

tesi di laurea

# Valutazione Comparativa Sperimentale di Tecniche di Testing di Applicazioni Android

2012/2013

**relatore**

Ch.mo prof. Porfirio Tramontana

**correlatore**

Ing. Domenico Amalfitano

**candidato**

Mariano Alfonso Biscardi

Matr. 534002866



## Situazione di mercato

- +43% vendite smartphone nel 2012 (700 milioni di dispositivi)
- +70% traffico rete internet *mobile* rispetto al 2011
- Crescente necessità di aumentare produttività e connettersi alla rete in mobilità
- Android leader del mercato con il 68,8% del *market share*

## Problematiche

- Numerose alternative ad ogni singola applicazione
- Team di sviluppo composti da pochi membri
- Immettere sul mercato applicazioni sufficientemente testate in tempi brevi

## Soluzione

- Automazione del testing di sistemi interattivi

## Scopo della sperimentazione

- Analisi di alcune tecniche di esplorazione dell'interfaccia utente di applicazioni Android
- Confronto dei risultati
- Contestualizzazione dei risultati all'interno di realtà di sviluppo aziendale

## Strumenti utilizzati

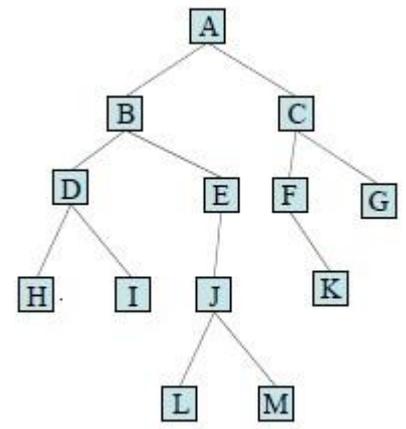
- Android Ripper (crawling + GUI Ripping)
- Emma (strumentazione del codice + valutazione copertura)
- R (analisi statistica + plotting grafici)

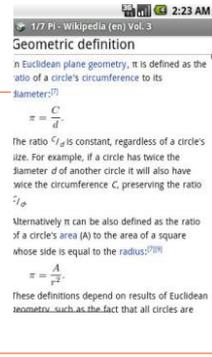
# Modalità di esplorazione del GUI-tree analizzate

- *Breadth* con criterio di equivalenza delle *activity*
- *Depth* con criterio di equivalenza delle *activity*
- *Breadth* senza criterio di equivalenza delle *activity*
- *Random* senza criterio di equivalenza delle *activity*

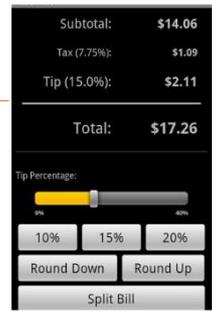
# Metriche di valutazione adottate

- Efficacia in termini di copertura (LOC/ Metodi/ Classi)
- Efficacia in termini di bug individuati
- Efficienza, intesa come numero di eventi scatenati
- Efficienza, intesa come tempo impiegato per portare a termine il test

