

Capítulo 2

Las Pruebas Durante Todo el Ciclo de Vida de Desarrollo de Software



Capítulo 2

- 2.1 Modelos de ciclo de vida de desarrollo de software
- 2.2 Niveles de prueba
- 2.3 Tipos de prueba
- 2.4 Pruebas de mantenimiento





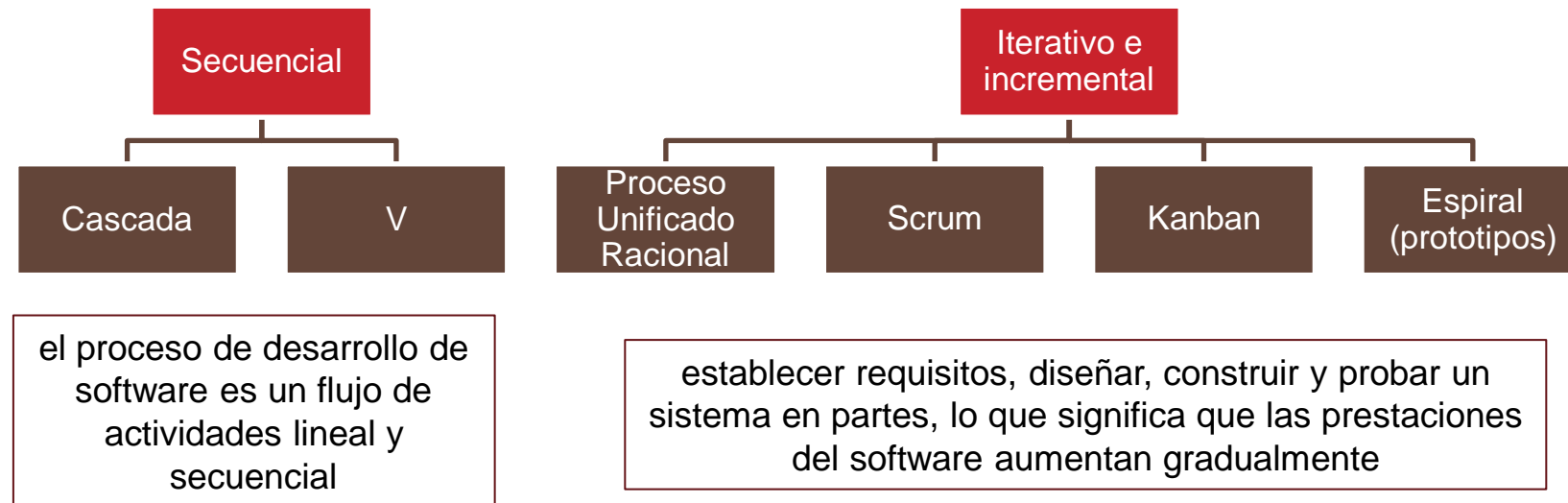
2.1 Modelos de ciclo de vida de desarrollo de software

Modelo de ciclo de vida de desarrollo de software

- Descripción de los tipos de actividades que se realizan en cada etapa de un proyecto de desarrollo de software, y cómo las actividades se relacionan entre sí de manera lógica y cronológica.

2.1.1 Desarrollo de Software y Pruebas de Software

Categorías del CVDS



2.1.1 Desarrollo de Software y Pruebas de Software

Modelo de Cascada (secuencial)

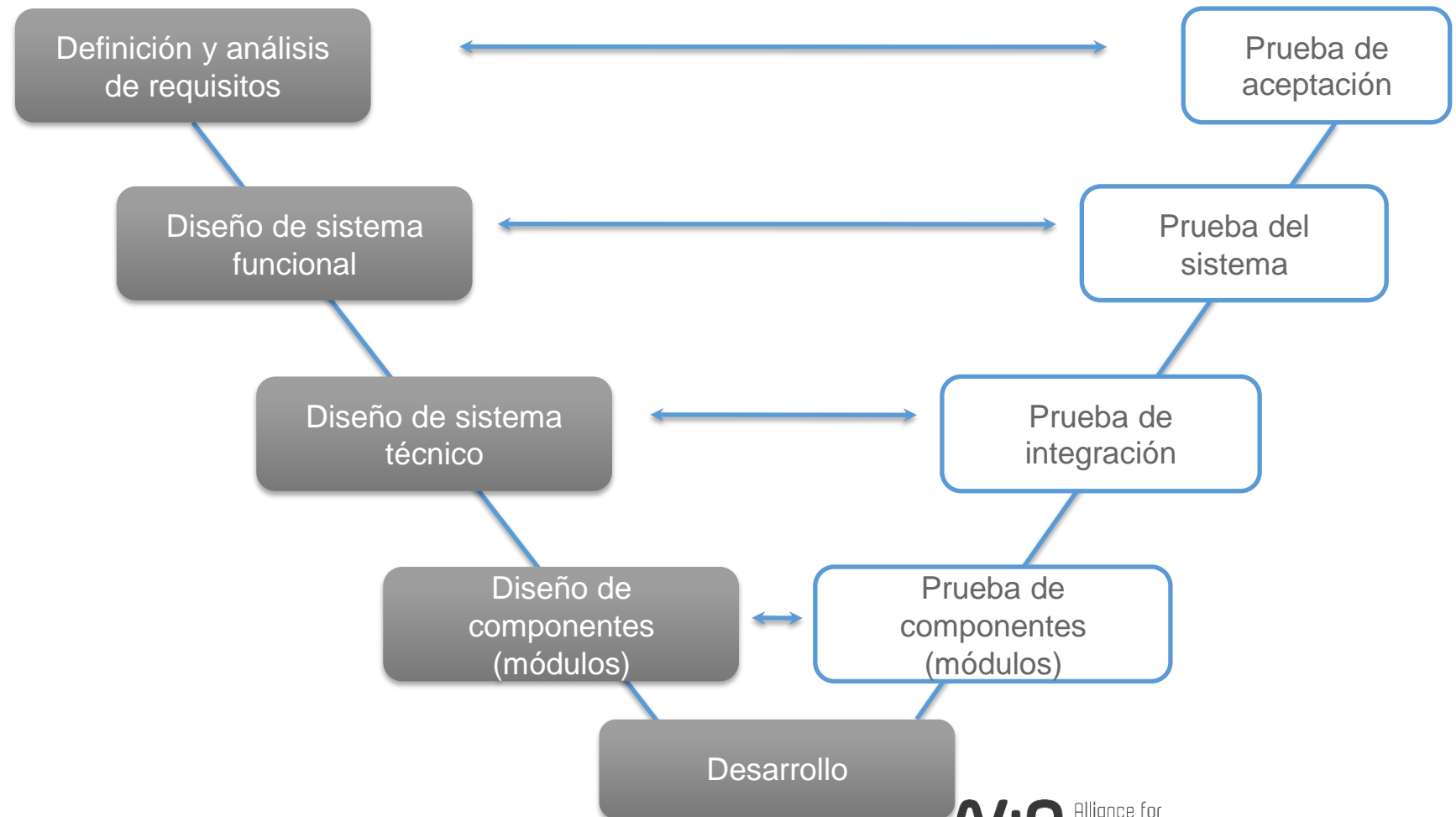


*Las actividades de prueba están al final del proyecto

2.1.1 Desarrollo de Software y Pruebas de Software

Modelo « V » (secuencial)

*Este modelo apunta al principio “*Pruebas Tempranas*”, cada nivel cuenta con sus actividades de prueba definidas

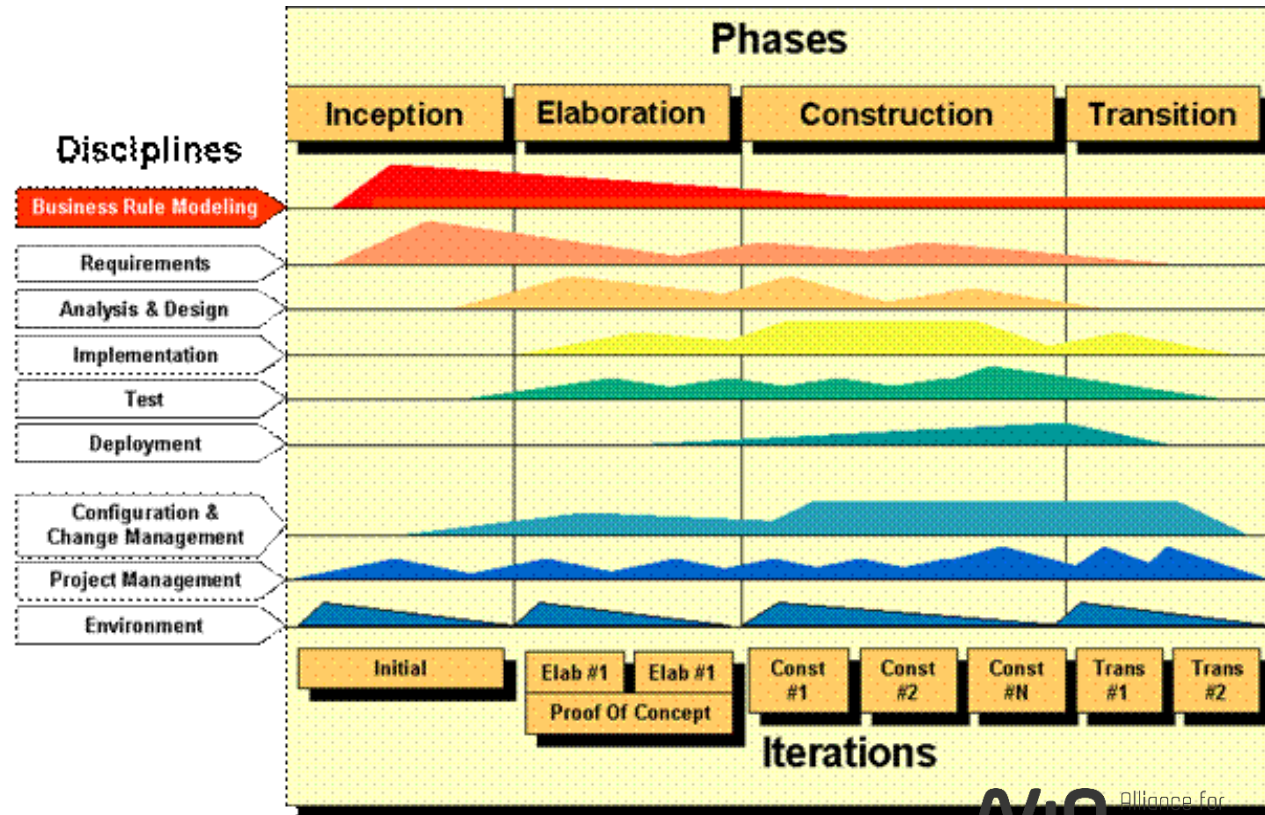


2.1.1 Desarrollo de Software y Pruebas de Software

- 9 disciplinas, cada una cubierta por 4 fases
- Las iteraciones tienen lugar dentro de las fases
- Iteraciones largas con piezas grandes de software

