



Università  
Ca'Foscari  
Venezia

# Piano di Testing

Data di consegna: 23/11/2017



## CRBM

Callegher Gianmarco [854612@stud.unive.it](mailto:854612@stud.unive.it)

Ragazzo Alessio [860550@stud.unive.it](mailto:860550@stud.unive.it)

Benvenuto Francesco [858843@stud.unive.it](mailto:858843@stud.unive.it)

Makaj Aurelio [859759@stud.unive.it](mailto:859759@stud.unive.it)

**Versione 1.0**

# Indice

- 1. Processo di testing adottato**
- 2. Elementi testati**
- 3. Schedule del testing:**
  - 3.1. Unite Test (Test di Modulo)**
  - 3.2. Integration Test (Test di Integrazione)**
  - 3.3. System Test (Test di Sistema)**
  - 3.4. User Acceptance Test (Collaudo, Test di Accettazione, Test di Validazione)**
  - 3.5. Alpha Test e Beta Test**
    - 3.5.1. Alpha Test**
    - 3.5.2. Beta Test**
  - 3.6. Regression Test (Test di non Regressione)**
  - 3.7. Stress Test (Test di Carico)**
- 4. Procedure di registrazione dei test**
- 5. Requisiti hardware e software utilizzati**
- 6. Vincoli che condizionano il testing**

# 1. Processo di testing adottato

La metodologia di testing adottata dal gruppo sarà di tipo bottom-up. Considerando la connessione tra i componenti del software come un albero, si inizierà, dunque, con il test delle foglie procedendo di padre in padre, il quale rappresenterà l'integrazione dei figli, verso la radice. Nel processo si realizzeranno dei “drivers” che simuleranno l'ambiente nel quale le componenti sono valutabili.

La scelta è ricaduta su questa metodologia in quanto è appropriata per lo sviluppo di sistemi object-oriented.

Alla fine della fase di sviluppo, quando il prodotto sarà nella fase “Alpha”, verranno utilizzate le metodologie di testing White Box e Black Box nel seguente modo: la prima servirà a testare l'applicazione con input generati *ad hoc* per testare determinate componenti.

Nello specifico, questi input sono scelti con la consapevolezza di come il sistema lavora e dei possibili edge cases. Mentre la seconda mirerà a simulare le azioni che un possibile utente andrà a compiere quando utilizzerà l'applicazione.

Inoltre dopo aver concluso il *White Box* testing l'applicazione entrerà nella fase “Beta” e verrà rilasciata nel Play Store (store ufficiale di google per scaricare le applicazioni) in questo modo il *Black Box* testing potrà essere fatto oltre che da tool appositi che simulano l'utilizzo dell'applicazione da utenti che proveranno il prodotto anche se non ancora completo.

## 2. Elementi testati

Considerata la connessione dei componenti come un albero, con la strategia precedentemente enunciata, il test di ogni singolo modulo verrà effettuato dai componenti del gruppo mentre il test del prodotto finale sarà affidato nuovamente al gruppo stesso oltre a degli utenti esterni (Beta tester) i quali avranno il compito di fornire feedback sull'usabilità dell'applicazione.

## 3. Schedule del testing

### 3.1. Unite test (Test di Modulo)

Ha l'obiettivo di individuare gli errori nel singolo modulo software, verificando non vi siano incongruenze tra le specifiche del singolo modulo e risultati ottenuti.

### 3.2. Integration (Test di Integrazione)

Poiché l'*unite test* non è adatto a testare *user interface* complesse o l'interazione tra componenti, dovremo sviluppare un *Integration Test*. Quindi dopo aver effettuato lo unit test sui singoli moduli, verrà testato il padre come integrazione software dei figli, seguendo, dunque, una linea gerarchica.

### 3.3. System Test (Test di Sistema)

Ha l'obiettivo di verificare il funzionamento dell'intera applicazione attraverso la verifica requisiti precedentemente stabiliti:

- Requisiti funzionali
- Requisiti non funzionali
- Requisiti di Documentazione (Aiuto all'uso dell'applicazione)

### 3.4. User Acceptance Test (Collaudo, Test di Accettazione, Test di Validazione)

I beta tester vengono chiamati a testare il prodotto finale, affinché possano controllare se le funzionalità dell'applicazione sono conformi alle direttive concordate.

### **3.5. Alpha Test & Beta Test**

#### **3.5.1. Alpha Test**

In questa fase il team di sviluppo farà un primo controllo globale dell'applicazione

#### **3.5.2. Beta Test**

Viene eseguito, prima del rilascio finale, da dei beta tester esterni al team di sviluppo, i quali hanno il compito di valutare la congruenza tra le funzionalità, la completezza e l'operatività dell'applicazione rispetto a quanto concordato dal documento di analisi e specifica, fornendo eventuali feedback.

### **3.6. Regression Test (Test di non Regressione)**

In seguito alla correzione di un bug verranno effettuati nuovi test al fine di verificare se l'errore è stato arginato e se ciò ha portato alla formazione di nuovi problemi

### **3.7. Stress Test (Test di Carico)**

L'applicazione viene portata ad un livello di computazione elevato, al fine di valutare i tempi di attesa nell'elaborazione di grandi quantità di informazioni. Si migliorerà l'applicazione se come risultato si avrà un basso livello di sopportazione.

## 4. Procedure di registrazione dei test

Verrà utilizzata una scheda, come da template, in cui ciascuna riga indica un requisito differente, per il quale verranno indicate le metodologie di testing, gli errori riscontrati e infine il responsabile del testing (se Alpha Test)

Template Scheda registrazione Test:

ID	Nome Requisito	Metodologie Testing	Bug	Responsabile

## 5. Requisiti hardware e software utilizzati

Il dispositivo mobile dovrà essere dotato di:

- una versione di Android pari o superiore alla 5.1
- il sensore GPS
- una connessione ad internet

## 6. Vincoli che condizionano il testing

I vincoli che condizionano il testing sono: di natura temporale, dettati dalla consegna del progetto in data 31/01/2017, e di natura quantitativa rispetto agli Open Data che potranno essere raccolti, rappresentati e rielaborati.