

Verificación y validación de software

Fundamentos de la verificación y validación del software

Fundamentos de la verificación y validación del software

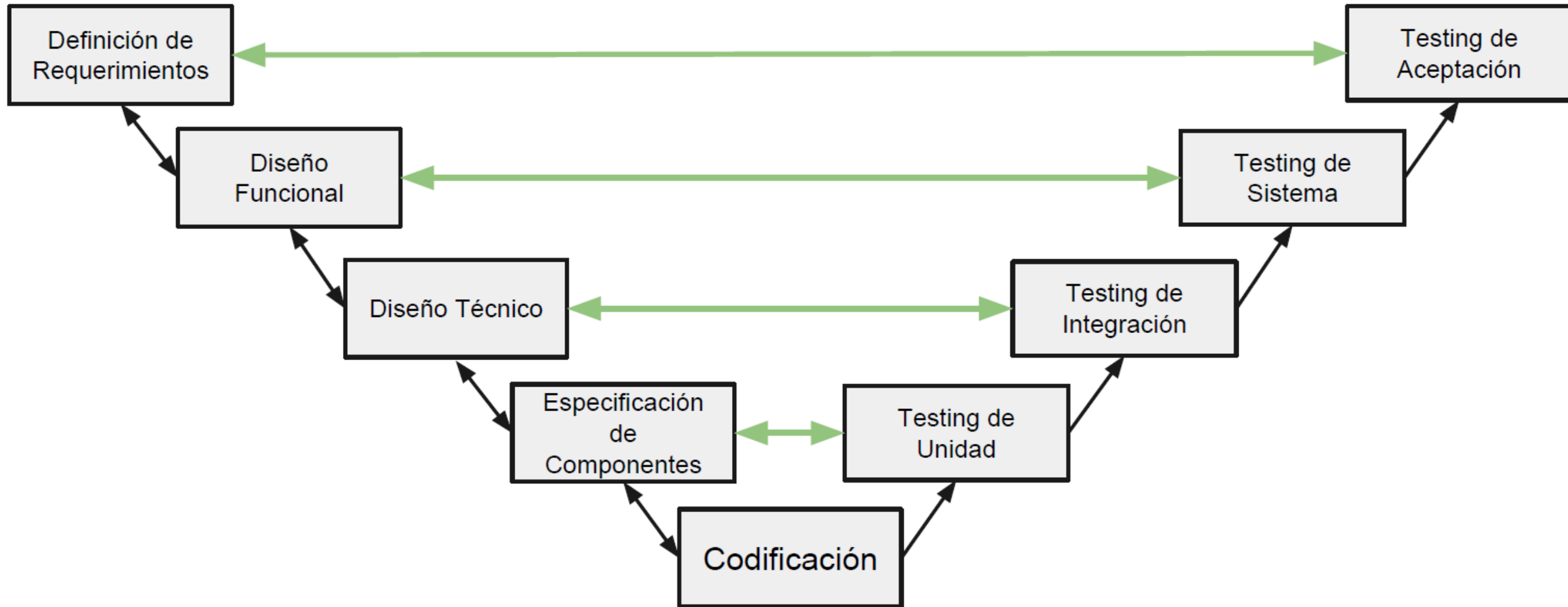
En la clase anterior...

- La verificación en las diferentes fases del proceso de desarrollo. *Parte II*

Fundamentos de la verificación y validación del software

La involucración temprana de las pruebas en el ciclo de vida del desarrollo de sistemas ayuda a detectar los defectos con mayor antelación y/o con más facilidad, e incluso ayuda a prevenirlos.

Fundamentos de la verificación y validación del software



Fundamentos de la verificación y validación del software

Prueba de Caja Negra (ignoro el código al momento de hacer la verificación)

Prueba de Caja Blanca (genero los casos de test en base a conocimiento que obtengo del código)

Fundamentos de la verificación y validación del software

- ❖ Se necesitan “estrategias” para seleccionar casos de prueba “significativos”
- ❖ Test Set de Significancia
 - Tiene un alto potencial para descubrir errores
 - La ejecución correcta de estos test aumenta la confianza en el producto
- ❖ Más que correr una gran cantidad de casos de prueba nuestra meta en las pruebas debe ser correr un suficiente número de casos de prueba significativos (alto porcentaje de posibilidades de descubrir un error)

Fundamentos de la verificación y validación del software

Los test de caja blanca detectan entre el 50% y el 75% de los errores, pero son los errores más fáciles de encontrar.

Los test de caja negra detectan un porcentaje menor de errores pero son los errores más problemáticos.

Fundamentos de la verificación y validación del software

Testing de Unidad

Component Testing

Es el primer test que se hace sobre código. La “Unidad” puede ser una clase, módulo, función, etc, depende del lenguaje/paradigma.

