



# Piano di testing



**IO, LEGO**

## Autori

- Volpe Davide 862989
- Melissari Luca 865700
- Jasari Besar 862699
- Marcolin Giulio 862611
- Bertin Eleonora 865370

# Sommario

<b>Introduzione</b>	<b>3</b>
Scopo del documento	3
Struttura del documento	3
Glossario	3
<b>Processo di testing</b>	<b>4</b>
<b>Tracciabilità dei requisiti</b>	<b>4</b>
<b>Elementi testati</b>	<b>6</b>
<b>Schedule del testing</b>	<b>7</b>
<b>Procedure di registrazione dei test</b>	<b>8</b>
<b>Requisiti hardware e software</b>	<b>8</b>
<b>Vincoli per il testing</b>	<b>8</b>

# 1. Introduzione

## 1.1. Scopo del documento

Questo documento definisce i metodi di testing che useremo durante lo sviluppo del progetto.

## 1.2. Struttura del documento

Il documento è strutturato nella seguente maniera:

- **Processo di testing:** specifica le tipologie di testing utilizzate durante la procedura di testing.
- **Tracciabilità dei requisiti:** si analizzano tutti i requisiti funzionali elencati nel documento di Analisi e Specifica [sezione 4].
- **Elementi testati:** elenco di tutte le funzionalità dell'applicazione e del robot che verranno testate.
- **Schedule del testing:** rappresenta tutta l'attività di testing con l'ausilio di un diagramma UML.
- **Procedure di registrazione dei test:** specifica le modalità utilizzate per la raccolta degli esiti dei test effettuati, sia per la versione Alpha che Beta. Include una tabella di testing.
- **Requisiti hardware e software:** indica tutti i requisiti hardware e software necessari per effettuare i test.
- **Vincoli per il testing:** mostra i vincoli da rispettare durante tutte le fasi di testing.

## 1.3. Glossario

- **Bug:** identifica in informatica un errore nella scrittura del codice sorgente di un programma software.
- **Strategia Bottom-UP:** è una strategia di elaborazione dell'informazione e di gestione delle conoscenze.
- **Strategia Stress testing:** è una strategia che ha lo scopo di provare a "rompere" il sistema sotto test sovraccaricando le sue risorse o sottraendogli risorse.
- **Diagramma UML:** Il linguaggio UML è un linguaggio di modellazione e specifica. Viene usato, in questo contesto, per descrivere il dominio applicativo di un sistema software e/o il comportamento e la struttura del sistema stesso.
- **Android:** è un sistema operativo per dispositivi mobili sviluppato da Google Inc.
- **Linux:** è un sistema operativo per dispositivi desktop.
- **Windows:** è un sistema operativo per dispositivi desktop.

- **Android Studio:** è un ambiente di sviluppo integrato per lo sviluppo per la piattaforma android.
- **Eclipse:** è un ambiente di sviluppo integrato multi-linguaggio e multiplatforma.

## 2. Processo di testing

Il processo di testing che intendiamo applicare è suddiviso in due parti:

- **Alpha Testing**

Prima fase di testing con strategia Bottom-UP. Verranno testate le funzionalità mano a mano che esse verranno implementate, cosicché eventuali errori possano essere corretti immediatamente. Grazie a questo avremo la possibilità di decidere quali funzionalità mantenere e quali invece riconsiderare.

- **Beta testing**

In questa seconda fase grazie a una strategia di stress testing andremo a verificare il funzionamento dell'applicazione e robot sotto forte carico. Un'ulteriore modalità di testing sarà data dal rilascio del progetto a utenti esterni in modo da raccogliere il feedback necessario per l'ulteriore debugging e miglioramento delle funzionalità.

## 3. Tracciabilità dei requisiti

La seguente tabella indica, per ogni requisito funzionale del documento di Analisi e Specifica, la relativa procedura di testing effettuata per sia per la versione Alpha che per la versione Beta.

Durante la fase Alpha, dopo la verifica del codice seguono sempre una o più prove reali, eseguite dal team, che corrispondono a quelle specificate nel campo Beta Test. Durante la versione Beta, invece, i test saranno effettuati da un numero maggiore di utenti.

