

# Verificación y validación de software

---

---

# Fundamentos de la verificación y validación del software

# Fundamentos de la verificación y validación del software

---

## Agenda:

- Conceptos de Verificación y Validación
- El papel de las pruebas de software y sus técnicas dinámicas en la verificación
- Técnicas estadísticas para la verificación

# Fundamentos de la verificación y validación del software

---

Pressman:

- **Verificación** se refiere al conjunto de tareas que garantizan que el software implementa correctamente una función específica.
- **Validación** es un conjunto diferente de tareas que aseguran que el software que se construye sigue los requerimientos del cliente.

■ Boehm:

**Verificación:** ¿Estamos construyendo el producto correctamente?

**Validación:** ¿Estamos construyendo el producto correcto?

# Fundamentos de la verificación y validación del software

---

Sommerville

## Verificación

- Busca comprobar que el sistema cumple con los requerimientos especificados (funcionales y no funcionales)
- ¿El software está de acuerdo con su especificación?

## Validación

- Busca comprobar que el software hace lo que el usuario espera.
- ¿El software cumple las expectativas del cliente?

# Fundamentos de la verificación y validación del software

---

Ghezzi

## Verificación

- o Todas las actividades que son llevadas a cabo para averiguar si el software cumple con sus objetivos

# Fundamentos de la verificación y validación del software

---

IEEE

## Verificación

- o The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of the phase

## Validación

- o The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements

# Fundamentos de la verificación y validación del software

---

Con la Verificación y Validación **V&V** se pretende certificar que el sistema desarrollado se ajusta a lo esperado de la forma mas satisfactoria posible.

Nunca se puede demostrar que está libre de defectos

# Fundamentos de la verificación y validación del software

---

La verificación y la validación incluyen un amplio arreglo de actividades SQA:

- Revisiones técnicas
- Auditorías de calidad y configuración
- Monitoreo de rendimiento
- Simulación
- Estudio de factibilidad
- Revisión de documentación
- Revisión de base de datos
- Análisis de algoritmos
- **Pruebas de desarrollo, pruebas de usabilidad, pruebas de calificación, pruebas de aceptación y pruebas de instalación.**

Aunque las pruebas juegan un papel extremadamente importante en V&V, también son necesarias muchas otras actividades.

# Fundamentos de la verificación y validación del software

---

- ❖ Las pruebas representan la última fortaleza desde donde puede valorarse la calidad y descubrirse errores.
- ❖ Pero las pruebas no deben verse como una red de seguridad.
- ❖ Como se dice: “no se puede probar la calidad. Si no está ahí antes de comenzar las pruebas, no estará cuando termine de probar”.
- ❖ La calidad se incorpora en el software a lo largo de todo el proceso de ingeniería del software.
- ❖ La adecuada aplicación de métodos y herramientas, revisiones técnicas efectivas, y gestión y medición sólidas conducen a la calidad que se confirma durante las pruebas.

# Fundamentos de la verificación y validación del software

---

**La prueba de software es un elemento de un tema más amplio que usualmente se conoce como verificación y validación**

# Fundamentos de la verificación y validación del software

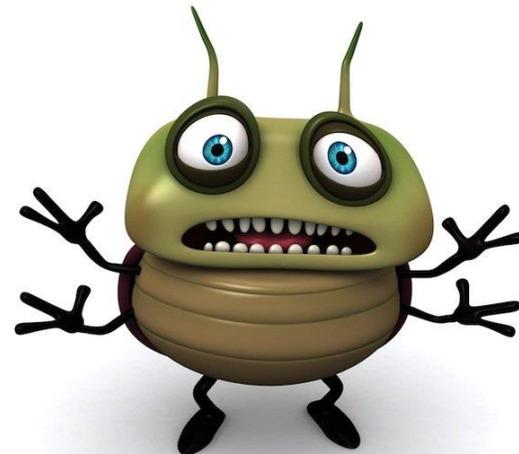
---

## El papel de las pruebas de software y sus técnicas dinámicas en la verificación

# Fundamentos de la verificación y validación del software

---

- En un proyecto de desarrollo de software los errores pueden presentarse en cualquiera de las etapas del ciclo de vida del software.
- Aún cuando se intente detectarlos después de cada fase utilizando técnicas como la inspección, algunos errores permanecen sin ser descubiertos.



