

PRUEBAS DE SOFTWARE

Presentación de la Asignatura

Antonio Garmendia

antonio.garmendia@uam.es

34

PRUEBAS DE SOFTWARE EN EL SIGLO XXI

- Aplicaciones definen un *comportamiento*
 - Enrutadores de red, finanzas, redes de conmutación, otras infraestructuras
- Mercado actual de aplicaciones
 - Mucho más grande
 - Más competitivo
 - Tiene más Usuarios
- Pruebas de Software para la Web (e.g., Google, Facebook, Youtube)
- Tenemos software empotrado (*Embedded Applications*) para controlar:
 - Aviones
 - Sistemas de control de tráfico aéreo
 - Naves espaciales
 - Relojes digitales
 - Control Remoto
 - Tablets
 - Teléfonos móviles
 - Coches Autónomos
 - Y muchos más!

PRUEBAS DE SOFTWARE EN EL SIGLO XXI



Las **Metodologías Ágiles** ponen atención especial en las pruebas de software



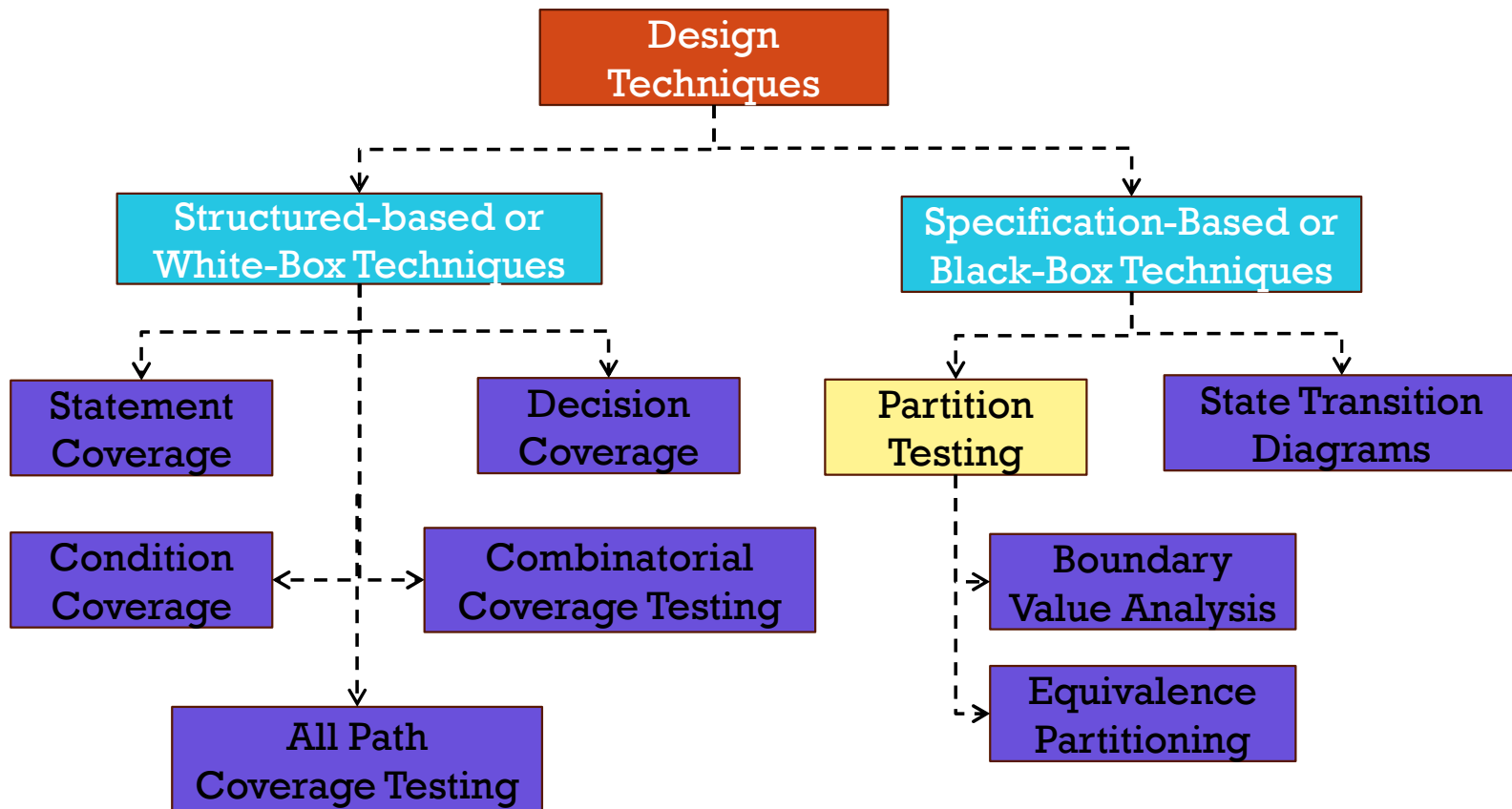
Los tests son fundamentales para comprobar los requisitos funcionales (¿Qué hace el sistema?)



Test-driven development (TDD) es una metodología de desarrollo ágil donde las pruebas se escriben antes de desarrollar el código

Programadores deben realizar test unitario (unit testing)

DISEÑO DE PRUEBAS DE SOFTWARE



TEMARIO

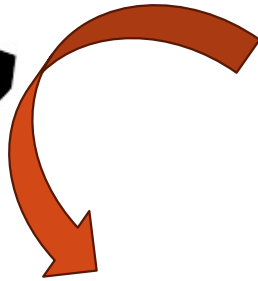
- Unidad 1.- Introducción a las pruebas de software
 - Concepto, Importancia, Problemas
- Unidad 2.- Testeo unitario y automatización del proceso de testeo
 - Diseño, Automatización, Detección de Fallos
- Unidad 3.- Criterios de Cobertura
 - Análisis de los diferentes tipos de cobertura, Diseño de pruebas basado en cobertura
- Unidad 4.- Generación automática de casos de prueba
 - Pruebas de mutación, pruebas metamórficas, y ¡**Mucho Más!**

PRÁCTICAS

- Práctica 1. Diseño de pruebas unitarias
 - Creación de pruebas unitarias JUnit
 - Cobertura de código
- Práctica 2. Diseño combinatorio de pruebas
 - Automatización de pruebas de software
 - Creación automática de Tests
 - <https://randoop.github.io/randoop/>
- Práctica 3. Pruebas de Mutación
 - Code-defenders
 - <https://code-defenders.org/>

EJEMPLO - PRUEBAS DE MUTACIÓN

Attacker



```
public class CountUtils {  
    public static int numZero(int[] x) {  
        int count=0;  
        for (int i=0; i<arr.length(); i++) {  
            if (arr[i] == 0)  
                count++;  
        }  
        return count;  
    }  
}
```

```
public class CountUtils {  
    public static int numZero(int[] x) {  
        int count=0;  
        for (int i=1; i<arr.length(); i++) {  
            if (arr[i] == 0)  
                count++;  
        }  
        return count;  
    }  
}
```



```
public class TestCount {  
    @Test  
    public void testNumZero() {  
        assertEquals(1, CountUtils.numZero([2, 7, 0]));  
    }  
}
```

Defender



EVALUACIÓN

Teoría 60%

Práctica 40%

- Dos exámenes de Teoría
- Cada examen cubrirá aproximadamente la mitad de los contenidos de la asignatura
- Tres prácticas en total
- Lenguaje de Programación: Java



PREGUNTAS

42

Antonio Garmendia

antonio.garmendia@uam.es

5/10/2024