

# Verificación y validación de software

# UNIDAD II

---

## Planeación 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 III*
  - Continuación Ciclo V.
    - ✓ Pruebas de Integración.
    - ✓ Pruebas de Sistema.
    - ✓ Pruebas de Aceptación.
- La verificación estática en las diferentes fases del proceso de desarrollo.
  - Validación
    - ✓ Validación en las diferentes fases del proceso de desarrollo.



# Fundamentos de la verificación y validación del software

---

## Pruebas de integración

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

# 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](http://mytoe.org/docs/Mars_closereading_art.pdf)

# 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

---

## Estrategias diferentes de integración incremental.

- 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

---

## Pruebas de sistema

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

---

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

---

<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

---

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

Deben estar claramente definidos.

# Fundamentos de la verificación y validación del software

---

## Pruebas de aceptación

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

---

- 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.

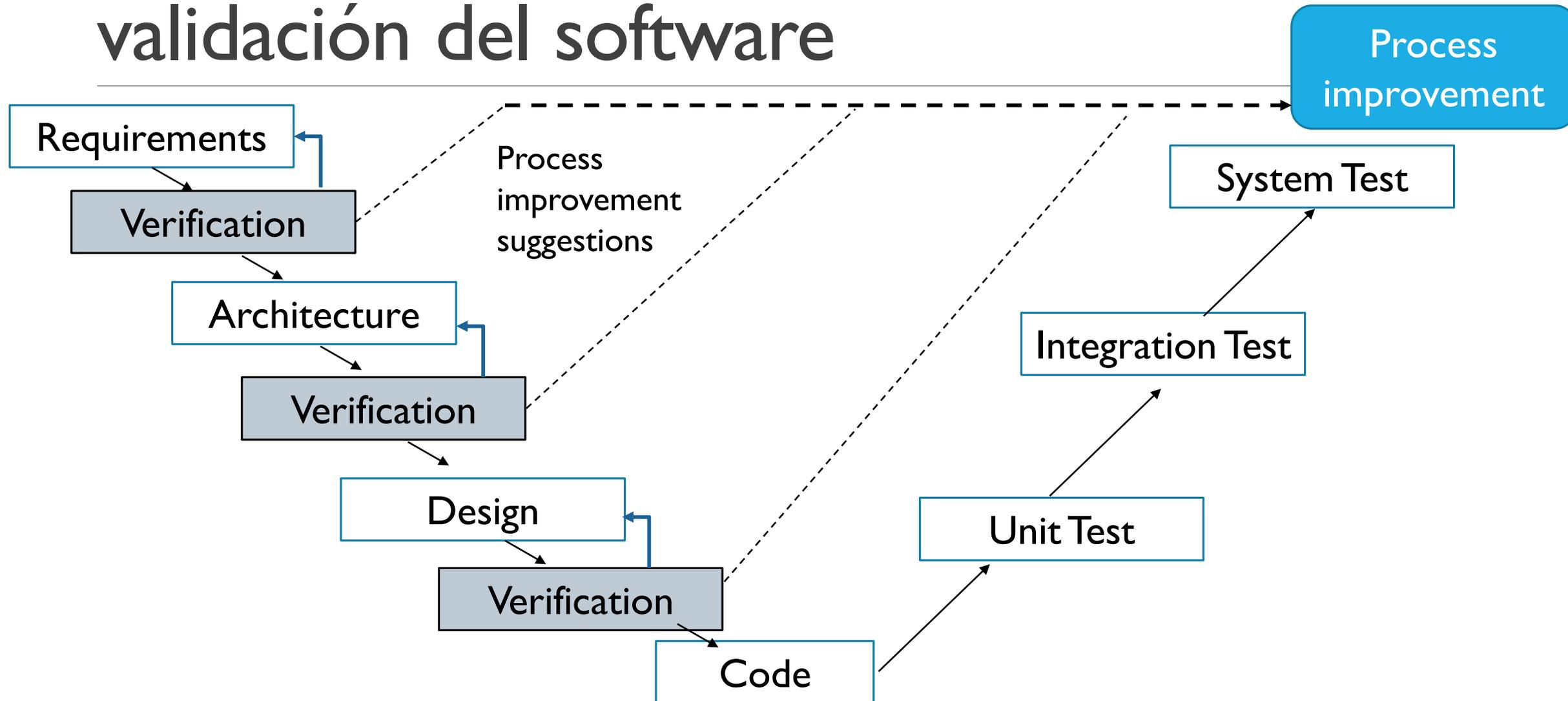
# Fundamentos de la verificación y validación del software