# Fundamentos de la verificación y validación del software

---

Por lo tanto es muy probable que el código final contenga errores de requerimientos y diseño, adicionales a los introducidos en la codificación.



# Fundamentos de la verificación y validación del software

---

- ❖ Las pruebas de software son una parte importante pero muy costosa del proceso de desarrollo de software
- ❖ Pueden llegar a representar entre el 30 y 50 % del costo total del desarrollo del software [Myers, 2004]
- ❖ Sin embargo, los costos de las fallas en un software en operación pueden llegar a ser mucho mayores (catastróficos)

# Fundamentos de la verificación y validación del software

---

Las pruebas de software tienen un rol muy importante en el **aseguramiento de la calidad**, ya que permiten detectar los errores introducidos en las fases previas del proyecto



# Fundamentos de la verificación y validación del software

---

## ■ ¿Cómo se determina que un sistema tiene mala calidad?

- Sólo se requiere de tres a cuatro defectos por cada 1, 000 líneas de código para que un programa tenga mal desempeño.
- La mayoría de los programadores cometen un error en cada 10 líneas de código que escriben.

Pressman, 2002

La corrección de los errores cuesta a los vendedores de software al menos la mitad de sus presupuestos de desarrollo durante las pruebas.

# Fundamentos de la verificación y validación del software

---

## ■ Costo de la calidad

*“sabemos que la calidad es importante, pero cuesta tiempo y dinero —demasiado tiempo y dinero— lograr el nivel de calidad en el software que en realidad queremos.”*

La mala calidad también tiene un costo

Usuarios

Fabricantes

# Fundamentos de la verificación y validación del software

---

- **Costos de la calidad**

- *Costos de prevención*
- *Costos de evaluación*
- *Costos de falla*
  - *Costos de falla internos*
  - *Costos de falla externos*

# Fundamentos de la verificación y validación del software

---

## ■ Costos de la calidad

- El costo promedio por corregir un defecto durante la **codificación** es aproximadamente de **US\$977** por error.
- El promedio del costo por corregir el mismo error si se descubre durante las **pruebas** del sistema es de **US\$7, 136**.
- **Según Cigital: una aplicación grande contiene 200 errores**

$$(200 \times \text{US\$}977) = \text{US\$}195\,400$$

Pressman 2002

# Fundamentos de la verificación y validación del software

---

**Algunos de los peores errores de la historia**

# Fundamentos de la verificación y validación del software

---

Se colapsa el aeropuerto de Los Angeles (2007).

Más de 17 mil personas se quedaron en tierra por un problema de software que provocó conflictos en una tarjeta de red que bloqueó toda la red informática



# Fundamentos de la verificación y validación del software

---

El lanzamiento comercial y la producción del Airbus A380 se retrasa más un año (2006)

Diferencias entre versiones de las herramientas CAD (Computer Aided Design) usadas en las fábricas de Hamburgo y Toulouse provocaron un problema en el cableado (530km de cables)



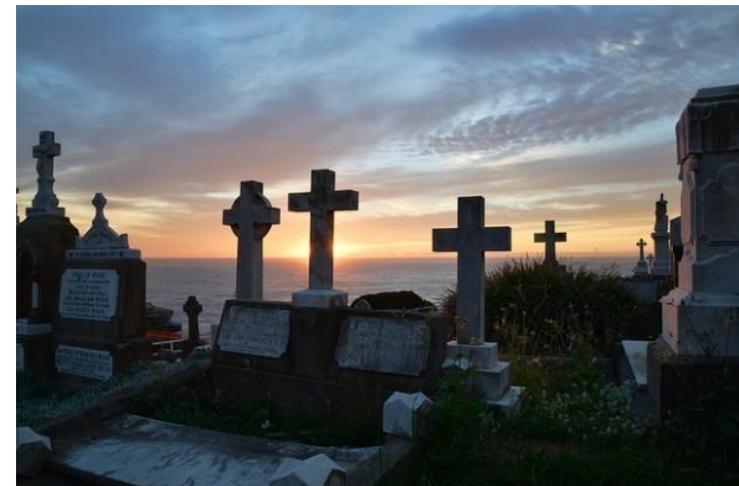
# Fundamentos de la verificación y validación del software

---

Sobredosis radiológica en el Instituto Nacional del Cáncer de Panamá (2000)

Errores en los procedimientos y un fallo de software causan que se apliquen dosis erróneas de radiación 8 personas murieron y 20 tuvieron problemas de salud graves.