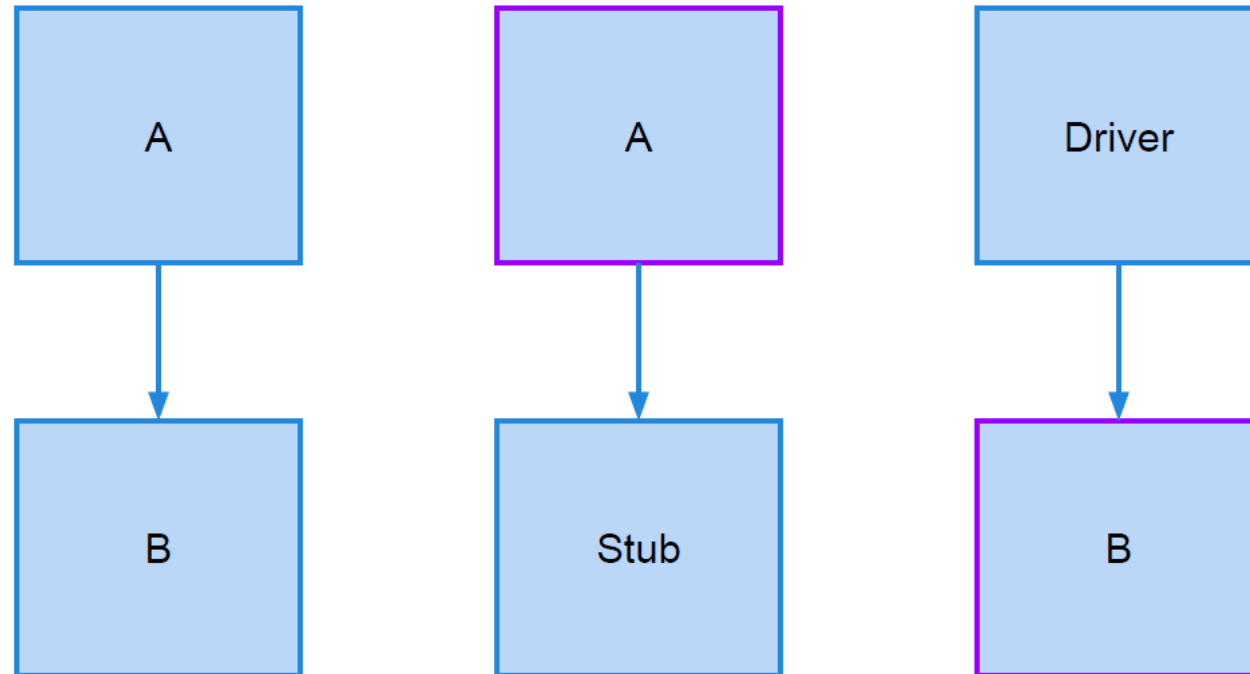
Fundamentos de la verificación y validación del software

- El test de unidad se basa en la especificación del componente. Es fundamental en esta etapa, que cada unidad se teste en **forma aislada**.
- La aislación nos permite estar seguros de que cualquier problema detectada es propio de la unidad. Aislar un componente no es trivial.

Fundamentos de la verificación y validación del software

Pruebas de Unidad

Stub
Driver



Fundamentos de la verificación y validación del software

El objetivo de este tipo de pruebas es asegurar funcionalidad y completitud. “El componente hace todo lo que requieren, ni más ni menos, y lo hace bien”.

Fundamentos de la verificación y validación del software

También se busca **solidez y robustez**.

El componente es sólido, “el componente no se rompe”.

