



Universidad
Tecmilenio®





Software Testing

Riesgos y estimación de pruebas



Semana 4





Las pruebas basadas en riesgos se empiezan a utilizar con mayor frecuencia en empresas de todos los tamaños y que se han adecuado a las metodologías ágiles, no precisamente por el riesgo alto de uso crítico de un sistema, sino por el hecho de que Agile Testing en sí mismo incrementa algunas situaciones de riesgo por la velocidad de implementación.

Por lo que se elige prevenir, en lugar de tener que corregir.

Los métodos de prevención de riesgos son un enfoque pragmático y ampliamente utilizado en la industria del software, ya que centra las actividades de prueba en aquellos escenarios que desencadenan las situaciones más críticas de un proyecto de desarrollo de software.



Taxonomía de RBT

Elementos del
riesgo.

Propiedades de
calidad.

Impulsor del
riesgo.



El sistema debe ser probado para que sea:

Funcional

Fiable

Usable

Eficaz

Seguro

Resiliente



Elementos para la evaluación de riesgos

Factor de riesgo.

Técnica de estimación.

Escala.

Grado de automatización.



Elementos del tercer eje de la estrategia

Planeación

- Se definen los objetivos y técnicas para las pruebas.
- Criterios que determinan que las pruebas se han completado.

Diseño o implementación

- Se determinan los casos de prueba.
- Se asigna la prioridad.
- Se seleccionan los casos de prueba a ejecutar.

Ejecución y evaluación

- Se realizan las mediciones y monitoreo.
- Se hace el reporte de riesgos.
- Se ejecutan pruebas y se realiza la revaloración con un criterio de salida.





¿Cuál es el método que nos permite adelantarnos a cualquier situación adversa que pudiera ocurrir durante el proyecto?

¿Cuáles son los elementos que consideras que garantizan la calidad del producto de software?



Las pruebas orientadas a los riesgos ayudan a prevenir cualquier falla que pueda ocurrir dentro de un proceso de software. Centran sus esfuerzos en la gestión, selección, priorización y uso de recursos de prueba correspondientes al riesgo analizado.

El método RBT es complejo para muchos profesionistas de TI, sin embargo, has aprendido a aplicarlo en cada uno de sus tres ejes o diferenciadores: la configuración conceptual, la forma en la que se determinan los riesgos y la estrategia.



Software Testing

Técnicas de Agile Testing

Semana 4





Para cumplir con los objetivos de Agile Testing, es necesario contar con técnicas diseñadas para ello y que puedan realizar las pruebas de software sin entorpecer el proceso de desarrollo.

Dicho de otro modo, encontrar métodos que se adapten a la continuidad y que se ejecuten de forma simultánea a otros procesos, en particular, al proceso de codificación.

Pruebas de caja negra

Pruebas funcionales

Se ocupan solo de los requerimientos funcionales de un sistema y cubren qué tan bien el sistema ejecuta sus funciones.

