UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Departamento de Informática
Universidad Técnica Federico Santa María

Ingeniería de Software

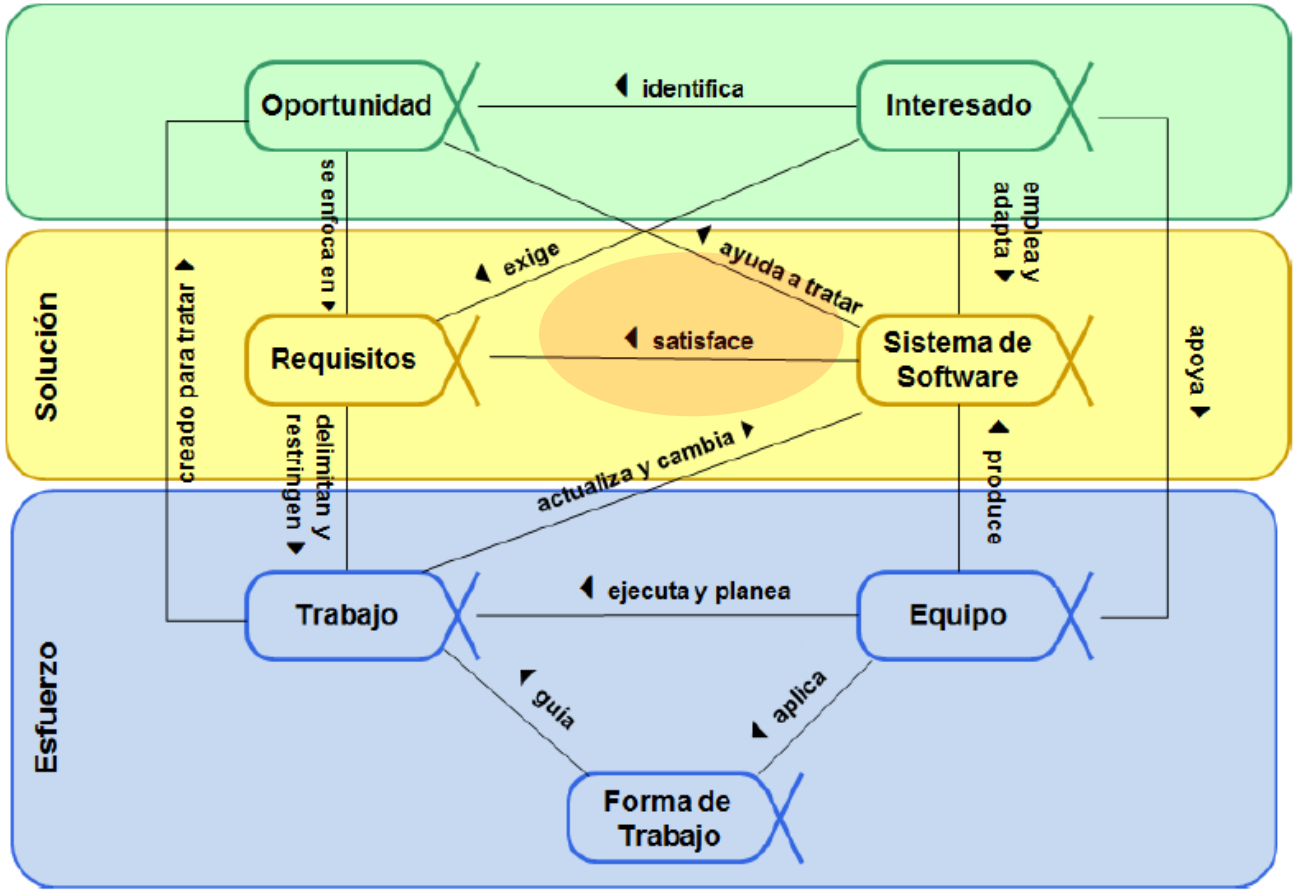
Testing

Hernán Astudillo & Gastón Márquez
Departamento de Informática
Universidad Técnica Federico Santa María

Contexto [1]

- ADSW
  - ¿Validación del proyecto?
  - Si fue validado, ¿Cómo se hizo?
  - Algunas ideas: prueba y error, múltiples accesos, otros
- ¿Existirá algún proceso/modelo que ayude a realizar pruebas y asegurarnos que nuestro sistema de software fallará lo menos posible?