---

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

# Planeación de la verificación y validación del software

---

## Agenda

### **Unidad II.**

#### Planeación de la verificación y validación del software

- Propósito de la planeación.
- Asignación de recursos, roles y responsabilidades

#### Artefactos relacionados

# Planeación de la verificación y validación del software

---

- La planificación de la verificación y validación del sistema empieza desde etapas tempranas.
- Los planes derivan de la especificación y diseño del sistema.

# Planeación de la verificación y validación del software

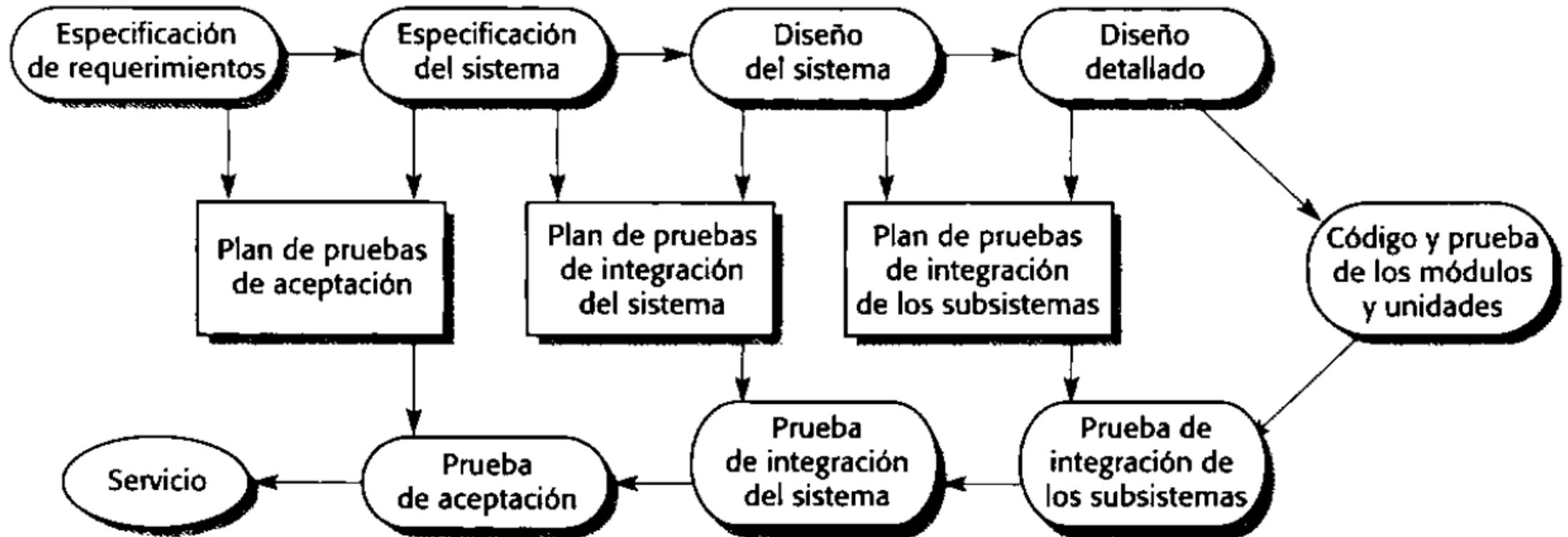
---

- La planeación de V&V:
  - ✓ Asegura que los recursos V&V sean identificados y asignados de forma eficiente.
  - ✓ Establece las metas y objetivos de la V&V.
  - ✓ Debe ser observada a lo largo del proceso de desarrollo, para mantener acciones realistas que puedan ser establecidas y llevadas a cabo.

