



Ingeniería de Software

22 - Quality Assessment

IIC2143-3

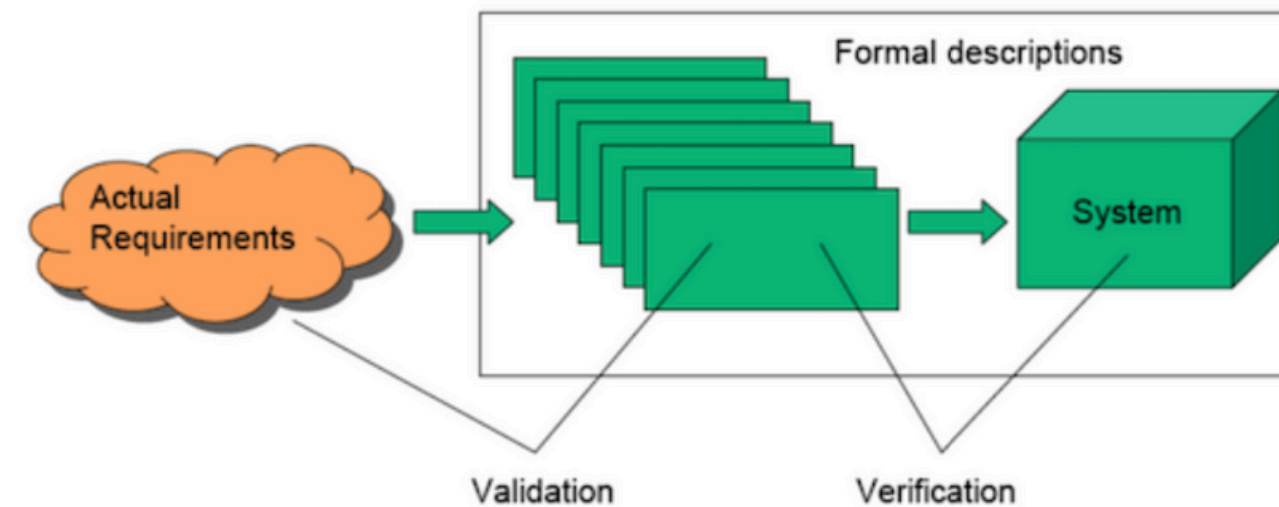
Josefa España

jpespana@uc.cl



¿Qué es la calidad?

- Asegurar que el producto cumple con los requisitos funcionales y no funcionales.
- Verificaciones: el producto hace lo que se especificó.
- Validaciones: el producto hace lo que el usuario necesita.





Formas de lograr la calidad

- No incorporar bugs al momento de construir el software (impracticable para productos grandes y complejos).
- Análisis estático del código para encontrar potenciales problemas. (automatizado)
- Inspección formal del código. (manual)
- Testing para diseñar casos de prueba y someter al software a estos.