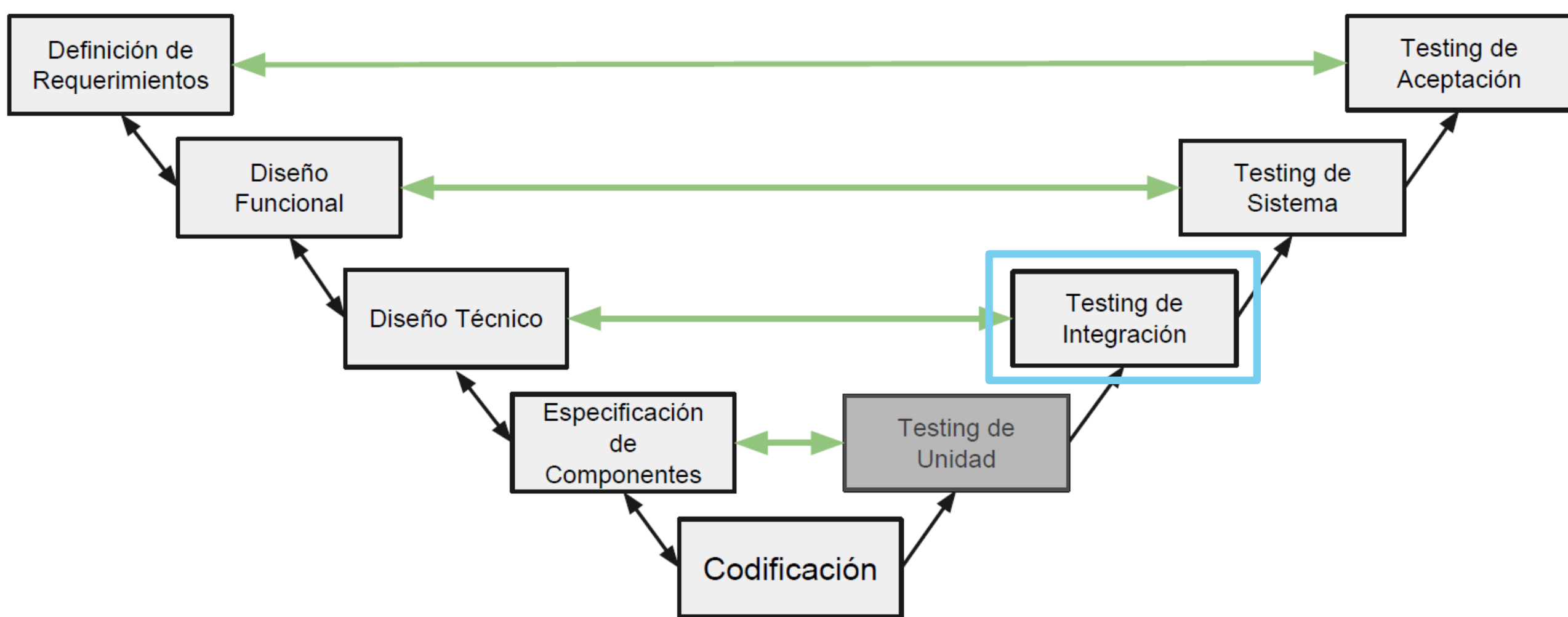
El componente es robusto, “si el componente se rompe no hace un desastre”.

Fundamentos de la verificación y validación del software

Agenda:

- Verificación
 - La verificación en las diferentes fases del proceso de desarrollo. *Parte III*
- Validación
 - Validación en las diferentes fases del proceso de desarrollo.
- Diferencias entre verificación y validación.

Fundamentos de la verificación y validación del software



Fundamentos de la verificación y validación del software

Segunda etapa de pruebas.

En esta etapa se asume que las componentes (unidades) fueron testeadas y sus defectos fueron resueltos.

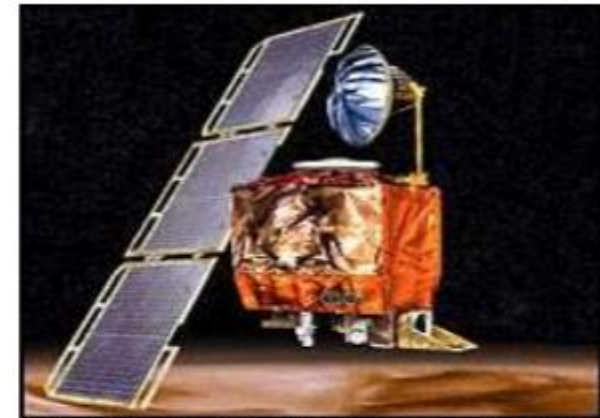


Fundamentos de la verificación y validación del software

Metric Mishap Caused Loss of NASA Orbiter

* By Kathy Sawyer & Robin Lloyd (Washington Post Staff Writer & CNN Interactive Senior Writer)

NASA's Mars Climate Orbiter was lost in space last week because engineers failed to make a simple conversion from English units to metric, an embarrassing lapse that sent the \$125 million craft fatally close to the Martian surface.



Scientists do not yet know what caused the Mars Orbiter to lose contact with Earth. (AP)

It now appears the error had affected the orbiter mission from its launching almost 10 months and 416 million miles before its Sept. 23 failure. And yet the problem was never caught and corrected by the system of checks and balances at the Jet Propulsion Laboratory (JPL) in California, which manages this and other interplanetary missions for NASA.

http://mytoe.org/docs/Mars_closereading_art.pdf

Fundamentos de la verificación y validación del software

The initial error was made by U.S. contractor Lockheed Martin Astronautics, which, like the rest of the U.S. launch industry, traditionally uses English measurements.

The JPL navigation team, on the other hand, uses metric measurements in the complex business of figuring out a spacecraft's position relative to moving planets and keeping it on course.

Fundamentos de la verificación y validación del software

The contractor is supposed to convert its measurements to metrics before delivery to NASA. (NASA has been using the metric system predominantly since at least 1990).

The initial error occurred in the files that were sent daily by Lockheed to JPL navigators. The wrong measurements were used to power the daily thruster firings.

Fundamentos de la verificación y validación del software

Algunos de los errores que se pueden encontrar son:

- Formatos de interfaces erróneos
- Errores en el intercambio de información

Fundamentos de la verificación y validación del software

El objetivo en esta etapa es

- Tomar los componentes probados de manera individual y construir una estructura de programa que se haya dictado por diseño.
- Detectar fallas en las interfaces e interacciones entre componentes.
- También se pueden incluir test de integración con sistemas externos.

