

TESTIRANJE SOFTVERA





TESTIRANJE SOFTVERA

Pojam softver-a prvi put se pominje u radovima John W. Tukey, 1957. godine.

Testiranje softvera se danas vidi kao aktivnost koja obuhvata ceo proces razvoja i održavanja i predstavlja važan deo kompletne konstrukcije softvera.

Kompletno testiranje softvera je nemoguće zbog ograničenja u resursima, budžetu, vremenu i zahtevima kvaliteta softvera.

Uvek postoji neki skriveni "bug"

Podrazumeva detaljna testiranja svih delova programa.

TESTIRATI< TESTIRATI I SAMO TESTIRATI SVE DELOVE
SOFTVERA



Automatsko testiranje softvera

Bitna pitanja kod automatskog testiranja softvera su:

- ❖ Šta automatizovati?
- ❖ Kada automatizovati?
- ❖ Kako automatizovati?



Programmable



Repeatability



Reusable



Cost Reduction



Quality





Automatsko testiranje softvera

ISO/IEC/IEEE 29119-1

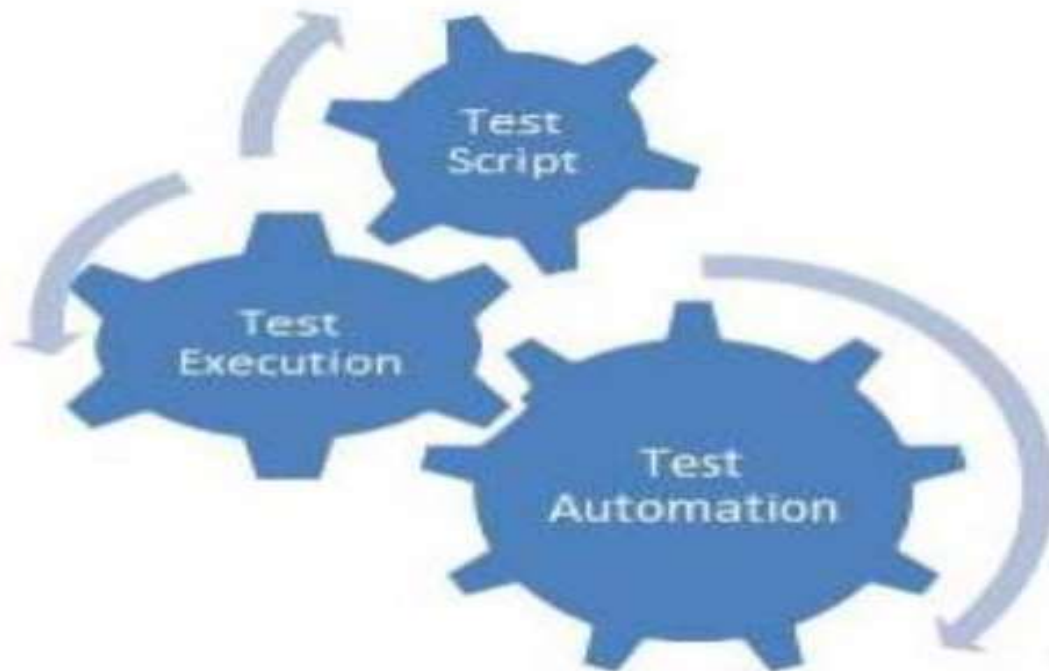
Neke od tema u ovom standardu su uvod u testiranje softvera, testni procesi i potprocesi, prakse u testiranju, **automatsko testiranje**.

Mark Fewster i Dorothy Graham navode kako je automatsko testiranja skuplje od manualnog (ručnog) testiranja. Kao najvažniji razlog su naveli potrebu za pažljivim izborom testova koji se mogu automatizovati.

Automatsko testiranje je proces kontrole i brze provere velikog skupa funkcionalnosti sistema te nam daje izveštaje o razlici izlaznih i ulaznih parametara.



