo orientado a procesos que se centra en la implementación y mejora de procesos.

Errores, Defectos, Fallas y Causas Raíz

Estos conceptos se relacionan directamente ya que en la mayoría de las ocasiones uno conlleva a otro y sino se soluciona o no se encuentra la causa raíz del error o el defecto, este llevara a una falla en el software, también se pueden presentar fallas por condiciones ambientales.

Principios de la Prueba

1. **Las pruebas muestran la presencia, no la ausencia de defectos:** Las pruebas reducen la probabilidad de que los defectos permanezcan sin descubrir en el objeto de prueba
2. **Las pruebas exhaustivas son imposibles:** En lugar de intentar realizar pruebas exhaustivas, se deben utilizar técnicas de prueba
3. **Las pruebas tempranas ahorran tiempo y dinero:** Los defectos que se eliminan al principio del proceso no causarán defectos posteriores en los productos de trabajo derivados
4. **Los defectos se agrupan:** Los defectos dentro del sistemas pueden estar agrupados la mayormente en uno o varios componentes que pueden causar la falla.
5. **Las pruebas se desgastan:** la repetición de las mismas pruebas a lo largo del tiempo se vuelve ineficaces para poder detectar defectos, se hace necesario modificar los datos de las pruebas o las pruebas existentes
6. **Las pruebas dependen del contexto:** No existe un enfoque único universalmente aplicable a las pruebas. Las pruebas se realizan de manera diferente en diferentes contextos (Kaner 2011).
7. **. Falacia de la ausencia de defectos:** Es una falacia esperar que la verificación del software garantice el éxito de un sistema.

Actividades y Tareas de la Prueba

Un proceso de prueba generalmente consiste en los principales grupos de actividades que se describen a continuación.

- La planificación
- El monitoreo y el control de prueba.
- El análisis de prueba
- El diseño de la prueba
- La implementación de prueba
- La ejecución de pruebas
- Las actividades de la finalización de pruebas

El Proceso de la Prueba en Contexto

la forma en que se lleva a cabo la prueba dependerá de una serie de factores contextuales que incluyen:

- Partes interesadas
- Miembros del equipo
- Dominio del negocio
- Factores técnicos
- Limitaciones del proyecto
- Factores organizativos
- Ciclo de vida del desarrollo de software
- Herramientas

Trazabilidad entre la Base de Prueba y el Testware

una buena trazabilidad permite determinar el impacto de los cambios, facilita las auditorías de prueba, también hace que el progreso de la prueba y los informes de finalización sean más fáciles de entender al incluir el estado de los elementos básicos de la prueba.

Habilidades Esenciales y Buenas Prácticas en las Pruebas

Son esas características esenciales que debe tener un probador que lo va a ayudar a desarrollar su labor de la mejor manera.

Habilidades Genéricas Requeridas para las Pruebas

- Conocimiento en pruebas
- Rigurosidad, cuidado, curiosidad, atención a los detalles, ser metódico
- Buenas habilidades de comunicación, escucha activa, ser un jugador de equipo
- Pensamiento analítico, pensamiento crítico, creatividad
- Conocimientos técnicos
- Conocimiento del dominio

Pruebas en el Contexto de un Ciclo de Vida de Desarrollo de Software

Un modelo SDLC define cómo las diferentes fases de desarrollo y los tipos de actividades realizadas dentro de este proceso se relacionan entre sí, tanto lógica como cronológicamente.

Las pruebas deben adaptarse al SDLC.

Pruebas como una Guía para el Desarrollo de Software

TDD, ATDD y BDD son enfoques de desarrollo similares, donde las pruebas se definen como un medio para dirigir el desarrollo.

Desarrollo Guiado por Pruebas (TDD)

Dirige la codificación a través de casos de prueba (en lugar de un extenso diseño de software) (Beck 2003). Las pruebas se escriben primero, luego el código se escribe para satisfacer las pruebas, y luego las pruebas y el código se refactorizan.

Desarrollo Guiado por Pruebas de Aceptación (ATDD)

Deriva las pruebas de los criterios de aceptación como parte del proceso de diseño del sistema (Gärtner 2011). Las pruebas se escriben antes de que se desarrolle parte de la aplicación, para satisfacer las pruebas.