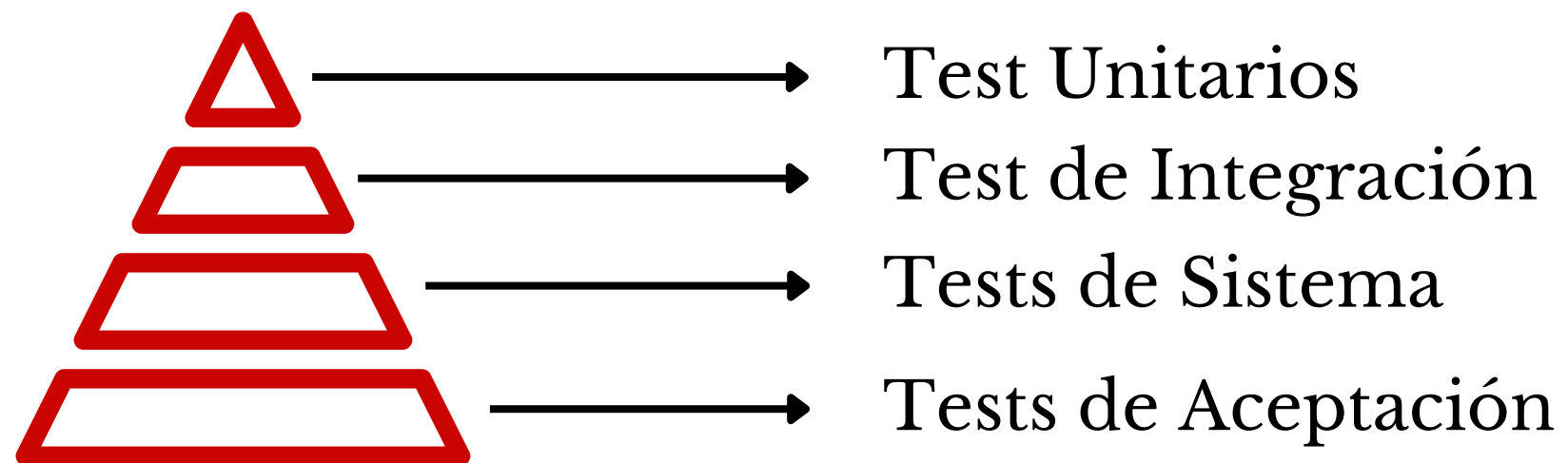
Testing



Testing

- Proceso que busca encontrar problemas sometiendo al producto a casos de prueba diseñados para maximizar la probabilidad de encontrar problemas.

Niveles de testing:





Niveles de testing

Tests Unitarios

- Testean componentes elementales (clases, módulos, servicios) usando herramientas como MiniTest y Rspec.

Tests de Integración

- Testear el flujo de dos o más componentes.

Tests de Sistema

- Testear el sistema completo como un todo. Verificación completa de los requisitos.

Tests de Aceptación.

- Validaciones por parte de los usuarios del producto.

