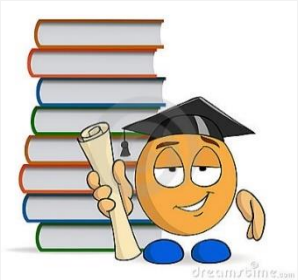

Fundamentos de Testing
Instructora: Ing. Gabriela Bagatello
INCLUIT
Part of harriague+asociados

Presentación



INCLUIT
Part of harriague+asociados

Agenda

Unidad 1: Principios básicos del proceso de prueba

1.1 ¿Porque es necesario el proceso de pruebas?

1.2 ¿Qué es el testing?

1.3 ¿En que consiste el proceso de pruebas?

1.4 Siete principios básicos del proceso de pruebas

1.5 Proceso básico de pruebas

INCLUIT
Part of harriague+asociados

Temas: *1.1 ¿Porque es necesario el proceso de pruebas?*

1.1.1 Contexto de los sistemas de software

- El software se utiliza en distintos tipos de aplicaciones: comerciales, productos de consumo, bancos, etc.
- A muchos nos ha pasado que un software no funcione según lo previsto.
- No todos los sistemas de software llevan el mismo nivel de riesgo.
- La cantidad de pruebas realizadas depende de los riesgos involucrados. Algunos problemas pueden ser bastantes triviales, pero otros pueden ser costosos y perjudiciales.

INCLUIT
Part of harriague+asociados

Temas: *1.1 ¿Porque es necesario el proceso de pruebas?*

1.1.2 Causas de los defectos de software

- Error /Mistake
- Defecto / Bug / Falta
- Fallo /Failure

- **Error:** Acción humana que produce un resultado incorrecto

- **Defect:** Desperfecto en un componente o sistema que puede causar que el componente o sistema falle en desempeñar las funciones requeridas.

- **Failure:** Manifestación física o funcional de un defecto.

"Un error introduce un defecto, un defecto causa un fallo"

INCLUIT
Part of harriague+asociados

Temas: *1.1 ¿Porque es necesario el proceso de pruebas?*

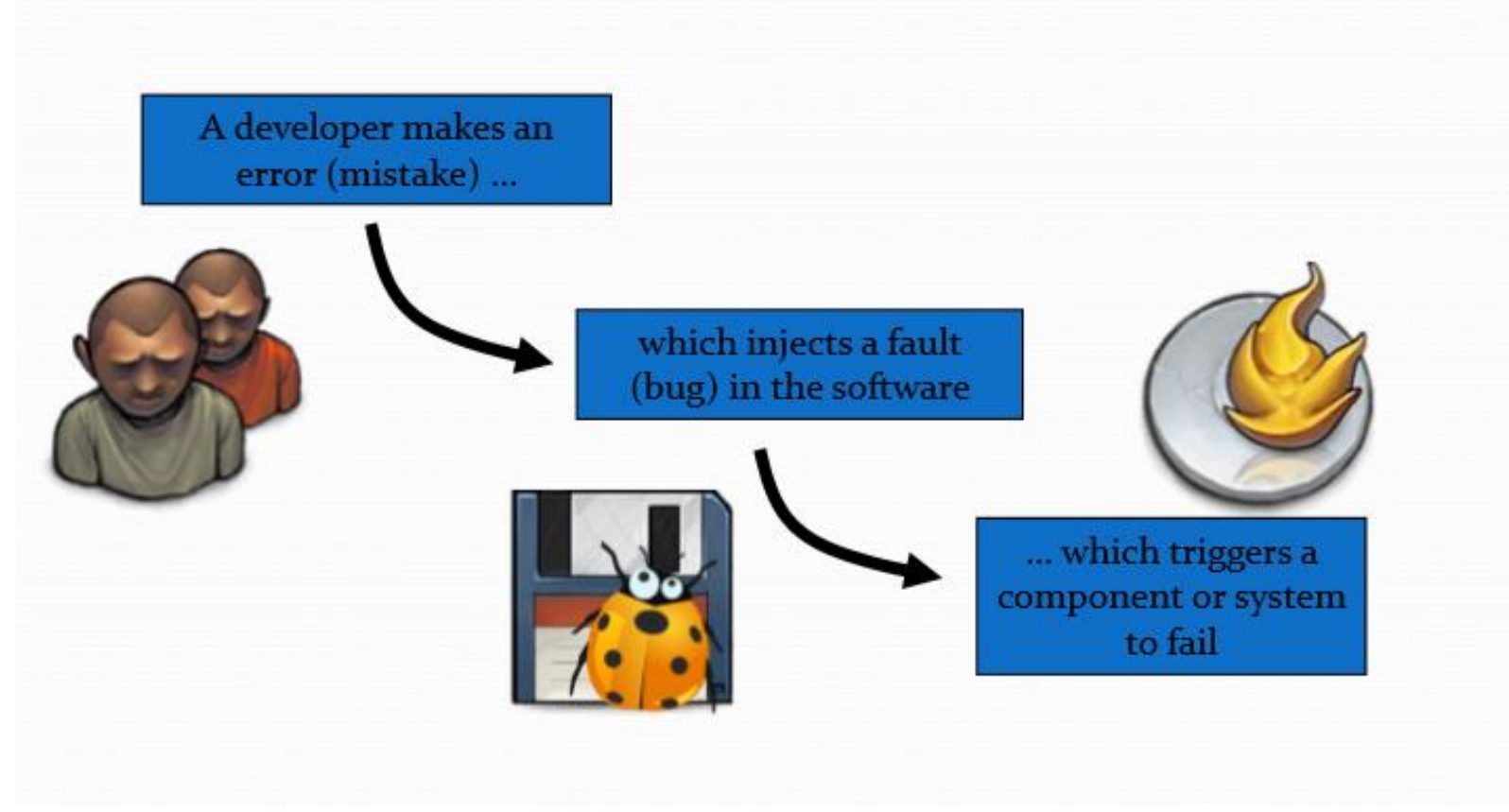
1.1.2 Causas de los defectos de software

- Errores en las especificaciones, diseños e implementación del software
- Errores en el uso del sistema
- Condiciones ambientales
- Daño intencional

"No todos los defectos dan lugar a fallas, algunos pueden permanecer en el software y es posible que nunca nos fijemos en ellos"

