



Objetivos

- Poner en práctica los conocimientos adquiridos a lo largo de la asignatura durante el proceso de desarrollo software
- Trabajar más a fondo de forma conjunta con las herramientas utilizadas en las EPDs.

Introducción

A lo largo de la asignatura ha ido adquiriendo conceptos y procedimientos a aplicar a lo largo del proceso de desarrollo software para garantizar la calidad de éste. Además, en las EPDs, ha ido asentando esos conocimientos a través de la práctica con herramientas muy utilizadas en la actualidad en las empresas del sector.

Durante el desarrollo de este trabajo, los alumnos tendrán que aplicar los conceptos adquiridos así como aplicarlos a un proyecto real utilizando las herramientas descritas en las EPDs. Dado que la asignatura cuenta con dos sesiones específicas de GitHub y, el resto de las EPDs giran alrededor de él, para la realización del proyecto, GitHub será pieza clave e indispensable.

Descripción del proyecto

El objetivo de proyecto no es más que aplicar todas las técnicas, conocimientos y destrezas adquiridas a lo largo de la asignatura para así afianzar los mismos.

La universidad Pablo de Olavide nos contacta para evaluar la calidad de su web, puesto que no tienen automatizado ningún test y no son capaces de garantizar de una forma rápida el paso de producción de nuevas releases. Se nos pide que, como vamos a ir a un concurso público, hacer el encargo a todos los equipos para, aquel equipo que mejor lo haga, ser contratado.

Instrucciones

En las EPDs ya ha adquirido los conocimientos necesarios para trabajar con GitHub como un gestor de proyectos basándonos en el código que existe en el repositorio. Utilizaremos la wiki, el gestor de issues, el panel ágil así como el uso de ramas y pull requests. Se solicita:

1. **En la wiki, por secciones:**
 1. **Descripción del problema a resolver:** Se deberá introducir de forma breve, de qué proyecto se trata. Infórmese sobre la empresa que tiene asignada a su grupo (de forma breve).
 2. **Desarrollo de una estrategia de pruebas e implementación de la misma:** A lo largo de la asignatura ha ido conociendo estrategias para testear su software. Defina una estrategia, documéntela y abórdela. Ej: Testing funcionales, de despliegue, de performance...
 3. **Definición de una guía de codificación:** Deberá establecer una guía de codificación, la cual será aplicada para los tests escritos en Java.
 4. **Implementación de los tests:** Implemente los tests con Selenium que crea oportuno.
 5. **Test cases:** Elabore una sección en la wiki con los test cases. Piense cómo organizar dicha wiki (Ej: por secciones de la página, etc).
 6. **Suites de testing definidas:** Como sabe, hay dos tipos de suites de tests:
 1. **Acceptance:** Tests que garantizan el core mínimo de la aplicación.
 2. **Regression:** Tests que garantizan todo el funcionamiento de la aplicación.
2. **Milestones (Releases):** Obligatoriamente habrá 3 (aunque, si lo necesita, podrá crear más).
 1. **Iteración 1:** Finalización fecha 6 de Noviembre. **No habrá presentación. Sólo tendrá que tener (como mínimo) los apartados 1 y 3 de la wiki realizado.** Además, será necesario tener en la wiki una sección con un calendario de roles (se recomienda que, cada componente del grupo esté, al menos, 2 semanas/rol). Si algún componente se excede de sus funciones durante el periodo asignado, será penalizado (ej: si la primera semana Alejandro es tester, no podrá diseñar test cases, sólo podrá codificar tests).
 2. **Iteración 2:** Finalización fecha 2 de diciembre a las 15:00h (presentación por parte del equipo en clase de EB). Aquí, los equipos deberán tener, como mínimo:
 1. Punto 2 de la wiki terminado (no cerrado (pero sí maduro), puede completarse a lo largo del proyecto).
 2. Punto 4 y 5 comenzado.
 3. Del punto 6, al menos, la suite de aceptación deberá estar definida.



3. **Iteración 3:** Finalización el día 16 de Diciembre a las 10:00h (presentación por parte del equipo durante la sesión de EB). Es decir: dispondremos de 1,5 horas para la presentación de proyectos.