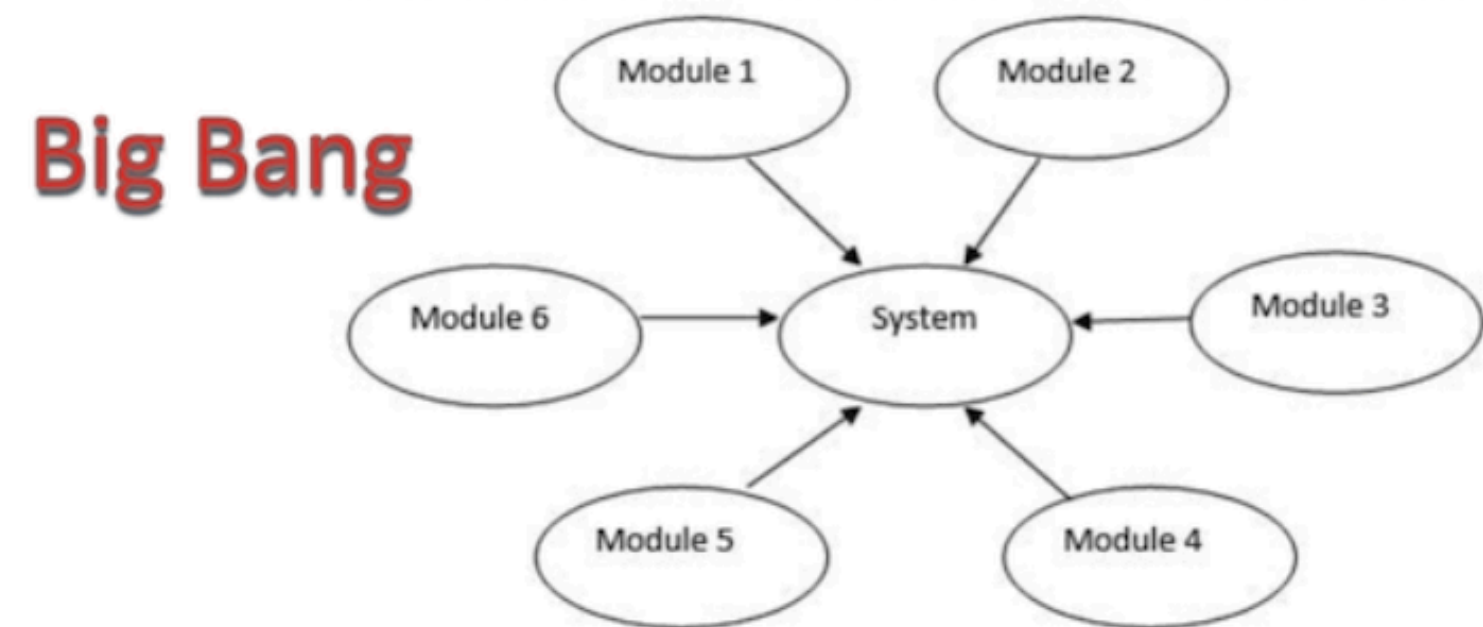


Tests de Integración



Tests de Integración

- Detecta los problemas que no tienen las unidades de forma individual, pero sí aparecen cuando se integran dichas unidades para formar una unidad mayor.
