# 4. Schedule del testing: tempo e risorse allocate

La gestione del tempo e delle risorse seguirà le seguenti specifiche:

* **Inspection e Walkthrough**: dal momento che il nostro team ha deciso di adottare il pair programming nel corso della fase di programmazione, è già intrinsecamente presente una fase di analisi statica. Quindi riteniamo che sarà sufficiente svolgere un lavoro puramente dedicato a un’analisi statica approfondita del codice (di circa 2 ore) solamente ogni 2 settimane.
* **Incremental Testing**: avremo cura di usarlo all’aggiunta di ogni nuova funzione, per assicurarci che non solo essa sia corretta, ma anche che non porti a una regressione del sistema nel suo complesso. Coerentemente con il linguaggio adottato, che fa uso del paradigma ad oggetti, la modalità di testing scelta è bottom-up.
* **Thread Testing**: data la natura asincrona delle applicazioni Android, abbiamo ritenuto che fosse necessario adottare il thread testing, per ridurre la probabilità di eventuali problemi dovuti alla concorrenza. Avremo cura di eseguirlo al termine dello sviluppo di ogni iterazione del modello evolutivo. In aggiunta svolgeremo dei test supplementare in itinere quando ci sembrerà necessario, per esempio in presenza di porzioni di codice che richiedono particolare attenzione alla concorrenza tra thread.
* **Back-to-back Testing**: ad ogni nuova versione dell’applicazione, questa verrà confrontata con le versioni precedenti in modo da controllare che non avvengano regressioni.
* **Black-box and White-box**: nel corso della fase di testing faremo principalmente uso dell’approccio black-box per controllare ogni nuovo modulo nel momento in cui viene aggiunto e il sistema nel suo complesso, riservandoci di usare in aggiunta anche l’approccio white-box per i casi più delicati e critici per il funzionamento complessivo dell’applicazione.

# 5. Procedure di registrazione dei test

Per la registrazione dei risultati dei test useremo una tabella con i seguenti campi:

* **CRF (Codice Requisito Funzionale)**: codice del requisito funzionale testato
* **Descrizione**: descrizione del test svolto
* **Data**: data di esecuzione del test
* **Tester**: membro del gruppo che ha eseguito il test
* **Risultato atteso *(% successi)***: numero di successi, in percentuale, che il test deve raggiungere per considerarsi superato
* **Risultato effettivo *(% successi)***: numero di successi, in percentuale, che il test ha raggiunto
* **Numero test**: numero di test svolti
* **Note**: osservazioni e eventuali misure correttive

| **CRF** | **Descrizione** | **Data** | **Tester** | **Risultato atteso**  ***(% successi)*** | **Risultato effettivo**  ***(% successi)*** | **N° test** | **Note** |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

# 6. Requisiti hardware e software utilizzati

* 4 dispositivi Android dotati di WI-FI, Bluetooth, accelerometro e GPS
* Strumenti di Android Studio (emulatore, debugger e profiler)
* Framework JUnit

Riteniamo che per una più esaustiva esecuzione dei test sia utile adoperare dispositivi con caratteristiche hardware e software diverse. Per questo motivo ciascun membro del team metterà a disposizione il proprio cellulare, fornendo al team un totale di 4 dispositivi dotati di wifi, bluetooth, accelerometro e GPS.

| Marca | Nome | Versione Android | CPU  (GHz) | RAM  (GB) | Display  (‘’) |
| --- | --- | --- | --- | --- | --- |
| Samsung | Galaxy A40 | 10 | 2x1.8 + 6x1.6 | 4 | 5.9 |
| Motorola | Moto X Play | 6.0 | 4x1.7 + 4x1.0 | 2 | 5.5 |
| Honor | 6X | 7.0 | 4x2.1 + 4x1.7 | 3 | 5.5 |
| OnePlus | One | 6.0 | 4x2.5 | 3 | 5.5 |

*Specifiche dei dispositivi di testing.*

In aggiunta useremo strumenti già presenti nell’ambiente di sviluppo Android Studio, quali emulatore, debugger e profiler.

Allo scopo di aumentare l’efficienza del nostro lavoro di testing, una parte di questi test sarà automatizzata usando il framework JUnit.

# 7. Vincoli che condizionano il testing

Con l’inizio della fase di programmazione, avrà inizio anche la fase di testing che continuerà fino alla consegna del progetto in data 15/01/2021.

Siccome il progetto e lo sviluppo dell’applicazione è no-profit, abbiamo deciso di limitarci a usare le risorse già a nostra disposizione (come ad esempio i nostri cellulari) per testarne il funzionamento senza fare grandi investimenti monetari.

Possibili termini glossario

* CRF (codice requisito funzionale)
* Pair programming
* JUnit
* Emulatore
* Debugger
* Profiler