Codice Requisito	Nome Requisito	Alpha Test	Beta Test
REQ-F0	Connessione Bluetooth automatica	Verifica del codice della connessione Bluetooth	Avvio dell'applicazione e verifica della connessione automatica al Robot
REQ-F1	Scelta della cella e del colore da controllare	Controllo delle funzioni relative al tap su una cella	Tap sulle celle e verifica del corretto posizionamento dopo la selezione
REQ-F2	Calcolo del percorso	Verifica del codice del calcolo del percorso fino alla cella selezionata	Selezione di svariate celle e controllo del movimento del robot fino alla cella desiderata
REQ-F3	Controllo del colore	Controllo del codice relativo al sensore del colore	-
REQ-F4	Feedback del robot	Verifica della comunicazione tra robot e applicazione	Verifica che l'app riceve un messaggio (visualizzato all'utente) dopo ogni controllo del colore

## 4. Elementi testati

Dei seguenti elementi testeremo il corretto funzionamento e le prestazioni:

- Avvio dell'applicazione
- Connessione del robot all'applicazione
- Tempo di risposta delle varie funzionalità della applicazione
- Invio informazioni dall'applicazione al robot e viceversa
- Start della funzione ricerca del robot

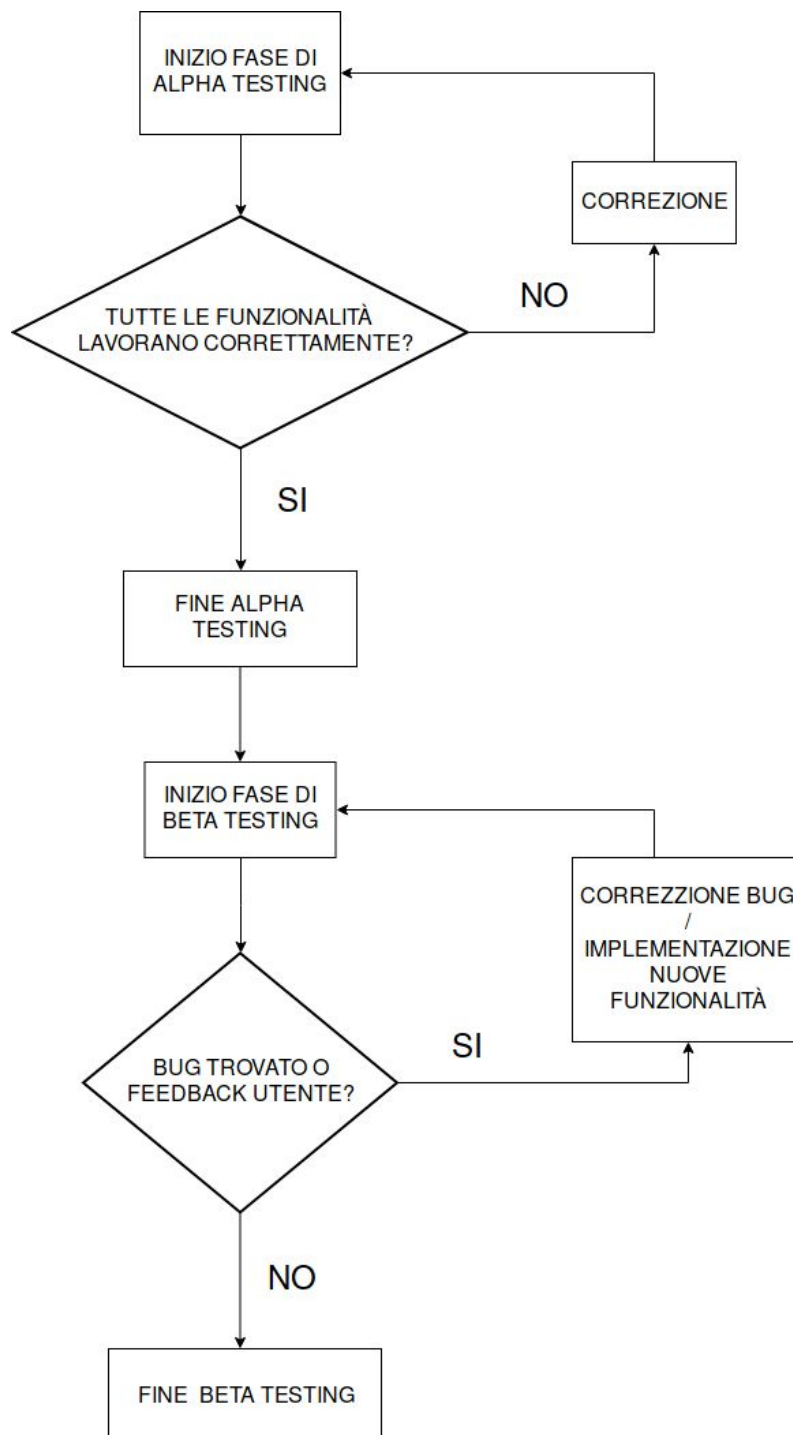
Alcune funzioni sono testabili solo dal team di sviluppo nella fase di Alpha testing poiché sono 'a basso livello', ossia funzionalità non visibili agli utenti e dunque non testabili durante la fase Beta.

