



Tecnológico de Monterrey

Integración de seguridad informática en redes y sistemas de software (Gpo 401)

Pruebas de software

Profesores:

*Luis Ignacio Bustillo Rubio
Faustino Bejarano Romero
Raime Alejandro Bustos Gardea
Enrique Antonio Sánchez Ordoñez*

**25 de agosto de 2022
Chihuahua, Chih.**

Investigación de los distintos tipos de casos de prueba

Pruebas unitarias

Las pruebas unitarias se realizan cerca de la fuente de la aplicación. Estas pruebas consisten en probar métodos y funciones individuales de las clases, componentes o módulos que usa tu software. En general, las pruebas unitarias son de bajo nivel y son fáciles de automatizar. Se pueden ejecutar rápidamente mediante un servidor de integración continua.

Pruebas de integración

Las pruebas de integración como lo indican en el nombre, son pruebas que integran diferentes métodos, funciones, componentes y/o módulos de software y se aseguran que estas trabajen correctamente en conjunto. Estos tipos de pruebas son más costosos de ejecutar, ya que requieren que varias partes de la aplicación estén en marcha.

Pruebas sistema

Las pruebas de sistema consisten en probar el correcto funcionamiento de todo un sistema/aplicación/producto de software. Estas se encargan de validar todas las interacciones que tienen todos los componentes del sistema.

Pruebas de regresión

Las pruebas de regresión son aquellas que se encargan de checar por bugs y errores cada vez que exista algún cambio. Estas pruebas no cambian ni se quitan solo se le agregan y se realizan cada vez que existe cualquier tipo de cambio.

Pruebas automatizadas

Las pruebas automatizadas son aquellas que se realizan por medio de alguna herramienta para automatizar el proceso manual de revisión y validación. Estas pruebas pueden incluir todas las pruebas mencionadas en esta sección.

Pruebas bajo condiciones frontera

Los escenarios de caso de prueba perimetral son aquellos que son posibles, pero que son características desconocidas o accidentales de los requisitos.

Herramientas de prueba

Nombre	Tipo	Licencia	Plataforma	Particularidades
VectorCAST	Integración	Comercial	C o C++	<ul style="list-style-type: none">• Valida los sistemas integrados críticos para la seguridad y el negocio• Se usa ampliamente en industrias financieras,

				dispositivos médicos, controles industriales, ferrocarriles.
Citrus	Integración	Open Source	Java	<ul style="list-style-type: none"> • Establece secuencia de mensajes y crea mensajes de error • Enviar y recibir mensajes • Espere el mensaje y active otro mensaje
LDRA	Integración	Open Source	C C++ Java Ada95	<ul style="list-style-type: none"> • Las pruebas se pueden generar y ejecutar fácilmente • Variedad de soporte para que las pruebas tengan un entorno común para una amplia gama de proyectos
Jasmine	Integración	Open Source	Javascript	<ul style="list-style-type: none"> • Para hacer las pruebas tiene una sintaxis simple • No necesita ningún framework extra de javascript
eZscript	Integración	Comercial	Delphi / Kylix	<ul style="list-style-type: none"> • Es compatible con las plataformas Windows y Linux. • Incluye funciones personalizadas que no existen en VB, como las funciones de administración de archivos FileRead y FileRename.
Xcode Cloud	Integración	Comercial	Swift y Objective C	<ul style="list-style-type: none"> • Permite tener un flujo de trabajo para integración continua • Se encarga de ejecutar las pruebas con cada commit • Prueba el código antes de ser publicado al App Store
Embunit	Unitarias	Comercial	C y C++	<ul style="list-style-type: none"> • Crea el código para las pruebas automáticamente
MochaJS	Unitarias	Open Source	JavaScript	<ul style="list-style-type: none"> • Implementado en nodejs • Soporte para

				diferentes navegadores <ul style="list-style-type: none"> • Proporciona una base limpia para desarrollar pruebas • Utilizar cualquier librería
JUnit	Unitarias	Open Source	Java	<ul style="list-style-type: none"> • Permite escribir casos de prueba mientras se desarrolla el software, esto puede ayudar a realizar pruebas tempranas y detectar problemas. • Soportado por todos los entornos de desarrollo integrados •
SimpleTest	Unitarias	Open	Php	<ul style="list-style-type: none"> • Permite crear los casos de prueba en scripts ejecutables
Módulo Unittest	Unitarias	Open Source	Python	<ul style="list-style-type: none"> • Permite hacer pruebas unitarias, pruebas integradoras, pruebas de un extremo a otro, pruebas de estrés y muchas otras por medio de la conexión de python a cliente.
SpiraTest	Sistema	Comercial	Ruby	<ul style="list-style-type: none"> • Maneja tareas y encuentra bugs de manera efectiva • Agrega problemas y tareas al usuario designado en cierta área
Rapise	Sistema	Comercial	Java, .NET, Flash, Qt, SWT, Ajax, WPF, Silverlight, etc.	<ul style="list-style-type: none"> • Admite la ejecución en paralelo. • Cuenta con automatización asistida. • Los test que realiza están basados en los requisitos y en modelos, son parametrizados y evalúan la seguridad. • Conformidad Unicode.