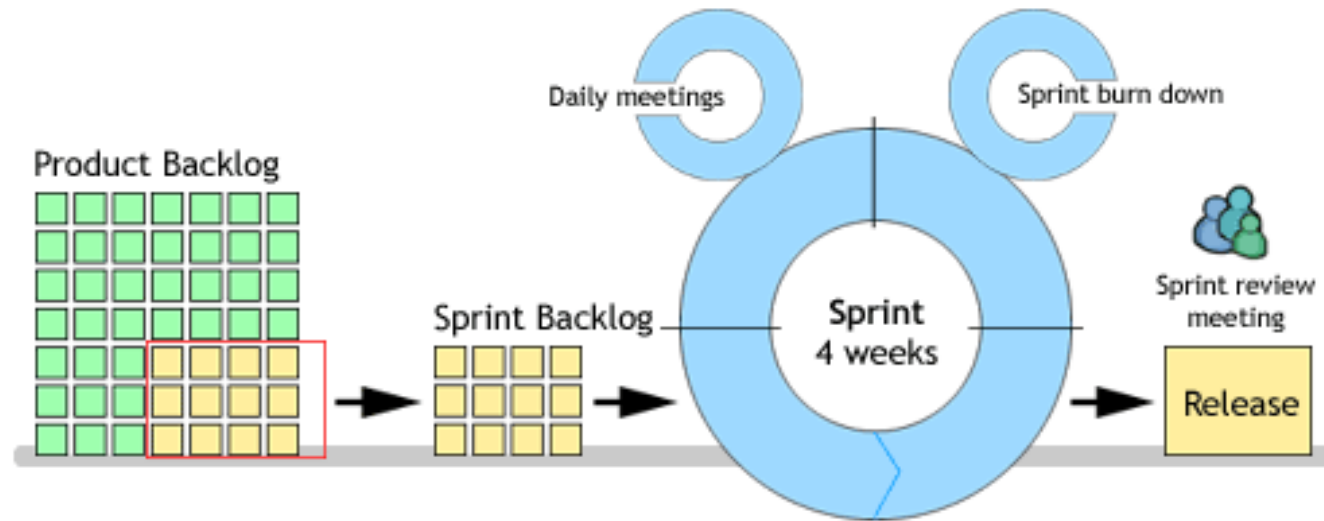
RPU en inglés (rational unified process)

•PRU (Proceso Racional Unificado), de la IBM, iterativo



2.1.1 Desarrollo de Software y Pruebas de Software

- Ágil (modelo SCRUM), iterativo



*Las actividades de prueba están incluidas en los diferentes pasos

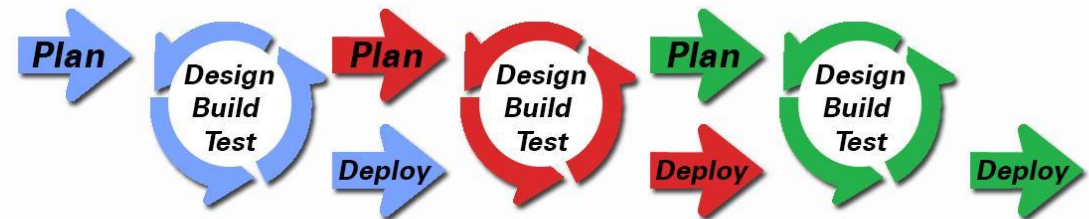
2.1.1 Desarrollo de Software y Pruebas de Software

Ejemplo:
Cascada vs Ágil

Waterfall
(Plan Driven)



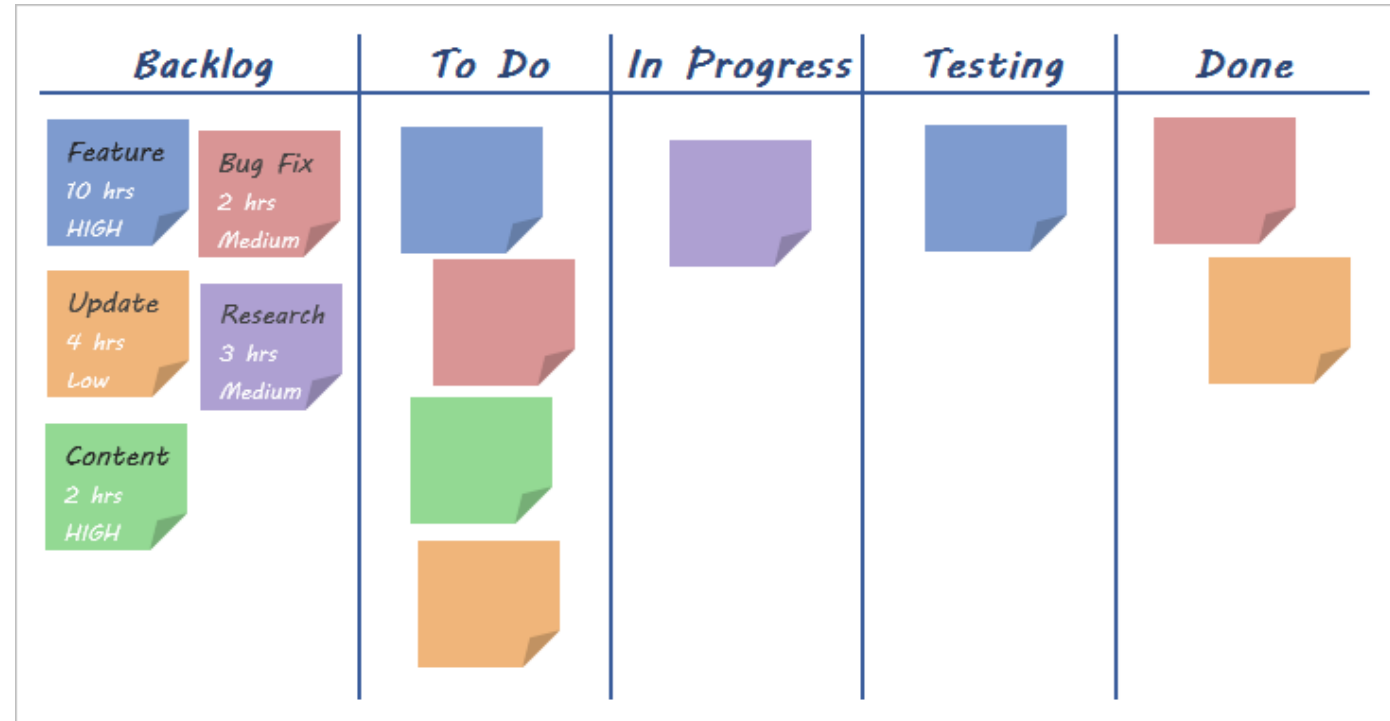
Agile
(Value Driven)



Project Timeline

2.1.1 Desarrollo de Software y Pruebas de Software

- Kanban (iterativo)



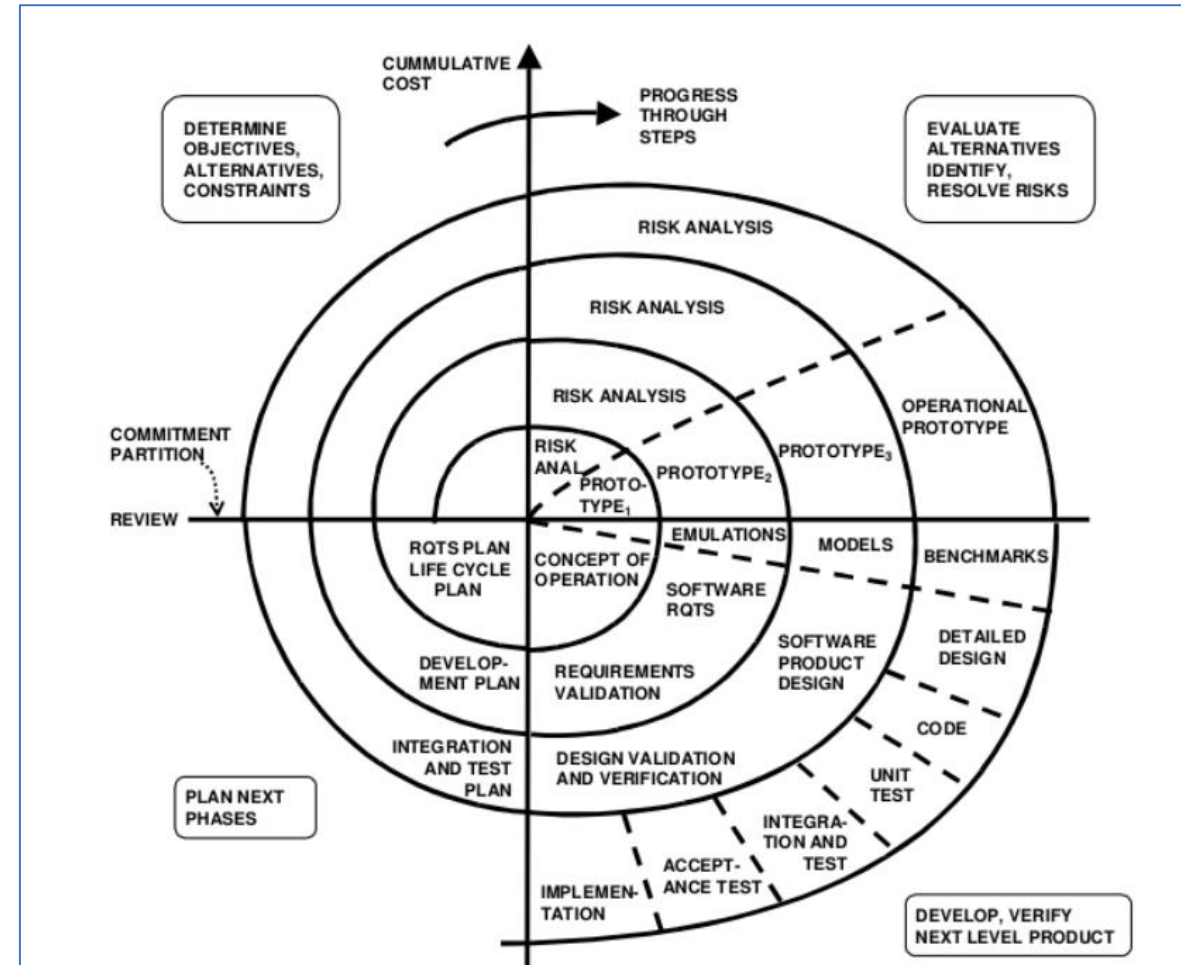
Implementado con o sin iteraciones de longitud fija, que pueden ofrecer una mejora o prestación única al finalizar, o pueden agrupar prestaciones para entregarlas a la vez

