

INGENIERÍA DEL SOFTWARE

PRÁCTICA 8: PROCESO DE PRUEBA

DESCRIPCIÓN

Planteamos el desarrollo completo de un plan de pruebas para el software. Para ello, comenzaremos desarrollando el primer incremento de la herramienta que hemos estado proyectando y a la que se le aplicará dicho plan de pruebas. El software se desarrollará en equipos y una vez construido aplicaremos los métodos vistos en el tema 6 para la generación de los casos de prueba que se organizarán y ejecutarán “contra” el software desarrollado por otro grupo. Para ello seguiremos las recomendaciones del estándar IEEE 829, para el diseño de pruebas. También se revisará el IEEE 1008, para la ejecución de pruebas.

OBJETIVOS

Esta práctica tiene por objetivos:

- Construir un software en equipo siguiendo un proceso que pase por: Definir su funcionalidad, diseñar su arquitectura y planificar la construcción repartiendo sus partes entre los miembros del equipo.
- Realizar, siguiendo un estándar (IEEE 829), un plan de pruebas para un software desarrollado por otro y que se nos entregará en partes. Primero su documento de análisis y diseño y posteriormente su código.
- Construir, de acuerdo al plan diseñado, un banco de pruebas automáticas codificadas usando Junit y Mockito con el grado de cobertura deseado.
- Ejecutar e informar del resultado de las pruebas construidas.

METODOLOGÍA

1. Desarrollo del Software (SUT).

- 1.1. **Documentación del proceso:** Se trata de cubrir la plantilla 02_PlantAnaDisPlan entregada en el curso virtual para construir un entregable llamado PrjGrupoXY.doc, con X=[1|2|3] según grupo ENSO e Y=[A|B|C|D|E] según el grupo acordado para la práctica con el profesor. En este entregable se describirá el alcance y diseño del software que se abordará en un primer incremento del proyecto. El documento también incluirá un plan de como la construcción del software se repartirá entre los integrantes del equipo

- 1.2.**Software del incremento.** El software será entregado como un zip que incluirá el workspace completo del eclipse con un único proyecto que se llamará igual que el documento PrjGrupoXY, tomando X e Y los mismos valores.
 - **RESTRICCIONES DEL SOFTWARE**
 - Las clases sólo pueden tener las interfaces descrita en el documento PrjGrupoXY.doc tal y como están descritas en él.
 - El software de ninguna forma puede incluir Interfaces Gráficas de USUARIO
 - El software debe compilar sin errores y no tener dependencias externas, bases de datos disponibles o similar
 - El software debe estar disponible en la fecha indicada
2. Revisión del entorno documental
 - 2.1.Revisar los apuntes al respecto de los estándares IEEE 829, IEEE 1008.
 - 2.2.Revisar los estándares definidos para los documentos a usar en el proceso, IEEE 829, IEEE 1008.
 - 2.3.Revisión de los ejemplos planes de pruebas.
 - 2.4.El objetivo es entender que campos de los propuestos en el estándar tienen sentido en nuestro problema y con que nivel de detalle deben cubrirse alcanzando una visión de lo que debe incluir nuestro documento en sus apartados de:
 - 2.4.1.1. *Plan de pruebas.*
 - 2.4.1.2. *Diseño de pruebas.*
 - 2.4.1.3. *Especificación de casos de prueba*
 - 2.4.1.4. *Especificación de procedimientos*
3. Construcción del documento de pruebas.
 - 3.1.Documento de pruebas (CN). Versión inicial.
 - 3.1.1.Se trata de crear la primera versión de nuestro documento de pruebas, de acuerdo a lo decidido en la fase anterior, pero con la restricción de disponer, únicamente, del documento PrjGrupoXY.doc. De esta forma sólo las técnicas de caja negra (CN) podrán ser propuestas y utilizadas en esta primera versión.
 - 3.1.2.Los detalles del documento cambiarán de acuerdo a la evolución de las prácticas.
 - 3.1.3.El documento sólo se entregará tras la revisión del equipo que construirá el software al que se le aplicarán estas pruebas. El equipo debe revisar si las interfaces, tal cual están pensadas, pueden probarse como sugiere el documento y si las pruebas diseñadas están acorde a la funcionalidad que se ha pensado.
 - 3.2.Documento de pruebas (Completo). Versión Final
 - 3.2.1.Se trata de crear la versión definitiva de nuestro documento de pruebas. Debe incluir tanto pruebas de caja negra como pruebas de caja blanca.
 - 3.2.2.Los detalles del documento cambiarán de acuerdo a la evolución de las prácticas.
 - 3.2.3.El documento sólo se entregará tras la revisión del equipo que construirá el software al que se le aplicarán estas pruebas. El equipo debe revisar si las interfaces, tal cual están pensadas, pueden probarse como sugiere el documento y si las pruebas diseñadas están acorde a la funcionalidad que se ha pensado.