Los médicos responsables del hecho fueron acusados de asesinato



# Fundamentos de la verificación y validación del software

---

## Aseguramiento de la calidad del software

Algunos desarrolladores de software todavía creen que la calidad del software es algo por lo que hay que empezar a preocuparse una vez generado el código.

# Fundamentos de la verificación y validación del software

---

**Proceso eficaz** de software que se aplica de manera que crea un **producto útil** que proporciona valor medible a quienes lo producen y a quienes lo utilizan.

# Fundamentos de la verificación y validación del software

---

- **Aseguramiento de la calidad del software**

- Es una actividad sombrilla. Incluye:

- 1) Un proceso de ACS,

- **2) Tareas específicas de aseguramiento y control de la calidad (incluidas revisiones técnicas y una estrategia de pruebas relacionadas entre sí)**

- 3) Prácticas eficaces de ingeniería de software (métodos y herramientas)

- 4) Control de todos los productos del trabajo de software y de los cambios que sufren

- 5) Un procedimiento para garantizar el cumplimiento de los estándares del desarrollo de software (cuando sea aplicable)

- 6) Mecanismos de medición y reporte.

# Fundamentos de la verificación y validación del software

---

## ■ Aseguramiento de la calidad del software

- El aseguramiento de la calidad del software es un “patrón planeado y sistemático de acciones” que se requieren para garantizar alta calidad en el software.
- La gente que realiza el ACS debe ver al software desde el punto de vista del cliente:
  - ¿El software cumple adecuadamente los factores de calidad mencionados anteriormente?
  - ¿El desarrollo del software se condujo de acuerdo con estándares preestablecidos?
  - ¿Las disciplinas técnicas han cumplido con sus roles como parte de la actividad de ACS?

# Fundamentos de la verificación y validación del software

---

- **Aseguramiento de la calidad del software**

- **Elementos:**

- **Estándares.**

- **Revisiones y auditorías**

- **Pruebas.**

- Colección y análisis de los errores.

- Administración del cambio.

- Educación.

- Administración de los proveedores.

- Administración de la seguridad.

- Seguridad.

- Administración de riesgos

# Fundamentos de la verificación y validación del software

---

- **Aseguramiento de la calidad del software**

- **Tareas:**

- Dos entidades diferentes ayudan a asegurar la calidad: los ingenieros de software que hacen el trabajo técnico y un grupo de ACS.
- El grupo ACS tiene la responsabilidad de planear, supervisar, registrar, analizar y hacer reportes acerca de la calidad.
- **Los ingenieros de software abordan la calidad (y ejecutan actividades para controlarla), aplicando métodos y medidas técnicas sólidas, realizando revisiones técnicas y haciendo pruebas de software bien planeadas.**

# Fundamentos de la verificación y validación del software

---

- **Aseguramiento de la calidad del software**
  - Un grupo independiente de ACS (auxiliar al equipo del software para lograr un producto final de alta calidad) realiza lo siguiente:

Planeación, supervisión, registro, análisis y elaboración de reportes para el aseguramiento de la calidad.

# Fundamentos de la verificación y validación del software

---

## ■ Aseguramiento de la calidad del software. El grupo ACS:

1. Prepara el plan de ACS para un proyecto.
2. Participa en el desarrollo de la descripción del software del proyecto.
3. Revisa las actividades de la ingeniería de software a fin de verificar el cumplimiento mediante el proceso definido para el software.
4. Audita los productos del trabajo de software designados para verificar que se cumpla con aquellos definidos como parte del proceso de software.
5. Registra toda falta de cumplimiento y la reporta a la alta dirección.