Fundamentos de la verificación y validación del software

El problema es “juntarlos todos”: conectarlos.

- Los datos pueden perderse a través de una interfaz;
- Un componente puede tener un inadvertido efecto adverso sobre otro;
- Las subfunciones, cuando se combinan, pueden no producir la función principal deseada;
- La imprecisión aceptable individualmente puede magnificarse a niveles inaceptables;
- Las estructuras de datos globales pueden presentar problemas.

Fundamentos de la verificación y validación del software

Se define como “integración” a la conexión entre componentes para formar un sistema o unidad mayor.

Fundamentos de la verificación y validación del software

¡Enfoque de *big bang*!



Fundamentos de la verificación y validación del software

Estrategias diferentes de integración incremental.

- El programa se construye y prueba en pequeños incrementos

Fundamentos de la verificación y validación del software

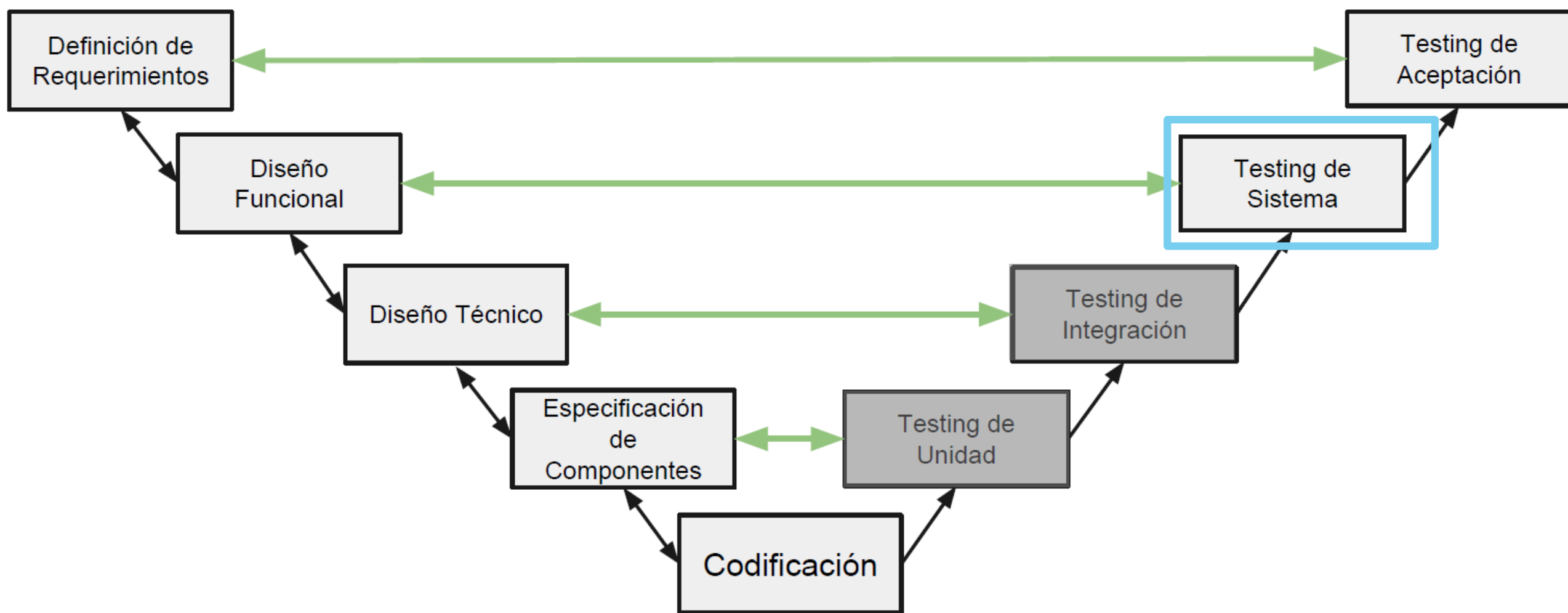
- Top-Down,
No requiere Drivers / Requiere Stubs
- Bottom-Up,
No requiere Stubs / Requiere Test Drivers
- Ad-hoc,
Se ahorra tiempo / Requiere Stubs y Drivers
- Backbone,
El orden no es importante / Buscar el Backbone
- Big-Bang

Fundamentos de la verificación y validación del software

Servidores de integración:

- Sistemas informáticos (Web y locales) → repositorios del proyecto (GitHub).
- El servidor de integración compila y ejecuta casos de prueba definidos por el desarrollador.
- Cada vez que el desarrollador integra algo nuevo, corre un test de integración

Fundamentos de la verificación y validación del software



Fundamentos de la verificación y validación del software

Tercera etapa del testing; esta chequea que el producto cumpla con lo especificado (funcional y no funcional).

Fundamentos de la verificación y validación del software

¿Cuál es la diferencia con el Testing de Integración?