Pruebas no funcionales

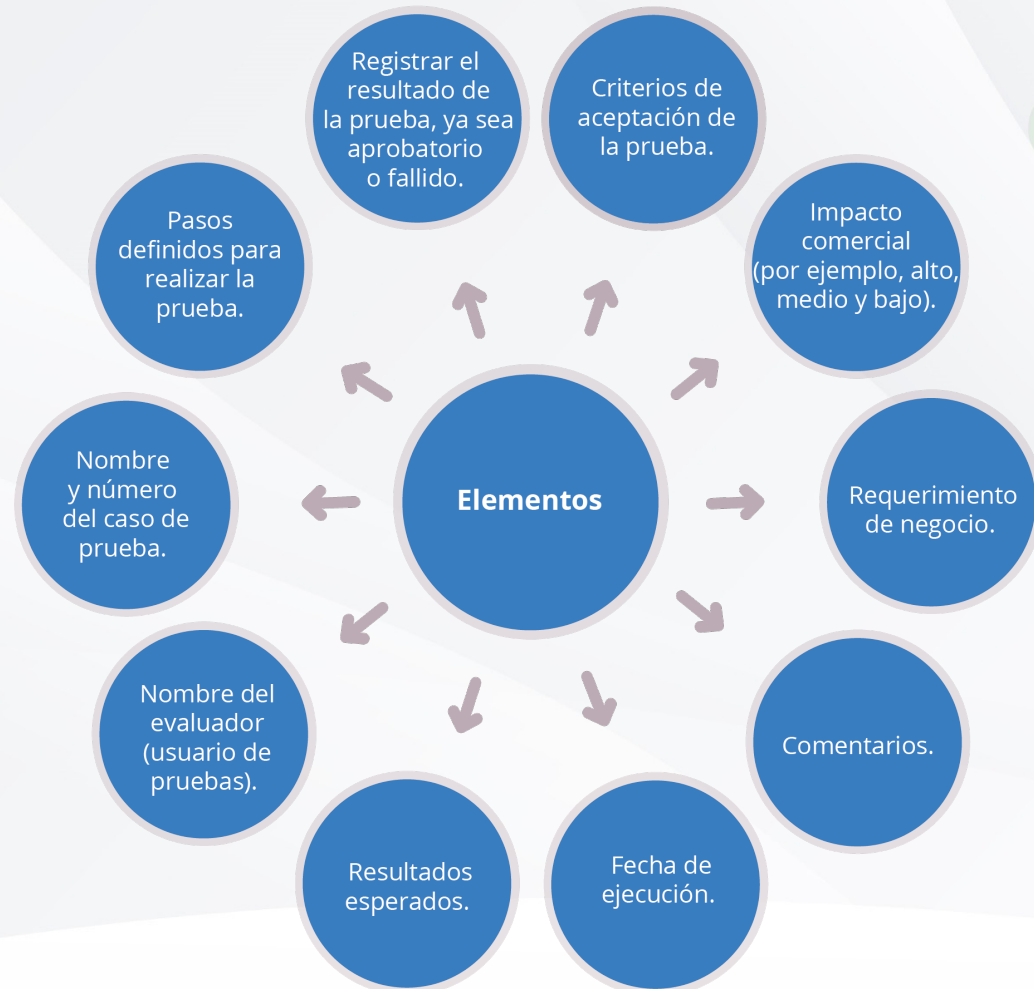
Son pruebas diseñadas para comprobar la parte técnica de un sistema y los elementos de software y diseño detrás de cada función.

Pruebas de regresión

Determinan si la última corrección no descompuso algo que ya se tenía bien.

