

Curso de Especialización en Sistemas Embebidos

TESTING DE SOFTWARE

12va Cohorte 2020

Trabajo práctico 1. Master test plan

“Implementación de aprendizaje por

refuerzo en un Robot”

Alumno:

Pablo Daniel Folino

Docentes:

Esp. Ing. Alejandro Permingeat

Esp. Ing. Esteban Volentini

Índice

**1.** [**Historial de cambios**](#Historial_de_cambios) 2

**2.** [**Introducción**](#Introduccións)

2.1. [Contenidos](#Contenidos) 3

**3.** [**Asignaciones**](#Asignaciones)3

3.1. [Responsable](#Responsable) 4

3.2. [Contratista](#Contratista) 4

3.3. [Alcances](#Alcances) 4 3.4. [Objetivos](#Objetivos) 4

3.5. [Precondiciones](#Precondiciones) 4

**4.** [**Bases del test**](#Bases_del_test) 4

**5.** [**Estrategia general del Test**](#Estrategia_general_del_Test) 5

5.1. [Características de calidad](#Características_de_calidad) 5

5.2. [Asignación de niveles de prueba a las características de calidad](#Asignacion_de_niveles_de_prueba) 6

**6.** [**Estrategia por nivel de prueba**](#Estrategia_por_nivel_de_prueba) 6

6.1. [Selección de características de calidad y determinación de la](#Selección_de_características_de_calidad)

[importancia relativa por nivel de prueba](#Selección_de_características_de_calidad) 6

6.2. [Dividir el sistema en subsistemas](#Dividir_el_sistema_en_subsistemas) 7

6.3. [Determinar la importancia relativa de los subsistemas](#Determinar_la_importancia_del_test) 8

6.4. [Determinar la importancia de test por combinaciones](#Determinar_la_importancia_del_test)

[de subsistema/característica de calidad](#Determinar_la_importancia_del_test) 8

6.5. [Establecer las técnicas de test a ser utilizadas](#Establecer_las_técnicas_de_test)| 10

**7.** [**Referencias**](#Referencias) **11**

1. Historial de cambios

|  |  |  |
| --- | --- | --- |
| **Fecha** | **Versión** | **Cambio** |
| 08/11/2020 | 1.0 | Versión 1 de Master Test Plan |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. Introducción

En el presente documento se detallarán todos los aspectos relacionados con la especificación del Master Test Plan (plan maestro de pruebas) referentes al desarrollo del software del Robot en donde se implementarán técnicas de Aprendizaje por Refuerzo (A.R.). Como fuentes de información se utilizan las historias de usuario y las especificaciones de requerimientos del Proyecto “Implementación de Aprendizaje por Refuerzo en un Robot” [1] del trabajo final de la Especialización de Sistemas Embebidos-FIUBA, como así también del documento de “Arquitectura y Diseño” [2] y “Especificaciones de Requerimientos de Software” [3]. Para tal fin se estructura el presente “Master Test Plan” según el ejemplo provisto en el *Apéndice E* del libro “Testing Embedded Software” de Bart Broekman y Edwin Notenboom.

El Robot consiste en una plataforma móvil tipo triciclo, y en él se instalan distintos sensores y actuadores, módulos que ayudan a la navegación y al posicionamiento del mismo. A grandes rasgos el “ROBOT\_RL” consta de los siguientes módulos de software:

* Módulo acelerómetro.
* Módulo control de motores.
* Módulos de comunicaciones WIFI-Bluetooth.
* Módulo de posicionamiento (GPS).
* Módulo sensor de ultrasonido.
* Módulo de administrador de energía.
* Módulo principal.
* Módulo encoder.

**2.1.** **Contenidos**

Los contenidos del presente *“Master Test Plan”* son:

• Asignaciones.

• Bases del test.

• Estrategia general del test.

• Estrategia por nivel de prueba.

**3.Asignaciones**

**3.1.** **Responsable**

El responsable de la elaboración de este documento es Pablo Daniel Folino, ingeniero a cargo del desarrollo del proyecto.

**3.2.** **Contratista**

La asignación es ejecutada bajo responsabilidad de Pablo Daniel Folino, jefe de testing del desarrollo del proyecto.

**3.3.** **Alcances**

El alcance del test de aceptación del dispositivo “Robot\_RL”.

