

# Verificación y validación de software

# UNIDAD II

---

## Planeación de la verificación y validación del software

# Planeación de la verificación y validación del software

---

## Unidad II.

### Planeación de la verificación y validación del software

- Propósito de la planeación.
  - Establecer el Plan V&V

# 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.

# Planeación de la verificación y validación del software

---

Una vez que se ha establecido el objetivo, los requerimientos y el alcance:

- ¿Cómo se alcanzarán estos objetivos?
- Diagrama de Red
- Diagrama de Gantt

# Planeación de la verificación y validación del software

---

<b>3.2</b>	<b>Análisis de Interfaz</b>
<b>3.2.1</b>	V&V debe asegurar que han sido identificados los elementos de interfaz correctos.
<b>3.2.2</b>	V&V debe asegurar que todos los elementos de interfaz están completamente definidos.
<b>3.2.3</b>	V&V debe asegurar que cada elemento de interfaz es empleado consistentemente.
<b>3.2.4</b>	V&V debe asegurar que los elementos de la interfaz mantienen el rendimiento necesitado por el sistema.
<b>3.2.5</b>	V&V debe asegurar que todos los elementos de la interfaz son testeables.

# Planeación de la verificación y validación del software

---

Cada vez que la información debe ser movilizada de un módulo a otro, un procedimiento a otro, etc. existe la posibilidad de que se corrompa o sea empleada de manera errónea.

Más aún cuando los módulos son desarrollados por diferentes equipos de desarrollo.

# Planeación de la verificación y validación del software

---

Normalmente, los módulos de un sistema son construidos bajo alguna especificación que dicta la forma en la que se comunicará con otros módulos, por ejemplo, el Documento de Control de Interfaz (Interface Control Document - ICD)

# Planeación de la verificación y validación del software

---

El equipo V&V debe asegurar:

- ❖ Que se han identificado los elementos correctos de la interfaz, se encuentran definidos completamente, son usados de forma consistente, mantienen el rendimiento que el sistema requiere y pueden ser verificados a través del testeo.

# Planeación de la verificación y validación del software

---

Tareas genéricas:

**Tarea 1.** Identificar los datos que deben ser transferidos entre módulos.

**Tarea 2.** Identificar las interfaces que manejan las transacciones de los datos.

**Tarea 3.** Comparar las interfaces definidas por la V&V con las de los desarrolladores y evaluar las inconsistencias.

# Planeación de la verificación y validación del software

---

**Tarea 4.** Analizar cada dato y determinar si se encuentra definido completamente.

**Tarea 5.** Graficar la ubicación de los datos y determinar si se están empleando de forma consistente.

**Tarea 6.** Identificar las necesidades de rendimiento del sistema.

# Planeación de la verificación y validación del software

---

**Tarea 7.** Modelar y simular la comunicación entre las interfaces para determinar si mantienen el rendimiento requerido.

**Tarea 8.** Desarrollar pruebas para las interfaces e identificar cuáles no son testeables.

# Planeación de la verificación y validación del software

---

La Tarea 1 y 2,

**Tarea 1.** Identificar los datos que deben ser transferidos entre módulos.

**Tarea 2.** Identificar las interfaces que manejan las transacciones de los datos.

Identifican los datos y las interfaces necesarias para el sistema y para los usuarios.

Se genera la información necesaria para que los analistas V&V comparen los datos y las interfaces identificadas por el equipo de desarrollo (Tarea 3).

**Tarea 3.** Comparar las interfaces definidas por la V&V con las de los desarrolladores y evaluar las inconsistencias.

# Planeación de la verificación y validación del software

---

**Tarea 4.** Analizar cada dato y determinar si se encuentra “definido completamente”.

Los analistas V&V deben asegurar que para cada dato, se provee la siguiente información:

- Las unidades de medida de los datos
- La precisión requerida de los datos
- El rango de valores que los datos pueden tomar.
- El tiempo en el que los datos deben ser procesados.
- La fuente de los datos.
- El destino de los datos.

# Planeación de la verificación y validación del software

---

Por ejemplo.

Lecturas de temperatura. Proyecto MUGSEY.

Si las lecturas de temperatura son transferidas entre módulos y el ICD define TEMPERATURA como el dato ítem, los analistas V&V deben asegurar que se ha identificado:

# Planeación de la verificación y validación del software

---

- El sistema de medida en el que la temperatura se encuentra representada (p.e. Celcius).
- La precisión requerida y los valores máximos y mínimos en los que la temperatura debe ser registrada.
- Requerimientos en los cuáles se indique el tiempo en el que las lecturas de temperatura deben ser guardadas en sistema.
- Por último, la fuente del dato y su destino.