Coordina el control y administración del cambio y ayuda a recabar y analizar métricas para el software.

# Fundamentos de la verificación y validación del software

---

- **Aseguramiento de la calidad del software**
  - **El plan ACS incluye:**
    - Las evaluaciones que se van a realizar
    - Las auditorías y revisiones por efectuar
    - Los estándares aplicables al proyecto
    - Los procedimientos para reportar y dar seguimiento a los errores
    - Los productos del trabajo que genera el grupo de ACS y la retroalimentación que se dará al equipo del software.

# Fundamentos de la verificación y validación del software

---

## Aseguramiento de la calidad

- Administración del cambio.
- Educación/Capacitación.
- Administración de la seguridad.
- Administración de riesgos
- Etcétera

## Verificación y Validación

Revisiones técnicas

Auditorías

**Pruebas**

Estándares

# Fundamentos de la verificación y validación del software

---

## Técnicas dinámicas de verificación

# Fundamentos de la verificación y validación del software

---

En el proceso de Verificación y Validación se utilizan dos técnicas de comprobación y análisis:

## **Inspecciones del Software (Técnicas estáticas de verificación):**

- Se contrasta estáticamente las diferentes representaciones del sistema (diagramas de requerimientos, diagramas de diseño y código fuente) en búsqueda de defectos.
- No requiere que el código se ejecute
- Debe realizarse durante todo el ciclo de desarrollo.

## **Las pruebas del Software (Técnicas dinámicas de verificación):**

- Se contrasta dinámicamente la respuesta de prototipos ejecutables del sistema con el comportamiento operacional esperado.
- Requiere disponer de prototipo ejecutables y esto sólo es posible en las fases finales del proceso de desarrollo.

# Fundamentos de la verificación y validación del software

---

- **Técnicas estáticas (analíticas)**

Analizar el producto para deducir su correcta operación

- **Técnicas dinámicas (pruebas)**

Experimentar con el comportamiento de un producto para ver si el producto actúa como es esperado

# Fundamentos de la verificación y validación del software

---

- ❖ El objetivo de las pruebas es detectar los defectos latentes de un sistema software antes de entregar el producto.
- ❖ Una prueba de defectos exitosa es aquella que descubre un fallo, esto es, un comportamiento contrario a la especificación.
- ❖ Las pruebas de defectos demuestran la existencia de un fallo, y no la ausencia de cualquier fallo.

# Fundamentos de la verificación y validación del software

---

## Tipos de defectos

- En algoritmos: Bifurcar a destiempo, Preguntar por la condición equivocada, No inicializar variables
- De sintaxis. Ejemplo: Confundir un 0 por una O
- De precisión y cálculo, Formulas no implementadas correctamente
- De documentación, Ejemplo: El manual de usuario tiene un ejemplo que no funciona en el sistema
- De estrés o sobrecarga. El sistema funciona bien con 100 usuarios pero no con 110
- De capacidad o de borde. El sistema funciona bien con importes <1000000
- De sincronización o coordinación. Ej. Comunicación entre procesos con faltas
- De capacidad de procesamiento o desempeño. No terminar el trabajo en el tiempo requerido
- De recuperación. No poder volver a un estado normal luego de una falla
- De estándares y procedimientos. No cumplir con la definición de estándares y/o procedimientos
- Relativos al hardware o software del sistema. Incompatibilidad entre componentes

# Fundamentos de la verificación y validación del software

---

- **Prueba (test)**

Proceso de ejecutar un programa con el fin de encontrar fallas (G. Myers)

**Ejecutar un producto para:**

- o Verificar que satisface los requerimientos

- o Identificar diferencias entre el comportamiento real y el esperado (IEEE)