Automatsko testiranje softvera





Automatsko testiranje softvera

Neki od pristupa automatskog testiranja su:

- **Testiranje API-ja** (application programming interface) - sa ulaznim podacima (gde prilikom testiranja softvera vidimo da li dobijamo na neki način očekivane izlazne podatke.)
- **Testiranje GUI-ja** (graphical user interface) - testiramo događaje na grafičkom korisničkom interfacu.
- **Testiranje web aplikacija** - testiramo web aplikacije.

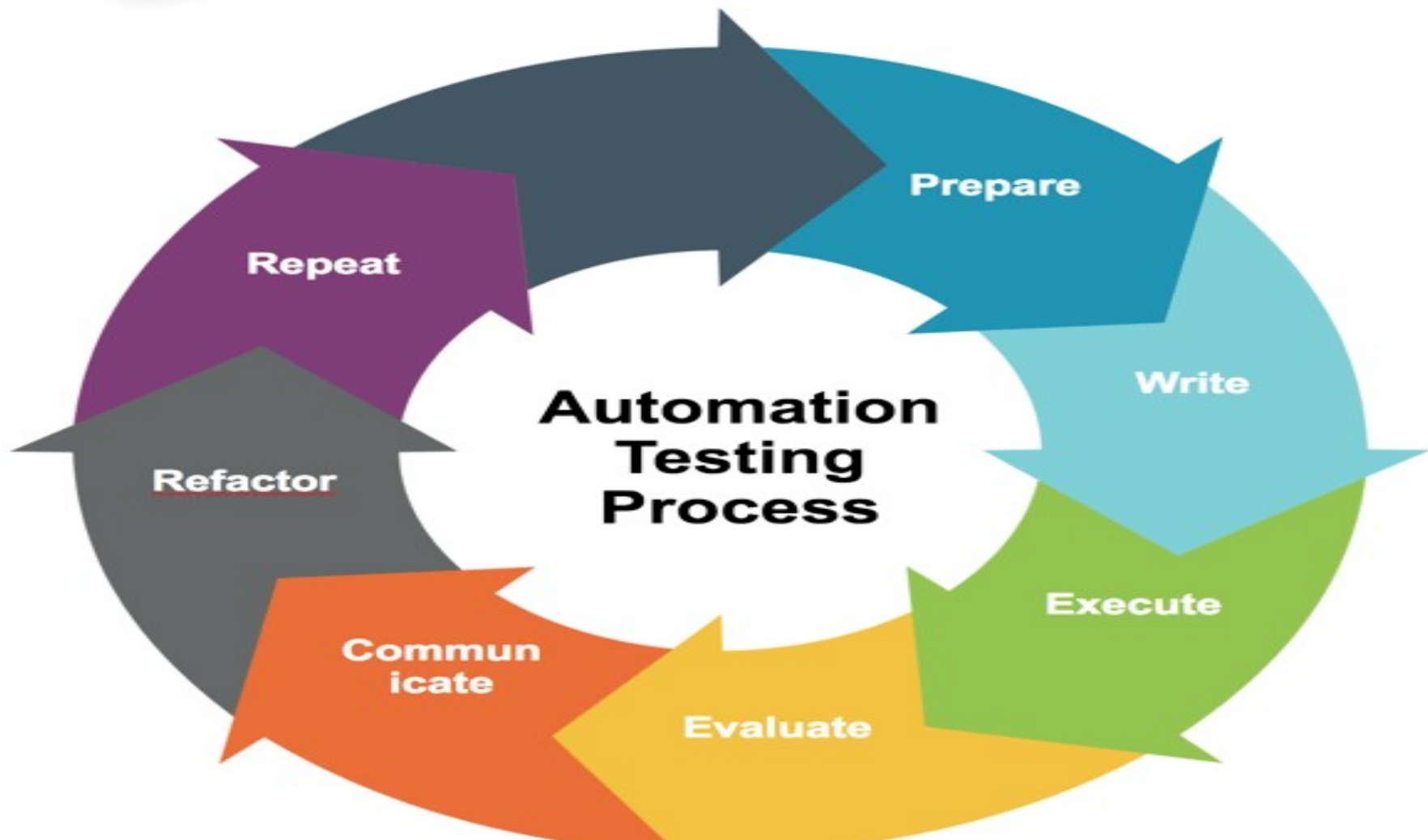
Na primer. Selenium je alat za testiranje web aplikacija, jedan je najzastupljenijih i dostaje jednostavan za korišćenje.