# Planeación de la verificación y validación del software

---

**Tarea 5.** Graficar la ubicación de los datos y determinar si se están empleando de forma consistente.

Tarea 5. Consiste en mapear todos los usos que se le dan a cada dato.

Una vez que los analistas han determinado dónde se está usando el dato, pueden evaluar si está siendo usado de forma consistente.

\* Uso de herramientas → Análisis estático

# Planeación de la verificación y validación del software

---

Las Tareas 6 y 7,

**Tarea 6.** Identificar las necesidades de rendimiento del sistema.

**Tarea 7.** Modelar y simular la comunicación entre las interfaces para determinar si mantienen el rendimiento requerido.

Son difíciles de definir de forma genérica. Problema a resolver:

**¿Las interfaces mantienen el rendimiento del sistema?**

# Planeación de la verificación y validación del software

---

Por ejemplo: Si el sistema procesa ciertos comandos 10 segundos después de recibirlos, el equipo V&V debe asegurar que las interfaces definidas pueden lograr este rendimiento.

Para ello, deben tener en cuenta el camino de ejecución completo:

- I. Recepción de la señal
- II. Reconocimiento del comando
- III. Manejo de interrupciones
- IV. Rutear el comando a la terminal apropiada.
- V. El tiempo de procesamiento en la terminal.

# Planeación de la verificación y validación del software

---

Si el analista espera a la fase de Desarrollo para testear este rendimiento, no habrá aportación alguna para la fase de Análisis y Diseño.

Por lo tanto, pueden empelarse enfoques de Análisis dinámicos que requieren el modelado y simulación de las interfaces. Aunque esto tiene un costo.

# Planeación de la verificación y validación del software

---

**Tarea 8.** Desarrollar pruebas para las interfaces e identificar cuáles no son testeables.

- ✓ Requiere todas las interfaces V&V para ser pre-testeadas.
- ✓ Pre-testear significa que el equipo V&V debe desarrollar pruebas para las interfaces de interés y poner banderas en aquellas que no pueden ser testeadas.

# Planeación de la verificación y validación del software

---

## Ejemplo Proyecto MUGSEY

- Requerimientos V&V para el análisis de interfaz.

# Planeación de la verificación y validación del software

---

<b>3.2</b>	<b>Análisis de Interfaz</b>
<b>3.2.1</b>	V&V debe asegurar que han sido identificados los elementos de interfaz correctos.
<b>3.2.2</b>	V&V debe asegurar que todos los elementos de interfaz están completamente definidos.
<b>3.2.3</b>	V&V debe asegurar que cada elemento de interfaz es empleado consistentemente.
<b>3.2.4</b>	V&V debe asegurar que los elementos de la interfaz mantienen el rendimiento necesitado por el sistema.
<b>3.2.5</b>	V&V debe asegurar que todos los elementos de la interfaz son testeables.

# Planeación de la verificación y validación del software

---

Ej: Objetivo V&V Project.

Proveer la seguridad de que el sistema de software **Identifica y gestiona las fallas de forma adecuada.**

# Planeación de la verificación y validación del software

---

Durante la fase de requerimientos el equipo V&V debe proveer la seguridad de que el software soporta las interfaces adecuadas para la identificación y manejo de errores.