2.1.1 Desarrollo de Software y Pruebas de Software

Espiral o creación de prototipos (iterativo)

- Implica crear incrementos experimentales

<https://xbsoftware.com/blog/software-development-life-cycle-spiral-model/>





2.1.2 Modelos de ciclo de vida de desarrollo de software en contexto

Criterios de selección de modelos

- Objetivo del proyecto
- Tipo de producto en desarrollo
- Prioridades comerciales (p. ej., tiempo de comercialización)
- Riesgos del producto y del proyecto

Capítulo 2

- 2.1 Modelos de ciclo de vida de desarrollo de software
- 2.2 Niveles de prueba
- 2.3 Tipos de prueba
- 2.4 Pruebas de mantenimiento





2.2 Niveles de Prueba

- Grupos de actividades de prueba que se organizan y administran juntas

Normas niveles de prueba de la ISTQB

- Pruebas de componente
- Pruebas de integración
- Pruebas de sistema
- Pruebas de aceptación

Los niveles se caracterizan por los siguientes atributos:

- Objetivos específicos
- Base de prueba, referenciada para derivar casos de prueba
- Objetos de prueba (es decir, lo que se está probando)
- Defectos y fallos típicos
- Enfoques y responsabilidades específicas

2.2.1 Prueba de Componentes/Unitaria o de módulo

*El foco de estas pruebas es probar los componentes separados o de forma atómica

*Estas pruebas son realizadas por los desarrolladores

Objetivos específicos

- Reducir el riesgo
- Crear confianza en la calidad del componente
- Encontrar defectos en el componente
- Prevenir que los defectos escapen a niveles de prueba más elevados

Base de prueba

- Diseño detallado
- Código
- Modelo de datos
- Especificaciones de componentes

Objetos de prueba (es decir, lo que se está probando)

- Componentes, unidades o módulos
- Código y estructuras de datos
- Clases

Defectos y fallos típicos

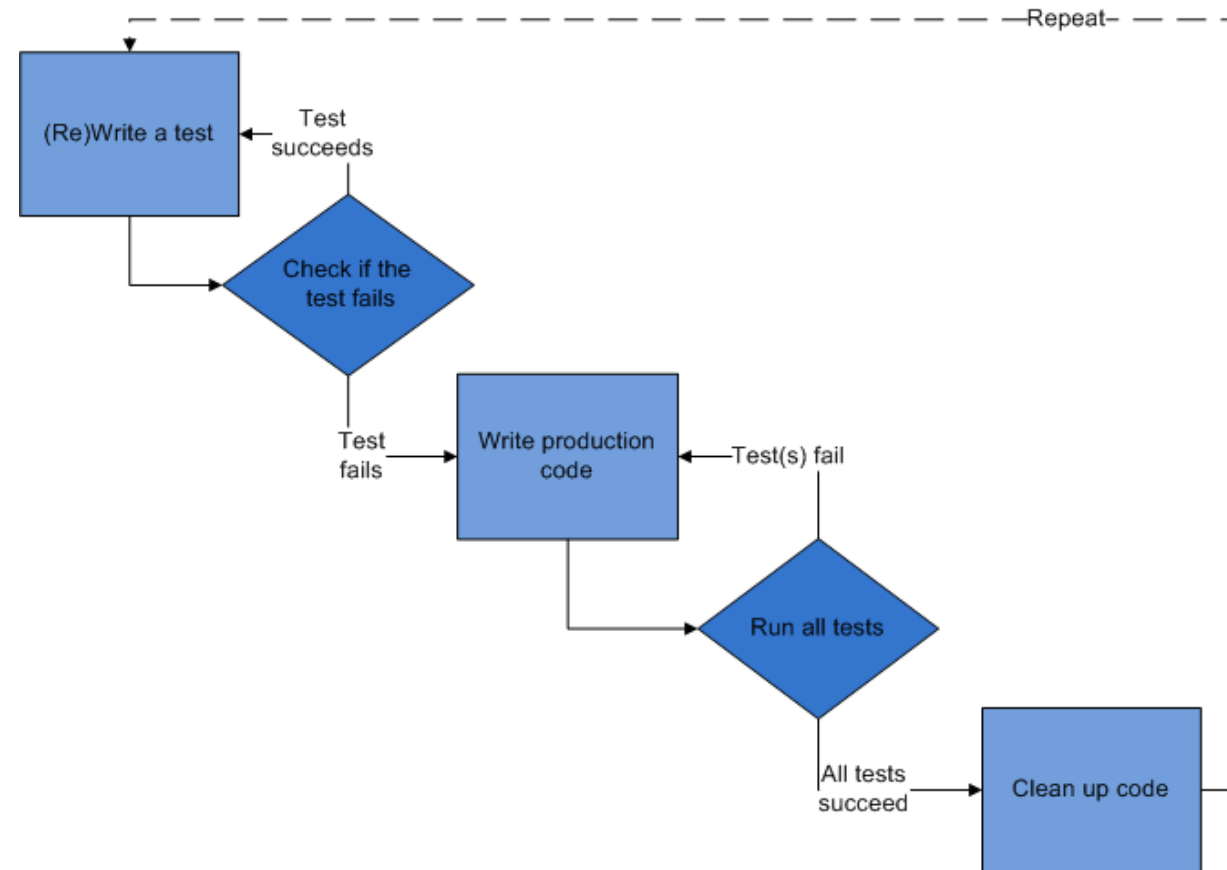
- Funcionalidad incorrecta (p. ej., no como se describe en las especificaciones de diseño)
- Problemas de flujo de datos
- Código y lógica incorrectos

Enfoques y responsabilidades específicas

- Se requieren habilidades de desarrollo
- Desarrollo Guiado por Pruebas (TDD) ☐ véa la página siguiente

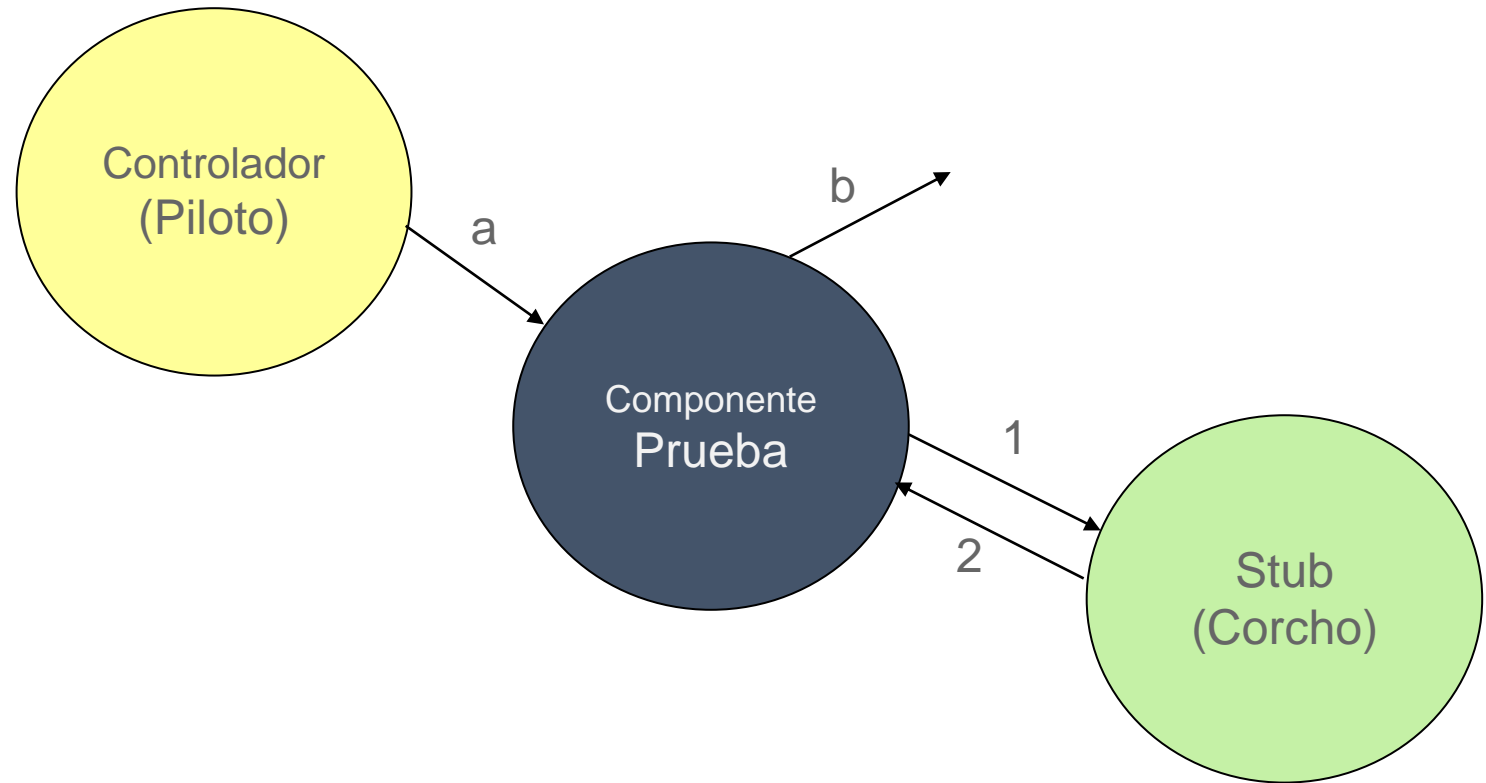
2.2.1 Prueba de Componentes

Ejemplo:
Desarrollo Guiado por Pruebas
(TDD)



2.2.1 Prueba de Componentes

- Dependiendo del enfoque de diseño e implementación, es posible utilizar stubs y/o controladores



2.2.2 Pruebas de Integración

Pruebas de interfaces entre componentes o sistemas

Objetivos específicos

- Verificar si los comportamientos funcionales y no funcionales del componente son como se diseñaron y especificaron
- Generar confianza en la calidad de las interfaces
- Encontrar defectos (que pueden estar en las interfaces o en los componentes o sistemas)