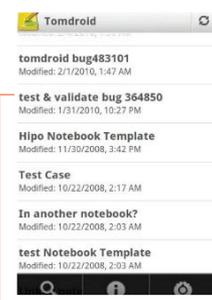




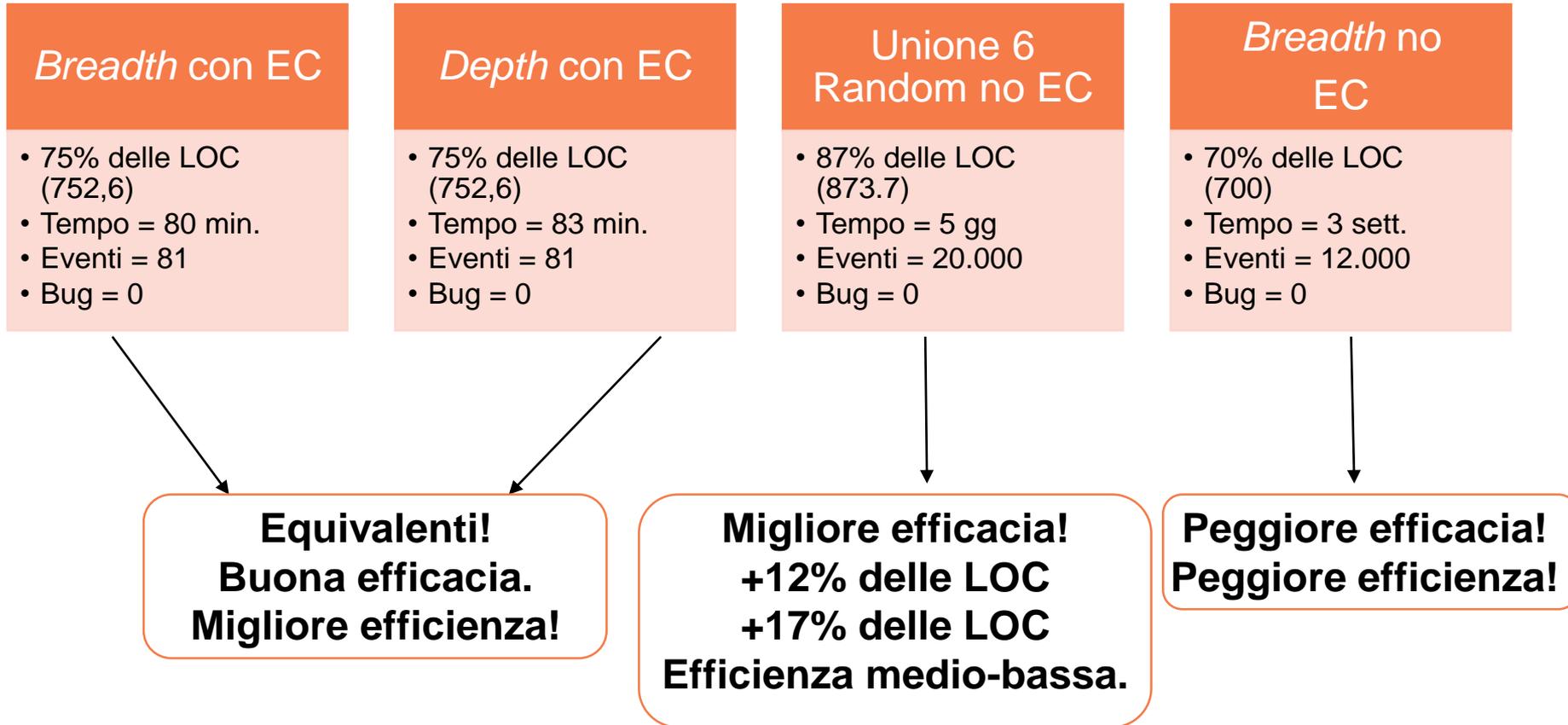
**Nome Applicazione:** *Aarddict*  
**Funzionalità:** Strumento di ricerca su dizionari offline  
**Caratteristiche:** 2097 LOC, 93 Classi, 424 Metodi



**Nome Applicazione:** *Tippy Tipper*  
**Funzionalità:** Strumento per il calcolo della mancia  
**Caratteristiche:** 999 LOC, 42 Classi, 225 Metodi

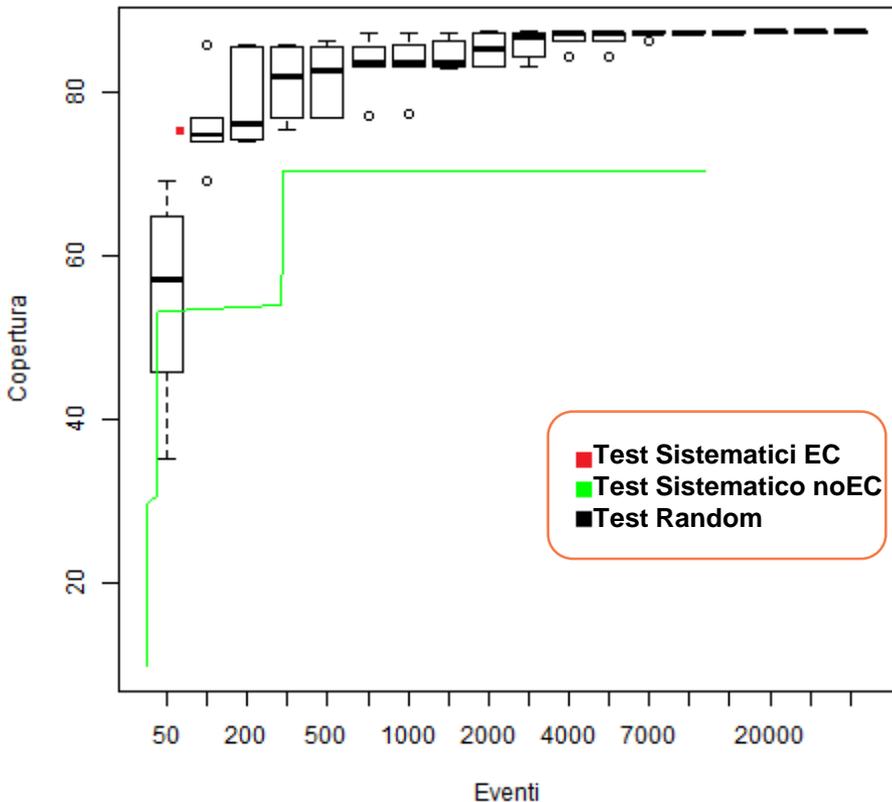


**Nome Applicazione:** *Tomdroid*  
**Funzionalità:** Client Tomboy per Android  
**Caratteristiche:** 3860 LOC, 133 Classi, 707 Metodi