- Estrategia de integración conocida como “Big Bang”, porque se integra todo al final.





Tests de Integración

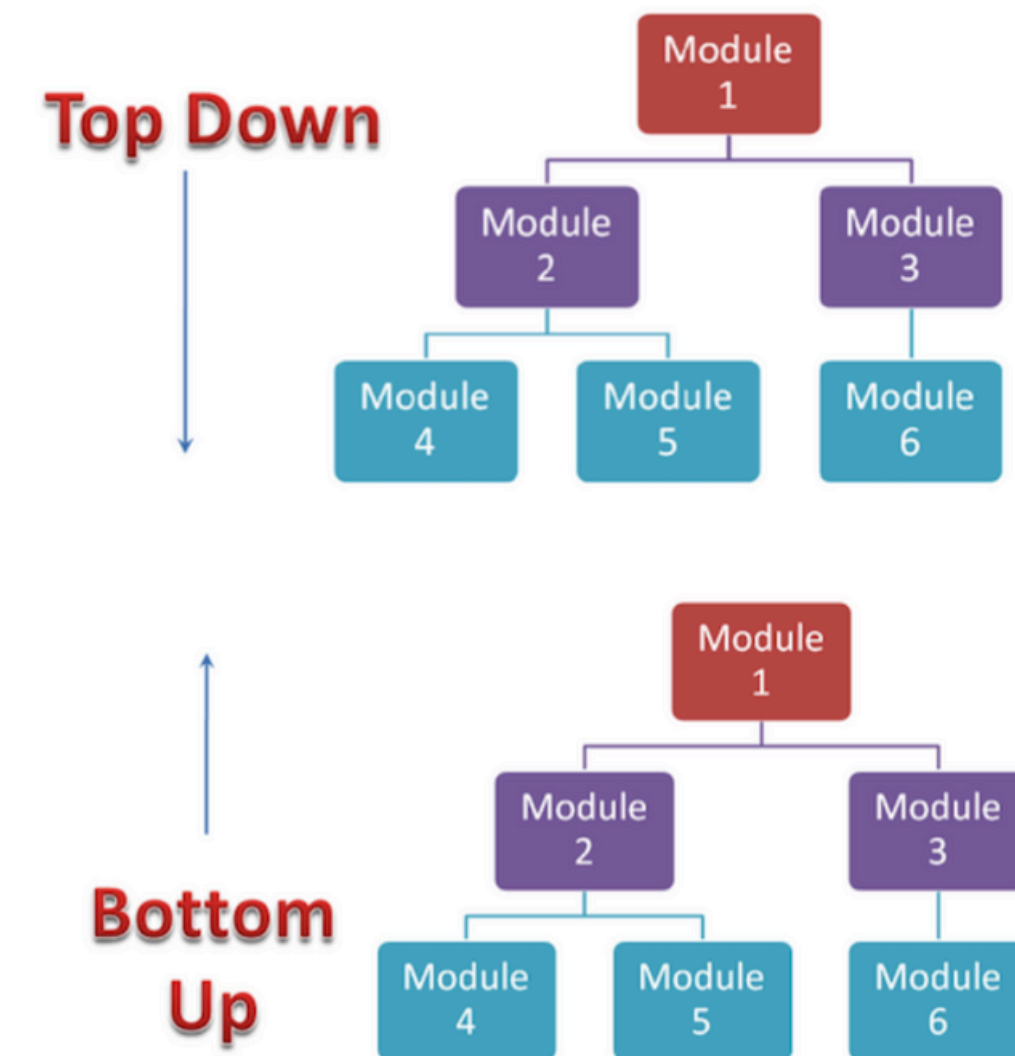
Estrategia “Bottom Up”:

- Integrar desde unidades más pequeñas a más grandes.

Estrategia “Top Down”:

- Se integran los componentes más grandes hacia los más pequeños.
- Mientras los pequeños no están disponibles, se hacen “stubs” para simular su comportamiento.

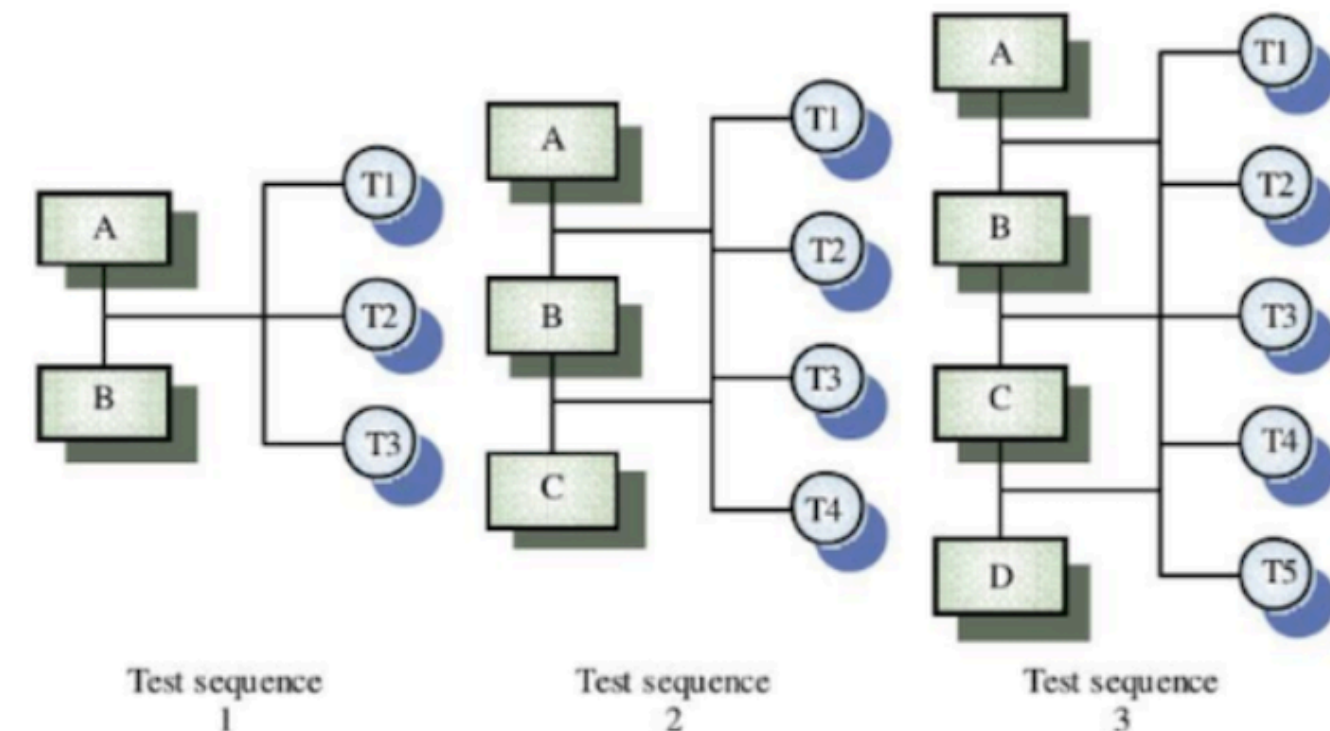
stub: Componente ficticio o que simula el comportamiento.