Contexto [2]

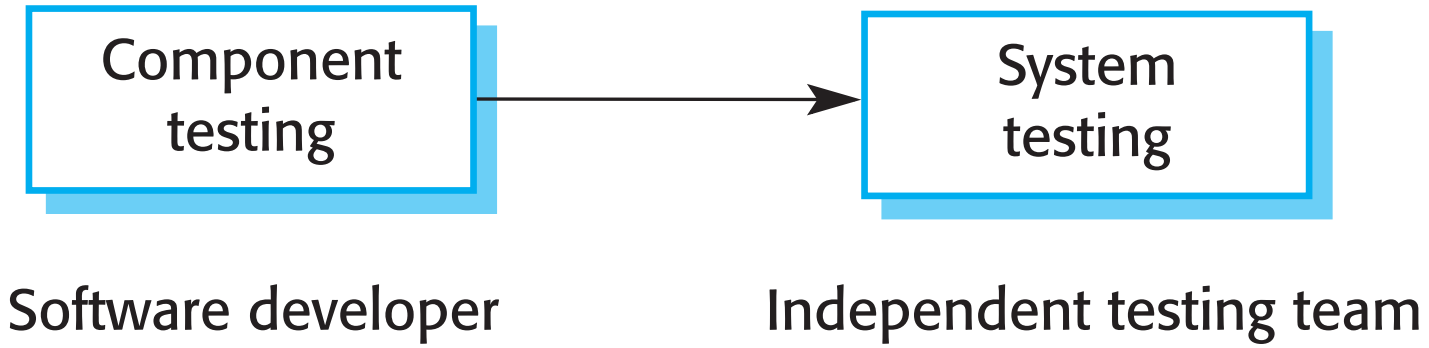

Oportunidad
identifica
Interesado
se enfoca en
exige
ayuda a tratar
empleа y adapta
Solución
creado para tratar
Requisitos
satisface
Sistema de Software
apoya
delimitan y restringen
actualiza y cambia
produce
Esfuerzo
Trabajo
ejecuta y planea
Equipo
guía
aplica
Forma de Trabajo

Testing Process

El proceso de Prueba [1]

- Existen dos enfoques a la hora de hablar de pruebas
  - Prueba de componentes
    - Se verifica individualmente cada componente
    - Se analiza la responsabilidad del desarrollo del componente
    - Las pruebas se realizan, generalmente, de la experiencia de los desarrolladores
  - Prueba del sistema
    - Se hace prueba a un grupo de componentes integrados para crear sistemas o sub-sistemas
    - La responsabilidad de la prueba se traspasa a un equipo
    - Los test se basan en la especificación del sistema

El proceso de Prueba [2]



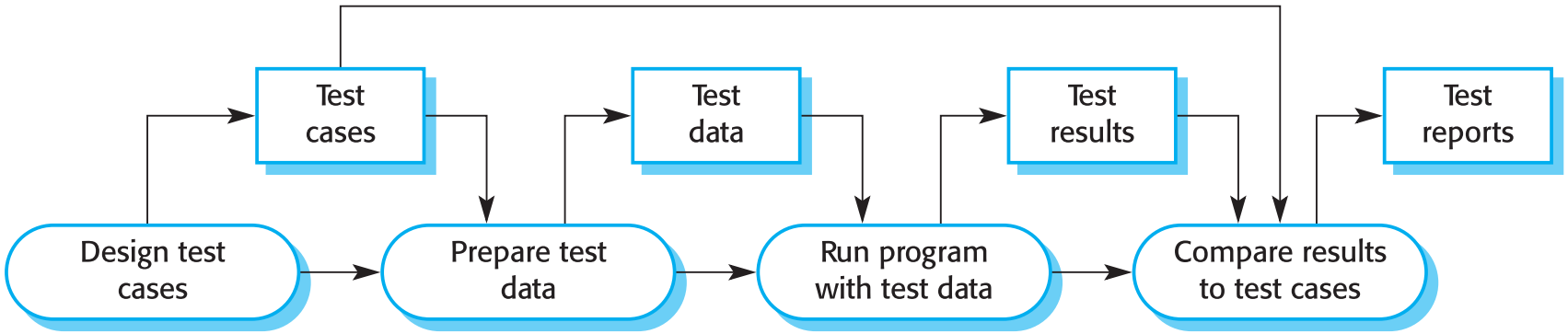
Defectos de las Pruebas

- El objetivo de las pruebas es encontrar defectos en los sistemas de software
- Una prueba exitosa es una prueba en donde se encuentran las causas que producen un comportamiento anómalo en un software
- Pero, la prueba sólo muestra la presencia de un defecto, no la solución