Testeremo inoltre:

- Usabilità della applicazione  
L'utente deve essere in grado di capire intuitivamente, dopo aver seguito il tutorial, come utilizzare le interfacce e le varie funzionalità che l'applicazione offre.
- Sensibilità della applicazione  
L'applicazione deve rispondere agli errori in modo rapido ed efficiente cosicché mantenga la sua stabilità.
- Test di compatibilità  
Verifica del funzionamento dell'applicazione con diverse versioni del sistema operativo android.

## 5. Schedule del testing

La pianificazione di tutta la fase di testing e debugging è riassunta nel seguente diagramma:



## 6. Procedure di registrazione dei test

La fase di test è divisa, come detto, nelle parti Alpha e Beta.

La seguente tabella sarà utilizzata per la raccolta dei test effettuati durante la versione Alpha.

Utente	Funzionalità testata	Data	Sistema operativo	Dispositivo	Esito	Osservazioni
Giulio Marcolin	Connessione e comunicazione Bluetooth tra App Android e dispositivo Lego EV3	1/12/2018	Windows	Smartphone Android e Lego EV3	Successo	Si deve attendere alcuni istanti per la riuscita
Davide Volpe	Presentazione e correttezza della App Android	3/12/2018	Windows	Smartphone Android	Successo	-
Davide Volpe & Luca Melissari	Algoritmo dei movimenti del robot sulla matrice	5/12/2018	Windows	Smartphone Android e Lego EV3	Insuccesso	A volte si riscontrano anomalie dovute al giroscopio
Besar Jasari	Riconoscimento dei colore	8/12/2018	Windows	Lego EV3	Successo	-
Eleonora Bertin	Calibrazione e misure della matrice e dei movimenti del robot su di essa	9/12/2018	Windows	Lego EV3	Successo	-

Per quanto riguarda i test della versione Beta, gli utenti avranno la possibilità di mandare un feedback agli sviluppatori tramite mail (compilata direttamente tramite applicazione) oppure potranno riferire di persona il problema riscontrato agli sviluppatori.

## 7. Requisiti hardware e software

Per effettuare il testing sarà richiesto l'utilizzo di:

- Pc personali degli sviluppatori.
- Smartphone con sistema operativo Android.
- Software di emulazioni di dispositivi Android.
- Sistemi operativi Linux e Windows.



- Android studio.
- Eclipse.

## 8. Vincoli per il testing

Gli unici vincoli riguardano la durata temporale complessiva di tutta la fase di testing. Si distinguono dunque in base ad essa :

- **Alpha testing**

Durante questa fase il vincolo da rispettare è il tempo della stesura del codice della funzionalità da testare.

- **Beta testing**

Questa fase invece ha bisogno che l'applicazione sia stata rilasciata in fase di beta (27/12/2018).

Tutti i test devono essere conclusi entro il rilascio ufficiale dell'applicazione e del robot, entro il 31/01/2019.