Integración continua

- Fundamental que a la hora de integrar una nueva funcionalidad se realicen los tests de integración necesarios.
- Actualmente es común hacer un “build” diario de la aplicación, por lo que la integración es continua.





Tests de Regresión

- Asegurar que después de agregar una funcionalidad, el sistema sigue funcionando como lo hacía antes de ser integrado
- Asegurar que al arreglar un error en un componente, no hayamos roto otras cosas a su paso.

Regression:
"when you fix one bug, you
introduce several newer bugs."





Tests de Regresión

Se realiza un set de regresión que nos permitirá asegurar la calidad:

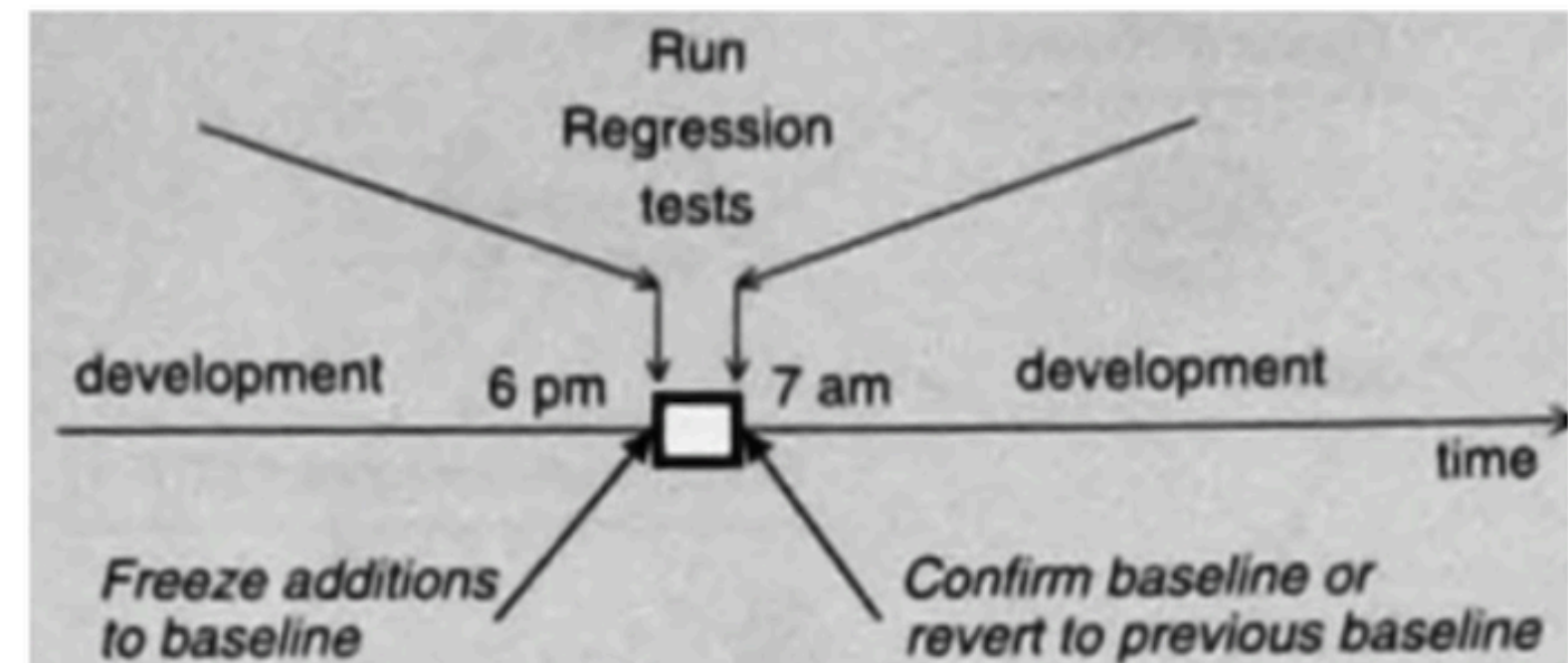
- Incluir tests sobre nuevas funcionalidades.
- Mostrar representativas que ejerciten todas las funciones.
- Tests que se centren en funcioncionalidades modificadas.
- ...