Leapwork	Sistema	Comercial	Cloud, SaaS, Web, Windows	<ul style="list-style-type: none"> • Admite la ejecución en paralelo • Gestión de requisitos • Tests basados en los requisitos • Tests de seguridad • Verificación de script de tests
LambdaTest	Sistema	Comercial	JIRA, Asana, Github, Trello, Slack	<ul style="list-style-type: none"> • Pruebas interactivas en vivo a través de VM alojado en la nube LambdaTest • Prueba automática de navegador • Integración con varias herramientas de CI / CD
SauceLabs	Sistema	Comercial	Cloud, Web, IOS, Android,	<ul style="list-style-type: none"> • Admite la ejecución en paralelo • Detecta regresiones visuales en tu aplicación UI • Te permite hacer tests automatizados basados en requisitos
Katalon Studio	Sistema	Comercial	Eclipse	<ul style="list-style-type: none"> • Permite automatizar aplicaciones web, aplicaciones móviles y pruebas API. • Grabación y reproducción para aplicaciones web y móviles. • Permite ejecutar pruebas en una configuración diferente.

Casos de pruebas

CASO DE PRUEBA 1

Identificador: Prueba 01

Nombre: testPiedravsPiedra

Escenario:

- Dado que el usuario elija manualmente piedra
- Cuando la computadora elija manualmente piedra
- Entonces el resultado arroja mensaje del ganador, con sus puntajes iguales

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 1, compu: 1, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p1", texto: "Empate", p1: 0, p2: 0

Entradas:

- valor: 1
- compu: 1
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p1"
- texto: "Empate"
- p1: 0
- p2: 0

CASO DE PRUEBA 2

Identificador: Prueba 02

Nombre: testPiedravsPapel

Escenario:

- Dado que el usuario elija manualmente piedra
- Cuando la computadora elija manualmente papel
- Entonces el resultado arroja mensaje del ganador, con sus puntajes actualizado

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 1, compu: 2, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p2", texto: "Ganador: CPU", p1: 0, p2: 1

Entradas:

- valor: 1
- compu: 2
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p2"
- texto: "Ganador: CPU"
- p1: 0
- p2: 1

CASO DE PRUEBA 3

Identificador: Prueba 03

Nombre: testPiedravsTijeras

Escenario:

- Dado que el usuario elija manualmente piedra
- Cuando la computadora elija manualmente tijeras
- Entonces el resultado arroja mensaje del ganador, con sus puntajes actualizado

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 1, compu: 3, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p3", texto: "Ganador: Jugador", p1: 1, p2: 0

Entradas:

- valor: 1
- compu: 3
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p3"
- texto: "Ganador Jugador"
- p1: 1
- p2: 0

CASO DE PRUEBA 4

Identificador: Prueba 04

Nombre: testPapelvsPiedra

Escenario:

- Dado que el usuario elija manualmente papel
- Cuando la computadora elija manualmente piedra
- Entonces el resultado arroja mensaje del ganador, con sus puntajes actualizados

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 2, compu: 1, p1: 0, p2: 1
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p1", texto: "Ganador: Jugador", p1: 1, p2: 1

Entradas:

- valor: 2
- compu: 1
- p1: 1
- p2: 1

Salida Esperada:

- imagen: "p1"
- texto: "Ganador: Jugador"

- p1: 1
- p2: 1

CASO DE PRUEBA 5

Identificador: Prueba 05

Nombre: testPapelvsPapel

Escenario:

- Dado que el usuario elija manualmente papel
- Cuando la computadora elija manualmente papel
- Entonces el resultado arroja mensaje del ganador, con sus puntajes actualizado

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 2, compu: 2, p1: 3, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p2", texto: "Empate", p1: 3, p2: 1

Entradas:

- valor: 2
- compu: 2
- p1: 3
- p2: 0

Salida Esperada:

- imagen: "p2"
- texto: "Empate"
- p1: 3
- p2: 0

CASO DE PRUEBA 6

Identificador: Prueba 06

Nombre: testPapelvsTijeras

Escenario:

- Dado que el usuario elija manualmente papel
- Cuando la computadora elija manualmente tijeras
- Entonces el resultado arroja mensaje del ganador, con sus puntajes actualizado

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 2, compu: 3, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p3", texto: "Ganador: CPU", p1: 0, p2: 1

Entradas:

- valor: 2

- compu: 3
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p3"
- texto: "Ganador CPU"
- p1: 0
- p2: 1

CASO DE PRUEBA 7

Identificador: Prueba 07

Nombre: testTijerasvsPiedra

Escenario:

- Dado que el usuario elija manualmente tijeras
- Cuando la computadora elija manualmente piedra
- Entonces el resultado arroja mensaje del ganador, con sus puntajes actualizados

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 3, compu: 1, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p1", texto: "CPU: Jugador", p1: 0, p2: 1

Entradas:

- valor: 3
- compu: 1
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p1"
- texto: "Ganador: CPU"
- p1: 0
- p2: 1

CASO DE PRUEBA 8

Identificador: Prueba 08

Nombre: testTijerasvsPapel

Escenario:

- Dado que el usuario elija manualmente tijeras
- Cuando la computadora elija manualmente papel
- Entonces el resultado arroja mensaje del ganador, con sus puntajes actualizado

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 3, compu: 2, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p2", texto: "Ganador: Jugador", p1: 1, p2: 0

Entradas:

- valor: 3
- compu: 2
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p2"
- texto: "Ganador: Jugador"
- p1: 1
- p2: 0

CASO DE PRUEBA 9

Identificador: Prueba 09

Nombre: testTijerasvsTijeras

Escenario:

- Dado que el usuario elija manualmente tijeras
- Cuando la computadora elija manualmente tijeras
- Entonces el resultado arroja mensaje de empate, con sus puntajes actualizado

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 3, compu: 3, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p3", texto: "Empate", p1: 0, p2: 0

Entradas:

- valor: 3
- compu: 3
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p3"
- texto: "Empate"
- p1: 0
- p2: 0

CASO DE PRUEBA 10

Identificador: Prueba 10

Nombre: testEntradaError1

Escenario:

- Dado que el usuario elija manualmente un valor fuera de rango
- Cuando la computadora elija manualmente papel
- Entonces el resultado arroja mensaje de error

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 4, compu: 2, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p3", texto: "Esas no existen", p1: 0, p2: 0

Entradas:

- valor: 4
- compu: 2
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p2"
- texto: "Esas no existen"
- p1: 0
- p2: 0

CASO DE PRUEBA 11

Identificador: Prueba 11

Nombre: testEntradaError2

Escenario:

- Dado que el usuario elija manualmente un valor fuera de rango
- Cuando la computadora elija manualmente tijeras
- Entonces el resultado arroja mensaje de error

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 4, compu: 3, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p3", texto: "Esas no existen", p1: 0, p2: 0

Entradas:

- valor: 4
- compu: 3
- p1: 0
- p2: 0

Salida Esperada:

- imagen: "p3"
- texto: "Esas no existen"
- p1: 0
- p2: 0

CASO DE PRUEBA 12

Identificador: Prueba 12

Nombre: testEntradaError3

Escenario:

- Dado que el usuario elija manualmente un valor fuera de rango
- Cuando la computadora elija manualmente un lugar fuera de rango
- Entonces el resultado arroja mensaje de error

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 15, compu: 5, p1: 0, p2: 0
- Se llama a la función Jugada con los valores ingresados manualmente
- Validar que el resultado arrojado de esa función haya sido igual a: imagen: "p3", texto: "Esas no existen", p1: 0, p2: 0

Entradas:

- valor: 15
- compu: 5
- p1: 0
- p2: 0

Salida Esperada:

- texto: "Esas no existen"
- p1: 0
- p2: 0

CASO DE PRUEBA 13

Identificador: Prueba 13

Nombre: testJuego

Escenario:

- Se crea manualmente una jugada
- Se validan que los datos de la jugada aparezcan correctamente

Instrucciones:

- Se ingresa manualmente una jugada con los valores: valor: 3, compu: 2, p1: 4, p2: 5
- Validar que la jugada sea igual a: valor : valor, compu : compu, p1 : p1, p2 : p2

Entradas:

- valor: 3
- compu: 2
- p1: 4
- p2: 5

Salida Esperada:

- valor: 3
- compu: 2
- p1: 4

- p2: 5

CASO DE PRUEBA 14

Identificador: Prueba 14

Nombre: testGanador

Escenario:

- Se crea manualmente una partida finalizada
- Se validan que los datos de la partida aparezcan correctamente

Instrucciones:

- Se ingresa manualmente una partida finalizada con los valores: imagen: "p1", texto: "Empate", p1: 4, p2: 5
- Validar que la jugada sea igual a: imagen: "p1", texto: "Empate", p1: 4, p2: 5

Entradas:

- imagen: "p1"
- texto: "Empate"
- p1: 4
- p2: 5

Salida Esperada:

- imagen: "p1"
- texto: "Empate"
- p1: 4
- p2: 5

CASO DE PRUEBA 15

Identificador: Prueba 15

Nombre: testJugadaValidaPapel

Escenario:

- Se presiona el botón de papel
- Se corre correctamente el juego

Instrucciones:

- Se presiona el botón de papel
- Validar que el texto del ganador sea igual a: "Ganador: Jugador", "Ganador: CPU" o "Empate"

Entradas:

- se presiona botonPapel

Salida Esperada:

- "Ganador: Jugador", "Ganador: CPU" o "Empate"

CASO DE PRUEBA 15

Identificador: Prueba 15

Nombre: testJugadaValidaPapel

Escenario:

- Se presiona el botón de papel
- Se corre correctamente el juego

Instrucciones:

- Se presiona el botón de tijeras
- Validar que el texto del ganador sea igual a: "Ganador: Jugador", "Ganador: CPU" o "Empate"

Entradas:

- se presiona botonPiedra

Salida Esperada:

- "Ganador: Jugador", "Ganador: CPU" o "Empate"

CASO DE PRUEBA 17

Identificador: Prueba 17

Nombre: testJugadaValidaTijeras

Escenario:

- Se presiona el botón de tijeras
- Se corre correctamente el juego

Instrucciones:

- Se presiona el botón de tijeras
- Validar que el texto del ganador sea igual a: "Ganador: Jugador", "Ganador: CPU" o "Empate"

Entradas:

- se presiona botonTijeras

Salida Esperada:

- "Ganador: Jugador", "Ganador: CPU" o "Empate"

Reporte gerencial

ID	NOMBRE	ESCENARIO	SALIDA ESPERADA	SALIDA REAL	P/F STATUS
01	restPiedra vsPiedra	El usuario empata con piedra	<ul style="list-style-type: none"> - imagen: "p1" - texto: "Empate" - p1: 0 - p2: 0 	<ul style="list-style-type: none"> - imagen: "p1" - texto: "Empate" - p1: 0 - p2: 0 	P
02	restPiedra vsPapel	El usuario pierde con piedra	<ul style="list-style-type: none"> - imagen: "p2" - texto: "Ganador: CPU" - p1: 0 - p2: 1 	<ul style="list-style-type: none"> - imagen: "p2" - texto: "Ganador: CPU" - p1: 0 - p2: 1 	P
03	restPiedra vsTijeras	El usuario gana con piedra	<ul style="list-style-type: none"> - imagen: "p3" - texto: "Ganador Jugador" - p1: 1 - p2: 0 	<ul style="list-style-type: none"> - imagen: "p3" - texto: "Ganador Jugador" - p1: 1 - p2: 0 	P
04	testPapel vsPiedra	El usuario gana con papel	<ul style="list-style-type: none"> - imagen: "p1" - texto: "Ganador: Jugador" - p1: 1 - p2: 1 	<ul style="list-style-type: none"> - imagen: "p1" - texto: "Ganador: Jugador" - p1: 1 	P

				- p2: 1	
05	testPapelvsPapel	El usuario empata con papel	<ul style="list-style-type: none"> - imagen: "p2" - texto: "Empate" - p1: 3 - p2: 0 	<ul style="list-style-type: none"> - imagen: "p2" - texto: "Empate" - p1: 3 - p2: 0 	P
06	testPapelvsTijeras	El usuario pierde con papel	<ul style="list-style-type: none"> - imagen: "p3" - texto: "Ganador CPU" - p1: 0 - p2: 1 	<ul style="list-style-type: none"> - imagen: "p3" - texto: "Ganador CPU" - p1: 0 - p2: 1 	P
07	testTijerasvsPiedra	El usuario pierde con tijeras	<ul style="list-style-type: none"> - imagen: "p1" - texto: "Ganador: CPU" - p1: 0 - p2: 1 	<ul style="list-style-type: none"> - imagen: "p1" - texto: "Ganador: CPU" - p1: 0 - p2: 1 	P
08	testTijerasvsPapel	El usuario gana con tijeras	<ul style="list-style-type: none"> - imagen: "p2" - texto: "Ganador: Jugador" - p1: 1 - p2: 0 	<ul style="list-style-type: none"> - imagen: "p2" - texto: "Ganador: Jugador" - p1: 1 - p2: 0 	P