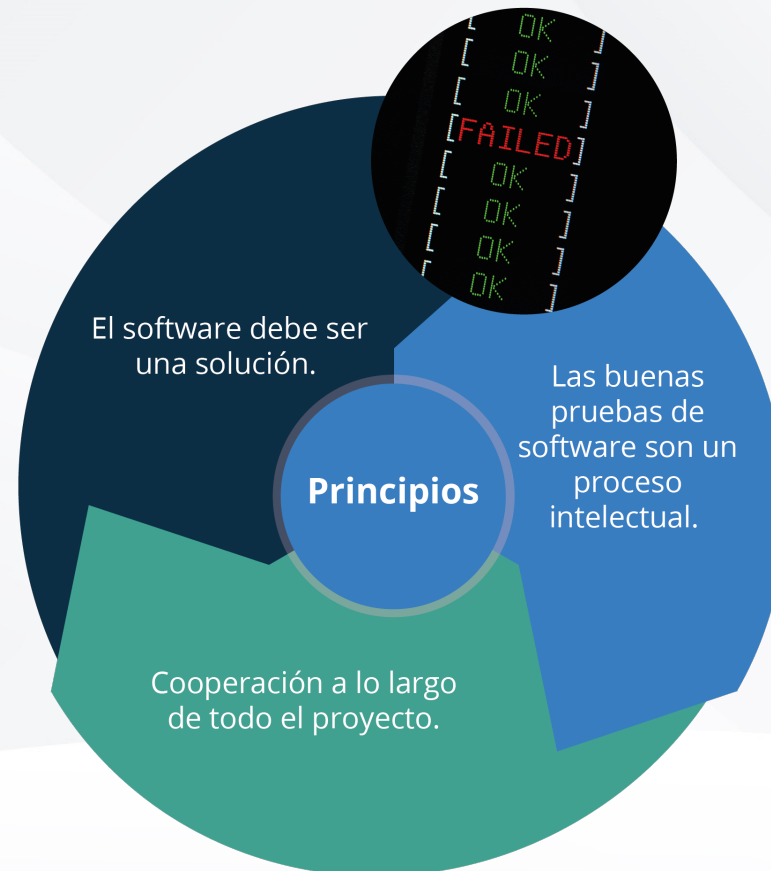


Pruebas de aceptación

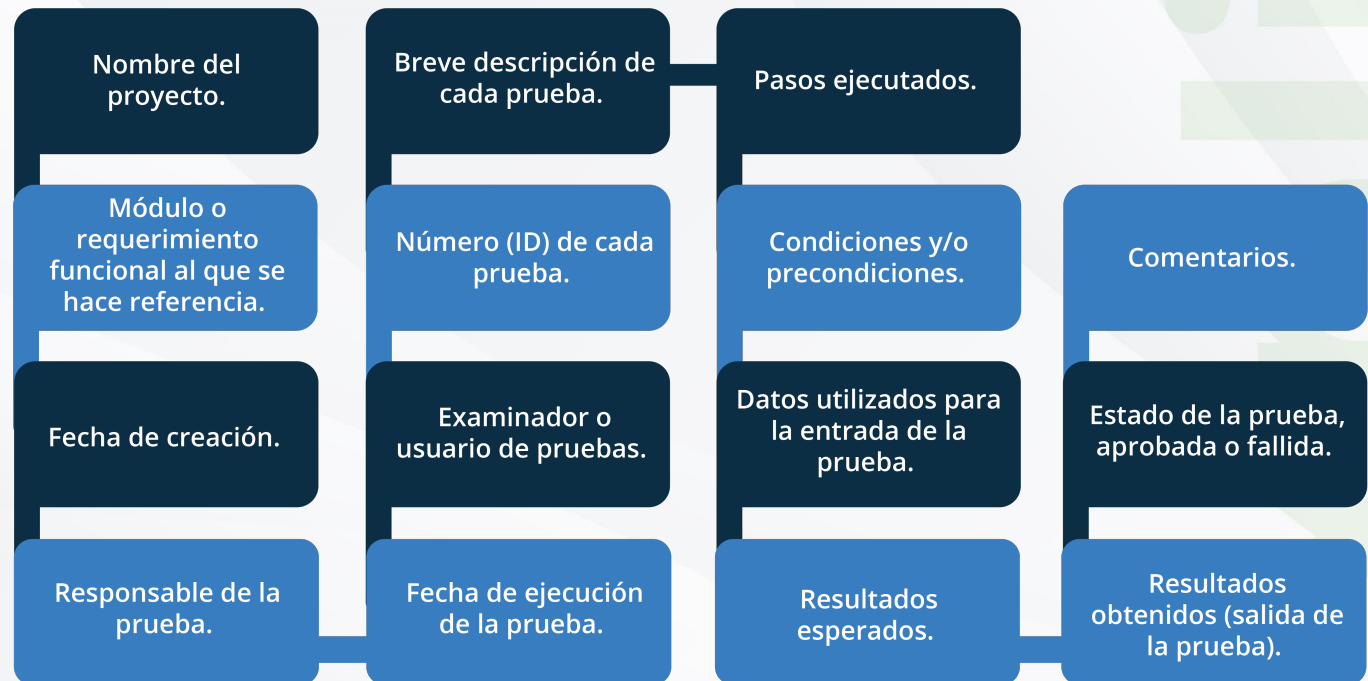


Pruebas exploratorias

Estas pruebas se ejecutan conforme se realiza el código; como dice su nombre, exploran la calidad del código y sus funciones sobre la marcha, por lo que es una técnica compatible con Agile Testing.



Bitácora de resultados





¿Consideras que las tres técnicas para Agile Testing son suficientes o consideras que se pueden diseñar todavía mejores o simplemente combinar las existentes para inventar una mejor?

¿Consideras que la bitácora de resultados para las pruebas de software es suficiente para garantizar la calidad del software o piensas que mantiene un nivel de riesgo alto?



