

**Università Ca'Foscari di Venezia**

***Dipartimento di Scienze Ambientali, Informatica e Statistica***

**Corso di Ingegneria del Software**

**A.A. 2018-2019**

**Docente Responsabile : prof. Agostino Cortesi**

## **Piano di Testing**



**Gruppo LegoStoneng**

**864553 - Marco Barnà**

**862653 - Pasquale Soldano**

**865154 - Alvise Battistich**

**864881 - Riccardo Denticò**

# SOMMARIO

---

<b>1. STRATEGIA DI TESTING</b>	<b>3</b>
1. Introduzione	
2. Incremental testing	
3. White-box testing	
4. Black-box testing	
<b>2. TRACCIABILITA' DEI REQUISITI</b>	<b>3</b>
<b>3. ELEMENTI TESTATI</b>	<b>4</b>
<b>4. SCHEDULE DEL TESTING</b>	<b>4</b>
<b>5. PROCEDURE DI REGISTRAZIONE DEI TEST</b>	<b>5</b>
<b>6. EVOLUZIONE DEL SISTEMA</b>	<b>5</b>
<b>7. REQUISITI HARDWARE E SOFTWARE</b>	<b>6</b>
<b>8. VINCOLI</b>	<b>6</b>

# 1. Strategia di Testing

---

## 1. *Introduzione*

A seguito di una attenta analisi, si è deciso di adottare le seguenti modalità di testing. Si utilizzeranno dei dispositivi Android che variano dalla versione 5.0 in su assieme all'emulatore virtuale presente su Android Studio.

Il seguente documento raccoglie una rapida panoramica delle attività svolte dal gruppo Lego StonEng in cui saranno esposte la metodologia di testing e come verrà fatto.

## 2. *Incremental Testing*

Procederemo al testing dell'applicazione mano a mano che verranno implementate le funzionalità (Dato che usiamo il metodo agile). In questo modo, verificheremo la solidità dell'applicativo durante il processo di sviluppo del software. Ogni volta che verrà implementata una delle classi specificate nel documento di progettazione, verranno effettuati dei test per rilevare tempestivamente eventuali errori di implementazione.

## 3. *White-Box Testing*

Contemporaneamente all'Incremental Testing adotteremo un approccio di test a White-Box. Questi consisteranno in test dettagliati che andranno a valutare la struttura dell'applicazione in modo da attuare una verifica finale per accettare la correttezza dell'intero prodotto software.

## 4. *Black-Box Testing*

Al completamento dell'implementazione dell'applicazione verranno effettuati dei test di tipo "scatola nera", ovvero utilizzando degli input definiti (validi e non validi) e confrontando i risultati ottenuti con quelli previsti. Questa tecnica di testing è molto utile per rilevare errori che utenti comuni, senza alcuna conoscenza implementativa dell'applicazione, potrebbero riscontrare durante l'utilizzo quotidiano. In questo modo testeremo che il robot può funzionare senza problemi.

# 2. Tracciabilità dei requisiti

---

I requisiti funzionali dell'applicazione verranno verificati seguendo le specifiche dei requisiti indicate nel Documento di Analisi e Specifica, nel quale sono ben elencati, sarà cura nostra fare un'ulteriore verifica dato l'utilizzo dell'approccio Black-box.