INCLUIT
Part of harriague+asociados

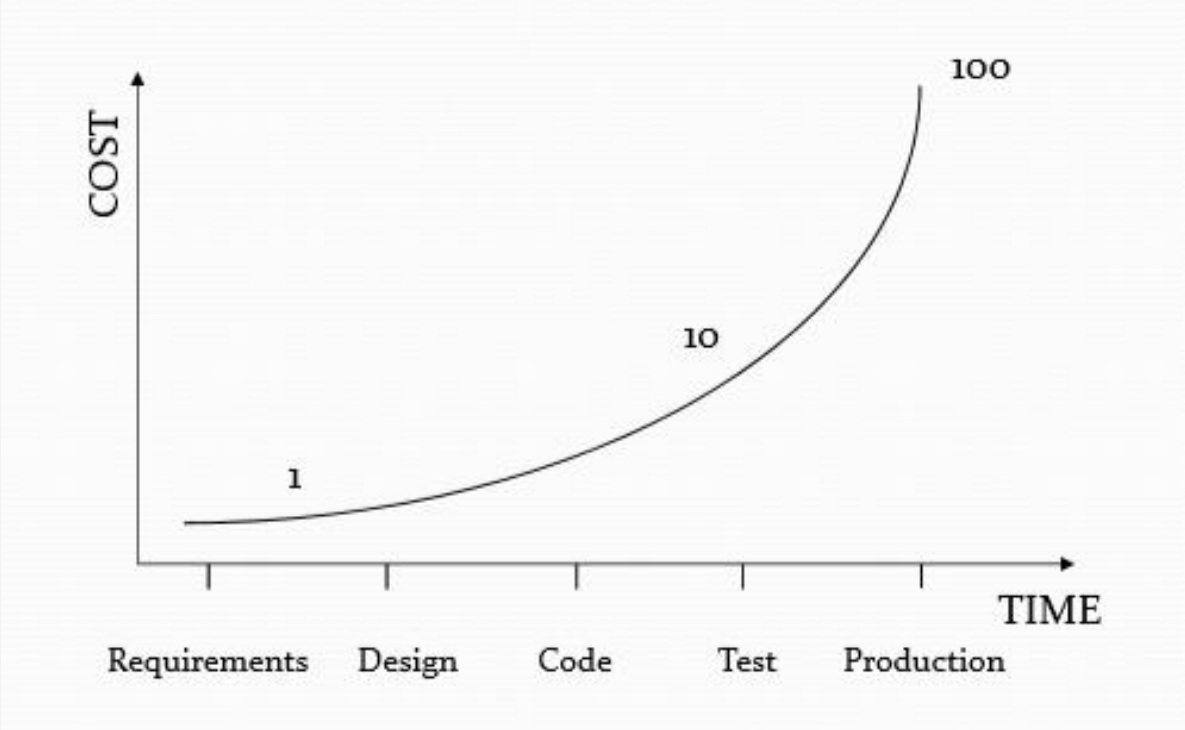
Temas: *Causas de los defectos*



INCLUIT
Part of harriague+asociados

Temas: *Costo de los defectos*

El costo de encontrar y corregir defectos aumenta considerablemente a lo largo del ciclo de vida



INCLUIT
Part of harriague+asociados

1.1.3 Proceso de pruebas y calidad

De acuerdo a la norma ISO/IEC 9126 la calidad del software está constituido por:

- *Atributos funcionales*
  - Funcionalidad, incluye:
    - ✓ Adecuación
    - ✓ Precisión
    - ✓ Interoperabilidad
    - ✓ Seguridad
    - ✓ Cumplimiento de funcionalidad

- *Atributos no funcionales*
  - Fiabilidad
  - Usabilidad
  - Eficiencia
  - Mantenibilidad
  - Portabilidad

INCLUIT
Part of harriague+asociados

1.1.4 ¿Con cuantas pruebas es suficientes?

- Niveles de riesgo:
  - riesgos técnicos
  - de seguridad
  - comerciales

- Limitaciones del proyecto: tiempo y presupuesto

“Cuando este seguro de que el sistema funciona correctamente”

INCLUIT
Part of harriague+asociados

Temas: *1.2 ¿En qué consiste el testing?*

La definición del ISTQB (International Software Testing Qualifications Board) es una organización de certificación de la calidad del software que opera internacionalmente:

- Esta presente en toda las actividades del ciclo de vida
- Un proceso
- Ambos estático y dinámico
- Relacionado con la planificación, preparación y evaluación de los productos de software
- Para determinar si cumplen los requisitos especificados
- Para demostrar que son aptos para el propósito
- Para detectar defectos

INCLUIT
Part of harriague+asociados

Temas: *1.2 ¿En qué consiste el testing?*

- **Casos de prueba** – “Test Case”:
  - Precondiciones
  - Conjunto de valores de entrada
  - Conjunto de resultados esperados
  - Forma en la cual se debe ejecutar el caso de prueba y verificar los resultados
  - Pos-condiciones esperadas

- **Base de prueba** – “Test Base”:

Conjunto de documentos que definen los requisitos de un componente o sistema. Utilizando como fundamento para el desarrollo de los casos de prueba.

INCLUIT
Part of harriague+asociados

Temas: *1.2 ¿En qué consiste el testing?*

Objetivos de las **Base de prueba** :

- Identificar defectos
- Aumentar la confianza en el nivel de calidad
- Facilitar información para la toma de decisiones
- Evitar la aparición de defectos

INCLUIT
Part of harriague+asociados

Resumen:

- Los fallos de software pueden causar importantes prejuicios.
- La calidad del software es la suma de los atributos que se refieren a la capacidad del software de satisfacer un conjunto de requisitos dados.
- El aseguramiento de la calidad constructivo se ocupa de la prevención de defectos.
- El aseguramiento de la calidad analítico se ocupa de detectar y corregir defectos.
- Los atributos de la calidad funcionales y no funcionales definen la calidad total del sistema.
- Cada prueba debe contar con un criterio para la finalización de la prueba (criterio de salida). Al alcanzar el criterio de salida de pruebas concluyen las actividades del proceso de pruebas.
- Los testers buscan fallos en el sistema e informan sobre los mismo (proceso de prueba – “testing”). Los desarrolladores buscan defectos y los corrigen (depuración – “debugging”).

INCLUIT
Part of harriague+asociados

Temas: *1.4. Proceso básico de pruebas*

1.4.1 Planificación y control de pruebas

1.4.2 Análisis y diseño de pruebas

1.4.3 Implementación y ejecución de pruebas

1.4.4 Evaluación de los criterios de salida e informes

1.4.5 Actividades de cierre de pruebas

INCLUIT
Part of harriague+asociados

Temas: *1.4. Proceso básico de pruebas*

1.4.1 Planificación y control de pruebas

- La planificación de las pruebas es la actividad de verificar que entendemos las metas y objetivos de los clientes, las partes interesadas y el proyecto, y los riesgos que se pretender abordar.

- El control de las pruebas es la actividad continua de comparar el progreso real con el plan e informar sobre el estado actual de las pruebas incluidas las desviaciones del plan.

INCLUIT
Part of harriague+asociados

Temas: *1.4. Proceso básico de pruebas*

1.4.2 Análisis y diseño de pruebas

Tareas principales:

- Revisar la base de pruebas
- Evaluar la testabilidad de la base de prueba y de los objetos de prueba
- Identificar y priorizar las condiciones de prueba en base al análisis
- Diseñar y priorizar los casos de prueba de alto nivel
- Identificar los datos de prueba
- Diseñar la configuración del entorno de pruebas e identificar cualquier infraestructura y herramienta necesaria

INCLUIT
Part of harriague+asociados

Temas: *1.4. Proceso básico de pruebas*

1.4.3 Implementación y ejecución de pruebas

Tareas principales:

- Finalizar, implementar y priorizar los casos de prueba
- Desarrollar y priorizar procedimientos de prueba
- Crear juegos de pruebas a partir de los procedimientos de prueba para lograr una ejecución de pruebas eficientes
- Verificar que el entorno de pruebas ha sido correctamente configurado
- Verificar y actualizar una trazabilidad bidireccional entre la base de pruebas y los casos de pruebas