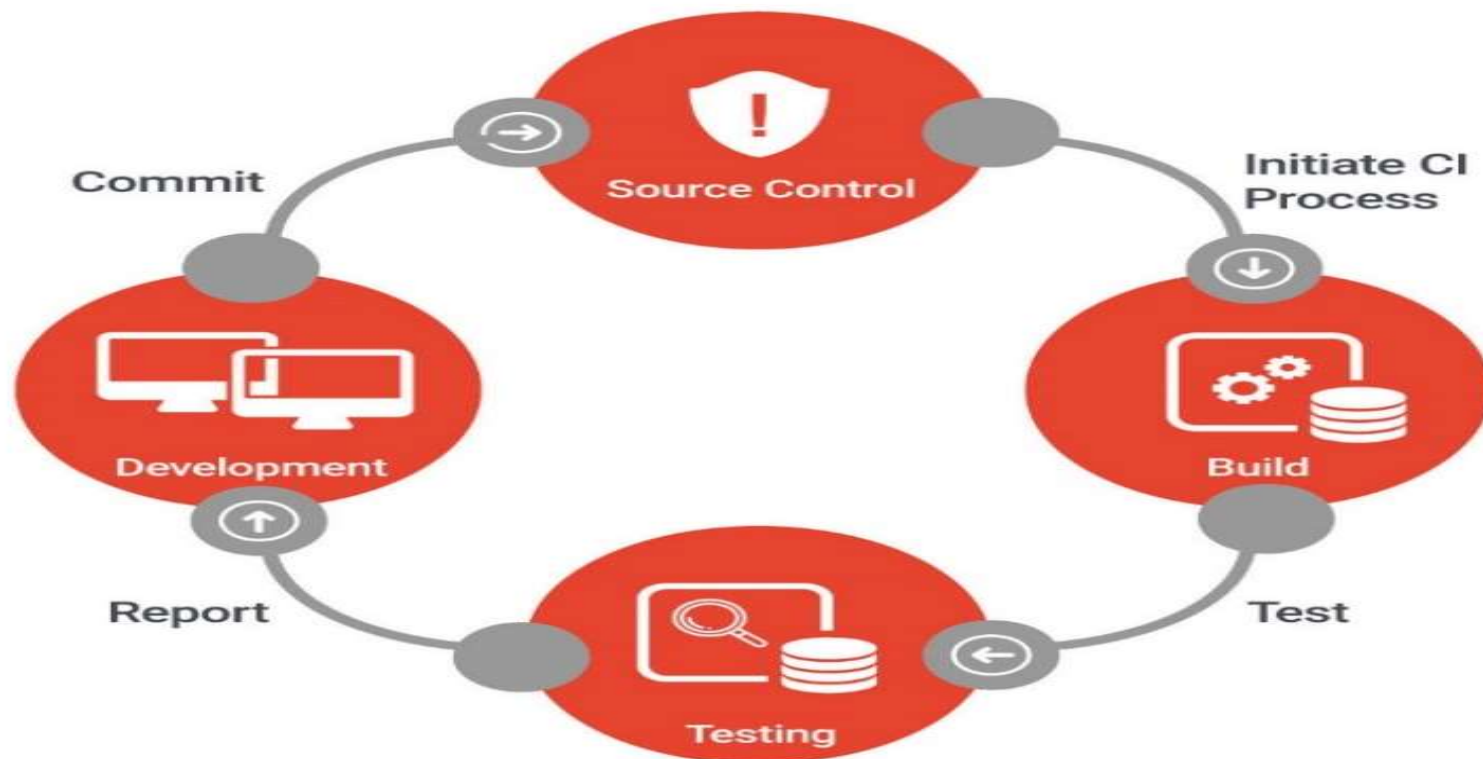


Automatsko testiranje softvera

Kako bi se odredilo koji testovi se mogu automatizovati potrebno je proveriti testove odnosno koji su od napisanih testova najbolji odnosno najpodložniji automatizaciji.