# Planeación de la verificación y validación del software

- Plan de pruebas. Somerville.

Planes de pruebas como un enlace entre las pruebas y el desarrollo



# Planeación de la verificación y validación del software

---

- Es parte del proceso V&V
  - Definición de estándares y procedimientos para las inspecciones y pruebas del sistema.
  - Listas de comprobación para conducir las inspecciones.
  - Definición del plan de pruebas del software.
  - Definición de recursos de hardware y software.
  - Planes de contingencia (para solucionar desajustes en implementación y diseño)

# Planeación de la verificación y validación del software

---

- Regla general:
- Entre más crítico el sistema, dedicar mayor esfuerzo a la verificación estática.

# Planeación de la verificación y validación del software

---

- Los planes de verificación y validación:
  - Ayudan a los gestores a asignar recursos.
  - Estimar calendario de pruebas.
  - Ayudan al personal técnico a obtener una panorámica general de las pruebas del sistema y ubicar su trabajo en este contexto.

# Planeación de la verificación y validación del software

---

- Principales componentes de un plan de verificación y validación para un sistema grande y complejo, Somerville:

# Planeación de la verificación y validación del software

---

- Estructura de un plan de pruebas (SOMMERVILLE)
- **El proceso de prueba**
  - Una descripción de las principales fases del proceso de pruebas.

# Planeación de la verificación y validación del software

---

- Estructura de un plan de pruebas
- **Trazabilidad de requerimientos**
  - Los usuarios son los más interesados en que el sistema satisfaga sus requerimientos y las pruebas deberían planificarse para que todos los requerimientos se prueben individualmente.

# Planeación de la verificación y validación del software

---

- Estructura de un plan de pruebas
- **Elementos probados**
  - Deberían especificarse los elementos del software que tienen que ser probados.

# Planeación de la verificación y validación del software

---

- Estructura de un plan de pruebas
- **Calendario de pruebas**
  - Un calendario de todas las pruebas y la asignación de recursos para este calendario se enlaza, con la agenda general del desarrollo del proyecto.

# Planeación de la verificación y validación del software

---

- Estructura de un plan de pruebas
- **Procedimientos de registro de pruebas**
  - No es suficiente con ejecutar simplemente las pruebas; los resultados de las pruebas deben ser registrados sistemáticamente. Debe ser posible auditar el proceso de pruebas para comprobar que se ha llevado a cabo correctamente.

# Planeación de la verificación y validación del software

---

- Estructura de un plan de pruebas
- **Requerimientos de hardware y software**
  - Esta sección debe determinar las herramientas software requeridas y la utilización estimada del hardware.

# Planeación de la verificación y validación del software

---

- Estructura de un plan de pruebas
  
- **Restricciones**
  - En esta sección deberían anticiparse las restricciones que afectan el proceso de pruebas como la escasez de personal.

# Planeación de la verificación y validación del software

---

- Estructura de un plan de pruebas
- El proceso de prueba
- Calendario de pruebas
- Elementos probados
- Trazabilidad de requerimientos
- Procedimientos de registro de pruebas
- Requerimientos de HW y SW
- Restricciones

# Planeación de la verificación y validación del software

---

- Para sistemas pequeños puede utilizarse un plan de verificación y validación menos formal.
- Aún así se requiere un documento formal que soporte el proceso de planificación.

# Planeación de la verificación y validación del software

---

- El plan de V&V evoluciona durante el proceso de desarrollo debido en retrasos en otras etapas.
- Si un componente necesario para el proceso de V&V no se encuentra listo se debe hacer un ajuste de recursos humanos.