INCLUIT
Part of harriague+asociados

Temas: *1.4. Proceso básico de pruebas*

- Ejecutar los procedimientos de prueba manualmente o recurriendo a herramientas de ejecución de pruebas
- Registrar los resultados de la ejecución de las pruebas y registrar las identidades y las versiones del software probado
- Comparar los resultados reales con los resultados esperados
- Reportar las discrepancias en forma de incidencias y analizarlas con vistas a establecer sus causas
- Repetir las actividades de pruebas como resultado de una medida adoptada para cada discrepancia

INCLUIT
Part of harriague+asociados

Temas: *1.4. Proceso básico de pruebas*

1.4.4 Evaluación de los criterios de salida e informes

Tareas principales:

- Comprobar los registros de pruebas con los criterios de salida previstos en la planificación
- Evaluar si se requieren más pruebas o si deberían modificarse los criterios de salida
- Elaborar un resumen de las pruebas para las partes interesadas

INCLUIT
Part of harriague+asociados

Temas: *1.4. Proceso básico de pruebas*

1.4.5 Actividades de cierre de pruebas

Tareas principales:

- Comprobar cuáles de los productos entregables previstos han sido efectivamente entregados
- Cerrar los informes de incidencias o aportar modificaciones a aquellos que siguen abiertos
- Documentar la aceptación del sistema
- Finalizar y archivar los productos de soporte de prueba, el entorno de pruebas y la infraestructura de pruebas para su posterior uso
- Entregar los productos de soporte de prueba a la organización de mantenimiento
- Analizar las lecciones aprendidas para determinar los cambios necesarios en futuras versiones y proyectos
- Utilizar la información recopilada para mejorar la madurez de las pruebas

INCLUIT
Part of harriague+asociados

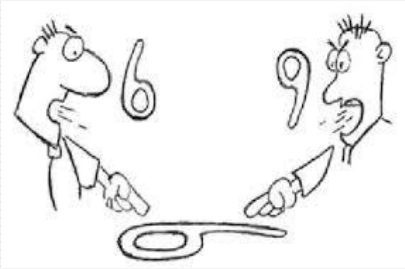
Resumen:

- **Planificación de pruebas** “Test Planning” abarca actividades como la definición de la estrategia de pruebas para todas las fases, así como la planificación de los recursos (tiempo, personal, maquinas).
- **Diseño de pruebas** (Especificación) abarca el diseño de casos de prueba y sus resultados esperados.
- **Ejecución de pruebas** abarca la definición de los datos de prueba, la ejecución de las pruebas y la comparación de resultados.
- **Evaluación de pruebas y generación de informes** abarca la evaluación del criterio de salida y el registro de los resultados de pruebas en forma escrita.
- **Control de pruebas** consiste en el control de las actividades que cubren todas las fases del proceso de pruebas.

INCLUIT
Part of harriague+asociados

Temas: *1.5 La psicología de las pruebas*

Desarrollador:



- Implementa requisitos
- Desarrolla estructuras
- Desarrolla el software
- Crea un producto

Tester:

- Planifica las actividades de prueba
- Diseña casos de prueba
- Su única preocupación es encontrar defectos
- Encontrar errores producidos por un desarrollador es su éxito

INCLUIT
Part of harriague+asociados

Temas: *1.5 La psicología de las pruebas*

- Percepción:
  - ❑ La actividad del desarrollador es constructiva
  - ❑ La actividad del tester es destructiva

- Debugging versus Testing
  - ✓ Debugging tiene por objetivo remover los defectos
  - ✓ Testing demuestra fallos causados por los defectos

"Las pruebas también constituyen una actividad constructiva,

su propósito es la eliminación de defectos en un producto "

INCLUIT
Part of harriague+asociados

Niveles de independencia:

- Pruebas diseñadas por las mismas personas que escribieron el software probado (bajo nivel de independencia)
- Pruebas diseñadas por terceros (ej: miembros del equipo de desarrollo)
- Pruebas diseñadas por una persona procedente de otro grupo de la organización (ej: un equipo de prueba independiente) o especialistas de pruebas (ej: especialistas en pruebas de usabilidad o rendimiento)
- Pruebas diseñadas por personas de otra organización o empresa (externalización o certificación por parte de un organismo externo)

INCLUIT
Part of harriague+asociados

Formas de mejorar la comunicación entre los testers y los demás:

- Empezar juntos en lugar de enfrentados
- Comunicar los hallazgos del producto de una manera neutral y centrada de los hechos
- Intentar comprender como se siente la otra persona y por qué reacciona de esa manera
- Confirmar que la otra persona ha entendido lo que usted ha dicho y viceversa

INCLUIT
Part of harriague+asociados

Agenda

Unidad 2: Pruebas durante todo el ciclo de vida del software

2.1 Modelo de desarrollo de software

2.2 Niveles de pruebas

2.3 Tipos de pruebas

2.4 Pruebas de mantenimiento

INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

2.1.1 Principios de todos los modelos

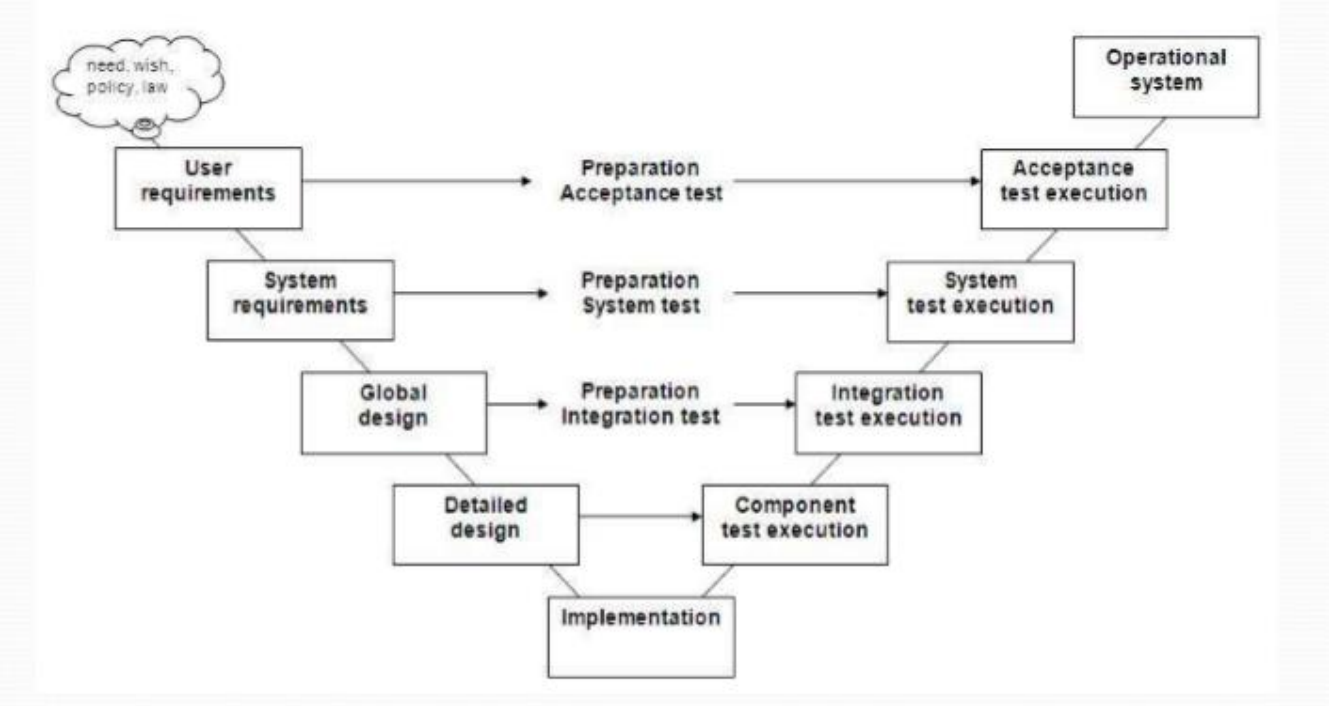
- Cada actividad de desarrollo debe ser probada
- Cada nivel de prueba debería ser probado de forma especifica
- El proceso de pruebas comienza con mucha antelación a la ejecución de pruebas
- Tan pronto como el desarrollo comienza puede comenzar la preparación de las pruebas correspondientes

INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

Modelo V o Modelo de Desarrollo secuencial

Cada nivel de desarrollo tiene su correspondiente nivel de pruebas



INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

Niveles del Modelo V:

- Pruebas de componentes (unidades)
- Pruebas de integración
- Pruebas de sistemas
- Pruebas de aceptación

INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

- Prueba de componentes:
  - Busca defectos en componentes de software.
  - Verifica que se puedan probar por separado

- Pruebas de integración:
  - Prueba interfaces entre componentes
  - Interacciones con diferentes partes del sistema

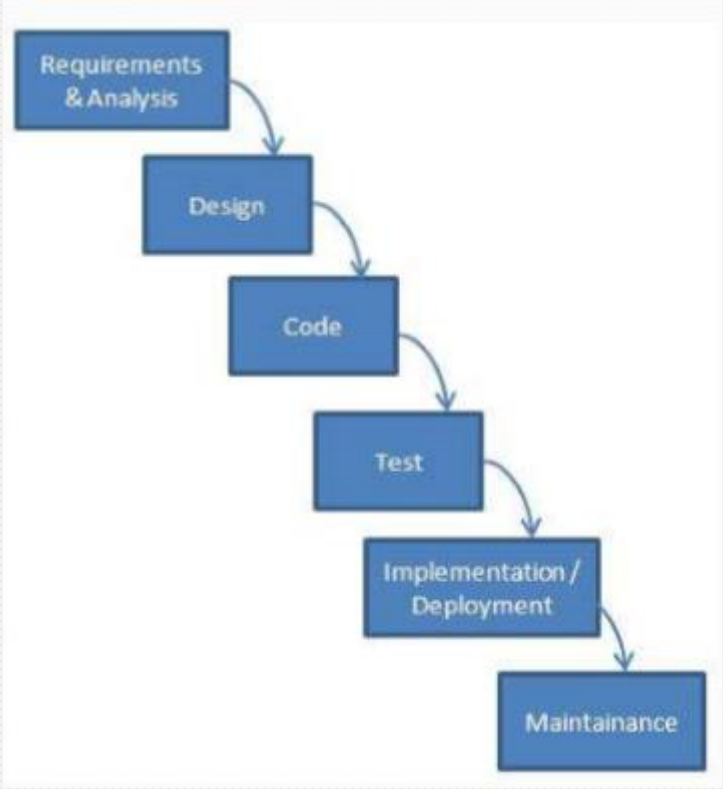
- Pruebas de sistemas:
  - Verificar los requisitos especificados

- Pruebas de aceptación
  - Pruebas de validación con respecto a las necesidades de los usuarios, los requisitos y los procesos de negocio realizados para determinar si se acepta o no el sistema

INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

Modelo de Desarrollo secuencial o en cascada



INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

Características de los modelos iterativos:

- Cada iteración contribuye con un característica adicional del sistema a desarrollar
- Cada iteración puede ser probada por separado
- Las pruebas de regresión y la automatización de pruebas son elementos de relevancia
- En cada iteración, la verificación (relación con el nivel precedente) y la validación (grado de corrección del producto dentro del nivel actual) se pueden efectuar por separado

INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

2.1.2 Modelos de desarrollo iterativo-incremental

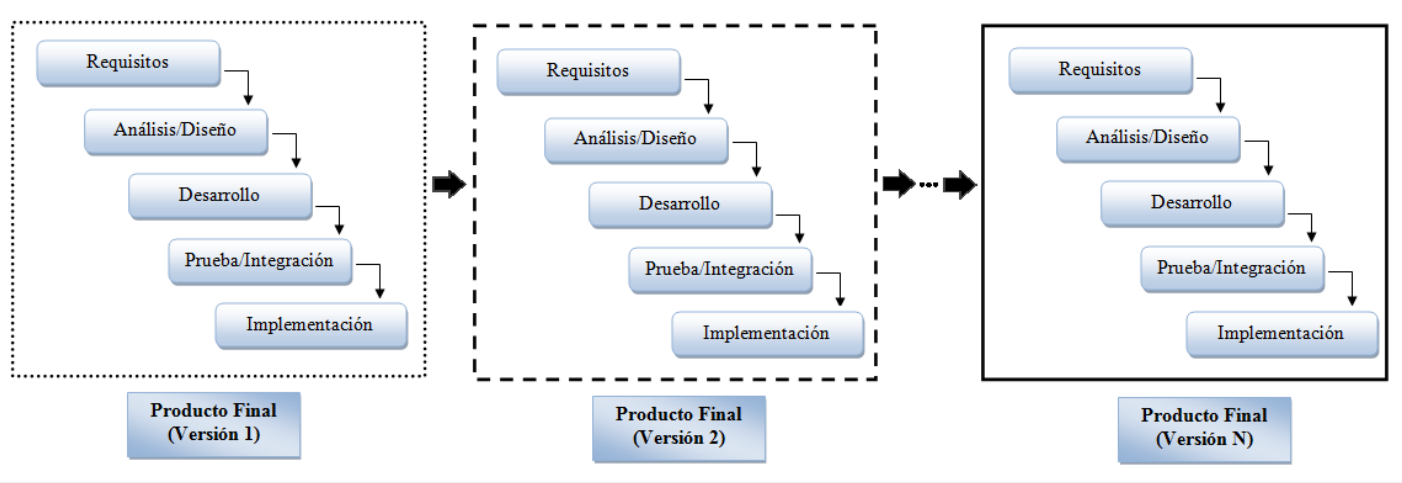
- Las actividades: definición de requisitos, diseño, desarrollo y pruebas se segmentan en pasos reducidos y se ejecutan de forma continua.
- Se debe alcanzar el consentimiento del cliente tras cada iteración con el objeto de modificar el rumbo del proyecto si fuese necesario.

INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

Modelo de desarrollo Iterativo

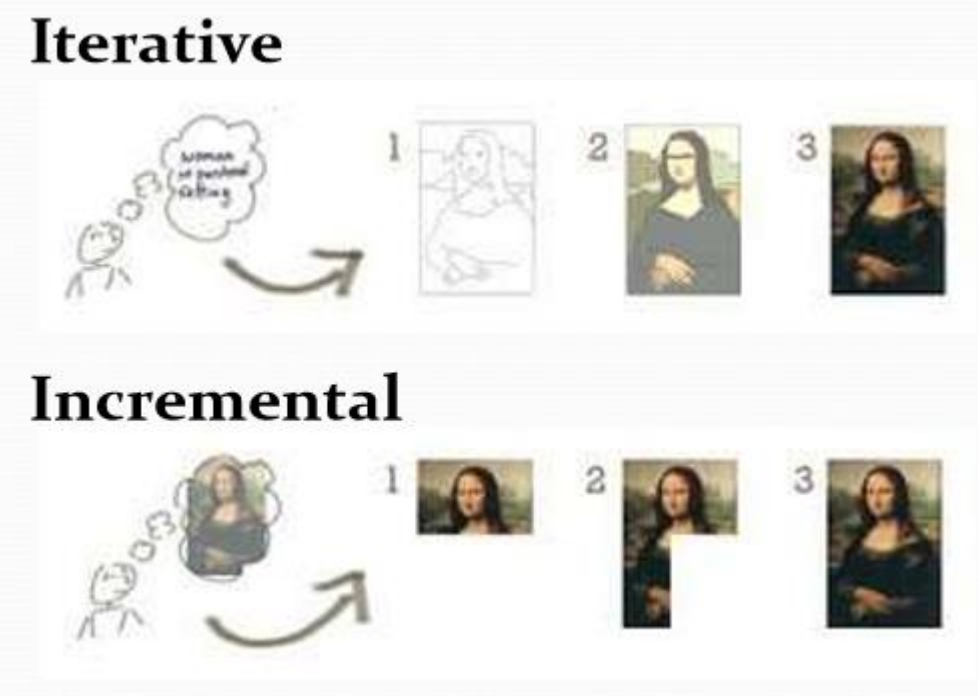
- Las entregas se dividen en incrementos
- Cada incremento añade una nueva funcionalidad
- El incremento producido por una iteración puede ser probado en varios niveles como parte de su desarrollo



INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

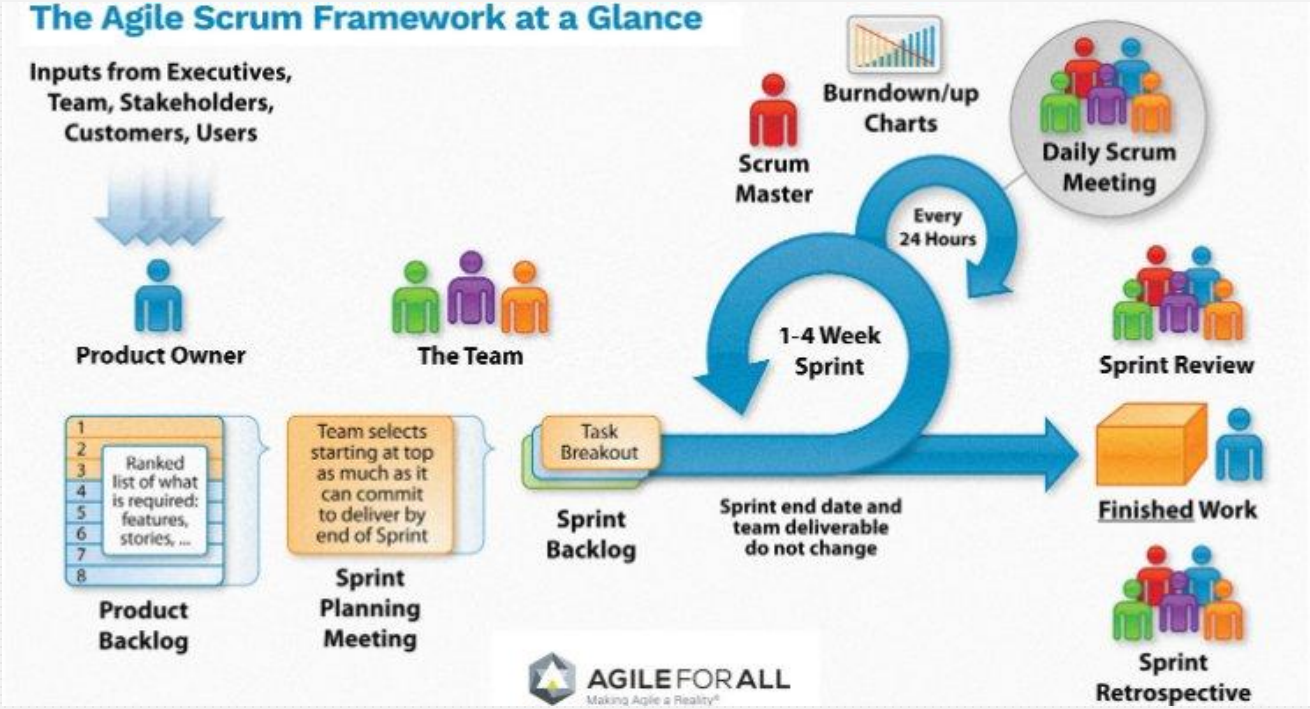
Iterativo vs Incremental



INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

Desarrollos Agile – Scrum



INCLUIT
Part of harriague+asociados

Temas: *2.1 Modelo de desarrollo de software*

2.1.1 Pruebas en un Modelo de Ciclo de Vida

- Para cada actividad de desarrollo existe una actividad de prueba
- Cada nivel de prueba tiene objetivos de prueba especifico
- Los procesos de análisis y diseño de las pruebas para un nivel de prueba deben inicializarse durante la actividad de desarrollo correspondiente
- Los testers deben iniciar su participación en la revisión de documentos en cuanto haya borradores disponibles

INCLUIT
Part of harriague+asociados

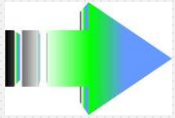
Temas: *2.2 Niveles de prueba*

2.2.1 Pruebas de componentes

- Pruebas de cada componente tras su construcción
- Los componentes son referidos como módulos, clases o unidades
- Alcance:
  - ✓ Solo se prueban componentes individuales
  - ✓ Cada componente es probado de forma independiente

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*



Pruebas de componentes

Base de pruebas:

- Especificación del componentes
- Diseño de detalle
- Código

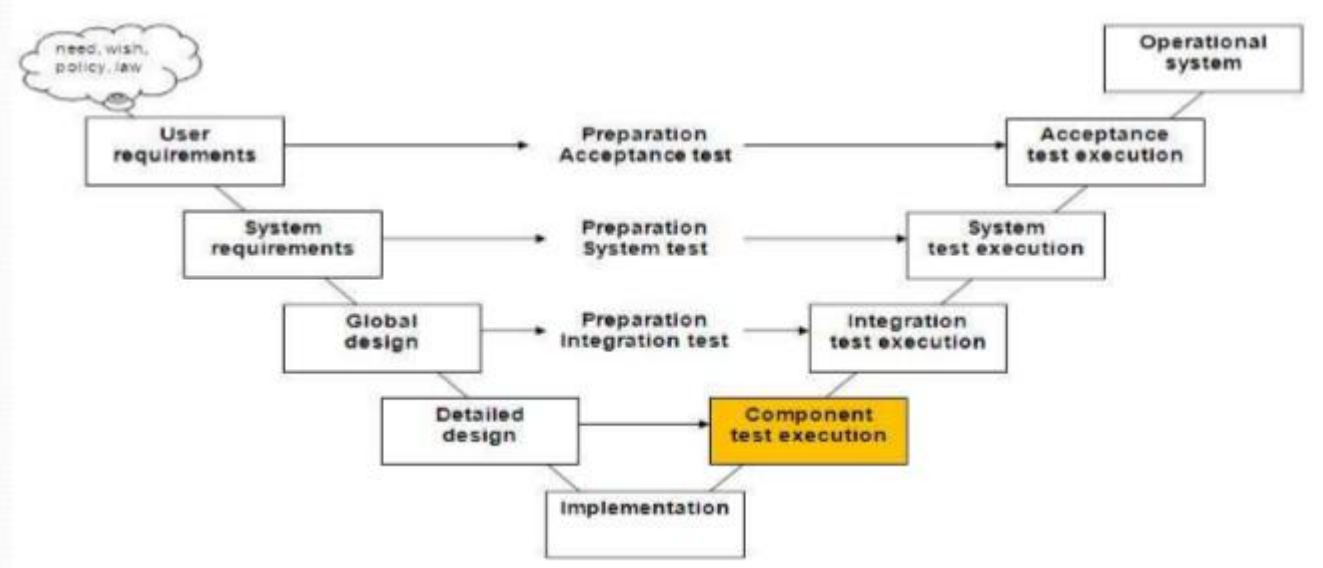
Objetos de prueba típicos:

- Componentes
- Programas
- Conversión de datos/programas de migración

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

2.2.1 Pruebas de componentes



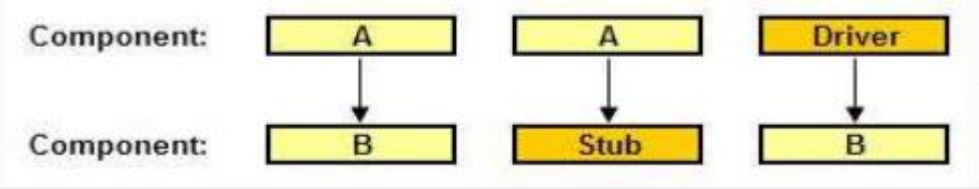
"La ejecución de pruebas de componentes requiere frecuentemente controladores "Drivers" y "Stubs"

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

Drivers y Stubs

- Un controlador “Driver” procesa la interfaz de un componente
- Estos simulan datos de entrada, registran datos de salida y aportan un arnés de pruebas (“test harness”)
- Un Stub reemplaza o simula un componente que aun no se encuentra disponible o que no es parte del objeto de prueba (“test object”)
- Para programar controladores y stubs:
    - Se debe contar con conocimientos de programación
    - Se necesita disponer de código fuente
    - Podrán ser necesarias herramientas especiales



INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

Pruebas de componentes: Desarrollo guiado por pruebas

- Se basa en:
  - ✓ Juego de casos de prueba (test case suite)
  - ✓ Preparación de ciclo de prueba (test cycle)
  - ✓ Pruebas automatizadas haciendo uso de herramientas de pruebas (test tool)

- Desarrollo de acuerdo con casos de prueba.
  - ➢ Preparación de versiones tempranas del componente para probar
  - ➢ Ejecución automática de pruebas
  - ➢ Corrección de defectos en versiones subsiguientes

INCLUIT
Part of harriague+asociados

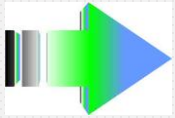
Temas: *2.2 Niveles de prueba*

2.2.2 Pruebas de integración

- Comprueba la interacción entre elementos software (componentes) tras la integración del sistema
- Es la actividad donde se combinan componentes individuales en subsistemas mas amplios
- Cada componente ya ha sido probado en su funcionalidad interna, ahora se comprueban las funciones externas.

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*



2.2.2 Pruebas de integración

Base de pruebas:

- ❑ Diseño de software y sistema
- ❑ Arquitectura
- ❑ Flujo de trabajo
- ❑ Casos de usos

Objetos de prueba típicos:

- Implementación de BD de subsistemas
- Infraestructura
- Interfaces

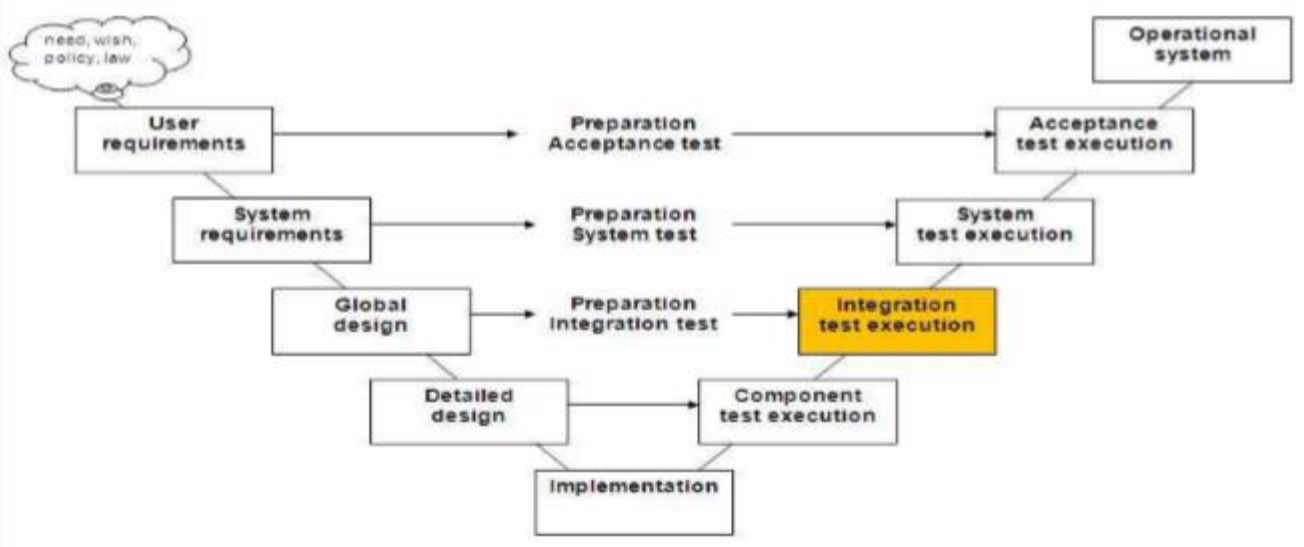
Configuración del sistema:

- Datos de configuración

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

2.2.2 Pruebas de integración



Pueden ser ejecutadas por desarrolladores, testers o ambos

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

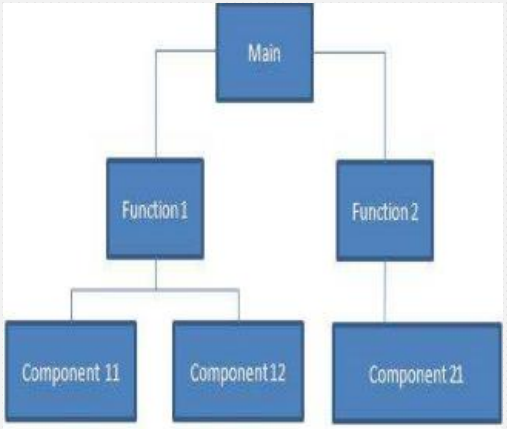
Estrategias para las pruebas de integración:

- El enfoque incremental es un elemento común a la mayoría de las estrategias (excepción: estrategia “Big Bang”)
- Estrategias ascendente (“Bottom up”) y descendente (“Top-Down”) son las utilizadas con frecuencia
- La elección de la estrategia debe considerar aspectos relativos a la eficiencia de las pruebas.
- La estrategia de integración determina la magnitud del esfuerzo requerido para las pruebas

INCLUIT
Part of harriague+asociados

Temas: *Big-Bang*

- Todos los componentes o sistemas se integran simultáneamente, después de lo cual todo se prueba en conjunto
- Ventaja: todo esta terminado antes de que comience la prueba de integración. No hay necesidad de simular partes
- Desventaja: en general es demorado y difícil de localiza la causa de los fallos

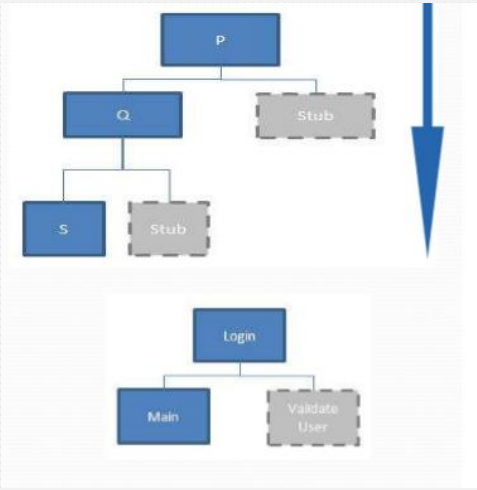


INCLUIT
Part of harriague+asociados

Temas: *Incremental Testing*

- Top-down :

✓ Una forma de prueba de integración incremental que comienza con el módulo superior en la jerarquía de ejecución.
✓ Componentes o sistemas son sustituidos por stubs.
✓ Ventaja: una versión esquelética de un programa puede existir temprano y permite demostraciones.

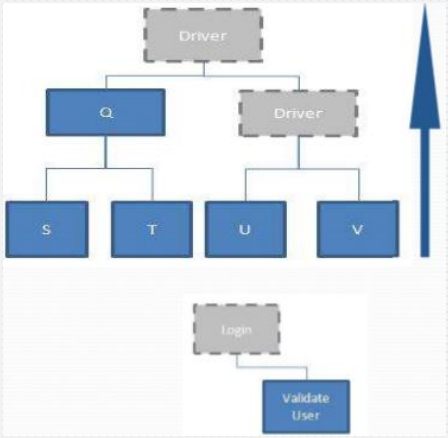


INCLUIT
Part of harriague+asociados

Temas: *Incremental Testing*

- Botton-up:

✓ Una forma de prueba de integración incremental donde los módulos de nivel más bajo de la jerarquía son probados primero
✓ Desventaja: el programa en su conjunto no existe hasta que se agregue el último módulo



INCLUIT
Part of harriague+asociados

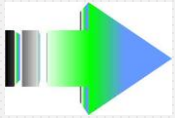
Temas: *2.2 Niveles de prueba*

2.2.3 Pruebas de sistema

- Pruebas del sistema software integrado con el objeto de comprobar el cumplimiento de requisitos especificados.
- Las pruebas de sistema se refieren a requisitos funcionales y no funcionales.

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*



2.2.3 Pruebas de sistema

Base de pruebas:

- ❑ Especificación de requisitos del sistema y software
- ❑ Casos de Uso
- ❑ Especificaciones funcionales
- ❑ Informes de análisis de riesgos

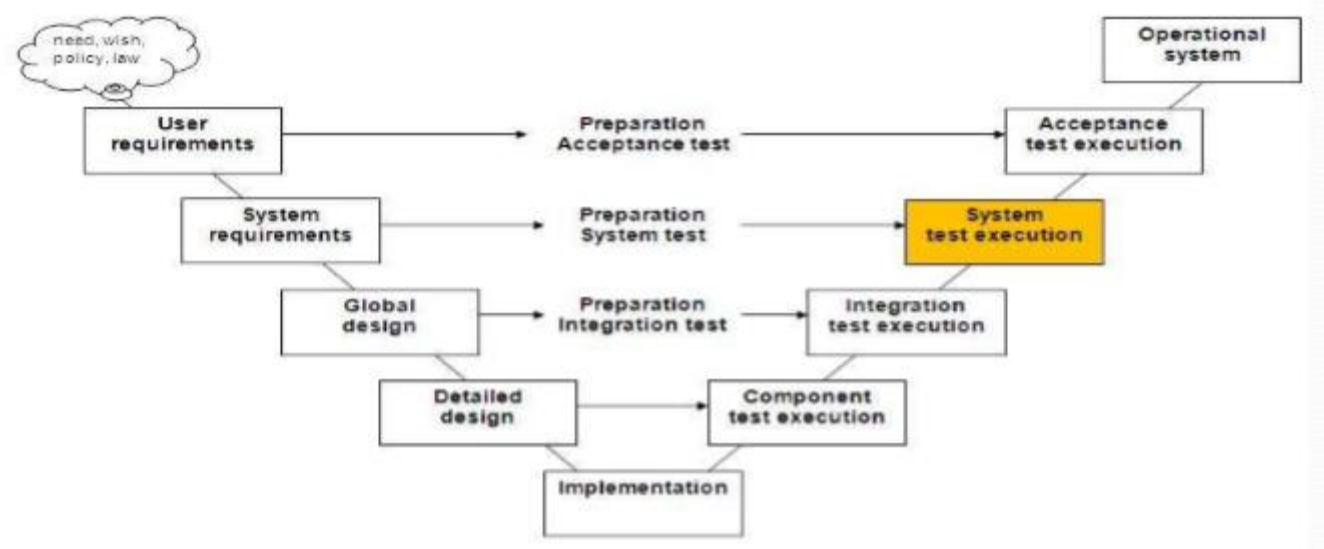
Objetos de prueba típicos:

- Manuales de sistema, usuario y funcionamiento
- Configuración del sistema

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

2.2.3 Pruebas de sistema



INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

2.2.3 Pruebas de sistema

Comprobar que la funcionalidad implementada expone las características requeridas:

- Adecuación
- Exactitud
- Interoperabilidad
- Seguridad
- Cumplimiento de la seguridad

INCLUIT
Part of harriague+asociados

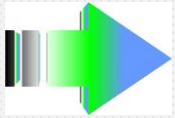
Temas: *2.2 Niveles de prueba*

2.2.4 Pruebas de aceptación

- Son pruebas formales llevadas a cabo con el objeto de verificar la conformidad del sistema con los requisitos.
- El objetivo es aportar justificación a la confianza en el sistema para que puede ser aceptado por el cliente (IEEE 610)

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*



2.2.4 Pruebas de aceptación

Base de pruebas:

- ❑ Requisitos del usuario
- ❑ Requisitos del sistema
- ❑ Casos de uso
- ❑ Procesos de negocio
- ❑ Informes de análisis de riesgos