Base de prueba

- Diseño de software y de sistema , iagramas de Secuentas, Casos de uso
- Especificaciones de interfaz y protocolo de comunicación
- Arquitectura a nivel de componente o de sistema
- Definiciones de interfaz externa

Objetos de prueba (es decir, lo que se está probando)

- Subsistemas, Bases de datos
- Infraestructura
- Interfaces, APIs
- Microservicios

Defectos y fallos típicos

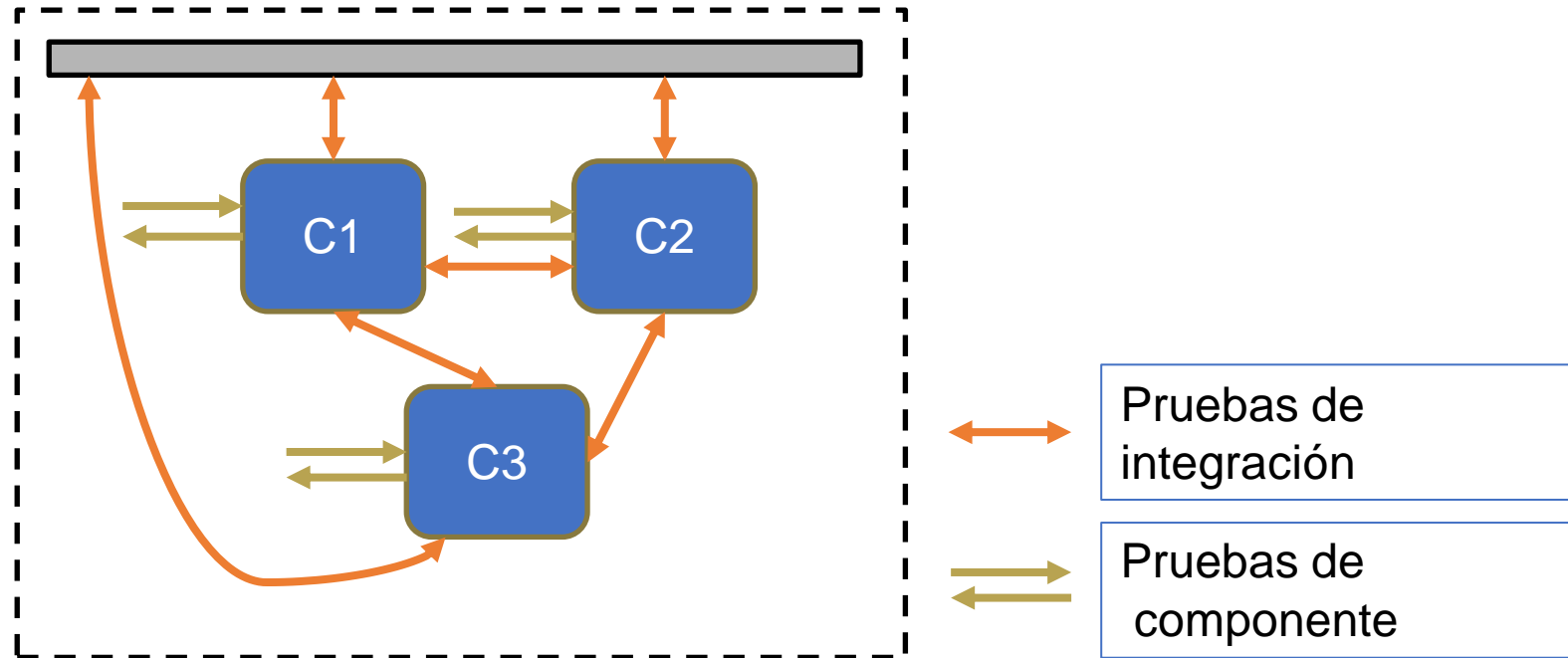
- Estructuras de mensaje inconsistentes entre sistemas
- Datos incorrectos, datos faltantes o codificación de datos incorrecta
- Incongruencias de la interfaz
- Fallos en la comunicación entre sistemas

Enfoques y responsabilidades específicas

- Se requieren habilidades técnicas tales como «arquitectura técnica»
- Big-Bang, Bottom-Up o Top-Down
- Pruebas de interfaces entre componente – RESPONSABILIDAD desarrollador
- Prueba re interafces entre sistemas – RESPONSABILIDAD probador

2.2.2 Pruebas de Integración

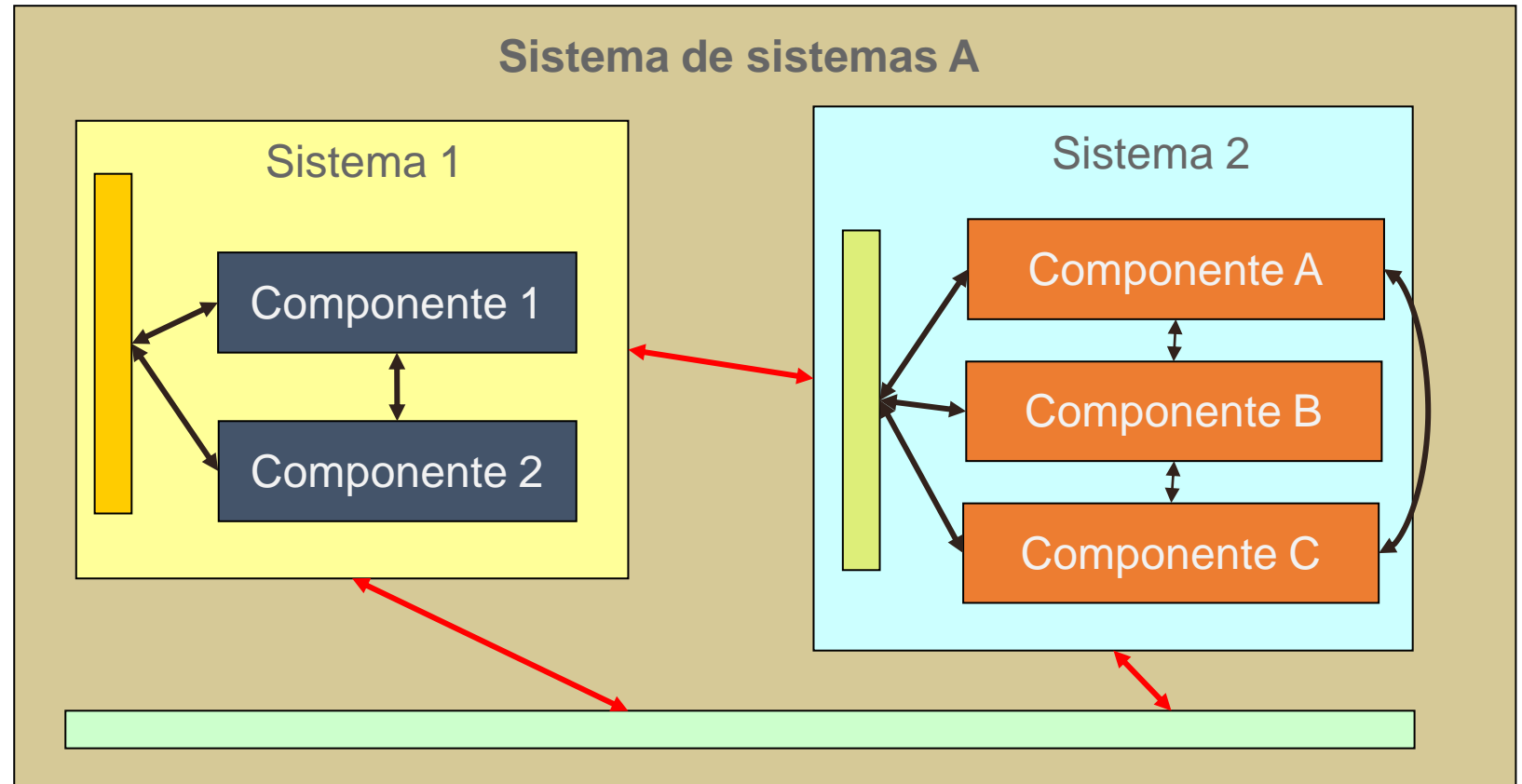
•Descripción General



2.2.2 Pruebas de Integración

Detalles específicos:
Varios niveles de
integración son
posibles

- Integración de componentes
(formando un sistema)
- **Integración de sistemas**
(formando un sistema de
sistemas)