**3.4.** **Objetivos**

Los objetivos son:

* Determinar si el sistema cumple con los requerimientos.
* Reportar las diferencias entre el comportamiento deseado y observado.
* Detectar si el software implementa funcionalidades que no forman parte de los requerimientos.
* Proveer una batería de pruebas automáticas para el firmware, que pueda ser reutilizado en futuras revisiones del dispositivo.

**3.5.** **Precondiciones**

* Contar con todas las herramientas de software y la infraestructura de hardware para realizar los test necesarios.
* Contar con el espacio suficiente para que el robot se pueda desplazar correctamente.

**4.** **Bases del test**

Las bases para desarrollar el test consisten en los siguientes documentos:

* [ROBOT\_RL-ARQ-002] Robot de Aprendizaje por Refuerzo- Documento de arquitectura y diseño del software. Revisión 2.
* [ROBOT\_RL-ER-001] Robot de Aprendizaje por Refuerzo- Documento de especificación y requerimientos de software. Revisión 1.
* [ROBOT\_RL-PROY-014] Robot de Aprendizaje por Refuerzo- Documento del proyecto general. Revisión 1.4.

**5.** **Estrategia general del Test**

La estrategia del test de aceptación está basada en la suposición de que el test del módulo y el test de integración serán realizados por el equipo de desarrollo.

**5.1** **Características de calidad**

La siguiente tabla muestra las características de calidad según el estándar ISO/IEC9126 que deben ser testeadas, y su importancia relativa.

|  |  |
| --- | --- |
| Características | Importancia Relativa [%] |
| Funcionalidad | 40 |
| Usabilidad | 20 |
| Eficiencia | 20 |
| Mantenibilidad | 20 |
| Total | 100 |

Funcionalidad:

El proyecto consiste en sumar nuevas funcionalidades a la placa EDU-CIAA-NXP, por lo tanto, es importante medir el cumplimiento de los requerimientos de funcionalidad.

Usabilidad:

Es la capacidad de que el software pueda ser entendido, aprendido y usado. Y como se trata de un proyecto orientado al ambiente educativo, es muy importante para cumplir el objetivo de funcionar como herramienta pedagógica. El dispositivo debe ser amigable, de forma tal que cualquier estudiante o docente con poco entrenamiento pueda usar uso de la plataforma robótica.

Eficiencia:

Es la capacidad del software tener una determinada prestación de los recursos del sistema bajo las condiciones de trabajo.

Mantenibilidad:

El software del Robot debe tener la posibilidad de cambiar ciertos parámetros de configuración, por lo tanto, es una de las características necesarias.

**5.2.** **Asignación de niveles de prueba a las características de calidad**

En la siguiente tabla se observan los niveles de prueba asignados a las distintas características de calidad detalladas en el punto anterior.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Funcionalidad | Usabilidad | Eficiencia | Mantenibilidad |
| Importancia Relativa [%] | 40 | 20 | 20 | 20 |
| Test de integración de Hardware | ++ |  | + |  |
| Test de unitario de Software | ++ | + | + |  |
| Test Integración de Software | + | ++ | + |  |
| Test de sistema | + |  |  | ++ |
| Test de Aceptación | ++ | + |  | + |
| Test de campo | ++ | + |  | + |

++ esta característica de calidad se cubrirá en su totalidad.

+ este nivel de prueba cubrirá esta característica de calidad.

(vacío) esta característica de calidad no es un problema en este nivel de prueba.

La tabla indica que de del total de pruebas el 40% serán de funcionalidad, 20% de usabilidad, de eficiencia otros 20% y de mantenibilidad el 20%.

**6.** **Estrategia por nivel de prueba**

Por cada nivel de prueba indicado en el [punto 5.2](#Asignacion_de_niveles_de_prueba). Asignación de niveles de prueba a las características de calidad, se evalúa la estrategia con la que se lo aborda.