Tests de Regresión

Desafíos

- ¿Con qué frecuencia correr los tests de regresión?
- ¿Cómo revertir los cambios en caso de que un test no pase?





Smoke tests y Sanity Test

Tests que nos permiten no tener que correr todos los tests de regresión para revisar comportamientos importantes:

Smoke tests

- Nos permiten asegurar que las funciones básicas de un programa funcionan correctamente después de alguna modificación.
 - Por ejemplo, nos permiten saber si la aplicación andará en producción.

Sanity tests

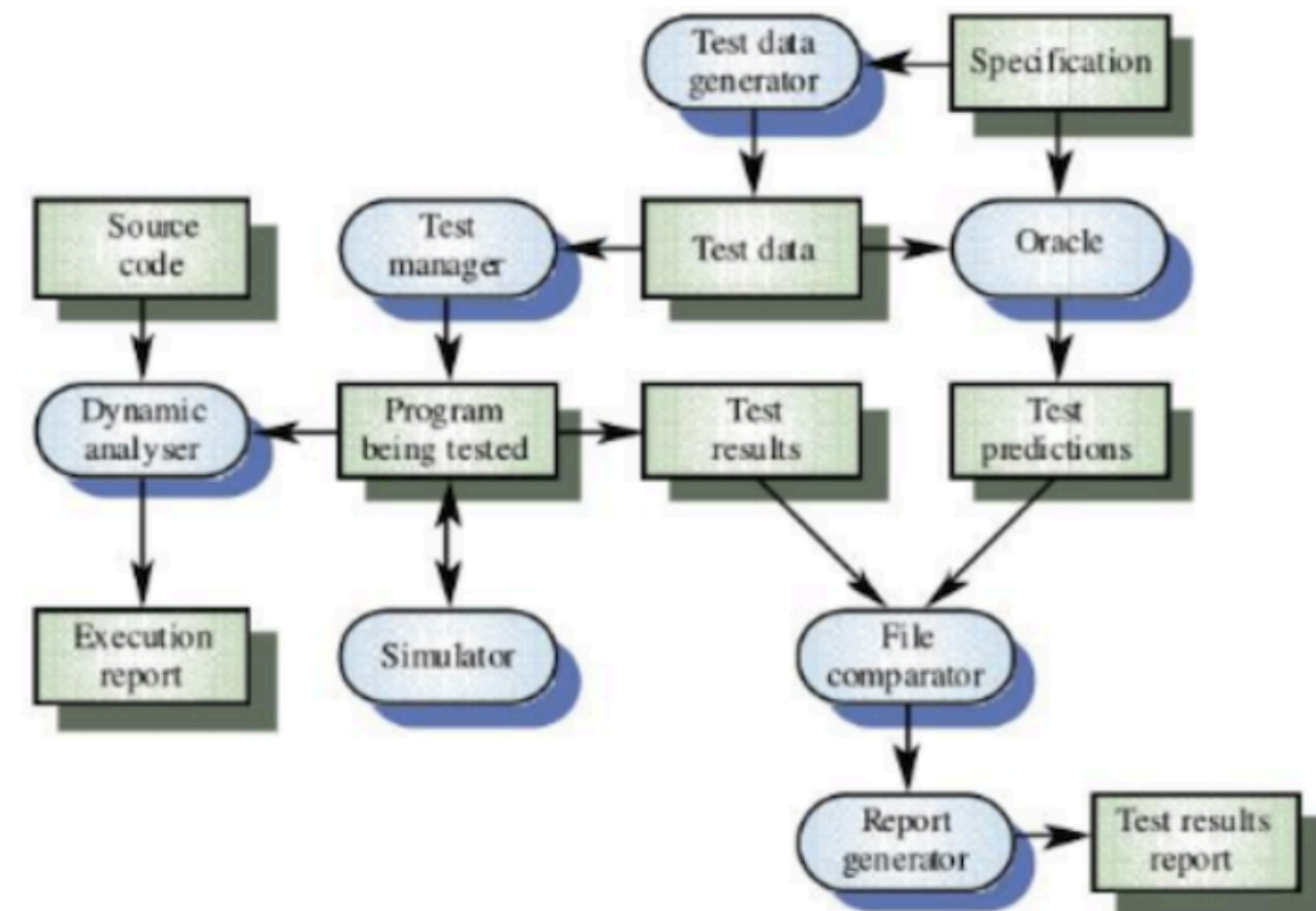
- Nos permite ver que un error o funcionalidad en específico ha sido corregida