Las pruebas de Sistema se hace desde el punto de vista del cliente y futuro usuario. Las etapas anteriores fueron etapas técnicas.

Fundamentos de la verificación y validación del software

Un objetivo de esta etapa es el de probar el sistema en un contexto de uso lo más cercano posible al real.

No se usan stubs ni drivers, todos los sistemas externos son reales y la plataforma es la real (o lo más real que sea posible).

Fundamentos de la verificación y validación del software

Por “plataforma real” no nos referimos a “en la del usuario”.

Recrear la plataforma del usuario en una máquina virtual.

Fundamentos de la verificación y validación del software

<https://www.virtualbox.org/>



VirtualBox

Welcome to VirtualBox.org!

VirtualBox is a powerful x86 and AMD64/Intel64 [virtualization](#) product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "[About VirtualBox](#)" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of [guest operating systems](#) including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.

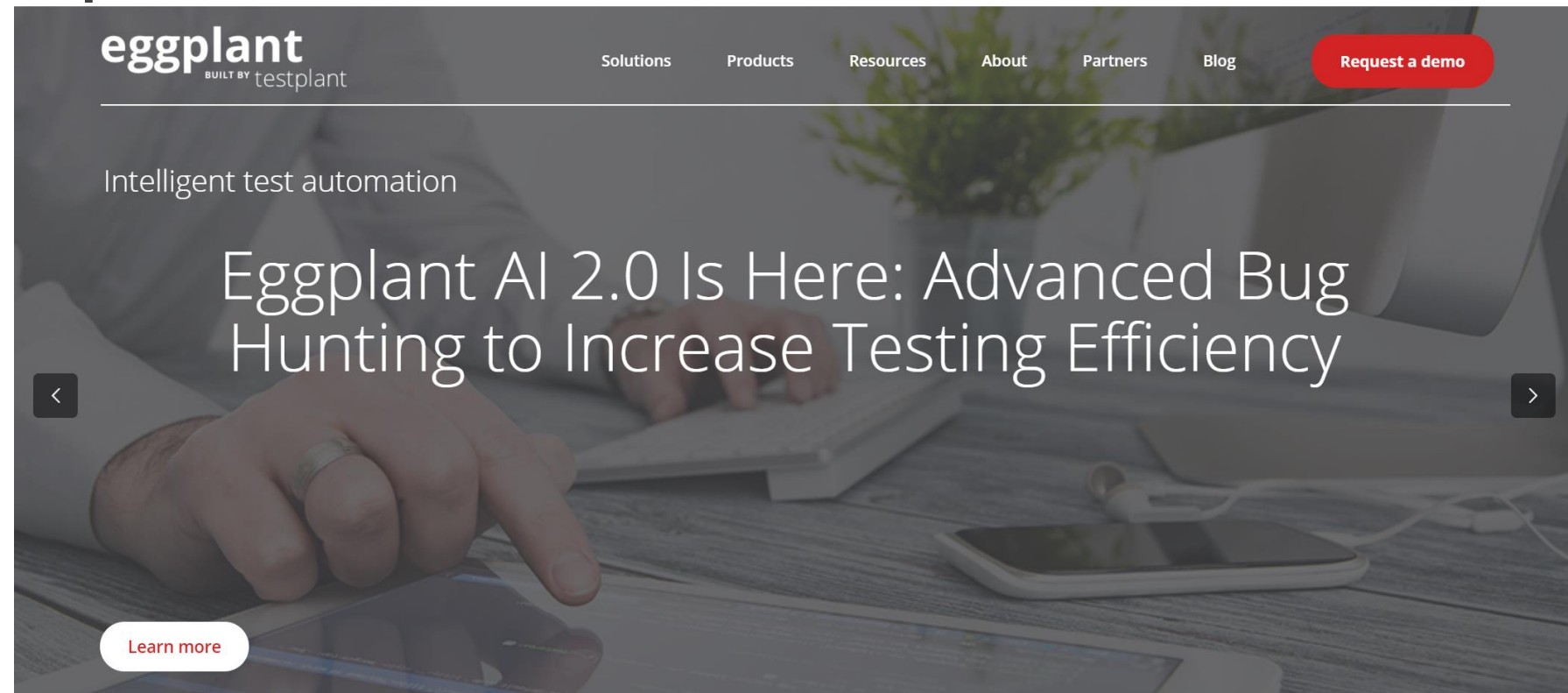
VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

[About](#)
[Screenshots](#)
[Downloads](#)
[Documentation](#)
 [End-user docs](#)
 [Technical docs](#)
[Contribute](#)
[Community](#)