# Planeación de la verificación y validación del software

---

- La planeación puede ser organizada en una serie de pasos, como se muestra a continuación, Fisher M:

# Planeación de la verificación y validación del software

Paso del proceso de planeación	Descripción
Establecimiento de la misión V&V	Desarrollar el objetivo del proyecto de V&V
Identificar los stakeholders de la V&V	Identificar aquellas entidades que tienen un interés establecido en el proyecto V&V.
Identificar los requerimientos de los stakeholders	Identificar las necesidades y expectativas de los stakeholders.
Establecer los objetivos de la V&V	Establecer de 5 a 7 resultados que definan el éxito del proyecto de V&V
Desarrollar un concepto	Desarrollar un enfoque de alto nivel que represente la forma en la que el proyecto V&V operará.

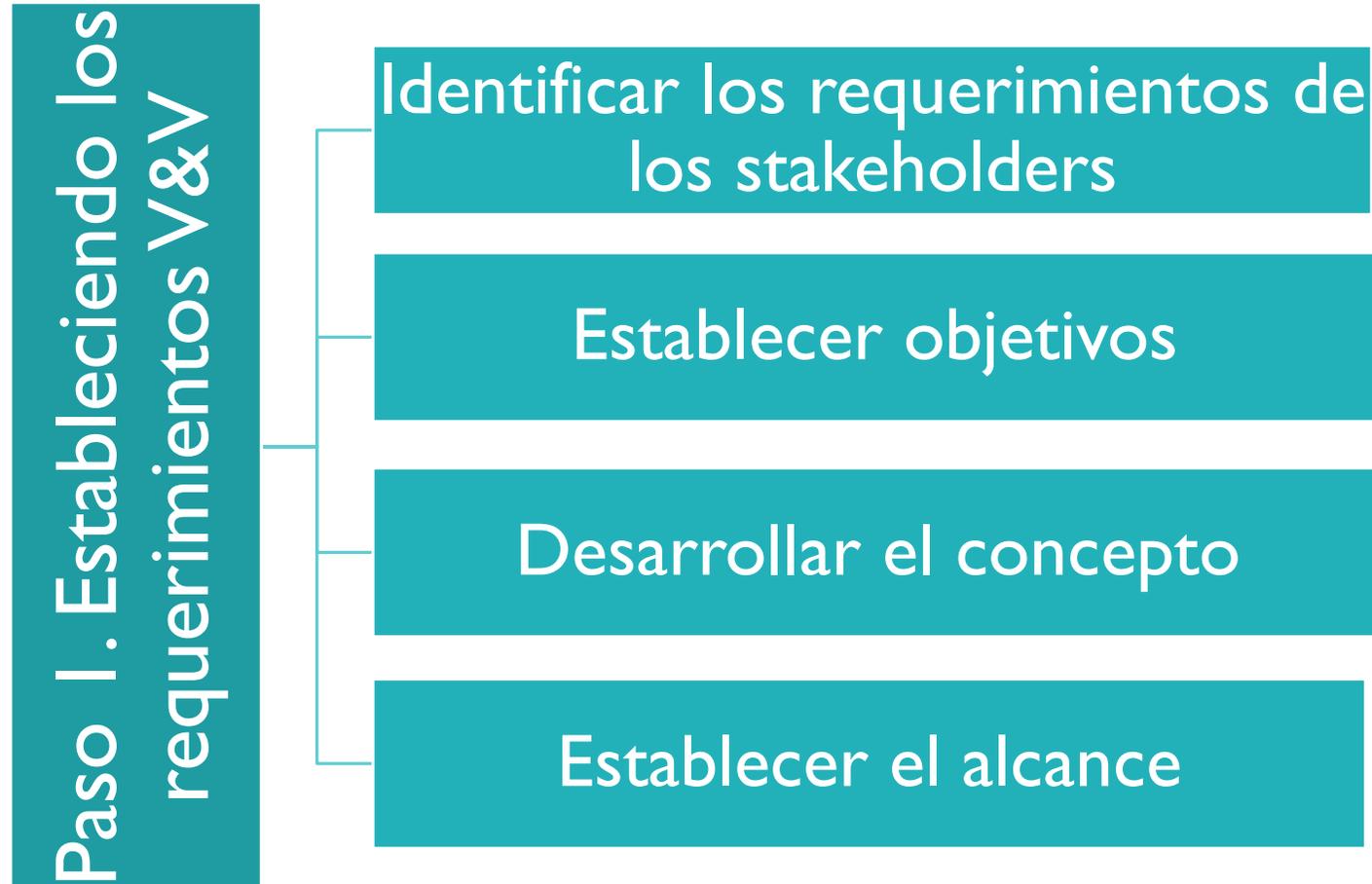
# Planeación de la verificación y validación del software