4. Construcción de las pruebas. **EL CÓDIGO NO SE CORRIGE NI SE MODIFICA**
 - 4.1. Construcción de las pruebas utilizando Junit.
 - 4.1.1.Trataremos de transformar todos los Casos de Prueba en Test de Junit
 - 4.1.2.Programaremos los procedimientos como Test complejos o a través de Suites de Test.
 - 4.2.Análisis de Cobertura
 - 4.2.1.Instalaremos el ECLEMMMA y analizaremos la cobertura de nuestros Test
 - 4.2.2.Añadiremos los Test que sean precisos para la cobertura que deseábamos tener.
 - 4.3.Mocks. Utilizaremos Mocks(simuladores) para evitar el uso de clases que colaboran en una funcionalidad cuando hacemos pruebas de unidad y los sustituiremos por la clase colaboradora cuando nos pasamos a las pruebas de integración.
5. Informe de ejecución. Debe indicar el grado de cumplimiento del plan de pruebas.
 - 5.1.Histórico de pruebas: Lo generan directamente las junit
 - 5.2.Informe de incidencia: Explica los problemas que hayan surgido en cada ejecución
 - 5.3.Informe resumen
 - 5.3.1.Indicar que aspectos del plan se abordaron de forma automática y como
 - 5.3.2.Resume de los test que no pasa el software ya que son ÉXITOS de las pruebas y deben destacarse en el informe.
 - 5.3.3.Se debe indicar que porcentaje de Éxitos hemos alcanzado con cada una de las técnicas de generación de casos de prueba.
 - 5.3.4.Qué grado real de cobertura alcanzamos con las pruebas de caja blanca generadas.
 - 5.3.5.Cualquier otra información que consideréis oportuna para destacar la calidad de vuestras pruebas.

MATERIAL

- Tema 6 de la asignatura: Pruebas del software
- Estándares
 - IEEE 829
 - IEEE 1008
- Otros documentos
 - Plantilla para apartado de análisis
 - Guías y ejemplos de Planes de pruebas.

ENTREGABLES

- Documento PrjGrp[1|2|3][A|B|C|D|E].doc
- Software PrjGrp[1|3]Y (zip workspace)
- Documento de Pruebas. Versión Inicial, pruebas de Caja Negra
- Documento de Pruebas. Versión Final
- Software PrjGrupo[1|2|3][A|B|C|D|E] con pruebas añadidas
- Informe de ejecución.

SE PUNTUARÁ

- Código: Es imprescindible su entrega.
- Casos de prueba: Se valorará la claridad del documento en la facilidad de distinguir los distintos casos de prueba y las técnicas seguidas para su obtención.
- Procedimientos: Se valorará la claridad con la que se identifiquen las secuencias de los bloques de casos de prueba y la precisión con la que queden definidos los procedimientos
- Informe de ejecución del plan de pruebas:
 - La claridad del documento
- Se penalizará la entrega tardía de documentos y será particularmente grave en el caso de que el retraso se produzca en la entrega del software y la especificación y diseño.