Zesium mobile d.o.o

Mičurinova 8

21000 Novi Sad

M

Anja Bešić

Branislav Manojlović

**Praktikum**

**Alati za testiranje mobilnih aplikacija 1**

Monkey testing, Monkey Runner

Novi Sad, 2017

**Sadržaj:**

Uvod ……. 1

1. Testiranje softvera …. 2

2. Monky …. 2

3. Monkey testing ….. 3

4. Alat Monkey Runner …. 4

5. Primeri …. 6

6. Zaključak … 10

Literatura …. 11

**Uvod**

Testiranje je proces evaluacije softvera i svih njegovih pratećih komponenata, kako bi se utvrdilo da li su kreirani u skladu sa zahtevima, da li odgovaraju svrsi za koju su namenjeni I da li postoje defekti (bagovi) u njima.

Testiranje podrazumeva puštanje sustema (softvera) u rad, sa ciljem da se identifikuju grešeke ili nedostaci tog sistema u odnosu na očekivane rezultate. Svako ponašanje softvera koje se ne poklapa sa originalnim zahtevima, predstavlja grešku koju je potrebno identifikovati i otkloniti.

U užem smislu, testiranje predstavlja proveru da li je softver u potpunosti implementiran prema zahtevima korisnika.

U širem smislu, testiranje predstavlja sistem kontrole kvaliteta kojim se proverava softver I sve njegove prateće komponente.

Kvalitet softvera je opisan zahtevima kojima se dfefiniše šta se najčešće očekuje od softverskih komponenti. Cilj testiranja softvera je provera da li su navedeni zahtevi ispunjeni.

Mobilno testiranje zahteva da se većina testova izvodi manuelno. Ipak, postoje delovi procesa mobilnog testiranja koji mogu biti automatizovani. Testeri mobilnih aplikacija trebaju težiti da automatizuju deo procesa testiranja kako bi dobili brže povratne informacije o stanju aplikacije.

Namena pratikuma je da objasni osnovne pojmove u iz oblasti testiranja mobilnih aplikacija , da opiše metodu automatizovanog testiranja na principu Monkey testing-a i da prikaže alat za automatsko testiranje mobilnih Andriod aplikacija Monkey Runner i da opiše način njegove upotrebe.

Prvo poglavlje, opisuje faze u životnom ciklusu testiranja softvera, proces testiranja, podele testiranja i strategije testiranja, sa posebnim naglaskom na Black box testiranje.

Drugo poglavlje, prikazuje osnovne vrste mobilnih aplikacija i razlike u njima.

Treće poglavlje, definiše principe i prikazuje metod Monkey testiranja native mobilnih aplikacija.

Četvrto poglavlje, opisuje alat za kreiranje automatizovanih testova, namenjenih za testiranje native mobilnih aplikacija po imenu Monkey Runner.

Peto poglavlje, sadrži praktične primere upotrebe Monkey Runner test alata sa ispisom progamskog koda, upotrebljenog u testiranju I napisanog na programskom jeziku Python.

Šesto poglavlje je zaključak u kome su izneta zapažanja o upotrebljenim alatima i navedene moguće tendencije daljeg razvoja u oblasti automatizovanja procesa testiranja native mobilih aplikacija.

**1. Testiranje softvera**

Proces testiranja obuhvata veliki broj aktivnosti i usko je povezan sa procesom razvoja softvera.

Testiranje se definiše kao proces evaluacije softvera i svih njegovih pratećih komponenti, kako bi se utvrdilo da li su kreirani u skaldu sa zahtevima, da li odgovaraju svrsi za koju su namenjeni i da li postoje defekti (bagovi) u njima.

Bag (engl. Bug) ili defekt, se definiše kao pogrešno ponašanje softvera. On počinje od trenutka kada se pronađe greška u sofveru i prestaje onda kada se ta greška otkloni.

**1.1. Faze životnog ciklusa u testiranju softvera**

Svaka kompanija kreira soj životni ciklus u testiranju softvera. U opštem slučaju, životni ciklus u testiranju softvera obuhvata sledeće faze:

1. Analiza zahteva – Requirements analysis,
2. Planiranje testiranja – Test planing,
3. Razvoj test slučajeva – Test case development,
4. Izvršavanje test slučajeva – Test case executing,
5. Generisanje izveštaja o rezultatima testa – Test results reporting,
6. Generisanje izveštaja o defektima – Defects reporting,
7. Regresiono testiranje – Regression testing,
8. Zaključivanje testova – Test closure.

**1.2. Proces testiranja**

Tokom procesa razvoja softvera, potrebno je prepoznati kada treba početi a kada treba završiti testiranje.