Objetivos de las Pruebas

- **Validación**
  - Demostrar a los desarrolladores y el sistema que los requerimientos establecidos han sido realizados exitosamente
  - Una prueba exitosa demuestra que el sistema de software funciona como se había establecido
- **Defectos**
  - Descubrir una falta o defecto en el sistema de software donde el comportamiento es incorrecto o no es conforme a las especificaciones establecidas

El proceso de Testing de Software



Políticas de Pruebas

- Solamente una prueba exhaustiva de Prueba puede mostrar que un sistema de software está libre de defectos. Sin embargo, una prueba exhaustiva es imposible
- Las políticas de Pruebas definen los enfoques que deben ser utilizados al momento de seleccionar las pruebas en un sistema
  - Todas las funcionalidades accedidas a través de menú deben ser testeadas
  - Las combinaciones de funcionalidad accedidas a través del mismo menú deben ser testeadas
  - Cuando el usuario ingresa peticiones al sistema, todas las funcionalidades deben ser testeadas

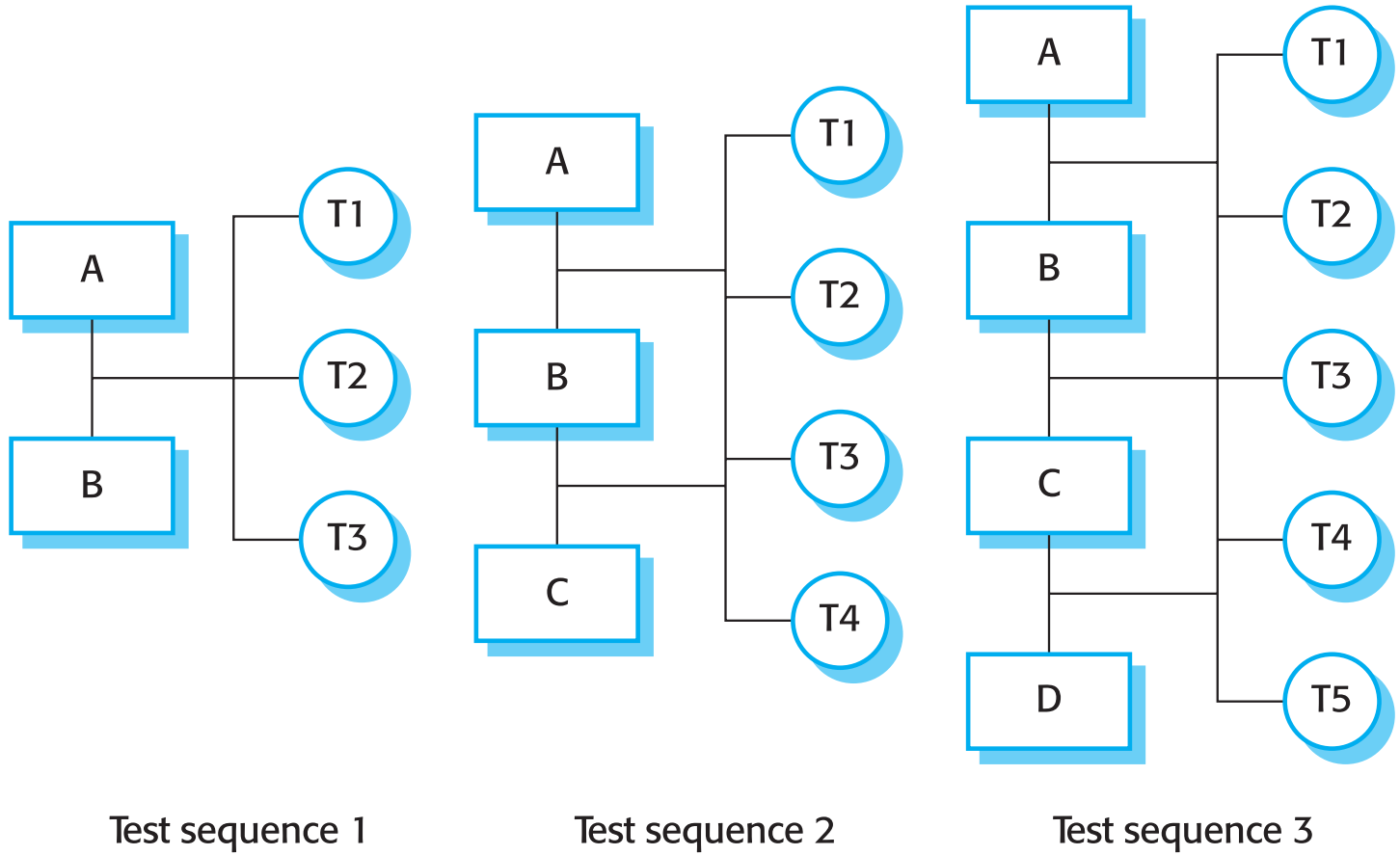
Prueba del sistema

- Involucra la integración de componentes para crear sistemas o sub-sistemas
- Se definen dos fases
  - Pruebas de Integración: el equipo de testing tiene acceso a todo el código fuente del sistema. El sistema es testeado en función de la integración de los componentes
  - Pruebas de Entrega: el equipo de testing prueba el sistema completo para ser entregado como una *caja negra*

Pruebas de Integración

- Involucra la creación del sistema desde sus componentes y se hacen pruebas para identificar problemas que puedan surgir desde la interacción de los componentes
- Integración top-down
  - Desarrollo del esqueleto del sistema y se agregan componentes
- Integración bottom-up
  - Integra la infraestructura de componentes y luego se le agregan funcionalidades
- Para simplificar las pruebas, el sistema debe ser integrado incrementalmente