El repositorio estará organizado por tecnologías:

- Ejemplo: selenium/ (donde tendrá el código fuente de los tests de Selenium)

Si utiliza alguna otra tecnología de testeo, no dude en generar otro directorio con el contenido necesario.

Entregas

Para cada entrega, deberán seguirse los siguientes procedimientos:

1. Código entregado: tendrá que tener un TAG con la signature: CAL_ITERACION_N.
2. Sólo se evaluará el contenido creado/editado/eliminado hasta la fecha de entrega de la iteración:
Iteración 1: 15 de noviembre a las 23:59:59
Iteración 2: 2 de diciembre a las 16:29:59
Iteración 3: 16 de diciembre a las 14:29:59

Requisitos

Los proyectos se realizarán en grupos[1] de, al menos, 6 personas. Excepcionalmente se considerarán grupos de otros tamaños previa consulta con el profesor.

[1] <https://docs.google.com/spreadsheets/d/1CFbmTWEvwoTGJxGWrtwFvN7XkUFSve0ox-xC4AIWKvc/edit?gid=0#gid=0>

A pesar de lo mínimo exigible, los grupos podrán incluir librerías de terceros **no vistas durante las EPDs** que aporten un plus extra a su trabajo de testeo. El uso de librerías, métodos y aspectos referentes a estas tecnologías no tratados en clase, resultarán en un incremento de la nota final del proyecto (incluso por encima de los diez puntos si fuese necesario, hasta 3 punto extras).

Los test mínimos exigibles de proyecto serán test funcionales programados con Selenium (se verá en EPD).

Aunque parece que en este tipo de proyecto sólo hay testers, hay 3 roles claros definidos:

1. **QA Lead/Scrum master:** Se encargará de ser el facilitador de las tareas del equipo y de organizar las releases (si su equipo desea usar sprints internos, uselos). Recuerde que, tras un Sprint o una release, en la wiki debe haber una sección con las retrospectivas que el equipo ha tenido. Puede realizar cualquier tarea.
2. **Tester:** Será el encargado de codificar los tests.
3. **Tester QA:** Será el encargado de validar que, cuando un tester entrega un set de tests a través de un Pull Request, estos son correctos y **será el encargado de aceptar/rechazar el pull request**. Además, será el encargado de diseñar el test plan con los test cases.

Recuerde que los roles rotarán en función del calendario establecido en la Iteración 1.

Como estamos utilizando una metodología ágil, deberá usar el tablero tal y como se vio en la EPD (ToDo, WIP, QA y Done).

1. **ToDo:** Tareas pendientes de ser realizadas.
2. **WIP:** Tareas en las que alguien esté trabajando.
3. **QA:** Tareas terminadas pendientes de ser validadas por el Tester QA.
4. **Done:** Tareas terminadas y validadas por QA (entregables).

Ejemplo de tareas: Crear sección de la wiki para los test cases, crear test cases para la sección XX, Implementar tests relativos a los test cases XX, etc.

Recuerde el equipo es el que marca la calidad del proyecto y esta abarca desde la profundidad del detalle de la documentación, la calidad del código (guía de codificación), calidad del software (pruebas) y el la evolución del software durante su desarrollo.

Valoración

El trabajo se valorará en dos exposiciones que se realizarán en las entregas citadas anteriormente. Dichas exposiciones deberán



abarcando todos los aspectos del trabajo, incidiendo en cómo ha trabajado el equipo para gestionar la calidad del software (incluso entrando en detalle en el uso de las herramientas utilizadas si lo cree conveniente). Será necesario que todos los miembros del equipo participen de una forma activa y equitativa en la presentación.

Los aspectos a evaluar son:

1. Gestión de la calidad durante el proceso software (ej: estrategias de testeo, guía de codificación, gestión de incidencias, etc).
2. Calidad del código fuente de los tests (ej: organización, aplicación de la guía de codificación, etc).
3. Calidad de la documentación (ej: organización, etc).
4. Uso adecuado de las herramientas proporcionadas (GitHub).
5. Calidad de la presentación (P. ej. calidad de la expresión oral, del contenido de las transparencias, etc.).
6. Extras incluidos: Uso de material no tratado en clase de las distintas tecnologías abarcadas durante el curso, librerías/herramientas de terceros (además de la obligatoria). Las valoraciones de este apartado significarán un aumento de la nota por encima de la valoración ordinaria.