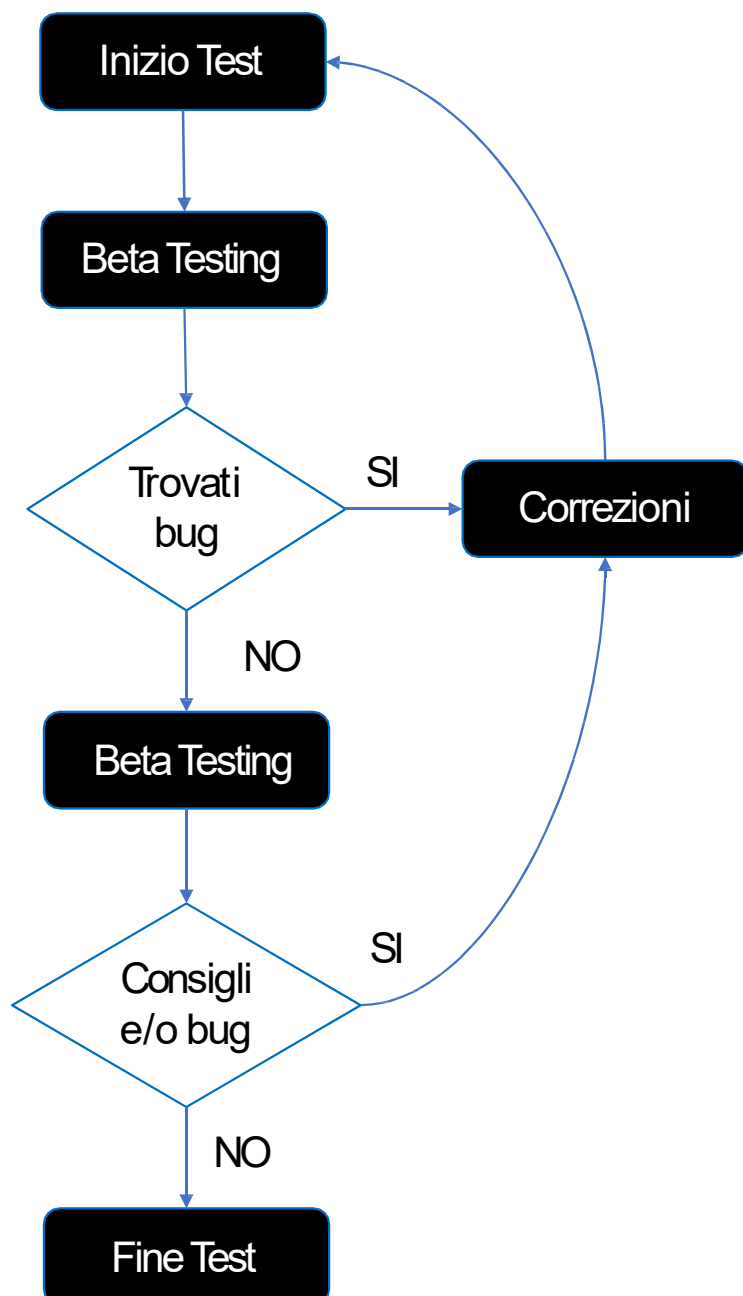
Allo stesso modo in corso di sviluppo avendo una metodologia agile verranno man mano verificate che le funzionalità aggiuntive a seguito di test (incremental testing) rispettino i requisiti richiesti.

### 3. Elementi testati

I test che verranno effettuati durante tutto il periodo di sviluppo di app, in particolare ad ogni sviluppo, correzione, aggiunta di una funzionalità. Le componenti testate sono principalmente relative ai sensori del LegoEV3 e alla sua connessione e comunicazione con lo smartphone. Testiamo ogni riga di codice in modo da riuscire ad avere un codice il più possibile privo di bug e malfunzionamenti. Ci occuperemo anche dei test relativi alla parte grafica con l'aiuto di persone esterne al team di progetto per la fase di beta-testing.

### 4. Schedule del testing

La pianificazione dei test può essere riassunta dal seguente diagramma:



## 4. Schedule del testing

Come si può notare dal diagramma di usso, la fase di testing passerà, anche più volte, attraverso alcuni passaggi fondamentali:

- Alpha-Testing: membri del team e riguarda l'implementazione del codice e dell'interfaccia grafica.
- Beta-Testing: utenti selezionati a valutare l'interfaccia e l'usabilità dell'applicazione.
- Correzione: modifiche eventuali in caso di bug e/o consigli.