# Planeación de la verificación y validación del software

---

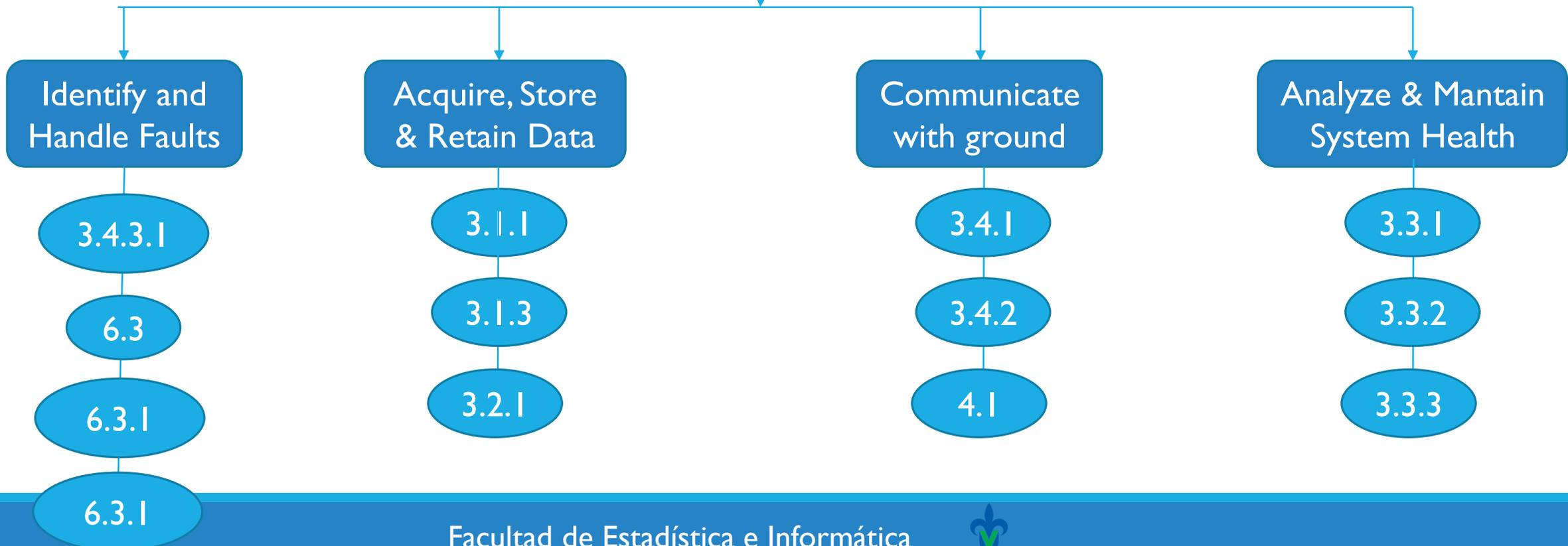
Los requerimientos del sistema que el equipo V&V debe evaluar son:

- 3.4.3.1 Comandos almacenados
- 6.3 Manejo de errores
- 6.3.1 Errores científicos
- 6.3.2 Abortar misión

# V&V Goal

The goal of the V&V Project is to assure that the software maintains the system's health, acquires and maintains operational data and is adequately developed to efficiently enable future missions.

## V&V Objectives



# Planeación de la verificación y validación del software

---

- Para los requerimientos del sistema MUGSEY se determinó que los siguientes requerimientos están relacionados con el objetivo:
  - ✓ Requerimiento 3.4.3.1 – **Stored Commanding**. MUGSEY 0x01 debe procesar una secuencia de comandos al ingresar un escenario de falla.
  - ✓ Requerimiento 6.3 – **Fault Handling**. MUGSEY 0x01 debe reconocer cuando sus subsistemas no están respondiendo y recuperarlos en 5 minutos.

# Planeación de la verificación y validación del software

---

- Para los requerimientos del sistema MUGSEY se determinó que los siguientes requerimientos están relacionados con el segundo objetivo:
  - ✓ Requerimiento 6.3.1 – **Science Faults**. Ninguna falla en el sistema evitará que se tomen las lecturas científicas (imagen, temperatura, altitud, posición) durante no más de 5 minutos.
  - ✓ Requerimiento 6.3.2 – **Abort Mission**. MUGSEY 0x01 debe reconocer cuando entra a una zona peligrosa y abortar la misión.

# Planeación de la verificación y validación del software

<b>V&amp;V Task</b>	<b>V&amp;V Requirement Fulfilled</b>	<b>V&amp;V Approach</b>	<b>Duration Days (Min, Avg, Max)</b>
Identify a potential faults	None	Static Analysis {Use cases and Scenarios}	(1,2,3)
Identify the interfaces required to identify and handle these faults	None	Static Analysis {Use cases and Scenarios}	(2,4,6)
Identify the data ítems that should be passed between interfaces	None	Static Analysis {Use cases and Scenarios}	(3,7,10)
Compare and contrast the results with the system and software interface requirements	3.2.1	Manual Analysis	(5,7,12)

# Planeación de la verificación y validación del software

<b>V&amp;V Task</b>	<b>V&amp;V Requirement Fulfilled</b>	<b>V&amp;V Approach</b>	<b>Duration Days (Min, Avg, Max)</b>
Determine if each data ítem is completely defined	3.2.2	Manual Analysis	(1,3,5)
Graph the locations of each data ítem and determine if they are used consistently	3.2.3	Static Analysis	(2,7,12)
Add temporal properties to the data ítems and simulate transactions	3.2.4	Dynamic Analysis	(7,14,24)
Develop test cases data ítems and fault scenarios and determine if any of them can not be tested	3.2.5	Static Analysis	(2,3,4)

# Planeación de la verificación y validación del software

---

Los 4 enfoques posibles para la tercera columna “V&V Approach” son:

- Análisis manual.
- Análisis estático.
- Análisis dinámico.
- Análisis formal.

# Planeación de la verificación y validación del software

---

- Casos de uso y escenarios (p.e. Diagramas de secuencia) son una excelente herramienta.
- Diagramas de flujo de datos.

---

# Gracias por su atención