- **Caso de Prueba (test case)**

Datos de entrada, condiciones de ejecución y resultado esperado

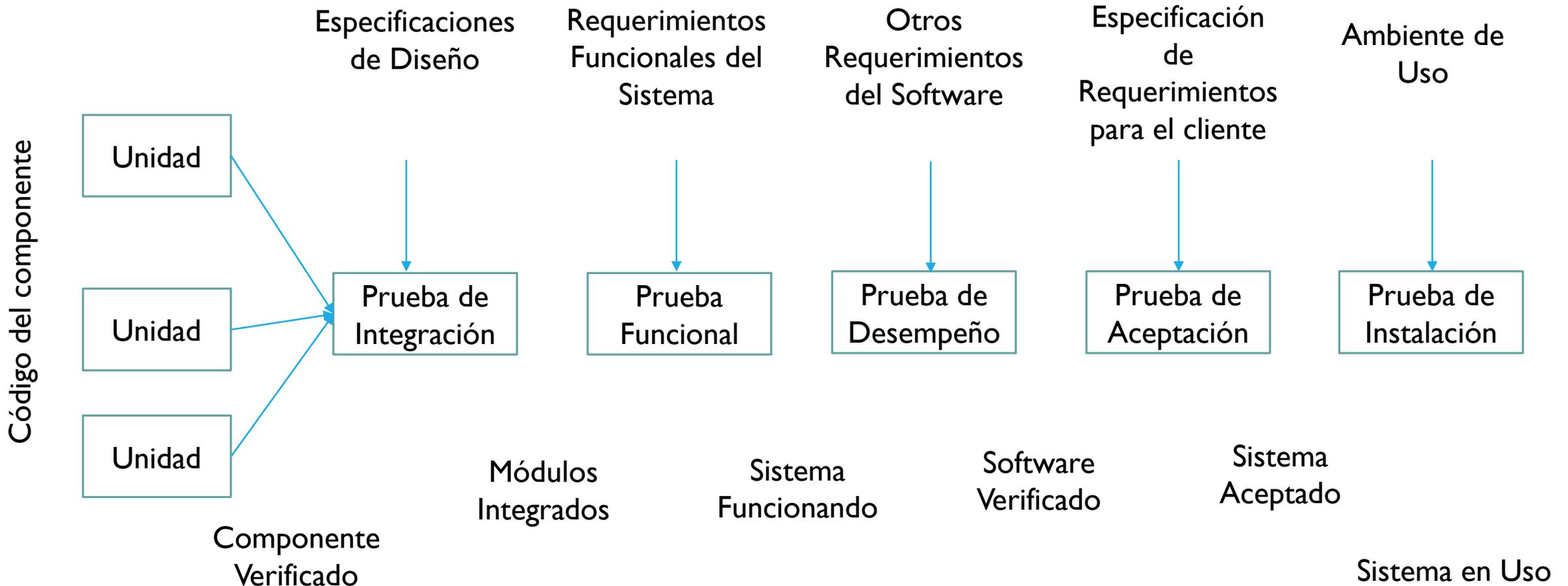