## 5. Tabella del testing

La tabella conterrà le informazioni riguardanti i test effettuati. Ogni test ha un nome, un tipo, il tester, la data e l'esito. L'esito potrà essere negativo o positivo. Negativo se il risultato non è quello atteso, positivo se lo è. La tabella del testing verrà aggiornata ad ogni test che effettueremo.

ID TEST	TIPO TEST	TESTER	DATA	ESITO
TE-01	Compatibilità libreria	Pasquale - Alvise Riccardo - Marco	25-11-2018	Risultato Atteso - OK
TE-02	Connessione Bluetooth	Pasquale - Alvise Riccardo - Marco	07-12-2018	Risultato Atteso - OK
TE-03	Test motori	Pasquale - Alvise Riccardo - Marco	09-12-2018	Risultato Atteso - OK
TE-04	Test sensori	Pasquale - Alvise Riccardo - Marco	09-12-2018	Risultato Atteso - OK
TE-05	Test Splash Screen	Pasquale - Alvise Riccardo - Marco	09-12-2018	Risultato Atteso - OK
TE-06	Test algoritmo Avoid Obstacles	Pasquale - Alvise Riccardo - Marco	16-12-2018	Risultato Atteso - OK
TE-07	Test algoritmo Calculate Distance	Pasquale - Alvise Riccardo - Marco	16-12-2018	Risultato Atteso - OK
TE-08	Test voice Commands	Pasquale - Alvise Riccardo - Marco	27-12-2018	Risultato Atteso - OK
TE-09	Final Test	Pasquale - Alvise Riccardo - Marco	30/01/2019	Risultato Atteso - OK

## 6. Procedura di registrazione del testing

---

Come già anticipato nel punto 4. La fase di testing si dividerà in due periodi ben distinti:

- Periodo di Alpha-Testing:  
Il team di sviluppo controlla e supervisiona l'esecuzione del codice per scoprire eventuali anomalie e bug nel codice attraverso test mirati in ambienti controllati. Successivamente un primo sguardo generale all'interfaccia presentata cercherà di togliere incoerenze e bug di visualizzazione.
- Periodo di Beta-Testing  
Persone esterne al team di sviluppo utilizzeranno l'applicazione in diversi periodi nella sua tonalità allo scopo di trovare bug e comportamenti anomali che possono essere segnalati per una correzione futura dal team di sviluppo.

## 7. Requisiti hardware e software

---

Per effettuare i test useremo dispositivi che rispettano i seguenti requisiti:

- Sistema operativo Android 5.0 o successivi
- Connessione Bluetooth

Nello specifico i dispositivi usati per il testing saranno:

- Xiaomi Redmi Note 5 Plus
- Xiaomi Pocophone F1
- Samsung Galaxy Note 3 NEO
- Xiaomi Mi 4C
- Emulatore Android studio

Che presentano le seguenti versioni di Android:

- 8.1 (Oreo)
- 8.0 (Oreo)
- 5.1 (Lollipop)

Ci potrebbero anche essere delle variazioni per quanto riguarda i dispositivi utilizzati. Potrebbero verificarsi delle aggiunte o sostituzioni di alcuni di essi o potrebbe variare la versione del sistema operativo in caso di aggiornamenti firmware.



## 8. Vincoli

---

La fase di testing ha avuto inizio il 20 Ottobre 2018, insieme alla fase di sviluppo dell'applicazione. Il 10 novembre i membri del team hanno ultimato una prima versione dell'applicazione con le seguenti funzionalità implementate: funzionalità di base dell'interfaccia, connettività Bluetooth, guida manuale. Lo sviluppo dell'applicazione continuerà fino al 31 dicembre, al termine di questa i componenti del gruppo saranno impegnati nelle ultime attività di testing, che si concluderà prima del rilascio su [apkmirror.com](http://apkmirror.com) e/o su Google Playstore, in data 31 gennaio 2019.