Danas automatsko testiranje softvera je sve češća pojava te skoro sve organizacije imaju bar deo funkcionalnosti pokriven sa automatskim testovima







Automatsko testiranje softvera

Automatsko testiranje softvera je posebno izraženo u radovima (Fernandes, Di Fonzo, 2017) kod takozvanog regresijskog testiranja (jednom pripremljeni regresijski testovi mogu se izvoditi svaki put kad je to potrebno)

Zatim u testiranju osnovnih funkcionalnosti (Smoke Testing) koje omogućuje kvalitetnu brzu procenu softvera na osnovu koje se zatim donosi odluka o detaljnijem testiranju; statičnih testova (mogućnost češćeg izvođenja većeg broja testova, što donosi pouzdanosti sistema);



Automatsko testiranje softvera

Nekoliko razloga da koristimo automatsko testiranje softvera:

- ❑ Ako postoje automatski testovi oni daju mogućnost da se češće pokrenu ti testovi.
- ❑ Automatskim testiranje softvera možemo izvršavati testove koje je teško ili nemoguće realizovati ručno.

Šta ako u sistemu imamo nekoliko hiljada korisnika.
Kako da manuelno simuliramo tu zadatu situaciju ?



Automatsko testiranje softvera

Nekoliko razloga da koristimo automatsko testiranje softvera:

- ❑ Automatskim testiranjem softvera omogućuje se bolja raspodela resursa
- ❑ Testove koji su automatizovani moguće je pokrenuti više puta. (videte da li postoji razlika u ulaznim i izlaznim parametrima)
- ❑ Automatskim testiranje softvera / smanjuje se vreme potrebno za testiranje



Automatsko testiranje softvera

Nekoliko razloga da koristimo automatsko testiranje softvera:

- ☐ Ukoliko je dobro definisan skup automatskih testova, povećava se sigurnost da u proizvodu neće biti "neugodnih iznenađenja nakon isporuke."
- ☐ Može da se pokrije i više test scenarija, čak i onih manje verovatnih koje mogu da se dese.
- ☐ Moguće je ponovno korišćenje automatskih testova.



Automatsko testiranje softvera

Jedina strategija koja nam garantuje dugoročno održivi tempo razvoja je nezavisnost od manualnih testera i istovremeni razvoj automatskih testova uporedo sa razvojem novih funkcionalnosti.

Kratkoročni cilj:

Brzina razvoja će se očigledno smanjiti jer osim aplikacijskog moramo napisati i testni kod, ali će zato kvalitet koda biti dugoročno obezbeđen.



Automatsko testiranje softvera

Test "automatizacije" je proces kada testerski tim piše skripte i koristi drugi softver za testiranje softvera.

Automatsko testiranje se koristi za ponovno pokretanje test scenarija koji se izvodi ručno, brzo i više puta.

Automatsko testiranje se koristi i za testiranje aplikacije sa stanovišta opterećenja, performansi i stresa.

Povećava se pokrivenost testom, poboljšava tačnost, štedi vreme i budžet u poređenju sa manuelnim testiranjem.



Automatsko testiranje softvera

Automatsko testiranje predstavlja tehniku testiranja softvera kojom se upoređuje stvarni ishod (rezultat) sa očekivanim ishodom (rezultatom) pisanjem automatskih testova.

Automatsko testiranje softvera se koristi sa test skriptima ili korišćenjem nekog alata za automatizaciju testova.

Ovaj način testiranja se koristi kako bi se automatizovali testovi koji se često ponavljaju pa i oni testovi koje je teško izvršiti ručno.