Paso del proceso de planeación	Descripción
Desarrollar los requerimientos V&V	Usando los requerimientos de los stakeholders, el concepto V&V, el concepto V&V y el estándar de los requerimientos V&V, desarrollar los requerimientos del sistema para el proyecto V&V.
Establecer el alcance de la V&V	Usando los objetivos V&V, identificar los componentes de software que serán valorados durante el proyecto.
Desarrollar la Estructura de Desglose de Trabajo (EDT)	Identificar los productos que resultarán del esfuerzo de la V&V y las tareas responsables del desarrollo de productos.
Desarrollar el diagrama de red	Usando los componentes de la EDT, desarrollar las relaciones temporales entre ellos.
Estimación de recursos	Usando el enfoque Bottom-Up, estimar el presupuesto usando los paquetes de trabajo.
Desarrollar el plan del proyecto	Usando los resultados de los pasos anteriores, desarrollar el plan del proyecto.

# Estableciendo los requerimientos V&V

# Planeación de la verificación y validación del software

---



# Planeación de la verificación y validación del software

---

- Paso I. Estableciendo los requerimientos V&V
  - El backbone de cualquier proyecto es el alcance técnico del trabajo a realizar, o requerimientos.
  - El alcance técnico puede crear proyectos o romper proyectos.
  - No basta con establecer el alcance apropiado, sino, hacerlo de acuerdo a las metas y objetivos.

# Planeación de la verificación y validación del software

---

- Paso I. Estableciendo los requerimientos V&V
  - Al menos el 10% o 15% del presupuesto del proyecto debería ser empleado a la planeación, incluyendo la definición de los requerimientos

# Planeación de la verificación y validación del software

---

- Paso 1. Estableciendo los requerimientos V&V
  - Marcus Fisher incluye en su libro *Software Verification and Validation*, dos enfoques para la planeación de un proyecto V&V:
    - Enfoque de ingeniería en sistemas
    - Estándar 1012 del Institute of Electrical and Electronics Engineers (IEEE).

# Planeación de la verificación y validación del software

---

- Paso I. Estableciendo los requerimientos V&V
  - Enfoque de Ingeniería en Sistemas

# Planeación de la verificación y validación del software

---

- El equipo V&V debe definir objetivos medibles para definir lo que significa un proyecto V&V exitoso.
- Asimismo debe identificar los stakeholders: se dividen en activos y pasivos.

# Planeación de la verificación y validación del software

---

- Stakeholders activos: Entidades que han establecido interés e interactuarán con el proyecto V&V.
- La interacción puede ser usando los resultados de la V&V para comunicarlos al equipo.

# Planeación de la verificación y validación del software

---

- Stakeholders pasivos: Entidades interesadas en el proyecto V&V y pueden influenciar de alguna manera el éxito del esfuerzo.
- La influencia puede variar desde estándares que afectan el desarrollo del trabajo hasta administradores de la red que determinan las herramientas que pueden ser instaladas.