**6.1.** **Selección de características de calidad y determinación de la importancia relativa por nivel de prueba**

Para cada nivel de prueba, se adoptan las siguientes importancias relativas:

* *Integración de Hardware*

|  |  |
| --- | --- |
| Característica de calidad | Importancia relativa [%] |
| Funcionalidad | 80 |
| Usabilidad | -- |
| Eficiencia | 20 |
| Mantenibilidad | -- |

* *Prueba unitaria de software*

|  |  |
| --- | --- |
| Característica de calidad | Importancia relativa [%] |
| Funcionalidad | 60 |
| Usabilidad | 20 |
| Eficiencia | 20 |
| Mantenibilidad | -- |

* *Prueba de integración de software*

|  |  |
| --- | --- |
| Característica de calidad | Importancia relativa [%] |
| Funcionalidad | 20 |
| Usabilidad | 60 |
| Eficiencia | 20 |
| Mantenibilidad | -- |

* *Prueba de sistema*

|  |  |
| --- | --- |
| Característica de calidad | Importancia relativa [%] |
| Funcionalidad | 20 |
| Usabilidad | -- |
| Eficiencia | -- |
| Mantenibilidad | 80 |

*.*

* *Prueba de aceptación*

|  |  |
| --- | --- |
| Característica de calidad | Importancia relativa [%] |
| Funcionalidad | 60 |
| Usabilidad | 20 |
| Eficiencia | -- |
| Mantenibilidad | 20 |

* *Prueba de campo*

|  |  |
| --- | --- |
| Característica de calidad | Importancia relativa [%] |
| Funcionalidad | 60 |
| Usabilidad | 20 |
| Eficiencia | -- |
| Mantenibilidad | 20 |

.

**6.2.** **Dividir el sistema en subsistemas**

De acuerdo con el documento de arquitectura [2], el sistema se encuentra dividido en los siguientes subsistemas, que para las estrategias de prueba se tomarán aquellos subsistemas que ameriten importancia:

Subsistemas principales

**Parte A.** Módulo control de motores.

**Parte B**. Módulos de comunicaciones WIFI.

**Parte C.** Módulo principal.

**Parte D**. Módulo encoder.

Subsistemas de baja importancia

* + Módulo acelerómetro.
  + Módulos de comunicaciones Bluetooth.
  + Módulo de posicionamiento (GPS).
  + Módulo sensor de ultrasonido.
  + Módulo de administrador de energía.

**6.3. Determinar la importancia relativa de los subsistemas**

|  |  |
| --- | --- |
| **Subsistema** | **Importancia relativa [%]** |
| **Parte A.** Módulo control de motores | 20 |
| **Parte B.** Módulos de comunicaciones WIFI. | 30 |
| **Parte C.** Módulo principal. | 30 |
| **Parte D.** Módulo encoder. | 20 |
| **Total** | **100** |

**6.4.** **Determinar la importancia de test por combinaciones de subsistema / característica de calidad**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Importancia relativa**  **[%] 100** | | **Parte A** | **Parte B** | **Parte C** | **Parte D** |
| 20 | 30 | 30 | 20 |
| Funcionalidad | 30 | ++ | + | ++ | + |
| Usabilidad | 30 | ++ | ++ |  |  |
| Eficiencia | 20 |  |  | ++ | + |
| Mantenibilidad | 20 | + |  | + | + |

++ esta característica de calidad se cubrirá en su totalidad.

+ este nivel de prueba cubrirá esta característica de calidad.

(vacío) esta característica de calidad no es un problema en este nivel de prueba.

**6.5.** **Establecer las técnicas de test a ser utilizadas**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Técnica de test aplicada | **Parte A** | **Parte B** | **Parte C** | **Parte D** |
| STT | + | + | + | + |
| CFT |  |  | + | + |
| ECT |  |  | + | + |
| CMT | + | + | + | + |

Referencias:

STT State transition testing

CFT Control flow test

ECT Elementary comparison test

CTM Classification-tree method

**7.** **Referencias**

[1][“ Implementación de Aprendizaje por Refuerzo en un Robot”](Test%20basis/ROBOT_RL-PROY-014.pdf) Ver1.4; Autor: Pablo D. Folino; Año 2020; páginas (8-9).

[2][“ Trabajo Práctico 4: Arquitectura y Diseño”](Test%20basis/ROBOT_RL-ARQ-002.docx) Ver 2.0; Autor Autor: Pablo D. Folino; Año 2020

[3[]” Especificaciones de Requerimientos de Software”](Test%20basis/ROBOT_RL-ER-001.docx) Ver1.0; Autor Autor: Pablo D. Folino; Año 2020