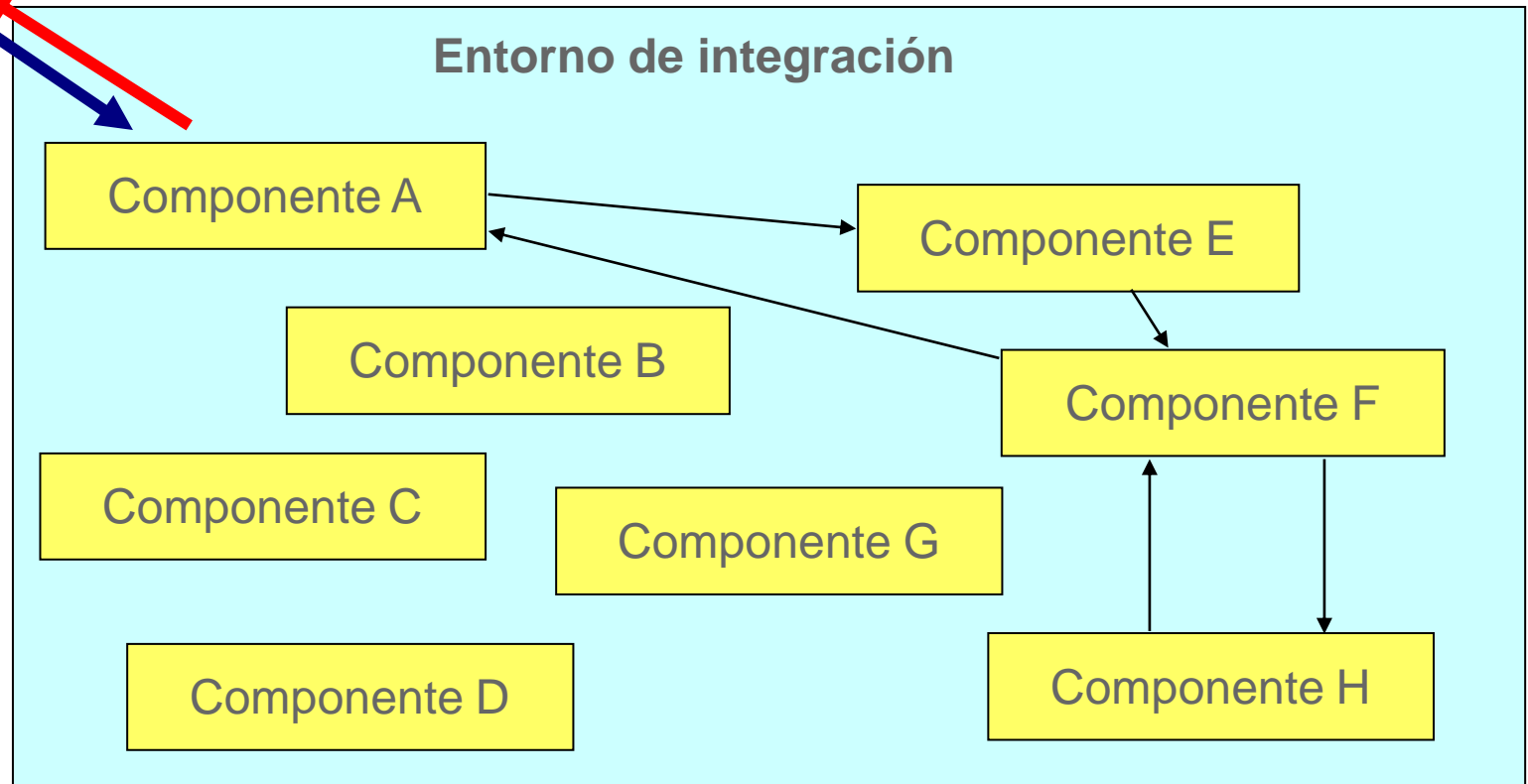


2.2.2 Pruebas de Integración

CAOS DE LA PRUEBA: ¡encontrar el error!

Estrategia de integración:
“**BigBang**” no es recomendable.

¡No integre todos los componentes a la vez!, mejor incremental.

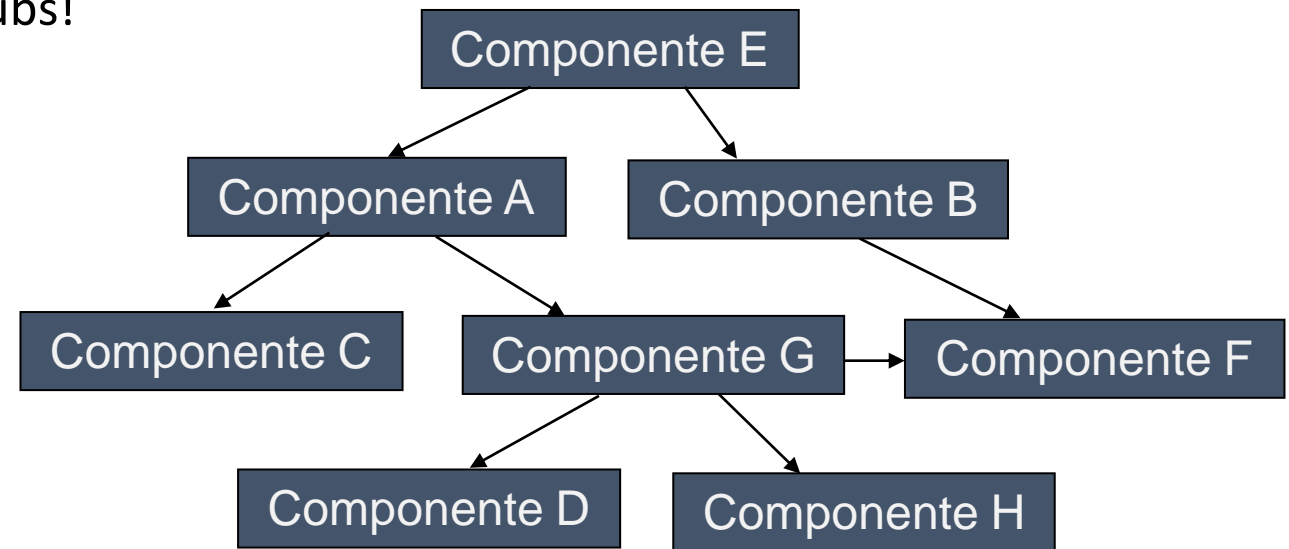


2.2.2 Pruebas de Integración

- Estrategia de integración:

“**Top-Down**” (de arriba hacia abajo) ¡usando stubs!

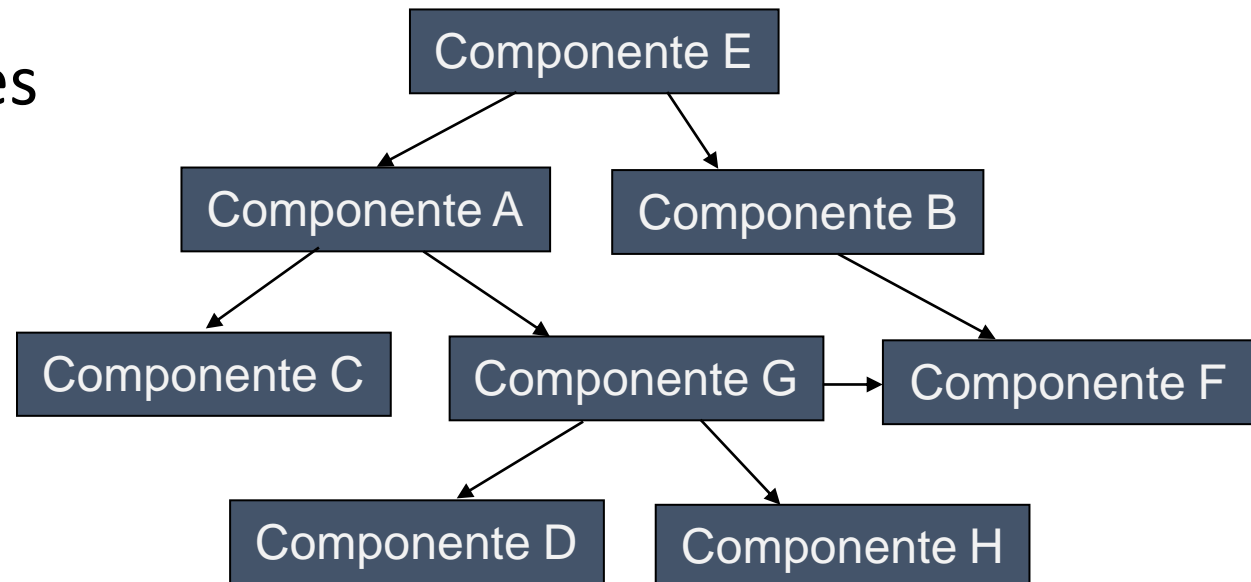
- Desde los componentes más bajos hacia arriba hasta cada componente que los llama y además
 - Haga un «Cuadro de llamadas»
 - ¡Responda la pregunta sobre con qué componente comenzar! (El que requiere menos controladores)



2.2.2 Pruebas de Integración

- Estrategia de integración
“Bottom-Up (de abajo hacia arriba)”: utilizando controladores

- Componente(s) “autosostenible(s)”
(que no llama a otro componente)
hacia los componentes de “llamada”



2.2.3 Pruebas de Sistema

•Pruebas de extremo a extremo en el producto

Objetivos específicos

- Verificar si los comportamientos funcionales y no funcionales del sistema son como se diseñaron y especificaron
- Validar que el sistema está completo y funcionará como se espera
- Generar confianza en la calidad del sistema como un todo
- Encontrar defectos

Base de prueba

- Especificaciones de requisitos del sistema y software (funcionales y no funcionales)
- Informes de análisis de riesgos
- Casos de uso
- Epopeyas e historias de usuario

Objeto de prueba (es decir, lo que se prueba)

- Aplicaciones
- Sistemas de hardware/software
- Sistemas operativos
- Sistema Sujeto a Prueba (SUT)

Defectos y fallos típicos

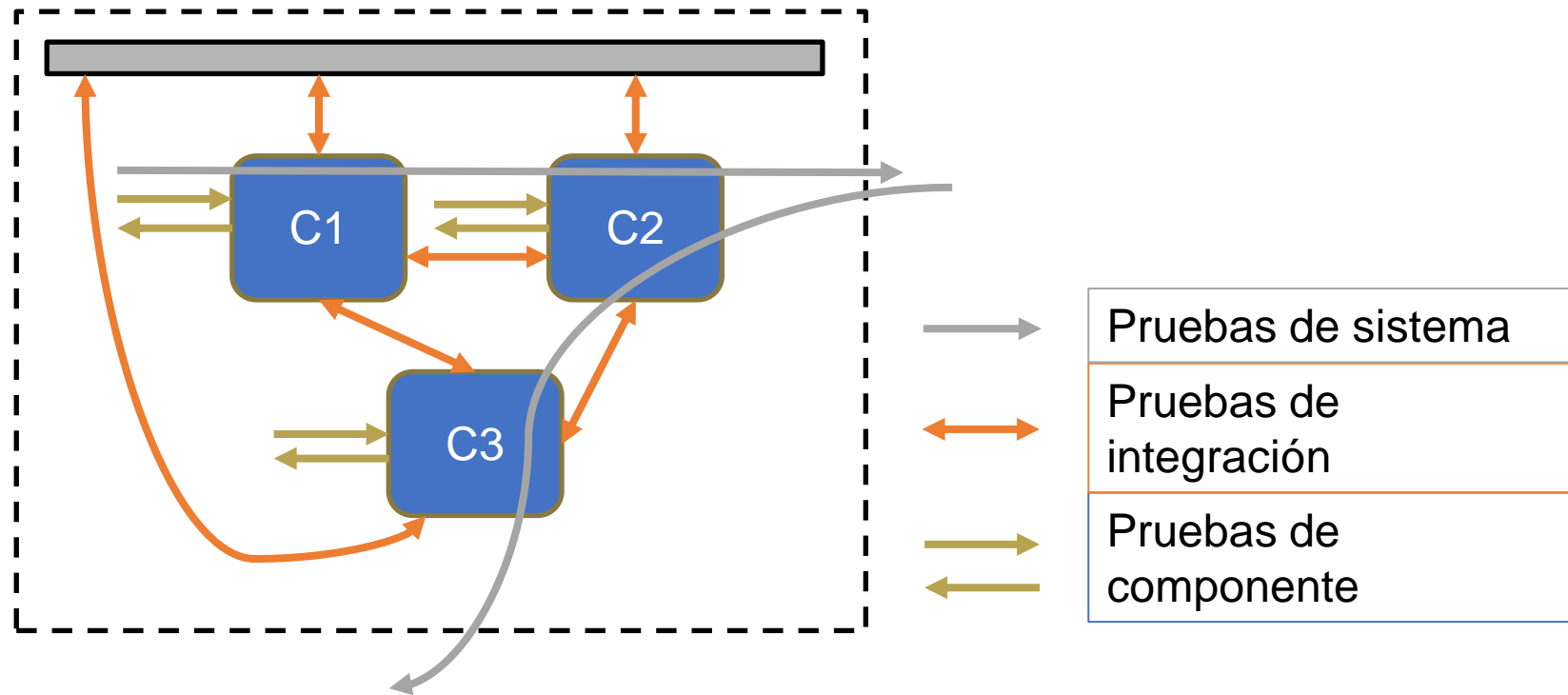
- Cálculos incorrectos
- Comportamiento funcional o no funcional del sistema incorrecto o inesperado
- Control incorrecto y/o flujos de datos dentro del sistema
- No realizar correctamente y completamente las tareas funcionales de extremo a extremo

Enfoques y responsabilidades específicas

- Use técnicas de diseño de prueba como se presenta en el capítulo 4
- Pruebas tempranas

2.2.3 Pruebas de Sistema

• Vista General



2.2.4 Pruebas de Aceptación

Probar el comportamiento y las capacidades de todo un sistema o producto.