Desarrollo Guiado por el Comportamiento (BDD):

Expresa el comportamiento deseado de una aplicación de manera simple

DevOps y Pruebas

es un enfoque organizacional que tiene como objetivo crear la coordinación entre desarrollo y operaciones para lograr unos objetivos.

Beneficios:

- Retroalimentación rápida sobre la calidad del código y si los cambios afectan negativamente al código existente
- La Integración Continua (CI) promueve un enfoque de desplazamiento hacia la izquierda en las pruebas (consulte la sección 2.1.5) alentando a los desarrolladores a presentar código de alta calidad acompañado de pruebas de componentes y análisis estático

- Promueve procesos automatizados como Integración Continua y Entrega Continua (CI/CD) que facilitan el establecimiento de entornos de prueba estables
- Aumenta la visión sobre las características de calidad no funcionales (por ejemplo, rendimiento, fiabilidad)
- La automatización a través de un canal (pipeline) de entrega reduce la necesidad de pruebas manuales repetitivas
- El riesgo en la regresión se minimiza debido a la escala y el rango de las pruebas de regresión automatizadas

Enfoque de Desplazamiento hacia la Izquierda

El desplazamiento hacia la izquierda normalmente sugiere que las pruebas deben hacerse antes, pero no significa que las pruebas posteriores en el SDLC deban descuidarse

Niveles de Prueba y Tipos de Prueba

Los niveles de prueba son grupos de actividades de prueba que se organizan y administran juntas.

- Las pruebas de componentes
- Las pruebas de integración de componentes
- Las pruebas del sistema
- Las pruebas de integración de sistemas
- Las pruebas de aceptación

Tipos de Prueba

Las pruebas funcionales: Evalúan que las funciones de un componente se realicen de manera efectiva

Las pruebas no funcionales: Estas evalúan el comportamiento del sistema

La prueba de caja negra: El objetivo principal de las pruebas de caja negra es comprobar el comportamiento del sistema contra sus especificaciones.

Las pruebas de caja blanca: El objetivo principal de las pruebas de caja blanca es cubrir la estructura subyacente mediante las pruebas al nivel aceptable.

Pruebas de Confirmación y Pruebas de Regresión

Las pruebas de confirmación: Confirman que un defecto se ha corregido exitosamente

Las pruebas de regresión: Confirman que no se hayan producido consecuencias por algún cambio

Pruebas de mantenimiento:

El alcance de las pruebas de mantenimiento generalmente depende de:

- El grado de riesgo del cambio
- El tamaño del sistema existente
- El tamaño del cambio

Pruebas estáticas

A diferencia de las pruebas dinámicas, en las pruebas estáticas no es necesario ejecutar el software bajo prueba. Las pruebas estáticas proporcionan la capacidad de evaluar la calidad y generar confianza en los productos de trabajo.

Diferencias entre Pruebas Estáticas y Pruebas Dinámicas

Algunos tipos de defectos solo pueden encontrarse mediante prueba estática o dinámica, las pruebas estáticas identifican el fallo directamente mientras que las dinámicas producen un error para posteriormente identificar donde esta el fallo.

Análisis y Diseño de Pruebas

Descripción General de las Técnicas de Prueba

Las técnicas de prueba ayudan al probador a definir las condiciones de la prueba, identificar los ítems de cobertura e identificar los datos de la prueba durante el análisis y el diseño de la prueba.

Las técnicas de prueba de caja negra: se basan en un análisis del comportamiento especificado del objeto de prueba sin referencia a su estructura interna.

técnicas de prueba de caja blanca: se basan en un análisis de la estructura interna y el procesamiento del objeto de prueba.

técnicas de prueba basadas en la experiencia: utilizan efectivamente el conocimiento y la experiencia de los probadores para el diseño y la implementación de casos de prueba.

Técnicas de Caja Negra

Las técnicas de prueba de caja negra comúnmente utilizadas discutidas en las siguientes secciones son:

- Partición Equivalente
- Análisis de Valores Límite
- Pruebas de Tabla de Decisión
- Pruebas de Transición de Estado

Técnicas de Caja Blanca

Debido a su popularidad y simplicidad, esta sección se centra en dos técnicas de prueba de caja blanca relacionadas con el código:

- Pruebas de sentencias
- Pruebas de ramas

El Valor de las Pruebas de Caja Blanca

Una fortaleza fundamental que comparten todas las técnicas de caja blanca es que toda la implementación del software se tiene en cuenta durante las pruebas, lo que facilita la detección de defectos incluso cuando la especificación del software es vaga, obsoleta o incompleta.