Pruebas de Integración incremental



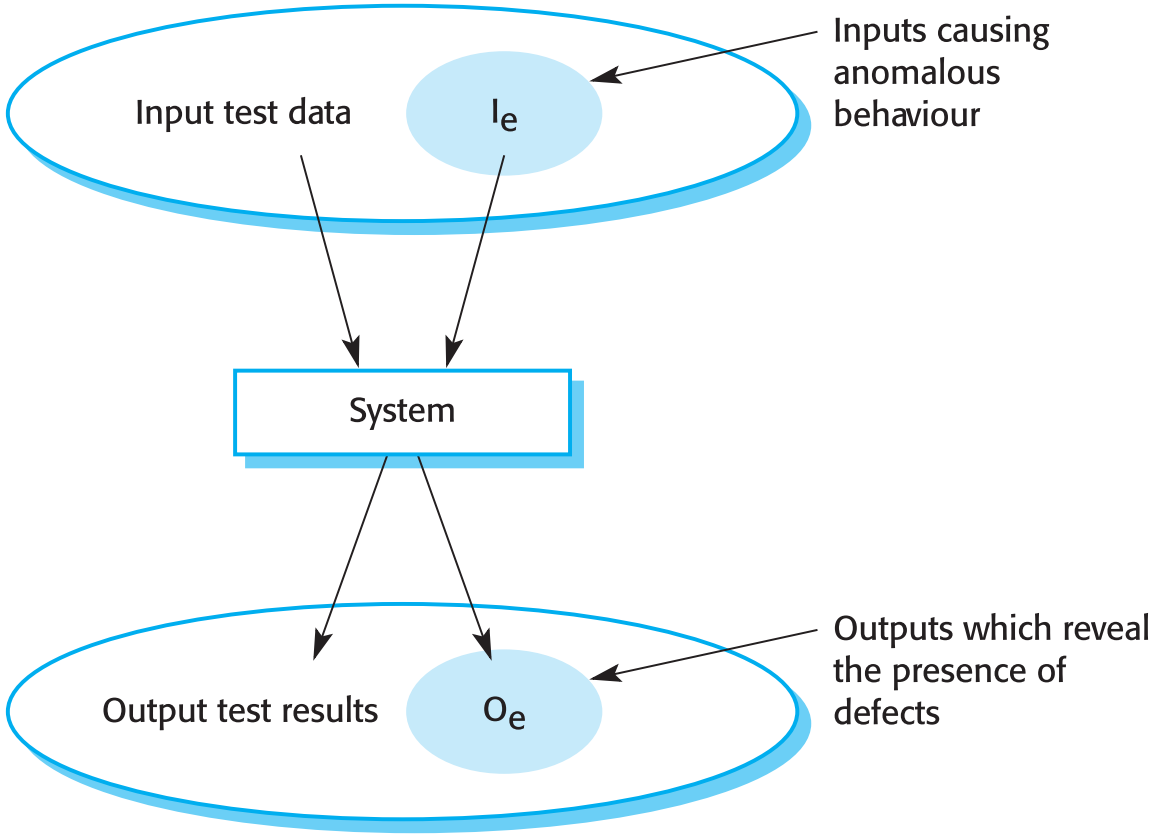
Enfoques de las Pruebas

- Validación de la arquitectura
  - La integración top-down es mejor para descubrir errores en una arquitectura de software
- Demostración del sistema
  - La integración top-down permite una demostración temprana en los inicios de la fase de desarrollo
- Test de implementación
  - La integración bottom-up es útil

Entregas de Pruebas

- El proceso de entregas de Pruebas de un sistema debe ser distribuido a los clientes
- El principal objetivo de lo anterior es entregar confianza al cliente de que el sistema ***cumple*** los requisitos establecidos
- Las entregas de Pruebas son usualmente cajas negras o funcionalidades
  - Basadas en la especificación del sistema
  - Los testers no saben lo que contiene la implementación del sistema