- **Conjunto de Prueba (test set)**

Conjunto de casos de prueba

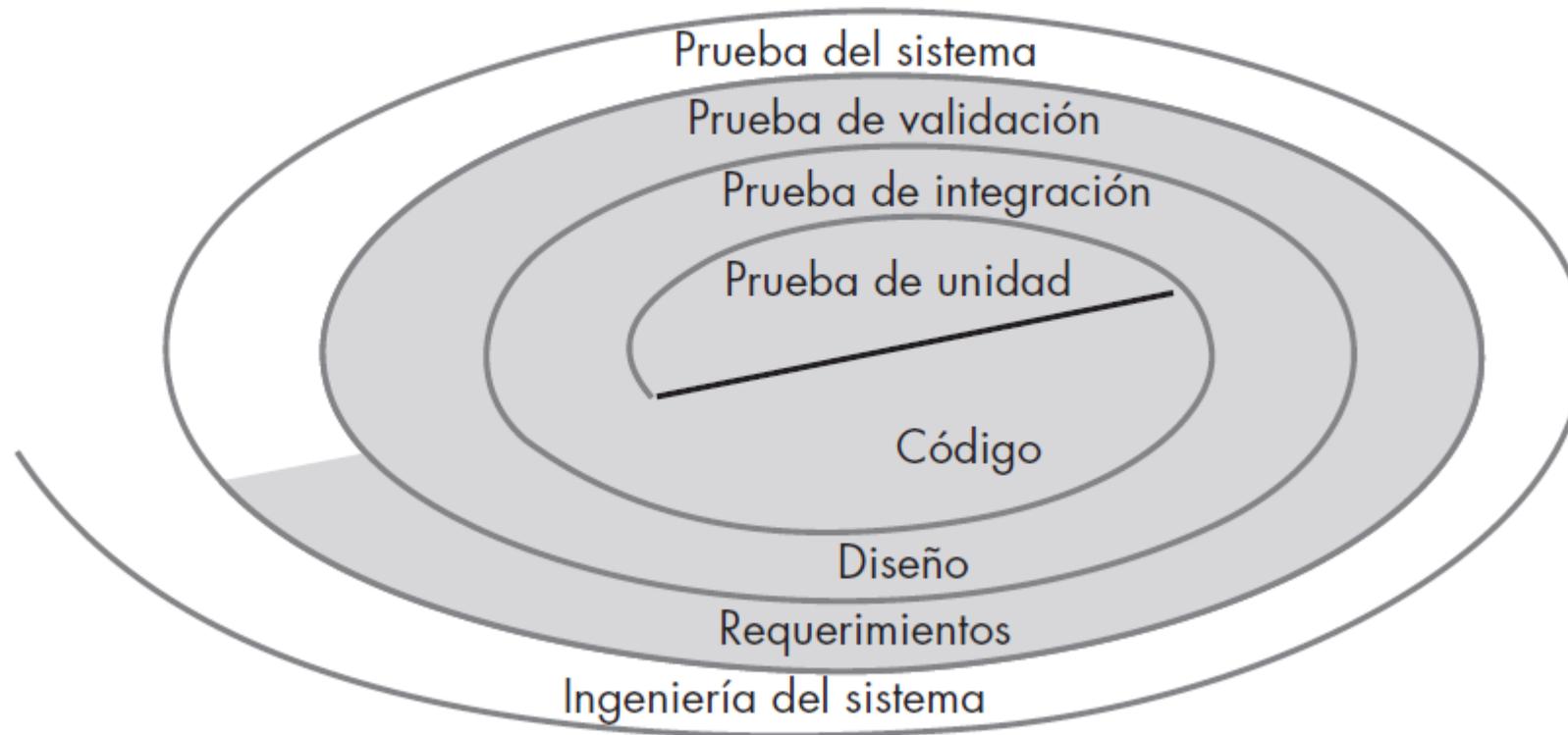
# Fundamentos de la verificación y validación del software.

