

tesi di laurea

Sperimentazione di Tecniche di Testing Automatico per applicazioni Android

Anno Accademico 2011/2012

relatore

Ch.ma Prof.ssa Anna Rita Fasolino

correlatore

Ing. Domenico Amalfitano

candidato

Gennaro Imperato

Matr. 534/867

La crescente richiesta di applicazioni per dispositivi mobili di qualità richiede una maggiore attenzione nei processi di sviluppo software.

In particolare, il testing e la sua automazione svolgeranno un ruolo strategico per garantire la qualità delle applicazioni sviluppate.

Attualmente Android è il più popolare sistema operativo per dispositivi mobili.

Garantire l'affidabilità delle applicazioni per Android è importante e sarà un fattore strategico per confermare e ampliare il suo successo.

Android si è affermato in poco tempo come primo market share con un numero di dispositivi su cui è installato pari a quasi il triplo rispetto a Symbian e iOS.

Android è uno stack software per dispositivi mobili il cui kernel deriva da quello di Linux.

Le applicazioni Android vengono realizzate componendo i quattro elementi fondamentali dell' Application Framework:

- **Activity;**
- **Broadcast Receiver;**
- **Content Provider;**
- **Service.**



Le Activity hanno la responsabilità di gestire l'interfaccia grafica. Ogni Activity è composta da oggetti, chiamati widget, con cui l'utente interagisce scatenando eventi e fornendo input.

Le applicazioni Android sono event-driven.

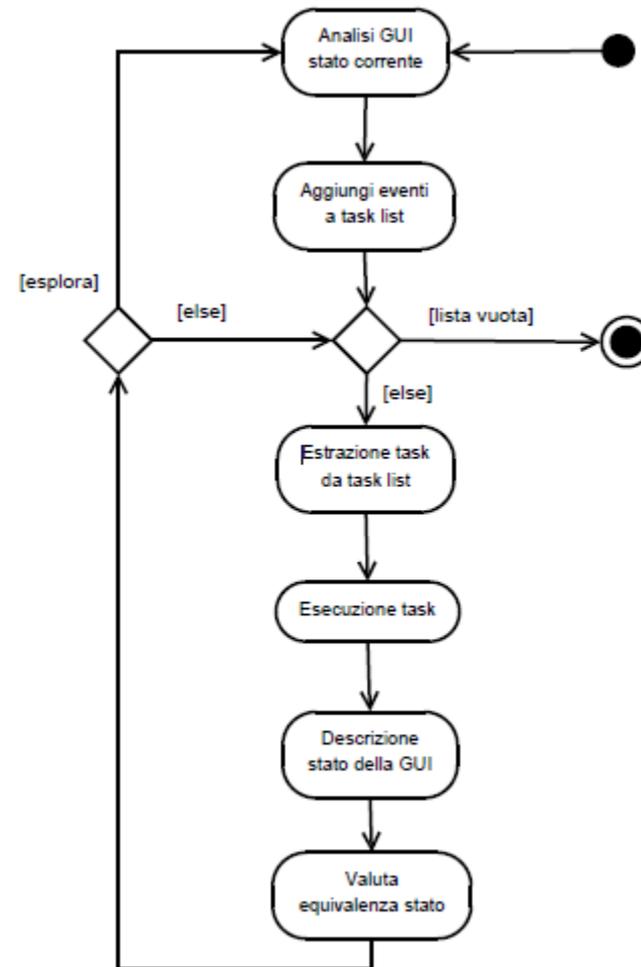
Il passaggio da un'Activity all'altra è scatenato dalle interazioni dell'utente con i widget.



Le tecniche di Gui Testing possono dividersi in tre categorie:

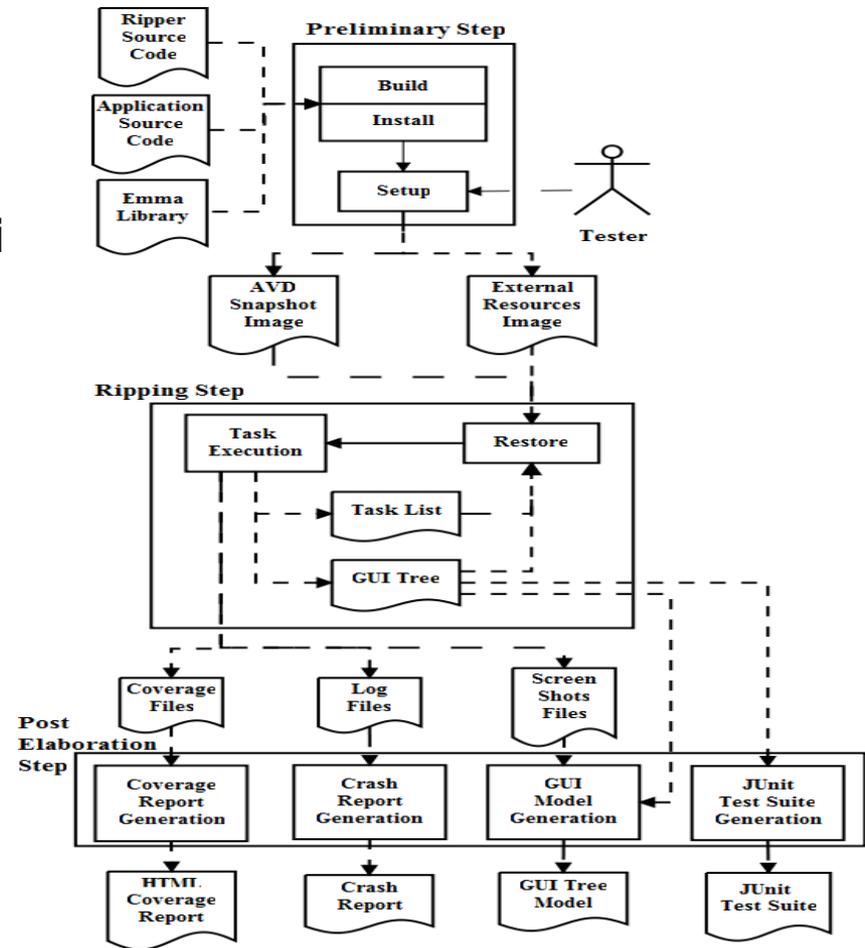
- **Tecniche Model-Based, in cui esiste una descrizione dell'applicazione sotto test sufficiente alla generazione automatica dei Test Case per la verifica del software;**
- **Tecniche Random Testing, in assenza di un modello, l'applicazione viene esercitata in maniera casuale alla ricerca di eventuali eccezioni non gestite;**
- **Tecniche Crawler-Based, in cui l'applicazione viene esplorata senza una conoscenza pregressa della sua struttura seguendo una strategia di navigazione.**

Un Gui Ripper solleciterà un'applicazione software allo scopo di individuare eccezioni non gestite e/o di raccogliere informazioni analizzandone la GUI.



Il processo di Test si compone di tre fasi:

- Fase Preliminare;
- Fase di Ripping;
- Fase di Post-Elaborazione.



App	Codice Coperto	Bug rilevati	Tempo di Esecuzione	Tracce Eseguite
AardDict	66%	2	2,5 ore	93
AndBible	40%	1	26,5 ore	287
Book Catalogue	51%	1	35 ore	580
Mileage	54%	0	28 ore	556
TomDroid	49%	1	2	86
WordPress	39%	4	5	257