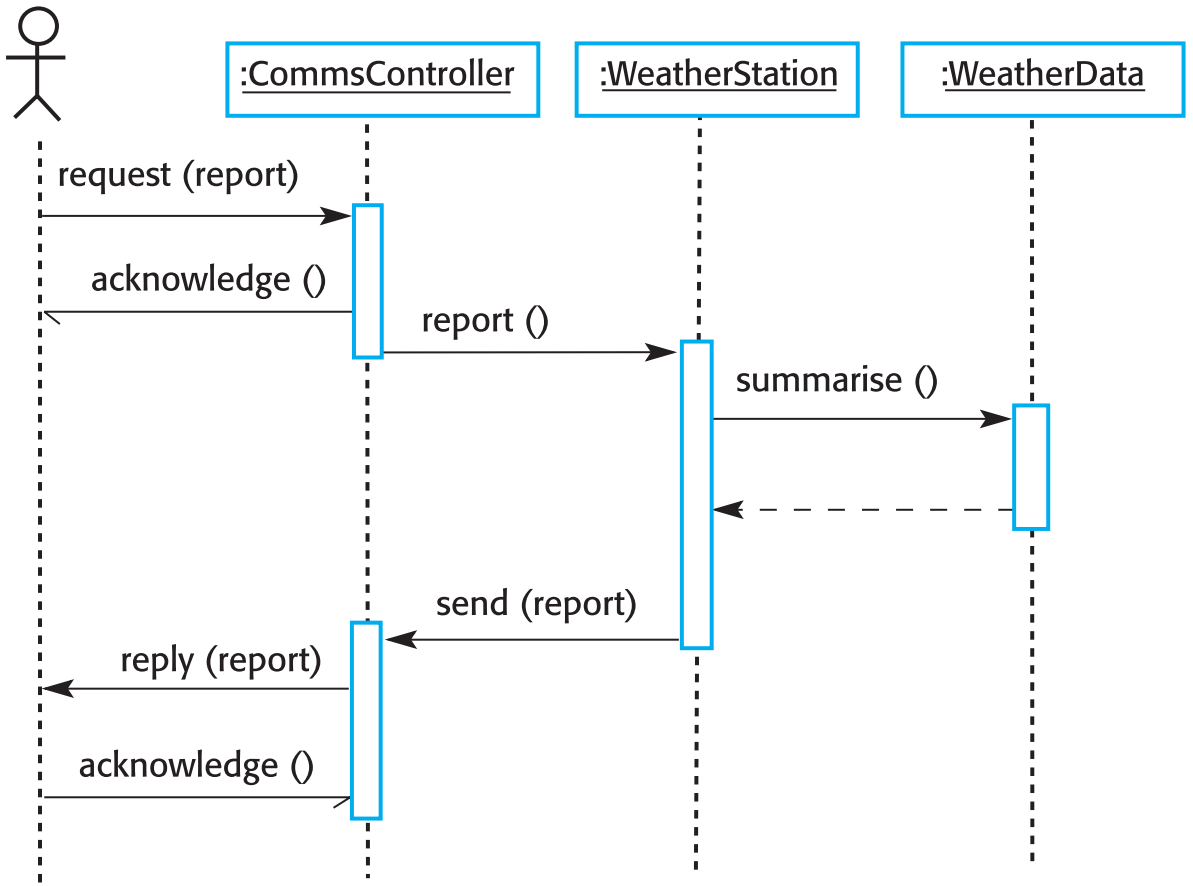
Black-box Testing



Casos de Uso [1]

- Los Casos de Uso pueden ser la base para derivar pruebas en el sistema de software
- Éstos ayudan a identificar operaciones que pueden ser testeadas y ayudan al diseño de los casos de prueba
- Además, con la ayuda de los diagramas de secuencias, se pueden realizar entradas y salidas para realizar pruebas de testing

Casos de Uso [2]



Guías para Testing [1]