Objetivos específicos

- Generar confianza en la calidad del sistema como un todo
- Validar que el sistema está completo y funcionará como se espera
- Verificar que los comportamientos funcionales y no funcionales del sistema son los especificados

Base de prueba

- Procesos de negocio
- Requisitos de usuario o empresariales
- Regulaciones, contratos legales y normas
- Casos de uso

Objetos de prueba (es decir, lo que se está probando)

- Sistema sujeto a prueba
- Configuración del sistema y datos de configuración
- Procesos de negocio para un sistema totalmente integrado

Defectos y fallos típicos

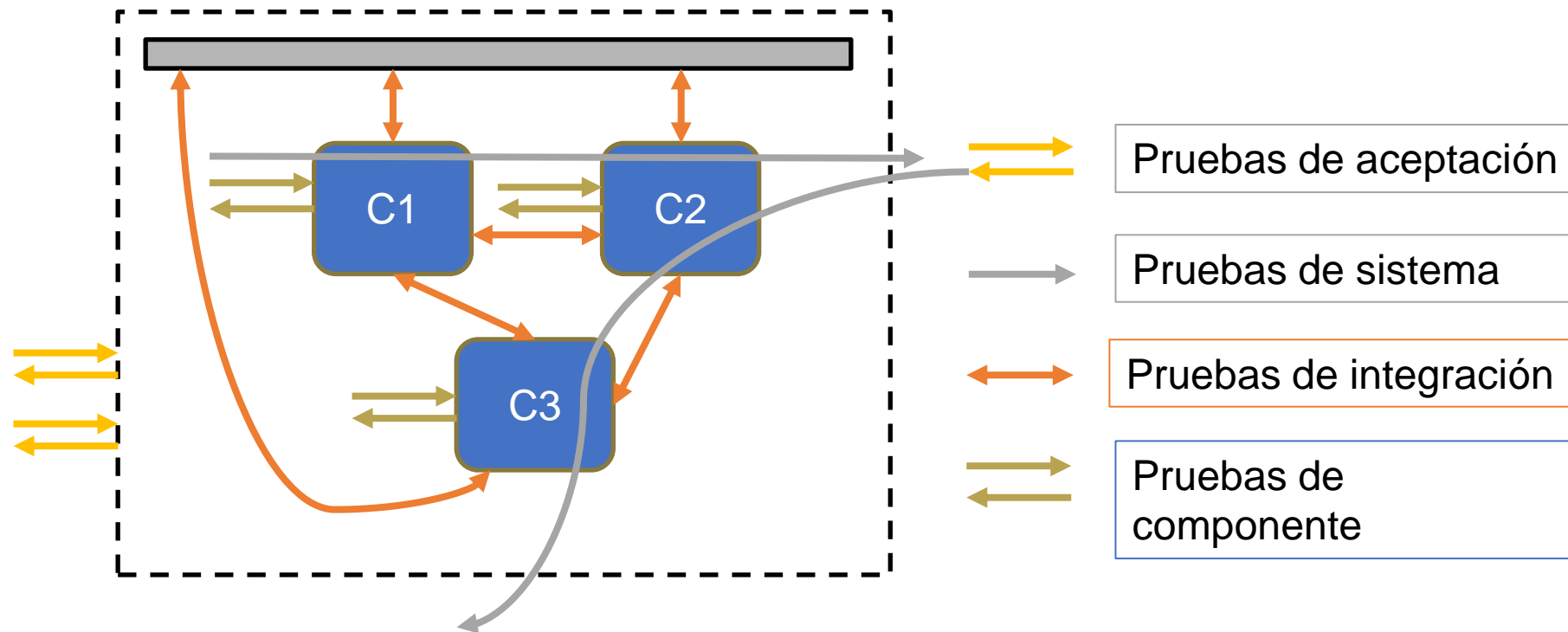
- Los flujos de trabajo del sistema no cumplen con los requisitos comerciales o del usuario
- Las reglas comerciales no se implementan correctamente
- El sistema no cumple con los requisitos contractuales o reglamentarios
- Fallos no funcionales tales como vulnerabilidades de seguridad

Enfoques y responsabilidades específicas

- A menudo son responsabilidad de los clientes, usuarios comerciales, propietarios de productos

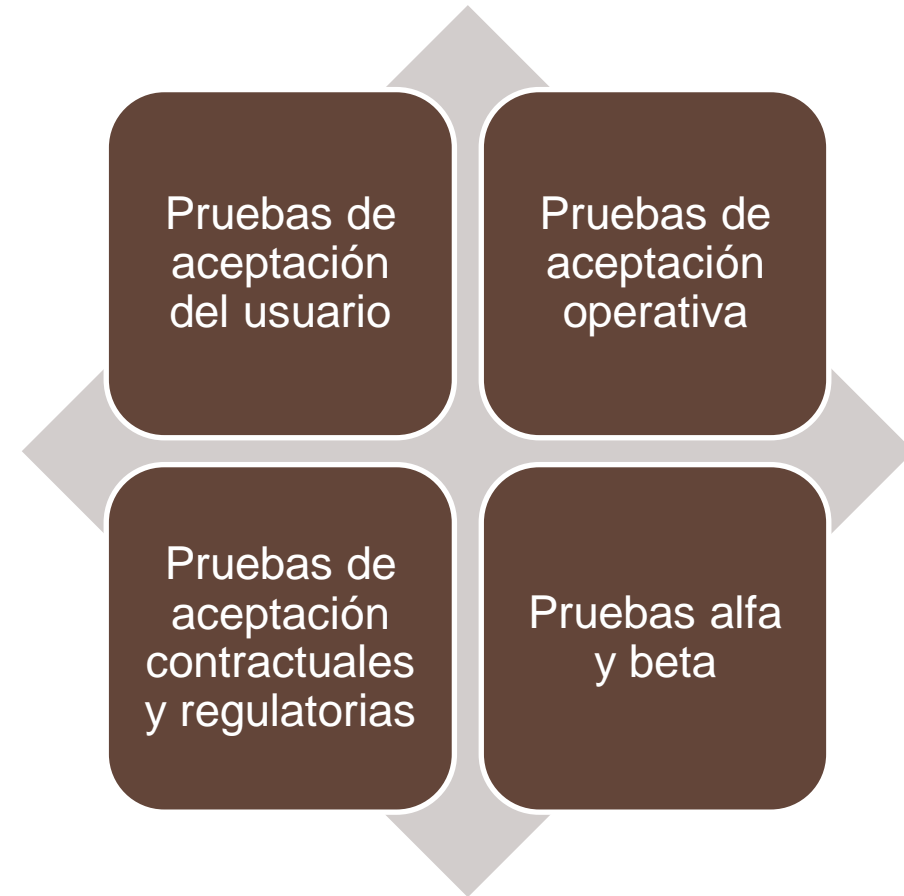
2.2.4 Pruebas de Aceptación

•Descripción General



2.2.4 Pruebas de Aceptación

- Principales formas de pruebas de aceptación





2.2.4 Pruebas de Aceptación

- Pruebas de aceptación del usuario (PAU)

Validar la aptitud para el uso del sistema por parte de los usuarios previstos en un entorno operativo real o simulado.

Generar confianza en que los usuarios pueden usar el sistema para satisfacer sus necesidades.

2.2.4 Pruebas de Aceptación

•Prestaciones especiales

- A menudo asumidas por los **clientes** o **usuarios**
- Ejemplo de un plan resumen de prueba de aceptación

CUSTOMER ID	CUSTOMER NAME	CUSTOMER DETAILS					

DATE OF TESTING	QC ENGINEER - CUSTOMER	QC ENGINEER – INTERNAL			SUPPORT TEAM - INTERNAL	

INSPECTION – DIMENSIONAL CHECK TESTING							
SR	MATERIAL / PARTS NUMBER	STANDARDS	SPECIFICAITONS	TEST RESULTS			OBSERVATIONS POINTS
				TEST -1	TEST -2	TEST -3	

VISUAL INSPECTION							
SR	MATERIAL / PARTS NUMBER	STANDARDS	SPECIFICAITONS	TEST RESULTS			OBSERVATIONS POINTS
				TEST -1	TEST -2	TEST -3	

RESULTS COMMENTS BY QC ENGINEER - CUSTOMER				RESULTS QC ENGINEER - INTERNAL			
ACCEPTED REJECTED				SIGNATURE		ACCEPTED REJECTED	



2.2.4 Pruebas de Aceptación

- Pruebas de aceptación operativa (PAO)

Pruebas de aceptación de sistemas por parte del personal de administración de operaciones o sistemas

- Pruebas de copia de seguridad y restauración
- Instalación, desinstalación y actualización
- Recuperación de desastres
- Gestión de usuarios
- Tareas de mantenimiento
- Carga de datos y tareas de migración
- Comprobaciones de las vulnerabilidades de seguridad
- Pruebas de rendimiento

Importante: recopilar los requisitos operativos al inicio del proyecto



2.2.4 Pruebas de Aceptación

• Pruebas de aceptación contractuales y regulatorias

Aceptación contractual

- Contra de los criterios de aceptación de un contrato para producir software desarrollado a la medida

DOCUMENT ED-12B / DO-178B

Aceptación regulatoria

- Contra cualquier regulación que se deba cumplir, como las reglamentaciones gubernamentales, legales o de seguridad
- Ejemplo: aceptación contra la DO 178B y los requisitos relacionados con la prueba

SOFTWARE CONSIDERATIONS IN AIRBORNE
SYSTEMS AND EQUIPMENT CERTIFICATION

2.2.4 Pruebas de Aceptación

Términos, según el glosario de la ISTQB



Pruebas Alfa

- Pruebas operacionales simuladas o reales por usuarios/clientes potenciales o un equipo de prueba independiente **en el sitio de los desarrolladores**, pero fuera de la organización de desarrollo, las pruebas Alfa se emplean a menudo para software de distribución masiva como una forma de prueba de aceptación interna.