Automatizar tests de regresión

Los tests de regresión al revisar muchas funcionalidades se vuelven imposibles de realizar manualmente, por lo que deben ser automatizados.

- Selenium: Aplicación opensource que permite testear aplicaciones web con una interfaz de usuario interactiva.
 - Ver que la interfaz funcione como un todo.





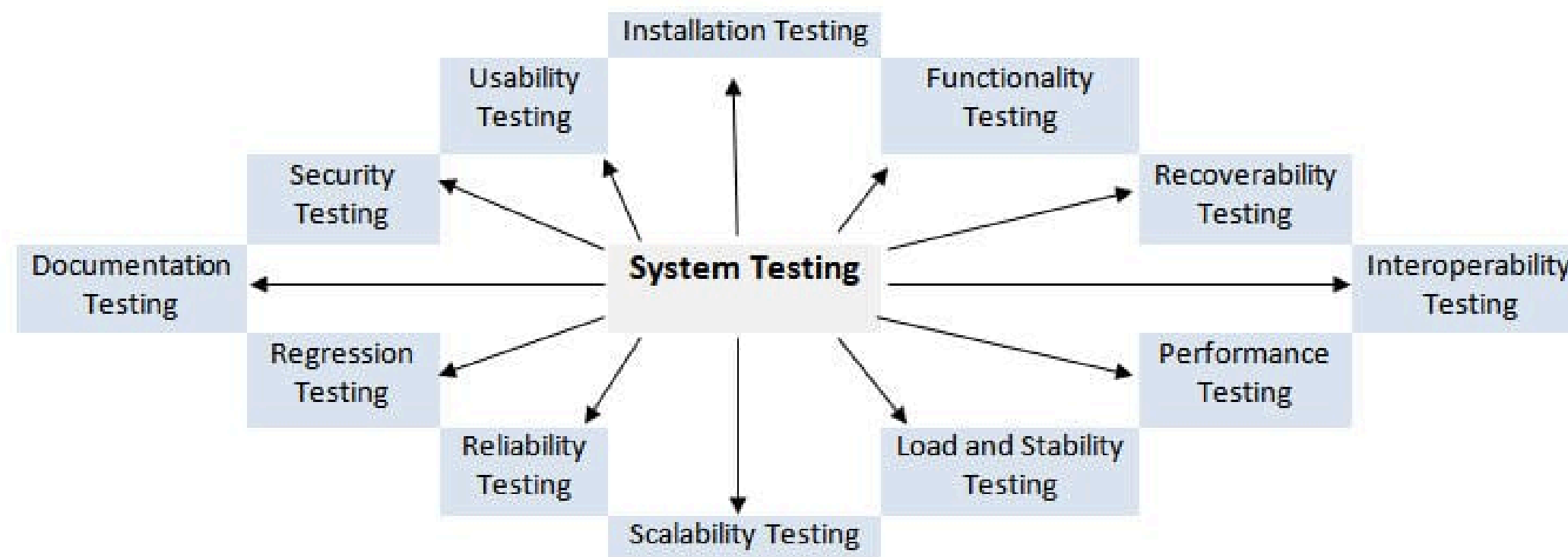
Tests de Sistema



Tests de Sistema

Verificar el producto como un todo, end-to-end.

- Revisar que el producto satisfaga los requerimientos funcionales y no funcionales.



System Testing - © www.SoftwareTestingHelp.com



Load Test y Stress Test

Tests de Carga (Load Test)

- Nos permite testear si nuestra aplicación resiste una cantidad esperada de usuarios o de datos.

Tests de Esfuerzo (Stress Test)

- Someter al sistema a una carga muy por encima de la carga para la que fue diseñada, con el objetivo de saber la estabilidad y confianza del sistema ante grandes cantidades de usuarios o datos.
 - Podría bloquearse completamente, caer con pantalla azul, etc. Debemos considerarlo para mantener nuestro software.



Tests de Usabilidad

- Tests para observar y registrar la forma en que un grupo de usuarios lleva a cabo algunas tareas específicas utilizando el software.
- Busca ver qué tan fácil les resultó llevar a cabo las tareas para las cual el software fue pensado.
 - Cuántos clicks realizaron en lugares incorrectos, cuanto demoró, etc.
- **NO** busca que los usuarios digan si les gustó o es amistoso, solo estudiar que tan usable es.



Test de Aceptación



Tests de Aceptación

- Único test para hacer una validación del producto final, porque está involucrada la satisfacción del usuario.
- Permite identificar problemas que no se detectaron antes porque no fue probado por usuarios reales.

Tipos de tests de aceptación:

- **Alpha testing:** Testear en el ambiente de desarrollo por equipo externo.
- **Beta testing:** Testear en el ambiente del cliente por los clientes.
- **Contract Acceptance Testing:** Comprobar lo esperado vs lo acordado en contrato.
- **Regulation Acceptance Testing:** Regulaciones legales.
- **Operational Acceptance Testing:** Backups, mantención, entrenamiento, y todo lo que se necesite para operar.



Quality Assessment

	System	Acceptance	Regression
Test for ...	Correctness, completion	Usefulness, satisfaction	Accidental changes
Test by ...	Development test group	Test group with users	Development test group
	Verification	Validation	Verification



Otro test: A/B Testing

- Evaluar dos versiones distintas del producto, versión A y B, y ver cuál es más efectiva según alguna métrica.
- Se elige la versión con la métrica más alta.

Se puede hacer en vivo, publicándole a un 50% de clientes la versión A, y al otro 50% la opción B. Posteriormente se comparan los resultados para ver cuál funciona mejor.



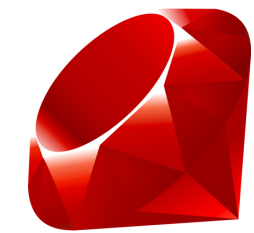


Bibliografía

- Apuntes de Testing del Libro del curso.



¿Consultas?



Ingeniería de Software

22 - Quality Assessment

IIC2143-3

Josefa España

jpespana@uc.cl