Download VirtualBox 5.2

Fundamentos de la verificación y validación del software

<http://www.testplant.com/>

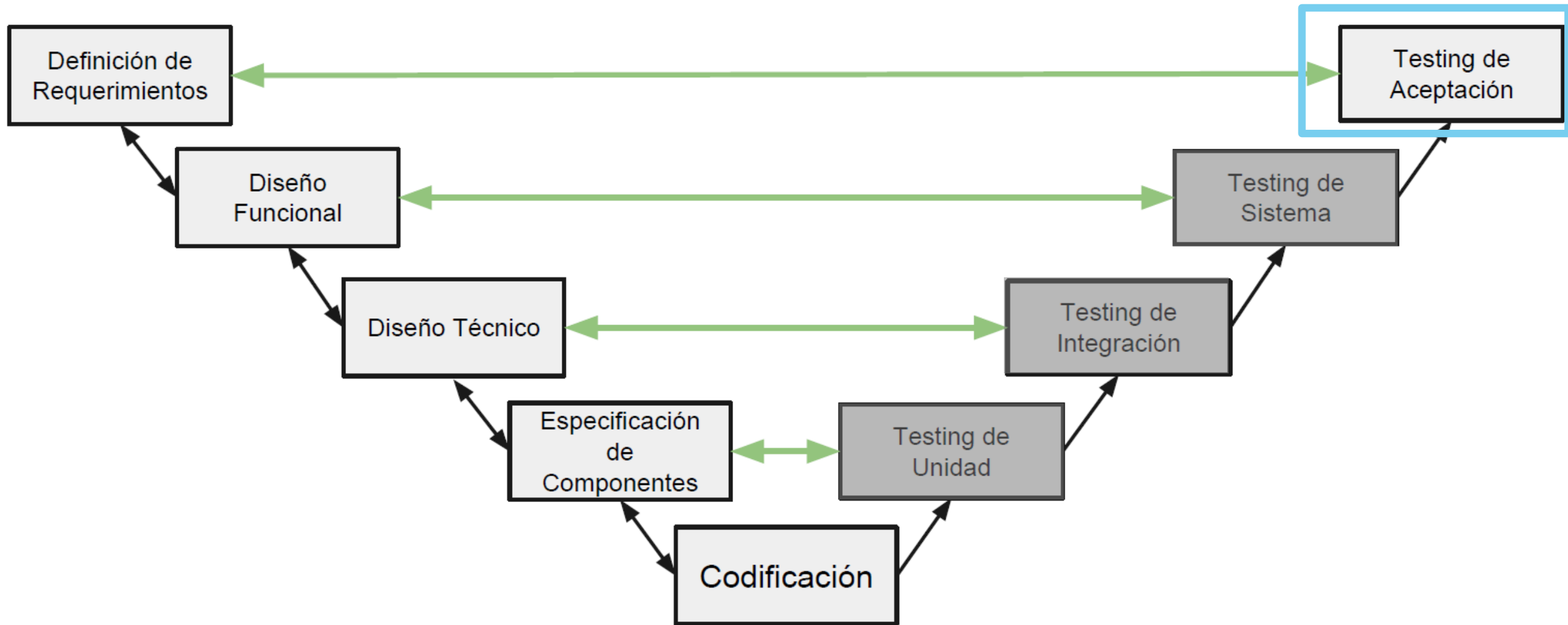


Fundamentos de la verificación y validación del software

Es fundamental la definición de los requerimientos del sistema, en esta etapa.

Deben estar claramente definidos.

Fundamentos de la verificación y validación del software



Fundamentos de la verificación y validación del software

Todas las etapas anteriores evalúan el software, antes de ser mostrado al cliente. El Test de Aceptación es uno de los pocos test que involucra al cliente.

Fundamentos de la verificación y validación del software

De acuerdo al tipo de desarrollo, el Test de Aceptación puede realizarse de diferentes formas y en diferentes momentos.