- Las guías que se describirán a continuación son propuesta por Sommerville [Sommerville, 2004] las cuales describen consejos para elegir la mejor prueba para encontrar defectos
  - Elegir las entradas que fuercen al sistema a generar un error
  - Diseñar entradas que cause saturación del buffer
  - Repetir los puntos anteriores varias veces
  - Forzar salidas inválidas del sistema
  - Probar dichas salidas con grandes y pequeños volúmenes de datos

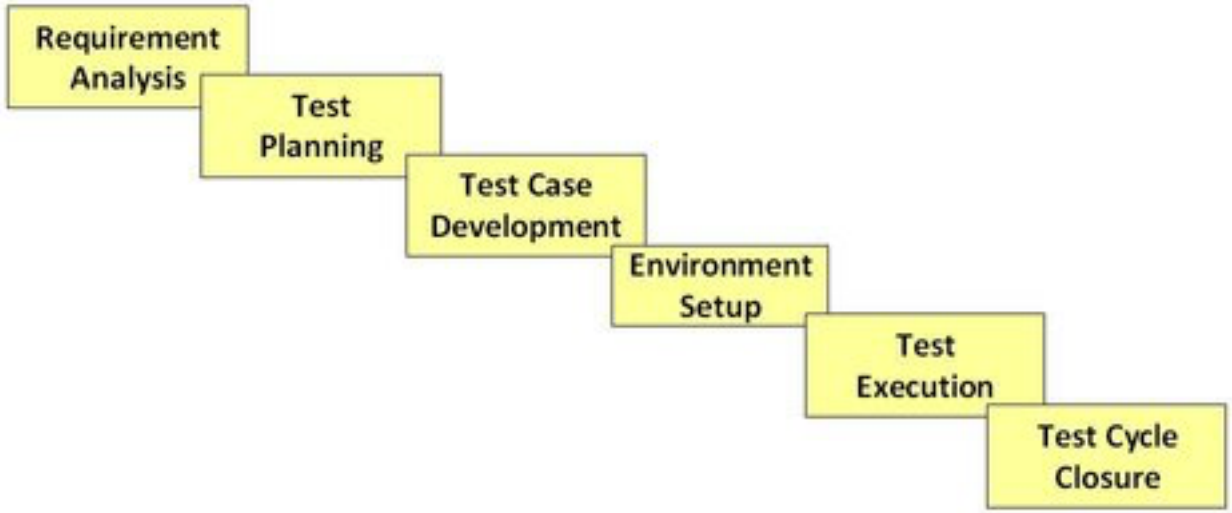
Guías para Testing [2]