* Početak testiranja – U životnom ciklusu razvoja softvera, testiranje može da počne od faze sakupljanja zahteva. Obično, od trenutka kada se ispuni ulazni kriterijum. Ulazni kriterijum (engl. Entry criteria) je minimalan set uslova koje treba ispuniti kako bi se počelo sa testiranjem.
* Završetak testiranja – Testiranje softvera može trajati sve do isporuke softvera krajnjim korisnicima. To zavisi od toga koji se model razvoja softvera koristi. Završetak testiranja se realizuje:
  + - dostizanjem roka, u smislu kraja faze testiranja ili kraja celog projekta,
    - uspešnim završavanjem svih predviđenih test aktivnosti,
    - znatnim smanjenjem broja bagova,
    - odlukom menadžmenta.

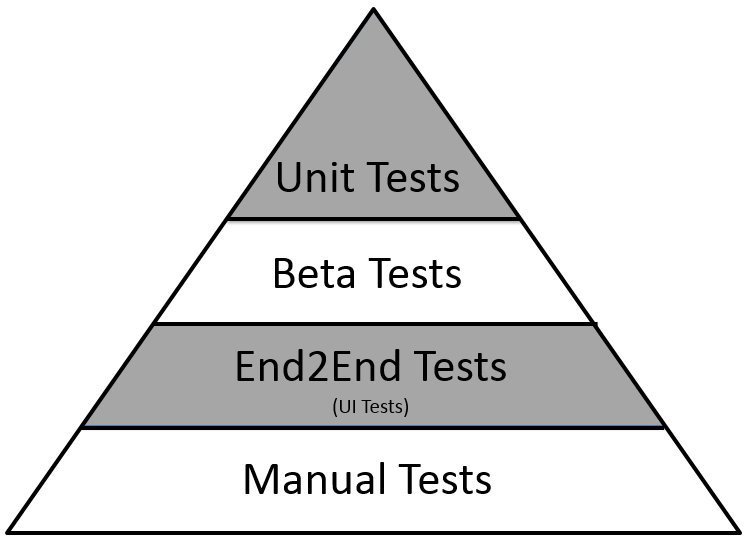
Drugim rečima, završetak testiranja, je definisan izlaznim kriterijumom. Izlazni kriterijum (engl. Exit criteria) je skup zahteva koji kada se ispune dozvoljavaju završetak testiranja.

U praksi, smatra se da je testiranje proces bez kraja, jer se ni za jedan softverski proizvod ne može reći da je u potpunosti 100% bez bagova (bug-free).

**1.3. Mobilno testiranje**

Mobilno testiranje zahteva da se većina testova izvodi manuelno. Ipak, postoje delovi procesa mobilnog testiranja koji mogu biti automatizovani. Testeri mobilnih aplikacija trebaju težiti da automatizuju deo procesa testiranja kako bi dobili brže povratne informacije o stanju aplikacije.

Piramida u mobilnom testiranju omogućava dobar miks manuelnog i automatizovanog testiranja i preporuka je da se on uzima u obzir pri planiranju testiranja mobilnih aplikacija.



Slika 1. Piramida u mobilnom testiranju

Najviši sloj u piramidi čini manuelno testiranje i ono predstavlja osnovu svakog projekta testiranja mobilnih aplikacija. Nakon manuelnog testiranja idu testiranja s kraja-na-kraj (end-to-end), beta testiranje i jedinično testiranje. Sloj beta testiranja je veoma važan, jer on omogućava da se ispune velika očekivanja krajnjih korisnika.

Sivi delovi piramide su testovi u kojima se mogu automatizovati test koraci. Beli delovi piramide podrazumevaju da se test koraci vrše manuelno.

**1.4. Podele testiranja**

Testiranje je višedimenzionalni proces koji možemo podeliti prema više aspekata I to prema: vrstama, nivoima, metodama i strategijama.

1. Vrste testiranja
   * Funkcionalno testiranje,
   * Testiranje opterećenja,
   * Testiranje sigurnosti,
   * Testiranje konfigracije,
   * Testiranje korisničkog interfejsa.
2. Nivoi testiranja
   * Jedinično testiranje,
   * Integraciono testiranje,
   * Sistemsko testiranje.
3. Metode testiranja
   * Manuelno testiranje,
   * Automatsko testiranje,
   * Testiranje po skriptovima,
   * Regresivno testiranje,
   * Istraživačko testiranje.
4. Strategije testiranja
   * Black box,
   * White box,
   * Grey box,
   * Uporedno testiranje.

**1.5. Strategije testiranja**

Strategije testiranja definišu na koji način će se vršiti testiranje. One određuju kako treba posmatrati sistem koji se testira I na koji način treba definisati test slučajeve. Primenjuju se na sve kompnente testiranja.