## Pruebas de Software

### Proceso de las Pruebas del Software



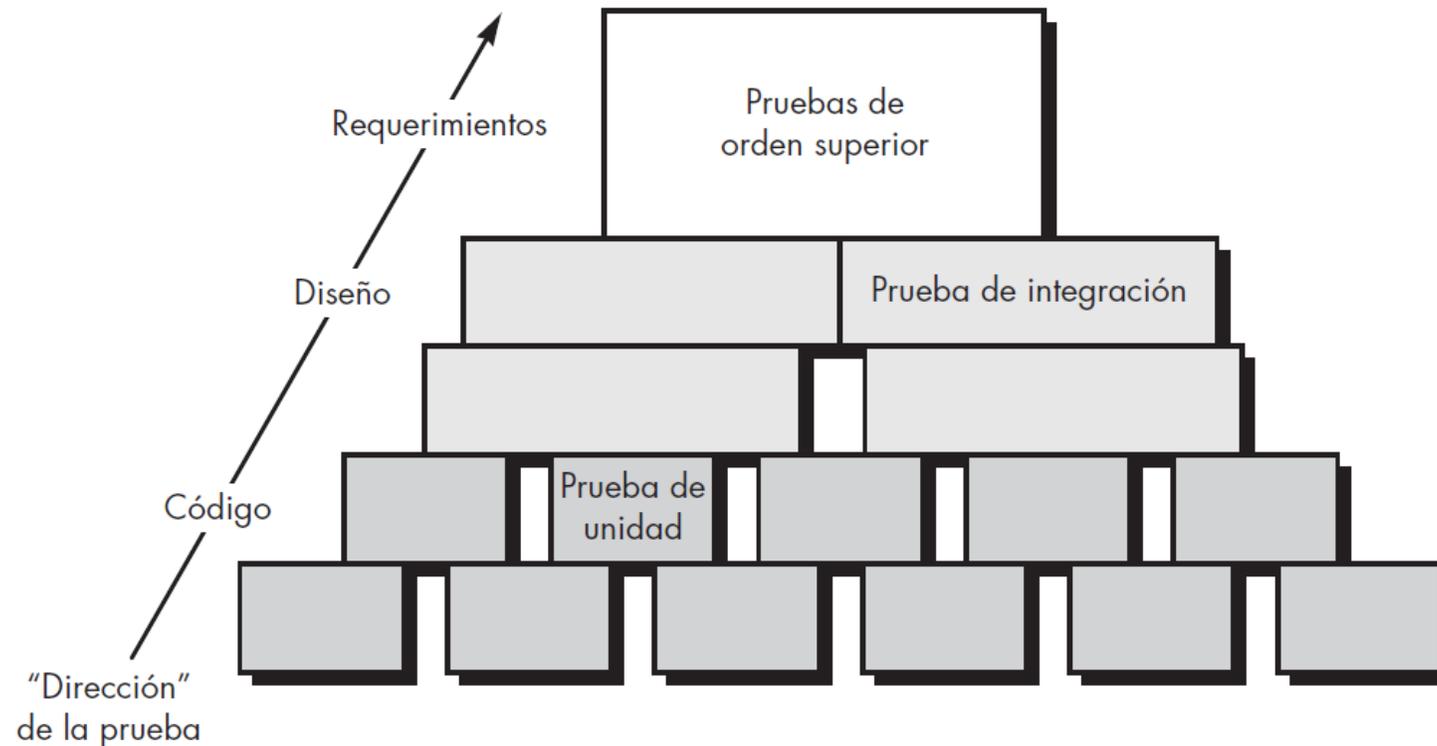
# Fundamentos de la verificación y validación del software



Estrategia de Pruebas

# Fundamentos de la verificación y validación del software

---



Pasos de las Pruebas del Software

# Fundamentos de la verificación y validación del software

---

- **Módulo, Componente o unitaria**

Verifica las funciones de los componentes

- **Integración**

Verifica que los componentes trabajan juntos como un sistema integrado

- **Funcional**

Determina si el sistema integrado cumple las funciones de acuerdo a los requerimientos

# Fundamentos de la verificación y validación del software

---

- **Desempeño**

Determina si el sistema integrado, en el ambiente objetivo cumple los requerimientos de tiempo de respuesta, capacidad de proceso y volúmenes

- **Aceptación/Validación**

Bajo la supervisión del cliente, verificar si el sistema cumple con los requerimientos del cliente (y lo satisface)

Validación del sistema, donde los requerimientos establecidos como parte de su modelado se validan confrontándose con el software que se construyó.

# Fundamentos de la verificación y validación del software

---

- **Instalación**

El sistema queda instalado en el ambiente de trabajo del cliente y funciona correctamente

# Fundamentos de la verificación y validación del software

---

## ¿Quién Verifica?

- **Pruebas Unitarias**

- Normalmente las realiza el equipo de desarrollo. En general la misma persona que lo implementó.
- Es positivo el conocimiento detallado del módulo a probar

- **Pruebas de Integración**

- Normalmente las realiza el equipo de desarrollo
- Es necesario el conocimiento de las interfaces y funciones en general

- **Resto de las pruebas**

- En general un equipo especializado (verificadores). Grupo de prueba independiente (GPI).
- Es necesario conocer los requerimientos y tener una visión global

# Fundamentos de la verificación y validación del software

---

Existen dos tipos diferentes de pruebas:

## **Las pruebas de defectos:**

- Buscan las inconsistencias entre un programa y su especificación.
- Las pruebas se diseñan para buscar los errores en el código.
- Demuestran la presencia, y no la ausencia, de defectos

## **Las pruebas estadísticas:**

- Buscan demostrar que satisface la especificación operacional y su fiabilidad.
- Se diseñan para reflejar la carga de trabajo habitual.
- Sus resultados se procesan estadísticamente para estimar su fiabilidad (contando el número de caídas del sistema) y sus tiempos de respuesta.