Técnicas Basadas en la Experiencia

Las técnicas de prueba basadas en la experiencia comúnmente utilizadas discutidas en las siguientes secciones son:

- Predicción de errores
- Pruebas exploratorias
- Pruebas basadas en listas de comprobación

Gestión de las Actividades de Prueba

Propósito y Contenido de un Plan de Prueba

Un plan de prueba describe los objetivos, recursos y procesos para un proyecto de prueba. Un plan de prueba:

- Documenta los medios y el cronograma para lograr los objetivos de la prueba

- Ayuda a garantizar que las actividades de prueba realizadas cumplan con los criterios establecidos

Criterios de Entrada y Criterios de Salida

Los criterios de entrada definen las condiciones previas para emprender una actividad determinada, Los criterios de salida definen lo que se debe lograr para declarar una actividad completada.

Priorización de Casos de Prueba

Las estrategias de priorización de casos de prueba más utilizadas son las siguientes:

Priorización basada en el riesgo: Los casos de prueba que cubren los riesgos más importantes se ejecutan primero.

Priorización basada en la cobertura: Los casos de prueba que logran la cobertura más alta se ejecutan primero.

Priorización basada en requisitos: Las prioridades de los requisitos son definidas por las partes interesadas.

Pirámide de Prueba

La pirámide de prueba es un modelo que muestra que diferentes pruebas pueden tener una **granularidad** diferente.

Cuadrantes de Prueba

En este modelo, las pruebas pueden estar orientadas al negocio o a la tecnología.

Gestión de Riesgos

La gestión de riesgos permite a las organizaciones aumentar la probabilidad de alcanzar objetivos, mejorar la calidad de sus productos y aumentar la confianza de las partes interesadas. Las principales actividades son:

- Análisis de riesgos (identificación y evaluación de riesgos)
- Control de riesgos (mitigación de riesgos y monitoreo de riesgos)

Control de Riesgos de Producto

El control del riesgo del producto comprende todas las medidas que se toman en respuesta a los riesgos del producto identificados y evaluados.

Monitoreo de Pruebas, Control de Pruebas y Finalización de Pruebas

El monitoreo de pruebas se refiere a la recopilación de información sobre las pruebas. El control de pruebas utiliza la información del monitoreo de pruebas para proporcionar, en forma de directivas de control, orientación y las acciones correctivas necesarias para lograr las pruebas más efectivas y eficientes. La finalización de la prueba recopila datos de las actividades de prueba completadas para consolidar la experiencia, el software de prueba y cualquier otra información relevante.

Métricas Utilizadas en las Pruebas

- Métricas de progreso del proyecto
- Métricas de progreso de las pruebas
- Métricas de calidad del producto
- Métricas de defectos
- Métricas de riesgo
- Métricas de cobertura
- Métricas de costos

Herramientas de Prueba

Las herramientas de prueba apoyan y facilitan muchas actividades de prueba.

Beneficios y Riesgos de la Automatización de Pruebas

El simple hecho de adquirir una herramienta no garantiza el éxito.

Los beneficios potenciales del uso de la automatización de pruebas incluyen:

- Tiempo ahorrado al reducir el trabajo manual repetitivo
- Prevención de errores humanos simples a través de una mayor consistencia y repetibilidad
- Evaluación más objetiva
- Acceso más fácil a la información sobre las pruebas para respaldar la gestión de pruebas y los informes de pruebas
- Reducción de los tiempos de ejecución de pruebas
- Más tiempo para que los probadores diseñen pruebas nuevas, más profundas y más efectivas

Los riesgos potenciales de usar la automatización de pruebas incluyen:

- Expectativas poco realistas sobre los beneficios de una herramienta
- Estimaciones inexactas de tiempo, costos, esfuerzo requerido para introducir una herramienta, mantener los scripts de prueba y cambiar el proceso de prueba manual existente.
- Usar una herramienta de prueba cuando la prueba manual es más apropiada
- Confiar demasiado en una herramienta, por ejemplo, ignorar la necesidad del pensamiento crítico humano.
- El uso de un software de código abierto que puede ser abandonado, lo que significa que no hay más actualizaciones disponibles.
- La herramienta de automatización no es compatible con la plataforma de desarrollo.