Pruebas Beta

- Pruebas operacionales por usuarios/clientes potenciales y/o existentes **en un sitio externo que de otro modo no estaría involucrado con los desarrolladores**, para determinar si un componente o sistema satisface las necesidades del usuario/cliente y se ajusta a los procesos de negocio. Las pruebas Beta se emplean a menudo como una forma de prueba de aceptación externa para software de distribución masiva con el fin de obtener retroalimentación del mercado.



2.2 Niveles de prueba

- Ejercicio: Identifique los diferentes niveles de prueba en su proyecto o en el que trabajó.

Capítulo 2

- 2.1 Modelos de ciclo de vida de desarrollo de software
- 2.2 Niveles de prueba
- 2.3 Tipos de prueba
- 2.4 Pruebas de mantenimiento



2.3 Tipos de Prueba

•Diferentes tipos de prueba basados en objetivos específicos de prueba

Tipo de prueba

- Un grupo de actividades de prueba destinadas a probar un componente o sistema enfocado en un objetivo de prueba específico, es decir, prueba funcional, prueba de usabilidad, pruebas de regresión, etc. Un tipo de prueba puede tener lugar en uno o más niveles de prueba o fases de prueba.





2.3.1 Pruebas Funcionales (Caja Negra)

• Pruebas que evalúan las funciones que debe realizar el sistema.

Niveles de prueba

- Todos

Base de prueba

- especificaciones de requisitos de negocio
- epopeyas
- historias de usuario
- casos de uso, o especificaciones funcionales

Técnicas de pruebas

- Principalmente caja negra

Medición de la cobertura

- Cobertura funcional (la medida en que se ha ejercido algún tipo de elemento funcional mediante pruebas)

Habilidades

- Conocimiento del problema empresarial específico que soluciona el software

2.3.2 Pruebas No Funcionales

• Pruebas para evaluar las características de los sistemas y el software, como son la usabilidad, la eficiencia del rendimiento o la seguridad.

Niveles de prueba

- Todos

Base de prueba

- especificaciones de requisitos de negocio
- epopeyas
- historias de usuario
- casos de uso
- Especificaciones no funcionales

Técnicas de pruebas

- Principalmente caja negra

Cobertura

- Cobertura no funcional (la medida en que se ha ejercido algún tipo de elemento no funcional mediante pruebas)

Habilidades

- Conocimiento específico (p. ej., tecnología o categorías de usuarios)



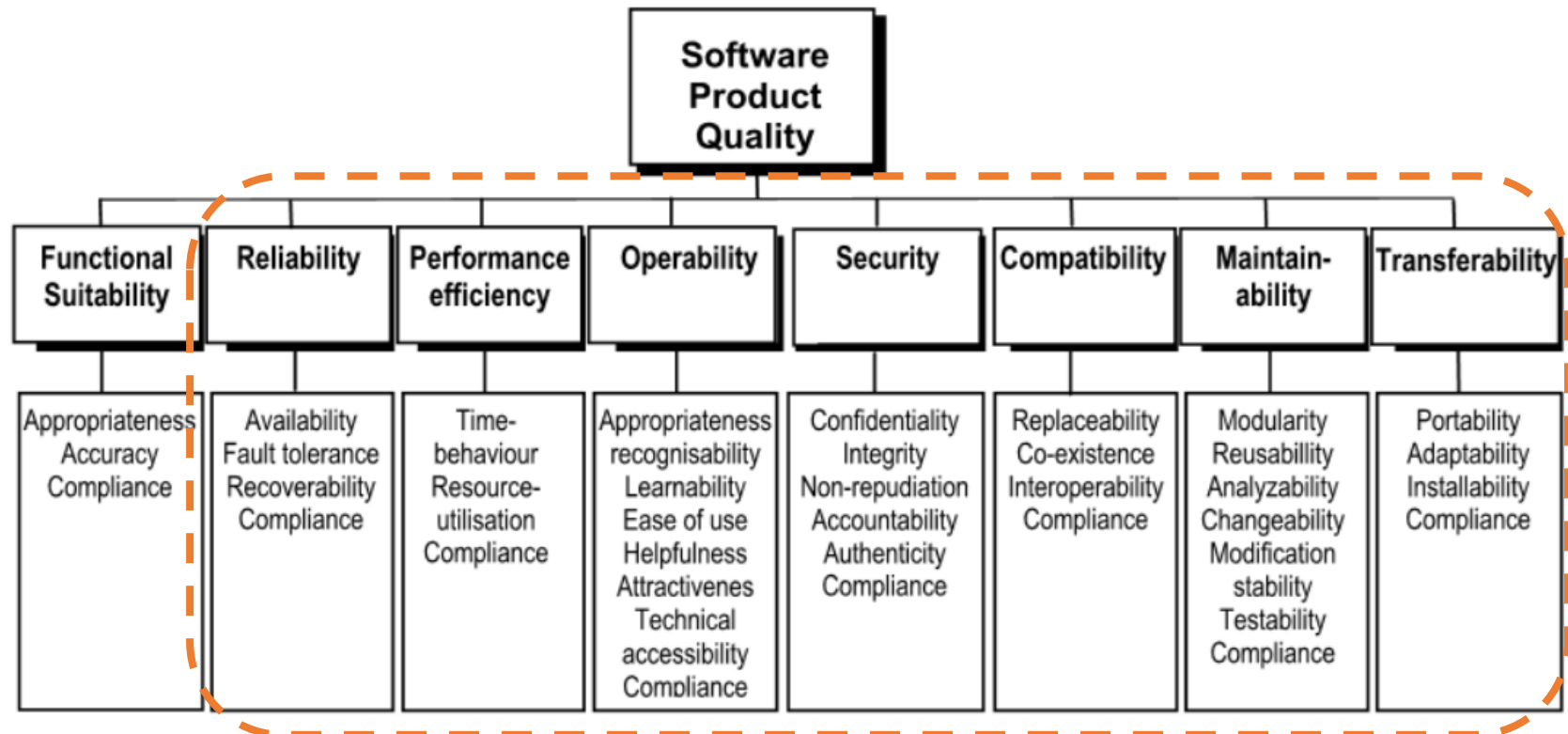
2.3.2 Pruebas No Funcionales

- Ejemplos de requisitos de rendimiento

- Tiempos de respuesta de la transacción
 - Con respecto a las consultas en la pantalla, el tiempo de respuesta debe ser de dos segundos entre la validación de la consulta y la visualización de los datos en la pantalla.
 - En cuanto a las consultas y las entradas en varias pantallas, la visualización de cada página está sujeta a los siguientes tiempos de respuesta:
 - Visualización de la primera página de pantalla: menos de 1 segundo
 - Desplazarse por la página/siguiente pestaña o anterior (página arriba/página abajo): menos de 1 segundo
 - Pasar a la línea siguiente o anterior (mediante la barra de desplazamiento a la derecha): menos de 500 ms

2.3.2 Pruebas No Funcionales

- La ISO (ISO/CEI 25010) identifica claramente los atributos no funcionales que se deben probar





2.3.3 Pruebas de Caja Blanca

- Evaluar si la estructura o arquitectura del componente o sistema es correcta, completa y según lo especificado.

Niveles de prueba

- Todos

Base de prueba

- Código, arquitectura, flujos de trabajo y/o flujos de datos dentro del sistema

Técnicas de pruebas

- Caja blanca

Medición de la cobertura

- Cobertura estructural: medida en que se ha ejercido algún tipo de elemento estructural mediante pruebas (instrucción, decisión, por ejemplo)

Habilidades

- Arquitectura, base de datos, codificación



2.3.4 Pruebas Relacionadas con el Cambio (confirmación o regresión)

- Cuando se realizan cambios en un sistema, ya sea para corregir un defecto o debido a una funcionalidad nueva o cambiante

- Las pruebas de regresión son buenas candidatas para automatización.