Fundamentos de la verificación y validación del software

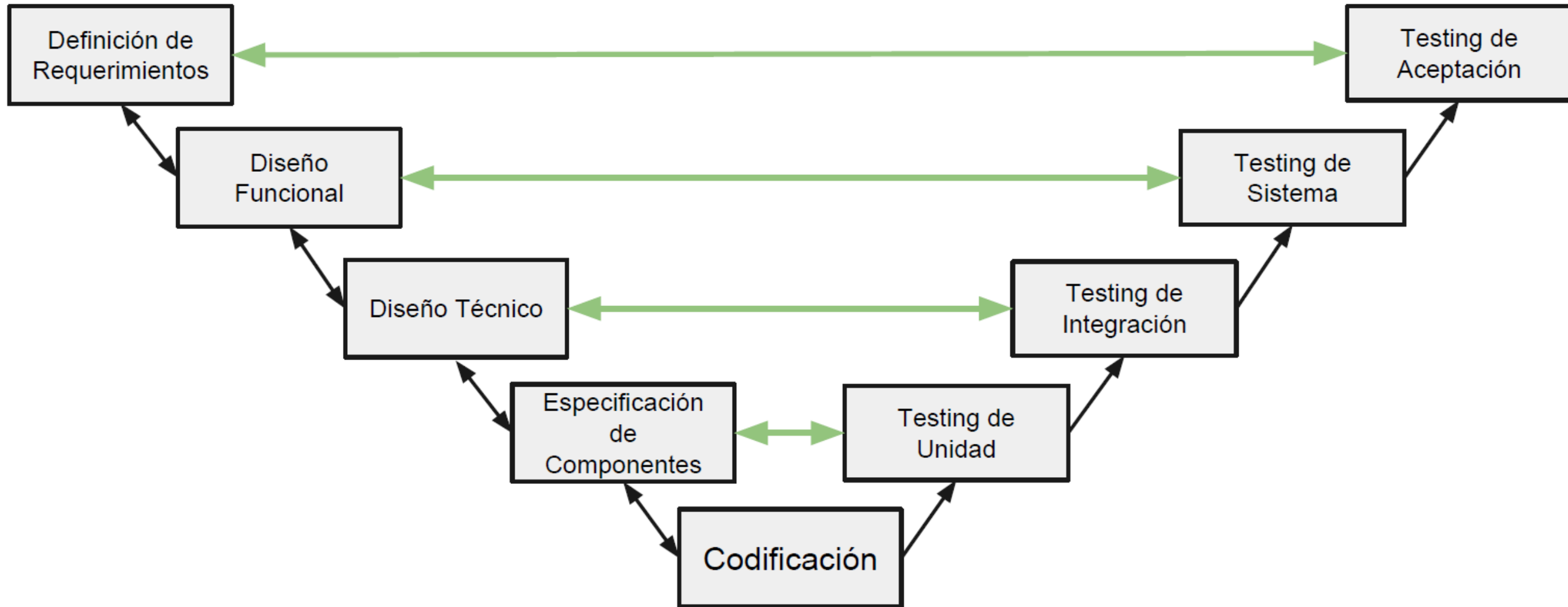
- El cliente y el usuario pueden ser dos personas diferentes.
- Aún teniendo la aceptación del cliente, se debe buscar la aceptación del usuario.
- Quienes administren el sistema (sysadmins) también
- deben aceptarlo.

Fundamentos de la verificación y validación del software

¿Cómo podemos evaluar si algo es útil? ¿Cómo definimos la utilidad de una cosa?

El concepto de Usabilidad en el software incluye muchos aspectos: Que sea fácil de usar, que sea fácil de aprender, que permita realizar tareas sin obstáculos.