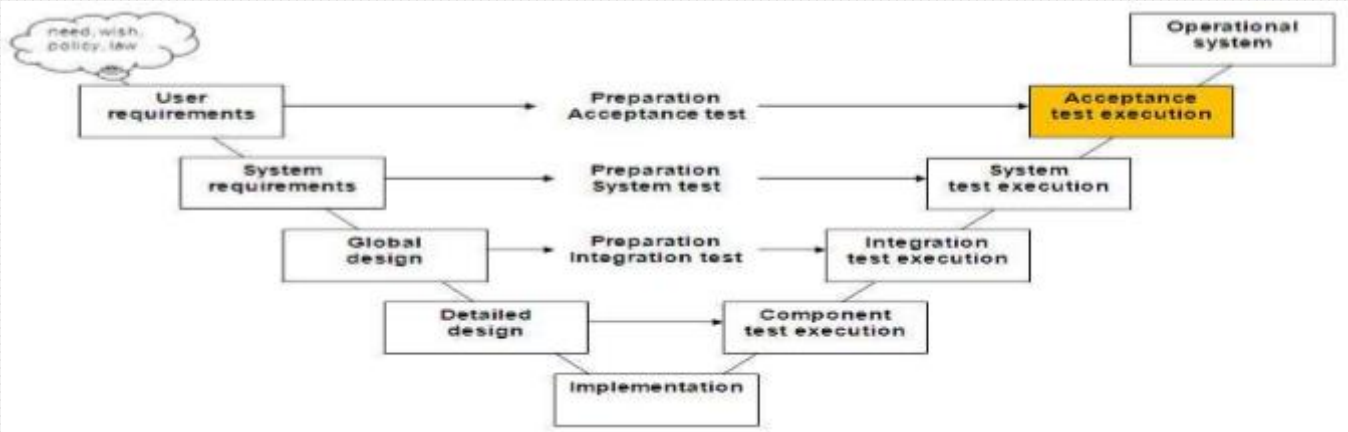
Objetos de prueba típicos:

- Procesos de negocio en sistema integrado
- Procesos operativos y de mantenimiento
- Procedimientos de usuario
- Formularios
- Informes

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

2.2.4 Pruebas de aceptación



Son las prueba de sistema por parte del cliente

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

Las Pruebas de aceptación pueden adoptar distintas formas:

- ❑ Pruebas de aceptación de usuario
- ❑ Pruebas operativas (de aceptación)
  - ✓ Pruebas de backup/restauración
  - ✓ Recuperación de desastres
  - ✓ Gestión de usuarios
  - ✓ Tareas de mantenimiento
  - ✓ Carga de datos y tareas de migración
- ❑ Pruebas de aceptación contractual y normativa

INCLUIT
Part of harriague+asociados

Temas: *2.2 Niveles de prueba*

- Pruebas Alfa:

Se llevan a cabo en el emplazamiento de la organización de desarrollo, pero no las realiza el equipo de desarrollo.

- Pruebas Beta:

O pruebas de campo las realizan los clientes o clientes potenciales en sus propias instalaciones.

INCLUIT
Part of harriague+asociados

Temas: *2.3 Tipos de prueba*

- En cada nivel de prueba los objetivos de las pruebas tienen un foco diferente
- Se aplicaran distintos tipos de pruebas
  - Funcionales -> probar la función
  - No funcionales -> probar las características del producto
  - Estructurales -> Probar la estructura/arquitectura del software
  - De confirmación/regresión -> probar después de cambios

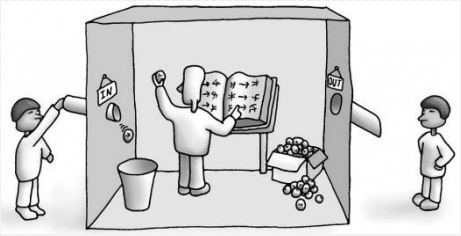
INCLUIT
Part of harriague+asociados

Temas: *2.3 Tipos de prueba*

2.3.1 Pruebas funcionales

Objetivo: La función del objeto de prueba

- La funcionalidad puede ser vinculada a los datos de entrada y salida de un objeto de prueba
- Los métodos de caja negra (“Black box”) se utilizan en el diseño de caso de prueba relevantes
- Proceso de pruebas para verificar los requisitos funcionales
- Se pueden llevar a cabo en todos los niveles de prueba



INCLUIT
Part of harriague+asociados

Temas: *2.3 Tipos de prueba*

2.3.2 Pruebas no funcionales

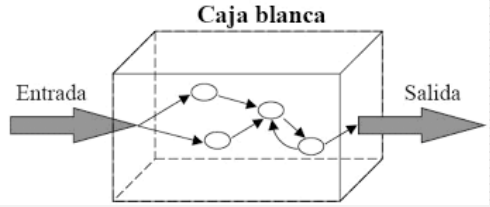
- ✓ Prueba de carga – load test
- ✓ Prueba de rendimiento – perfomance test
- ✓ Prueba de volumen – volumen test
- ✓ Prueba de estrés – stress test
- ✓ Prueba de estabilidad – stability test
- ✓ Prueba de robustez – test for robustness
- ✓ Pruebas de cumplimiento
- ✓ Pruebas de usabilidad

INCLUIT
Part of harriague+asociados

Temas: *2.3 Tipos de prueba*

2.3.3 Pruebas estructurales

- Análisis de la estructura de un objeto de prueba (enfoque: caja blanca)
- La finalidad es medir el grado en el cual la estructura del objeto ha sido cubierto por los casos de prueba
- Son posibles en todos los niveles de pruebas
- Todos los elementos estructurales identificados deberán estar cubiertos por casos de pruebas



INCLUIT
Part of harriague+asociados

Temas: *2.3 Tipos de prueba*

2.3.4 Pruebas asociadas a cambios: Repetición de pruebas y pruebas de regresión

- Repetir una prueba de funcionalidad que ha sido verificada previamente se denomina prueba de regresión
- Es necesario volver a probar “retest” zonas adyacentes debido a efectos colaterales
- Pueden ser realizadas en todos los niveles
- Pruebas de confirmación: repetición de pruebas luego de una corrección
- Pruebas de regresión: pruebas para descubrir nuevos defectos introducidos en funcionalidad previamente sin fallos

INCLUIT
Part of harriague+asociados

Temas: *2.4 Pruebas de mantenimiento*

- El mantenimiento de software cubre dos campos:

  - Mantenimiento: corrección de errores, que han sido parte de la versión inicial del software
  - Extensión/Ampliación: adaptaciones como resultado de un cambio del entorno o nuevos requisitos del cliente.

INCLUIT
Part of harriague+asociados

Temas: *2.4 Pruebas de mantenimiento*

- Alcance:
  - ✓ La ampliación de la funcionalidad requiere nuevos casos de prueba
  - ✓ La corrección de errores requiere la repetición de las pruebas (re-test)
  - ✓ La migración a otra plataforma requiere pruebas operativas
  - ✓ Adicionalmente, son necesarias pruebas de regresión intensivas
  - ✓ El alcance de las pruebas depende del impacto del cambio

INCLUIT
Part of harriague+asociados

INCLUIT
Part of harriague+asociados

¡Muchas Gracias!

INCLUIT
Part of harriague+asociados