# Fundamentos de la verificación y validación del software

---

- En ocasiones la confiabilidad del software se basa en evidencia indirecta, como la calidad del proceso de desarrollo; sin embargo, aunque las buenas prácticas son necesarias para obtener software confiable, no lo garantiza.
- En muchas situaciones es importante asegurar que un programa en particular ha alcanzado la confiabilidad propuesta como objetivo.
- Y se requiere medir la confiabilidad alcanzada hasta el momento.

# Fundamentos de la verificación y validación del software

---

- Si observamos la ejecución de un programa libre de errores por suficiente tiempo, es razonable aceptar que es lo suficientemente confiable. Los datos permitirán calcular mediciones cuantitativas de confiabilidad.
- Sin embargo, algunos sistemas en especial los críticos, la operación real no puede ser usada para obtener datos de confiabilidad.
- Es necesario conocer que un sistema es lo suficientemente confiable antes de su puesta en operación.

# Fundamentos de la verificación y validación del software

---

- Por lo que será necesario simular su uso operacional en un ambiente de prueba, a esto se le llama pruebas operacionales, pruebas estadísticas o pruebas de ingeniería de confiabilidad del software (SRE software reliability engineering testing).
- La idea principal es seleccionar casos de prueba que serían seleccionados en operación; de esta forma la confiabilidad calculada en las pruebas será la misma en operación real.
- La dificultad de llevar esto a cabo varía de un sistema a otro

# Fundamentos de la verificación y validación del software

---

- Durante las pruebas estadísticas es importante registrar la exactitud de los datos.
- Los datos pueden ser de dos tipos: tiempo y número de eventos fallidos.
- La variable de tiempo que se utilizará dependerá de la aplicación. Por ejemplo, en aplicaciones de control de procesos, el tiempo probablemente será el tiempo real de operación y la fiabilidad expresada como una tasa de ocurrencia de fallas, o el tiempo promedio de falla.

# Fundamentos de la verificación y validación del software

---

- En un sistema de seguridad que solo opera cuando es requerido, el tiempo será el número de demandas o llamadas y la confiabilidad puede ser expresada como la probabilidad de fallo sobre demanda.
- **Las pruebas estadísticas pueden ser utilizadas principalmente de dos formas:**
  - Para estimar el crecimiento de confiabilidad.
  - Para estimar la confiabilidad a partir de evidencia de falla.
  - Ambos dependen crucialmente de la exactitud con la que la simulación representa el comportamiento real.

# Fundamentos de la verificación y validación del software

---

Pressman, R.S., *Ingeniería del Software. Un enfoque práctico*, quinta edición, 2002, España.

Sommerville I., *Ingeniería de Software*, Addison-Wesley, 6ª. Edición, 2002, México.

FISHER, M. S., *Software Verification and Validation: An Engineering and Scientific Approach* Springer 2007

B. Littlewood, L. Strigini. (1997). Centre for Software Reliability, City University, London, United Kingdom  
Guidelines for Statistical Testing. Consultado el 09 de febrero de 2018, desde:  
[http://openaccess.city.ac.uk/254/2/StatsTesting\\_TN12-3.1distrib2.pdf](http://openaccess.city.ac.uk/254/2/StatsTesting_TN12-3.1distrib2.pdf)

Burnstein, I. (2003). *Practical Software Testing*. Springer, New York, USA.

Grindal, M., Offutt, J., and Andler, S. (2005). Combination testing strategies: a survey. *Software Testing, Verification, and Reliability* , 15(3):167–199.

Hartman, A. (2005). Software and hardware testing using combinatorial covering suites. In *Graph Theory, Combinatorics and Algorithms* , chapter 10, pages 237–266. Springer-Verlag. Kuhn, D. R., Wallace, D. R., and Gallo, A. M. (2004).

Software fault interactions and implications for software testing. *IEEE Transactions on Software Engineering* , 30(6):418–421.

Myers, G. J. (2004). *The art of software testing*. Wiley, second edition

---

# Gracias por su atención