Fundamentos de la verificación y validación del software

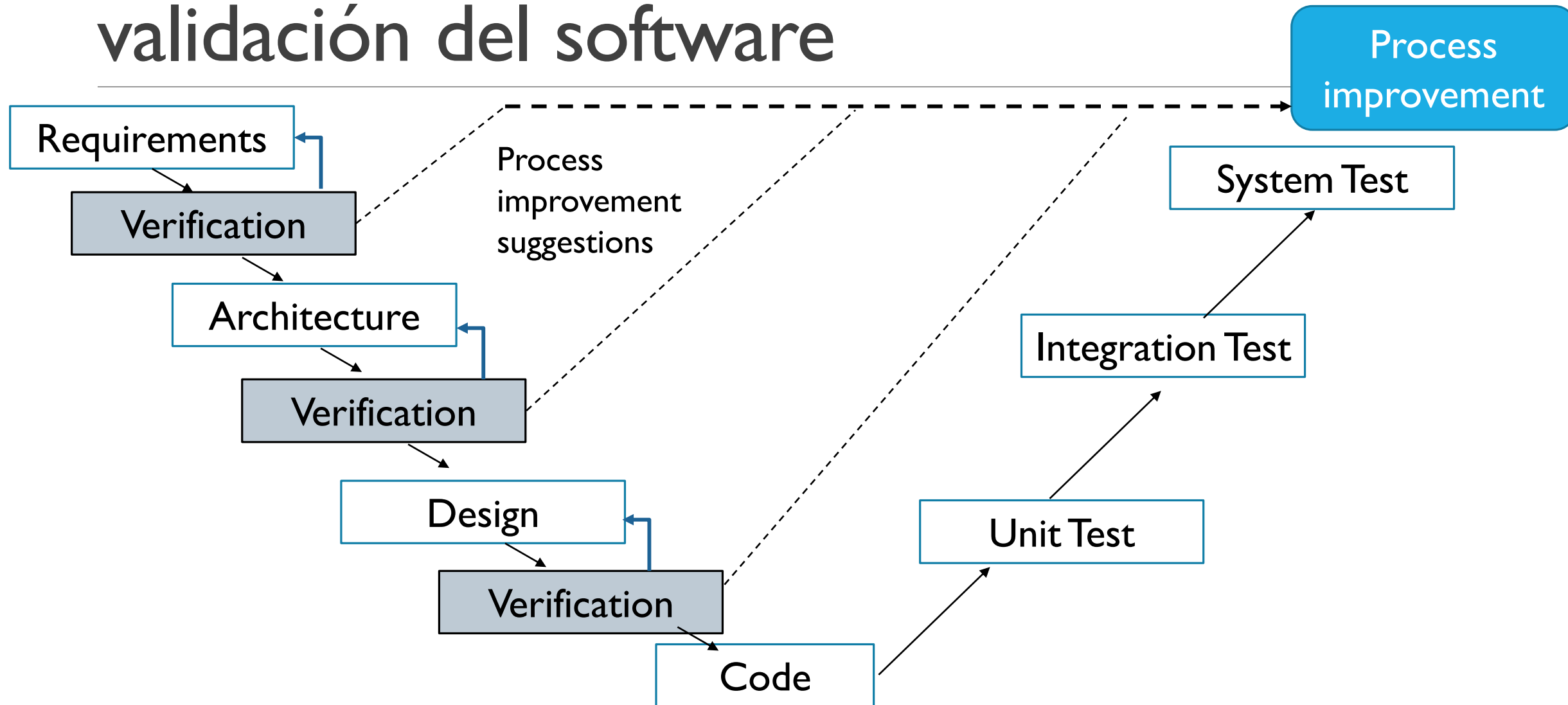


Fundamentos de la verificación y validación del software

Incorporando al Modelo V procesos de Verificación estática.

Gilb.T, Graham D.

Fundamentos de la verificación y validación del software



Fundamentos de la verificación y validación del software

Validación en las diferentes fases del proceso de desarrollo

Modelo de Fisher M.

Software Engineering Life-Cycle

Verification and Validation Life-Cycle

Requirements Phase

Design Phase

Implementation Phase

Test Phase

Requirements Phase

Design Phase

Implementation Phase

Test Phase

- Traceability Analysis
- Interface Analysis
- Requirements Analysis

- Traceability Analysis
- Interface Analysis
- Design Analysis

- Traceability Analysis
- Interface Analysis
- Code Analysis

- Traceability Analysis
- Interface Analysis
- Test Analysis

Fundamentos de la verificación y validación del software

Análisis de trazabilidad de requerimientos para la fase de requerimientos

Requirement No.	V&V Requirement
3.1.1.R	V&V debe asegurar que los requerimientos del sistema apropiados y los requerimientos del software se encuentren relacionados
3.1.2.R	V&V debe asegurar que los requerimientos del sistema se encuentren relacionados con los requerimientos del software correctos
3.1.3.R	V&V debe asegurar que las relaciones sean consistentes con el nivel de detalle

Fundamentos de la verificación y validación del software

Análisis de trazabilidad de requerimientos para la fase de diseño

Requirement No.	V&V Requirement
3.1.1.D	V&V debe asegurar que los requerimientos del software y los elementos del diseño se encuentren relacionados
3.1.2.D	V&V debe asegurar que los requerimientos del software se encuentren relacionados con los elementos del diseño correctos.
3.1.3.D	V&V debe asegurar que las relaciones sean consistentes con el nivel de detalle

Fundamentos de la verificación y validación del software

Análisis de trazabilidad de requerimientos para la fase de implementación

Requirement No.	V&V Requirement
3.1.1.1	V&V debe asegurar que los elementos del diseño y los elementos del código se encuentren relacionados
3.1.2.1	V&V debe asegurar que los elementos del software se encuentren relacionados con los elementos del código correctos.
3.1.3.1	V&V debe asegurar que las relaciones sean consistentes con el nivel de detalle

Fundamentos de la verificación y validación del software

Análisis de trazabilidad de requerimientos para la fase de Pruebas

Requirement No.	V&V Requirement
3.1.1.T	V&V debe asegurar que los elementos del código y los componentes de prueba se encuentren relacionados
3.1.2.T	V&V debe asegurar que todos los requerimientos de software apropiados y las pruebas de integración se encuentran relacionados
3.1.3.T	V&V debe asegurar los requerimientos apropiados del sistema y las pruebas del sistema están relacionadas.

Fundamentos de la verificación y validación del software

Análisis de trazabilidad de requerimientos para la fase de Pruebas

Requirement No.	V&V Requirement
3.1.4.T	V&V debe asegurar que todos los requerimientos conceptuales y las pruebas de aceptación están relacionadas.
3.1.5.T	V&V debe asegurar que los elementos del código están relacionados a los componentes de pruebas correctos.
3.1.6.T	V&V debe asegurar que los requerimientos de software están relacionados a las pruebas de integración correctas.
3.1.7.T	V&V debe asegurar que los requerimientos de sistema están relacionadas a las pruebas del sistema correctas

Fundamentos de la verificación y validación del software

Análisis de trazabilidad de requerimientos para la fase de Pruebas

Requirement No.	V&V Requirement
3.1.8.T	V&V debe asegurar que los requerimientos conceptuales están relacionados a las pruebas de integración correctas
3.1.9.T	V&V debe asegurar que las relaciones sean consistentes con el nivel de detalle

Gracias por su atención