- QA Framework: Test Guidelines (http://www.w3.org/TR/2004/WD-qaframe-test-20040225/)
- Según Software Testing Mentor [http://www.softwaretestingmentor.com/articles/guidelines-for-software-testing/, 2016], algunas guías prácticas para Testing son:
  1. La Prueba debe descubrir defectos y mejorar la calidad del software
  2. La Prueba debe ser realizado a través del Ciclo de Vida del Software
  3. La Prueba se debe hacer mediante la caja negra y blanca
     - White-box Testing consiste en pruebas estructurales del sistema de software, las cuales prueban la lógica interna del sistema
  4. La Prueba debe ser ejecutado bajo la lógica del negocio
  5. La Prueba debe ser ejecutado eficientemente con el objetivo de reducir el riesgo

Software Testing Life Cycle STLC

STLC

- Hasta el momento, se ha visto que Testing es una serie de actividades
- Pero, son actividades metodológicas



Análisis de Requisitos

- Durante esta fase se estudian los requisitos desde el punto de vista del testing con el objetivo de identificar requisitos testeables.
- **Actividades**
  - Identificar tipos de prueba
  - Recopilar datos sobre el establecimiento de prioridades y el enfoque
  - Identificar ambientes de prueba en donde se detalle qué se va a probar
- **Entregables**
  - Requirement Tracebility Matrix (RTM, http://www.guru99.com/traceability-matrix.html)
- **Necesidad**
  - Curiosidad

Plan de Prueba

- En esta fase, se discute el plan estratégico de las pruebas
- Típicamente, en esta etapa el analista QA determina el esfuerzo y costos estimados del proyecto y cómo se prepara y termina el plan
- **Actividades**
  - Preparación del plan
  - Selección de herramienta
  - Test de esfuerzo y estimación
  - La planificación de recursos y la determinación de funciones y responsabilidades
  - Entrenamiento
- **Entregables**
  - Plan de pruebas (http://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html)
  - Estimación del esfuerzo (http://www.guru99.com/an-expert-view-on-test-estimation.html)
- **Necesidad**
  - Visión holística

Desarrollo de Casos de Prueba

- Esta fase involucra la creación, verificación y trabajo de los casos de prueba
- Se identifican los datos de prueba
- **Actividades**
  - Creación de casos de prueba
  - Revisar las bases de los casos de prueba
  - Creación de datos de prueba (si es necesario)
- **Entregables**
  - Casos/scripts de prueba
  - Datos de prueba
- **Necesidad**
  - Creatividad

Configuración del ambiente de Prueba

- La configuración del ambiente de prueba decide las condiciones del hadware o software donde el sistema será probado
- **Actividades**
  - Entender la arquitectura, configuración del ambiente y preparar hadware y software por cada requerimiento que será probado
  - Realizar pequeñas pruebas del sistema (para ver si todo está bien con el ambiente de prueba)
- **Entregables**
  - Ambiente listo con los datos de prueba establecidos
  - Resultados de las pequeñas pruebas
- **Necesidad**
  - Conocer la tolerancia del sistema de software

Ejecución de la prueba

- Durante esta fase, el equipo se preocupa del Plan de Prueba y los casos de prueba
- Cualquier detalle encontrado en las pruebas, se debe reportar al equipo de testing
- **Actividades**
  - Ejecutar pruebas acordes al plan
  - Documentar los resultados
  - Mapear defectos a casos de pruebas establecidos en RTM
  - Seguir los defectos
- **Entregables**
  - RTM completa
  - Casos de pruebas actualizados
  - Reporte de defectos
- **Necesidad**
  - Paciencia

Cierre del ciclo

- El equipo de testing conoce, discute y analiza los artefactos del testing con el objetivo de identificar estrategias que se pueden implementar a futuro
- **Actividades**
  - Evaluar el ciclo completo basado en atributos como: tiempo, costo, software, objetivos del negocio, otros
  - Preparar métricas de prueba
  - Documentar lo aprendido en el proyecto
- **Entregables**
  - Reporte de cierre de pruebas
  - Métricas de prueba
- **Necesidad**
  - Diplomacia

UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

Departamento de Informática
Universidad Técnica Federico Santa María

Ingeniería de Software

Testing

Hernán Astudillo & Gastón Márquez
Departamento de Informática
Universidad Técnica Federico Santa María