**Tippy Tipper**

## 6 Test Random con 6 seed diversi →



#Events	CoverageUnion	Average	StDev
50	82.5425425425426	54.9015682349016	12.768417623629
100	87.3573573573574	75.980980980981	5.48385861761331
200	87.3573573573574	78.7103770437104	5.55612155822039
300	87.3573573573574	81.3446780113447	4.33598540180973
500	87.3573573573574	81.9402736069403	4.15643492590729
700	87.3573573573574	83.4901568234902	3.41908104714735
1000	87.3573573573574	83.5735735735736	3.38691761793636
1500	87.3573573573574	84.5829162495829	1.85038945224343
2000	87.4574574574574	85.3186519853186	2.02093813967744
3000	87.4574574574574	86.021021021021	1.83763446418022
4000	87.4574574574574	86.7067067067067	1.20909982335654
5000	87.4574574574574	86.7233900567234	1.21886687293399
7000	87.4574574574574	87.2238905572239	0.3833541972962
10000	87.4574574574574	87.4074074074074	0.05482708281972
15000	87.4574574574574	87.4074074074074	0.05482708281972
20000	87.4574574574574	87.4240907574241	0.051691469410171

Test di normalità di *Shapiro*



*Wilcoxon Mann Whitney U test*

Da 300 eventi: “viene rifiutata l’ipotesi che il test random sia peggiore dei sistematici in ampiezza/profondità con EC”

Analisi del trend e valutazione statistica

## Test Random e Test Sistemati con EC: differenze di copertura

```
String RoundType = PreferenceManager.getDefaultSharedPreferences().getString("round_tip");
if (RoundType.equals("round_tip"))
{
    return true;
}
else
{
    return false; /ramo non coperto dai sistematici
}

public static float getExcludeTaxRate(Context context)
{
    return PreferenceManager.getDefaultSharedPreferences().getFloat("exclude_tax_rate", 0.0f);
}
```

Specifiche sequenze di esplorazione portano alla copertura di nuove linee di codice

## La presenza di *dead-code* impedisce di valutare correttamente la copertura

```
@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    super.onCreateOptionsMenu(menu);

    //MenuInflater inflater = getMenuInflater();
    //inflater.inflate(R.menu.menu, menu);
    FlurryAgent.onEvent("Disabled Menu Button");
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    switch(item.getItemId())
    {
        case R.id.settings:
            startActivity(new Intent(this, Settings.class));
            FlurryAgent.onEvent("Settings Button");
            return true;
    }
    return false;
}
```

In assenza di *dead-code* i livelli di copertura raggiungibili sono più alti

*Breadth con EC*

- 44% delle LOC (928)
- Tempo = 33 min.
- Eventi = 49
- Bug = 1

*Depth con EC*

- 44% delle LOC (928)
- Tempo = 36 min.
- Eventi = 49
- Bug = 1

Unione 6  
Random no EC

- 45% delle LOC (949,5)
- Tempo = 7 gg
- Eventi = 30.000
- Bug = 1

*Breadth no EC*

- 45% delle LOC (949,5)
- Tempo = 3 sett.
- Eventi = 11.000
- Bug = 0

Aarddict

*Breadth con EC*

- 36% delle LOC (1396,2)
- Tempo = 140 min.
- Eventi = 115
- Bug = 1

*Depth con EC*

- 33 % delle LOC (1306)
- Tempo = 125 min.
- Eventi = 115
- Bug = 2

Unione 6  
Random no EC

- 59% delle LOC (2292.3)
- Tempo = 9 gg
- Eventi = 30.000
- Bug = 3

*Breadth no EC*

- 40% delle LOC (1544)
- Tempo = 5 sett.
- Eventi = 27.500
- Bug = 1

Tomdroid

## Predicibilità dei valori di copertura

- Non predicibili!
- Dipendono dalla complessità dell'applicazione

## Test sistematici con EC

- I più efficienti in assoluto
- I più efficaci in tempi brevi
- Adatti a metodologie di sviluppo agili (XP o TDD)

## Test sistematico *breadth* no EC

- Meno efficiente in assoluto
- Meno efficace in assoluto
- I valori di copertura raggiunti non giustificano il tempo di esecuzione

## Test Random

- I più efficaci in assoluto
- Efficienza medio-bassa
- Adatto al test di accettazione finale