Las técnicas más utilizadas para Agile Testing son *Black Box Testing*, *User Acceptance Testing (UAT)* y *Exploratory Testing*. A pesar de que tienen características muy distintas, todas se adaptan perfectamente a los modelos ágiles de la actualidad.

También se lograron identificar los elementos necesarios para hacer tus casos de prueba y el proceso hacia la técnica para documentarlos.



Software Testing

Herramientas de Agile
Testing

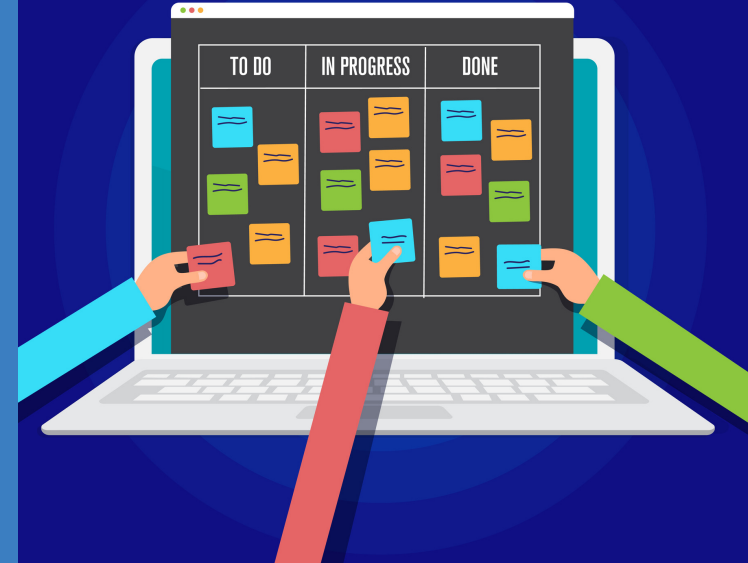
Semana 4



Las pruebas de software son una tarea que requiere de la participación de un equipo de profesionales, así como de experiencia, creatividad e ingenio.

Por estas razones resulta difícil, mas no imposible, automatizar el proceso de pruebas.

Todos los aspectos administrativos o repetitivos pueden automatizarse en cierto grado, incluso en un 100% en Agile Testing.



Pruebas unitarias.

Pruebas para aplicaciones web.

Pruebas de servicios web.

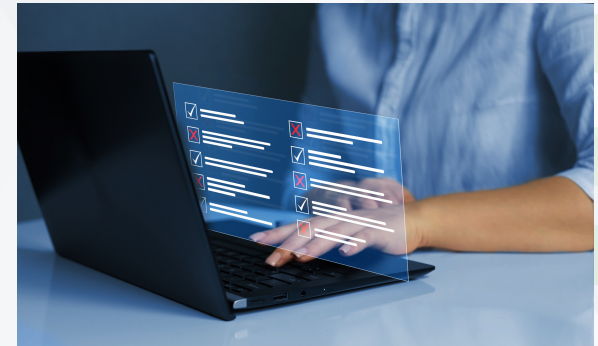
Pruebas de bases de datos relacionales.

Pruebas de lógica.

Pruebas de API.

Pruebas de carga.

Pruebas de estrés.



Herramientas para diseño, implementación y ejecución de pruebas.



Herramientas de administración, seguimiento y configuración.

Explicación

El software testing te ayudará a resolver los siguientes problemas:

- Detección de defectos de arquitectura (bases de datos).
- Bugs (fallas) en el código de programación.
- Funcionalidad de componentes no válida.
- Vulnerabilidades en la seguridad.
- Falta de escalabilidad.





AGILE

¿Consideras que el software de código abierto aumenta la competitividad de las herramientas de software testing a nivel global?

¿Por qué crees que las soluciones de Microsoft e IBM siguen siendo líderes en el mercado de pruebas de software, a pesar de que es necesario invertir una gran suma de dinero para adquirirlas?

Existen muchas herramientas para realizar pruebas de software, algunas son grandes *suites* y abarcan cada función imaginable, mientras que también existen otras con usos muy específicos.

No importa si se trata de aplicaciones comerciales o de código abierto, las opciones son infinitas.