Niveles de prueba

- Todos

Para ser utilizados con

- Pruebas funcionales
- Pruebas no funcionales
- Pruebas estructurales

Pruebas de confirmación

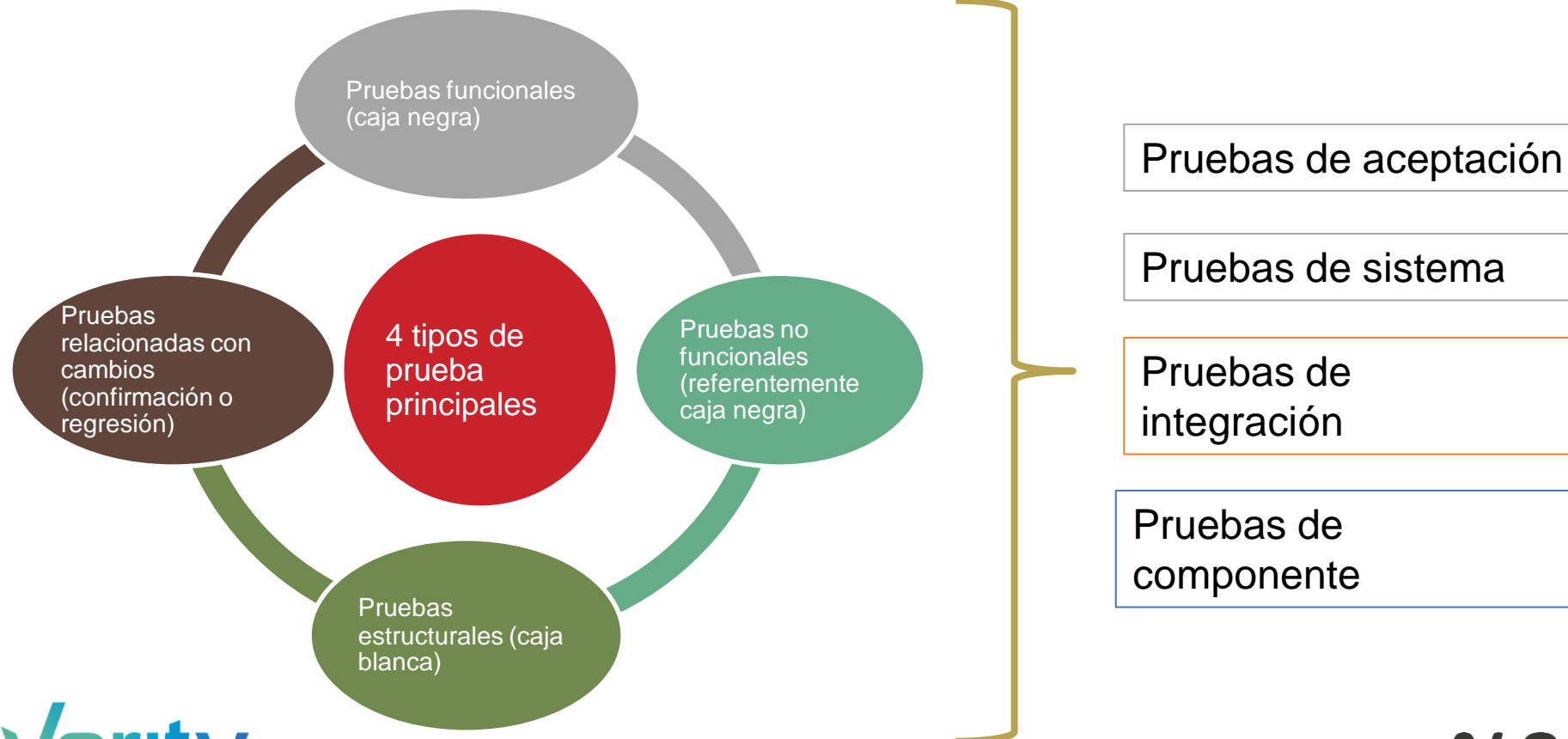
- Para confirmar que los cambios han corregido el defecto o implementado la funcionalidad correctamente.

Pruebas de regresión

- Para confirmar que los cambios no han causado consecuencias adversas imprevistas

2.3.5 Tipos y Niveles de Prueba

- Es posible realizar cualquiera de los tipos de prueba mencionados anteriormente en cualquier nivel de prueba





2.3.5 Tipos y Niveles de Prueba

- Ejercicio: para el nivel de prueba de su elección, dé un ejemplo de cada tipo de prueba; compare con el programa de estudio (sección 2.3.5) y con los demás participantes después.

Nivel:

Pruebas funcionales

•...
•...
•...

Pruebas no
funcionales

•...
•...
•...

Pruebas estructurales

•...
•...
•...


Pruebas relacionadas
con los cambios

•...
•...
•...

Capítulo 2

- 2.1 Modelos de ciclo de vida de desarrollo de software
- 2.2 Niveles de prueba
- 2.3 Tipos de prueba
- 2.4 Pruebas de mantenimiento





2.4.1 Causas para el Mantenimiento

• Cuando se realizan cambios como parte del mantenimiento

Cambios relacionados con el sistema en sí

- Corrección
- Nueva funcionalidad
- Modificaciones en la parametrización

Cambios en el entorno

- Sistema operativo
- Bases de datos y migración de datos
- Interfaces

Retirada

- Cuando una aplicación llega al final de su vida útil.
- Cómo almacenar y respaldar los datos.

Quando se realizan cambios como parte del mantenimiento, se debe realizar una prueba de mantenimiento, tanto para evaluar el éxito con el que se realizaron los cambios como para comprobar los posibles efectos secundarios (por ejemplo, regresiones) en las partes del sistema que permanecen inalteradas



2.4.2 Análisis del Impacto para el Mantenimiento

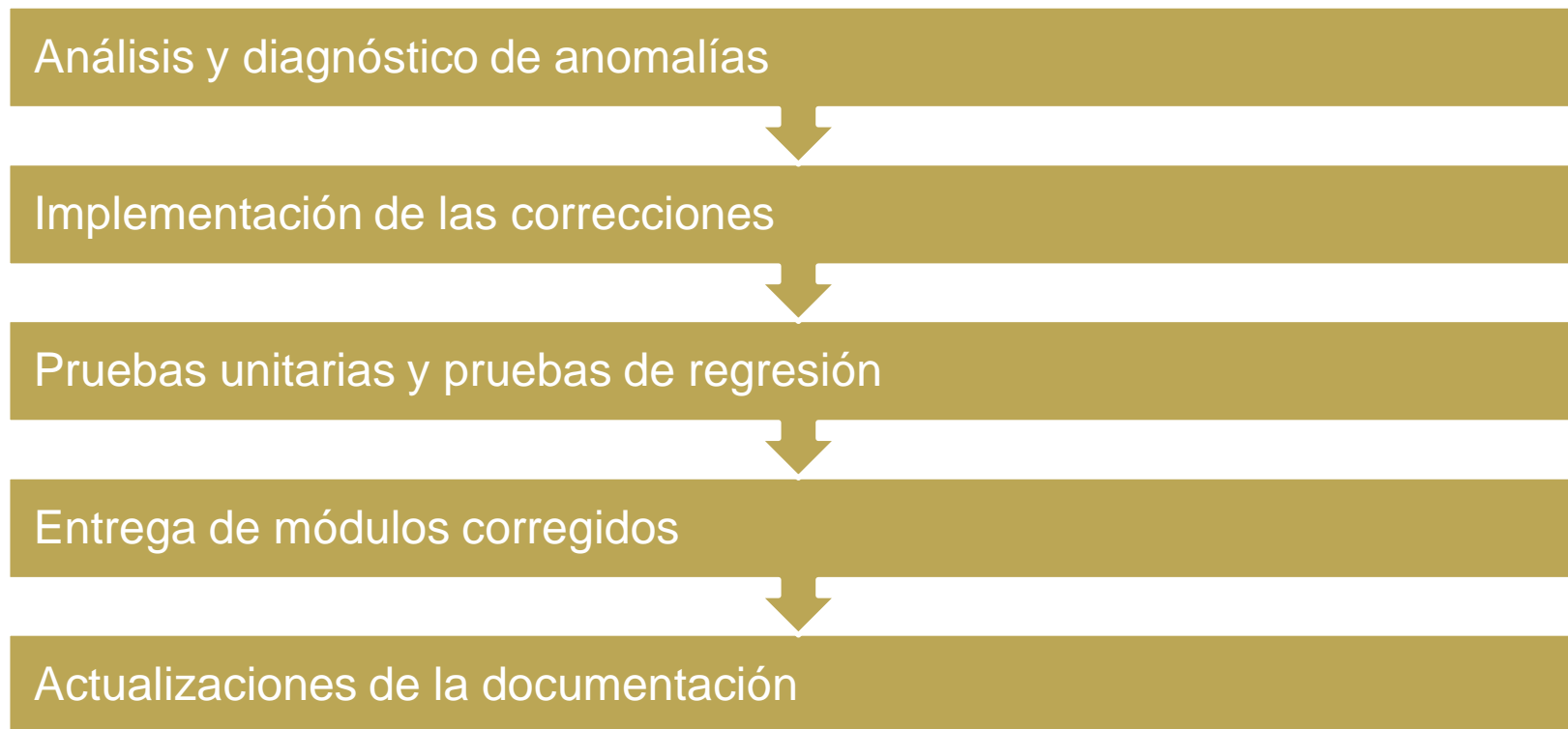
- El objetivo del análisis de impacto es para determinar como el sistema existente puede verse afectado por los cambios.

Análisis del impacto

- La evaluación de una modificación en las capas de la documentación de desarrollo, documentación de pruebas y componentes, a fin de poner en práctica un cambio dado de los requisitos especificados
- El análisis del impacto puede ser difícil si:
 - Las especificaciones están desactualizadas o faltan
 - Los casos de prueba no están documentados o están desactualizados
 - La trazabilidad bidireccional entre pruebas y la base de pruebas no se ha mantenido
 - El soporte de la herramienta es débil o inexistente

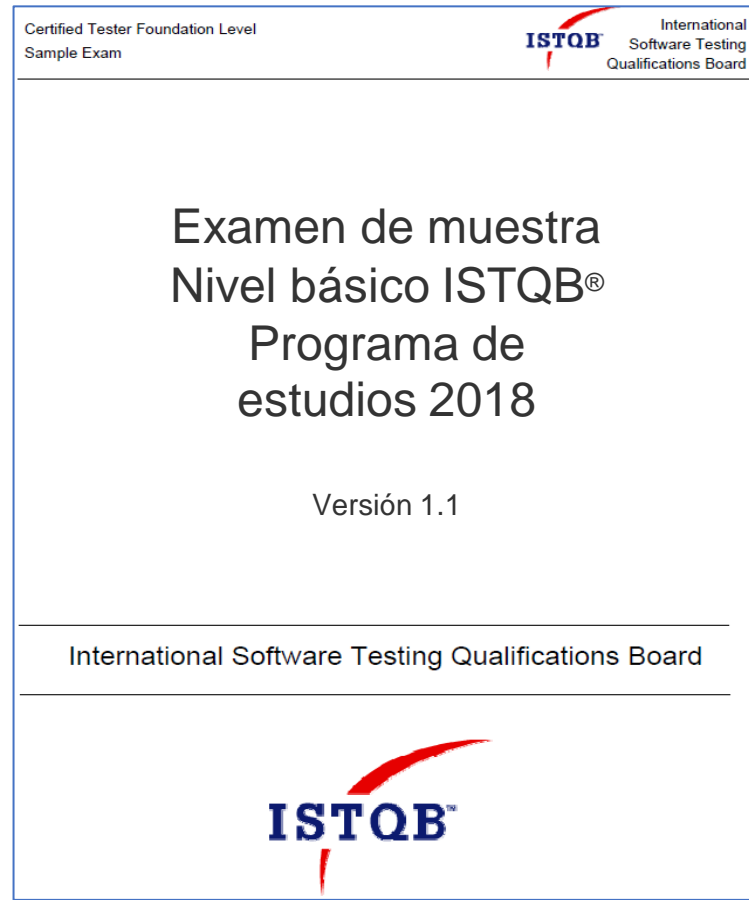
2.4.2 Análisis del Impacto para el Mantenimiento

- Ejemplo de actividades de mantenimiento subcontractadas



Ejercicio: ISTQB Examen de Muestra A

- Preguntas de la 9 a la 13



Capítulo 2

Las Pruebas Durante Todo el Ciclo de Vida de Desarrollo de Software

