

Pruebas y Confiabilidad del Software

Objetivo

El curso de pruebas de confiabilidad de software permitirá al alumno comprender las técnicas necesarias para mejorar la calidad del software. Cualquier sistema de software con fallas en su funcionamiento o que se ejecute fuera de su especificación tendrá poca duración o será rechazado por el cliente.

Descripción

De acuerdo a estudios recientes en cientos de sistemas de software usados en los Estados Unidos, la gran mayoría de estos presentan numerosos defectos de software que perjudican la integridad de los sistemas que controlan o que perjudican a la economía o seguridad del personal que utiliza este software.

Es claro, que en la actualidad el software es utilizado para controlar una gran parte de la economía de los países altamente industrializados y se encuentran presentes en la mayoría de los aparatos electrónicos que utilizamos diariamente. Es por esto que los sistemas de software cuenten con una alta confiabilidad, que permita que el software este siempre disponible, libre de fallos catastróficos, seguro, íntegro y que se ejecute siempre de acuerdo a como fue especificado.

En este curso se presenta primero una introducción a la confiabilidad de software, la cual permitirá introducir técnicas para una programación confiable que ayude a la eliminación de defectos. Así mismo, en este curso se presentarán distintas técnicas para la implementación de pruebas de software, inspecciones y depuración. Finalmente se presentará el mecanismo de diseño por contrato como un mecanismo para reducir los fallos en el diseño del software.

Cada capítulo de este curso tendrá una introducción teórica, un ejemplo de implementación y distintos ejercicios prácticos que permitan al alumno implementar las técnicas aprendidas.

A pesar de ser un curso impartido dentro de la especialidad de Ingeniería de Software, no se requiere de haber llevado el curso de Introducción a la Ingeniería de Software

Contenido

- a. INTRODUCCION A LAS PUEBRAS Y CONFIABILIDAD DEL SOFTWARE
 1. Introducción a las pruebas y confiabilidad del software.
 - a. Pruebas de software como proceso de Ingeniería del Software.
 - b. Confiabilidad
 - c. Definiciones de error, fallo y fallo de sistema.
 - d. Importancia de la detección de fallos en el software.
 2. Programación Modular: Primer paso hacia la programación confiable.
 - a. introducción a la programación modular.
 - b. Programación modular en la práctica
 - c. Programación el lenguaje C estilo tradicional
 - d. Mecanismos del lenguaje C para soporte del encapsulado.
 3. Metodología para la programación modular modelo de módulos.
 - a. estructura de archivos asociada al modelo de módulos
 - b. modelo de la interfaz y las dependencias de los módulos y sus archivos

- c. estructura de los archivos del encabezado
- d. directivas para el diseño de módulos

4. Ejemplos y actividad práctica.

b. FUNDAMENTOS DE CORRECCION Y CONFIABILIDAD

1. Conceptos básicos sobre defectos del software
2. Programación disciplinada
3. Tipos de datos abstractos
4. Diseño por contrato
5. Invariantes
6. Ventajas del uso consciente de aseveraciones
7. Estilo de codificación del software
8. Inspecciones, pruebas y depuración del software
9. Disciplinas de desarrollo
10. Ejemplos y actividad práctica

c. FUNDAMENTOS DE PRUEBAS DE SOFTWARE

1. Conceptos básicos sobre pruebas del software
2. Pruebas de Caja Negra
3. Pruebas estructurales o de Caja Blanca
4. Pruebas como mecanismo de evitación de defectos
5. Papel de las aseveraciones ejecutables durante las pruebas

d. AUTOMATIZACION DE LAS PRUEBAS

- a. Proceso de Automatización de las pruebas de unidad
- b. Requerimientos, Diseño e Implementación del Módulo de Pruebas
- c. Ejemplos de Prueba del Módulo
- d. Actividad Práctica

e. INSPECCIONES DE SOFTWARE

- j) Inspecciones del software para la detección de defectos
- k) Inspecciones formales
- l) Técnicas de detección de defectos durante la inspección
- m) Aspectos psicológicos de la inspección
- n) Inspecciones formales como mecanismo de prevención de defectos
- o) Otras prácticas de inspección menos formales
- p) Conclusiones

f. FUNDAMENTOS DE DEPURACION DEL SOFTWARE

- a) Conceptos básicos sobre depuración del software
- b) observación de los hechos
- c) Bitácora de la ejecución para la depuración
- d) Trazas o historia de la ejecución

- e) Verificación automática del estado del programa
- f) Ejemplos y Actividad Práctica

g. FUNDAMENTOS DE DISEÑO POR CONTRATO

- a. Diseño por contrato
- b. Invariables.
- c. Ventajas del uso consistente de aseveraciones
- d. Aseveraciones vs. Pruebas de Unidad
- e. Diseño de módulo para el soporte del diseño por contrato en lenguaje C
- f. Ejemplos y Actividad Práctica