09	testTijeras vsTijeras	El usuario empata con tijeras	<ul style="list-style-type: none"> - imagen: "p3" - texto: "Empate" - p1: 0 - p2: 0 	<ul style="list-style-type: none"> - imagen: "p3" - texto: "Empate" - p1: 0 - p2: 0 	P
10	testEntradaError1	Se ingresa un valor fuera del rango en una partida	<ul style="list-style-type: none"> - imagen: "p2" - texto: "Esas no existen" - p1: 0 - p2: 0 	<ul style="list-style-type: none"> - imagen: "p2" - texto: "Esas no existen" - p1: 0 - p2: 0 	P
11	testEntradaError2	Se ingresa un valor fuera del rango en una partida que da una diferencia de 2 (para checar que si da error y no manda ganador)	<ul style="list-style-type: none"> - imagen: "p3" - texto: "Esas no existen" - p1: 0 - p2: 0 	<ul style="list-style-type: none"> - imagen: "p3" - texto: "Esas no existen" - p1: 0 - p2: 0 	P
12	testEntradaError3	Se ingresan varios valores fuera de rango en una partida	<ul style="list-style-type: none"> - texto: "Esas no existen" - p1: 0 - p2: 0 	<ul style="list-style-type: none"> - texto: "Esas no existen" - p1: 0 - p2: 0 	P
13	testJuego	Se crea una manualmente un objeto Jugada	<ul style="list-style-type: none"> - valor: 3 - compu: 2 - p1: 4 - p2: 5 	<ul style="list-style-type: none"> - valor: 3 - compu: 2 	P

				<ul style="list-style-type: none"> - p1: 4 - p2: 5 	
14	testGanador	Se crea manualmente un objeto GanadorJuego	<ul style="list-style-type: none"> - imagen: "p1" - texto: "Empate" - p1: 4 - p2: 5 	<ul style="list-style-type: none"> - imagen: "p1" - texto: "Empate" - p1: 4 - p2: 5 	P
15	testJugadorValidaPiedra	Se inicia dando click al botón piedra	<ul style="list-style-type: none"> - "Ganador: Jugador", "Ganador: CPU" o "Empate" 	<ul style="list-style-type: none"> - "Ganador: Jugador", "Ganador: CPU" o "Empate" 	P
16	testJugadorValidaPapel	Se inicia dando click al botón papel	<ul style="list-style-type: none"> - "Ganador: Jugador", "Ganador: CPU" o "Empate" 	<ul style="list-style-type: none"> - "Ganador: Jugador", "Ganador: CPU" o "Empate" 	P
17	testJugadorValidaTijeras	Se inicia dando click al botón tijeras	<ul style="list-style-type: none"> - "Ganador: Jugador", "Ganador: CPU" o "Empate" 	<ul style="list-style-type: none"> - "Ganador: Jugador", "Ganador: CPU" o "Empate" 	P

Repositorio

<https://github.com/Abdielz-z/actividad>

Referencias:

Applause App Quality, Inc. (2022, 1 agosto). How to Find and Test Edge Cases. Recuperado de

<https://www.applause.com/blog/how-to-find-test-edge-cases#:~:text=Edge%20test%20case%20scenarios%20are,specific%20and%20calculated%20value%20fields>.

REHKOPF, M. R. (s. f.). Pruebas de software automatizadas para la entrega continua. Recuperado de

<https://www.atlassian.com/es/continuous-delivery/software-testing/automated-testing>

colaboradores de Wikipedia. (2022, 20 febrero). Pruebas de regresión.

Recuperado de https://es.wikipedia.org/wiki/Pruebas_de_regresi%C3%B3n

IBM. (2021). Visión general de casos de prueba y suites de pruebas.

Recuperado de

<https://www.ibm.com/docs/es/elm/6.0.3?topic=testing-test-case-test-suite-overview>

PITTET, S. P. (s. f.). Los distintos tipos de pruebas en software. Recuperado de

<https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>