# Planeación de la verificación y validación del software

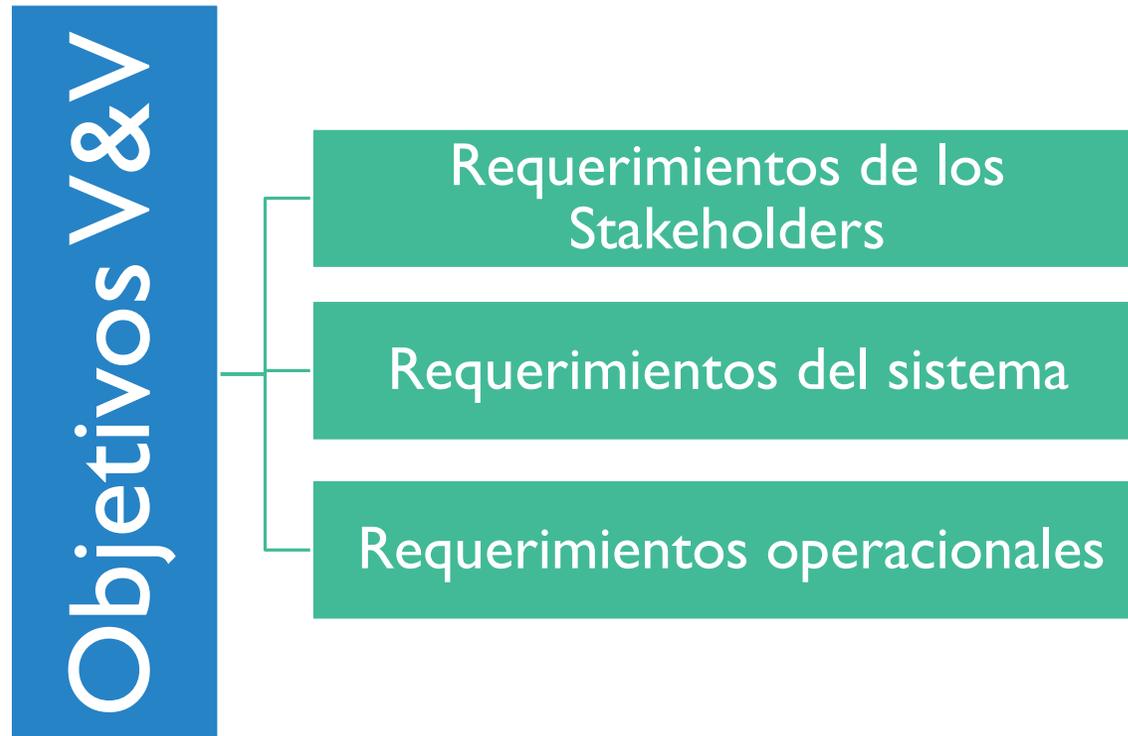
---

- Una vez que los stakeholders fueron identificados, es necesario obtener los requerimientos o expectativas del esfuerzo de V&V.
- A través de entrevistas, encuestas o investigación de proyectos V&V previos.

# Planeación de la verificación y validación del software

---

## Generación de objetivos V&V



# Planeación de la verificación y validación del software

---

- Ejemplo:
  - Proyecto MUGSEY 0x01

Mugsey 0x01 es un proyecto universitario que experimenta con el uso de la tecnología convencional para alcanzar la órbita baja de la Tierra.

Consiste básicamente en una plataforma unida a un globo de helio.

# Planeación de la verificación y validación del software

---

- Identificar el objetivo → Identificar la necesidad o el problema que el equipo V&V debe resolver.

# Planeación de la verificación y validación del software

---

- Objetivo V&V MUGSEY

El objetivo del proyecto es garantizar que el software mantenga la salud del sistema, adquiera y mantenga datos operativos y esté adecuadamente desarrollado para permitir de manera eficiente misiones futuras.

# Planeación de la verificación y validación del software

---

- Stakeholders identificados para el proyecto MUGSEY 0x01:
  - ✓ Administrador
  - ✓ Desarrolladores
  - ✓ Equipo de operación
  - ✓ Científicos universitarios
  - ✓ Sociedad de Estudios Gravitacionales
  - ✓ Administradores de la red de la Universidad
  - ✓ Departamento de Ingeniería de Software

# Planeación de la verificación y validación del software

---

- Actividad:

Identificar el objetivo y los stakeholders de su proyecto V&V.

---

# Gracias por su atención