1. Black box testiranje – je testiranje u kojoj se sistem posmatra kao jedinstvena celina koja vrši neku funkcionalnost I za neke ulazne vrednosti vraća odrešene rezultate. Tester u ovom pristupu nema pristup izvornom kodu (source code) same aplikacije. Ovo testiranje se vrši tako što se sistemu šalju neki ulazni podaci I proverava se da li je rezultat koji se vraća u skladu sa zahtevima, bez poznavanja interne strukture sistema. Prednost ovog testiranja je velika efikasnost, nije potreban pristup izvornom kodu, jasno su razdvojene perspektive korisnika I developera. Mana ovog testiranja je ograničena pokrivenost po broju test scenarija, test slučajevi su komplikovani za dizajniranje.
2. White box testiranje – je strategija u kojoj se vidi unutrešanj stuktura sistema a testovi se prave tako da se proveri da li određeni algoritam ili struktura rade. Ova strategija je detaljnija I zahteva mnogo više vremena za pravljenje testova. Ovde testeri moraju donekle imati zanje o samom izvornom kodu, moraju ga pregledati kako bi pronalšli koji deo koda ne radi kako bi trebalo. Prednosti ove strategije je u tome što je moguća optimizacija koda, zahvaljujući poznavanju koda testeri mogu da ostvare bolju pokrivenost aplikacije pri testiranju, pišući bolje test scenarije. Mane su povećani troškovi, potreba za obučenijim testerima, teže realizovanje ovog testiranja jer su potrebni specijalni alati za testiranje.
3. Grey box testiranje – je strategija u kojoj se koristi ograničn nivo znanja o internom radu same aplikacije. Ovde testeri imaju pristup dokumentima za dizajniranje aplikacije i bazama podataka. Zahvaljujući tome, testeri mogu bolje da pripreme test scenarije i test podatke. Prednost ove strategije je što ona pruža prednosti I Black box I Grey box testiranja, testeri se ne oslanjaju na izvorni kod već na dokumentaciju, testovi se vrše sa aspekta korisnika a ne dizajnera. Mane se ograničena pokrivenost, testovi mogu postati redudantni, nije pogodno za algoritamsko testiranje.
4. Uporedno testiranje – je strategija u kojoj se testiranje vrši tako što se rezultati porede sa etalonom ili sa drugim funkcionalnostima u sistemu u kojima se očekuje da će rezultati biti isti. Etalon je funkcionalnost za koju se sigurno zna da je tačna. Uporedno testiranje se često koristi u slučaju da se neki sistem zamenjuje novim sistemom. Pošto novi sistem mora da sadrži veliki broj funkcionalnosti iz starog sistema koje su samo re-implementirane, najbolji test koji se može izvršiti je poređenje sa starim sistemom.

**2. Monkey testing**

Monkey testing podrazumeva test metodu u testiranju native mobilnih aplikacija na Android platformi u kojoj se aplikacija podvrgava nizu nasumičnih akcija u cilju provere stabilnosti same aplikacije. Ova metoda se bazira na upotrebi monkey alata, razvijenog isključivo za testiranje mobilnih aplikacija razvijenih za rad na Android platformi.

Monkey tool generiše niz nasumičnih koraka koji obuhvataju simulaciju klaktanja i dodirivanja pojedinih elemenata na ekranu aplikacije kao što su klik na dugme, dodir na link, sliku ili tekstualno polje za unos podataka. Izvođenje ovih koraka se odvija velikom brzinom i praktično nekontrolisano, čime se aplikacija testira na način koji tester ne može da reprodukuje.

Prednost primene ovake metode u testiranju je ta što se aplikacija testira na rad u stresnom radnom scenariju, čime se prektično proverava stabilnost aplikacije i utvršuje da li aplikacija može da doživi pad (engl. crash) ukoliko bi se koristila mnogo intenzivnije nego što je to predviđeno klijentskim zahtevima.

Testiranje metodom Monkey testing, se u praksi primenjuje tek nakon što je cela aplikacija potpuno istestirana i uklonjeni i korigovani svi pronađeni bagovi i defekti. Ukoliko se nakon uspešno sprovedenog Monkey testiranja, aplikacija pokazala kao stabilna i nije padala u toku testiranja, ona se može kao završen i stabilan proizvod isporučiti klijentu.

Osnovne komande koje se primenjuju prilikom Monkey testiranja obuhvataju sledeće:

**3. Monkey Runner alat za automatsko testiranje**

**4. Primeri**

1) Primer za uspostavljanje konekcije sa uredjajem ili emulatorom

2) Primer za touch() metodu

3) Primer za swipe() i drag() evente

4) Primer za komunikaciju sa korisnikom i koriscenje while ili for petlji i slicno.

5) Primer za testiranje Log in forme i popunjavanje text field-ova

6)

7)

**5. Zaključak**

**Literatura**

[1] Popović J., “Testiranje softvera u praksi”, Računarski fakultet, Beograd, 2012.

[2] Knott D.,: “Hands-on Mobile App Testing”, eBook, Indiana USA, 2015.

[3] <https://developer.android.com/studio/test/index.html> sajt “Android Developers”.

[4] <http://blogs.wittwer.fr/whiler/2011/07/01/editeur-monkeyrunner/> blog Wittwer William