Automatsko testiranje softvera

- **Procedura koja se prati prilikom automatskog testiranja:**
 - Donesite odluku
 - Pribavite alate
 - Uvedite testiranje automatizacije
 - Planirajte dizajn i razvijajte skripte
 - Izvršite testove
 - Procenite rezultate testa
 - Sve dokumentovati



Automatsko testiranje softvera

Automatsko testiranje treba koristiti kada su :

Veliki i kritični projekti, kada postoje česti (promenljivi) zahtevi koje naručilac softvera želi da se implementira , pristup aplikaciji za opterećenje i preformanse kod mnogih virtuelnih korisnika.

Faktor vremena.

Planovi **za automatsko testiranje** se razvijaju zajedno sa redovnim razvojem specifikacija softverskih proizvoda, a zatim se automatski izvršavaju pomoću alata za kontinuiranu integraciju softvera.



Automatsko testiranje softvera

- ☐ Pisanje koda zahteva stručnost testera i poznavanje nekog programskog jezika i određenih alata.
- ☐ **Složeni testovi i funkcionalnosti** mogu biti jako teški za automatizaciju.
- ☐ Napomena: Programi i alati koji se koriste vrlo često mogu biti skupi.
- ☐ Potrebno je održavati automatske testove da bi imali validne rezultate, što će reći potrebno je i dodatno vreme za to.

Automatsko testiranje softvera

Neke od prednosti automatskog testiranja.

Automation Testing



Fast



Reliable



Reusable



Improves Accuracy



Saves time and money



Reduces Human-generated error



Supports the execution of repeated test cases





Automatsko testiranje softvera

Povećana preciznost izvođenja testova:

Jedna od glavnih prednosti automatskog testiranja jeste **povećanje preciznosti izvođenja testova**.

Ljudi greške prilikom izvršavanja manualnih testova mogu biti problematične.

- ❑ Za razliku od izvršavanja manualnih testova, kod izvođenja automatskih testova manja je verovatnoća da će ljudska greška uticati na automatizovano testiranje.
- ❑ Kada su testovi automatizovani, oni se pokreću češće i sa većom preciznošću izvođenja nego kada se testovi pokreću ručno.
- ❑ Automatizacija testova je korisna kada se radi sa velikom aplikacijom koja ima dosta funkcionalnosti ili kada se dodaju nove funkcije.
- ❑ Pored toga, pomaže da se sve greške ili defekti u kodu identifikuju i poprave što je pre moguće.



Automatsko testiranje softvera

Brže izvršenje

- ☐ Automatsko testiranje takođe omogućava brže izvršavanje testova.
- ☐ To je zato što se testovi izvode istovremeno, paralelno umesto serijski.
- ☐ Paralelno pokretanje testova znači da se više testova pokreće istovremeno za kraće vreme.

Pouzdan rezultati testiranja

- ☐ Još jedna prednost automatskog testiranja je ta što obezbeđuje pouzdane rezultate testiranja.
- ☐ Ovo dolazi kao rezultat toga jer se testovi pokreću često i automatski.
- ☐ Automatsko testiranje softvera pomaže u brzom identifikaciji grešaka i nedostataka, što olakšava timu da rešavanje problema onda kada se problemi otkriju.



Automatsko testiranje softvera

Smanjeni troškovi

- ☐ Automatsko testiranje takođe može dovesti do smanjenja troškova.
- ☐ Kada su testovi automatizovani, smanjena je potreba za manuelnim testiranjem.
- ☐ Pored toga, vreme potrebno za izvođenje testova je smanjeno, što dovodi do uštede u vremenu i novcu.
- ☐ Štaviše, automatski testovi mogu pomoći u smanjenju troškova razvoja softvera otkrivanjem i ispravljanjem grešaka ranijim fazama razvoja softvera.
- ☐ Oni takođe mogu pomoći u smanjenju troškova podrške aplikacije, jer će automatizovanim testovima biti potrebno manje vremena za pronalazak grešaka.



Automatsko testiranje softvera

Povećana efikasnost

Automatsko testiranje može pomoći u poboljšanju produktivnosti test inženjera automatizacijom testova koji bi inače morali da se izvršavaju manuelno.

Povećana saradnja između programera

Automatsko testiranje može pomoći da se poboljša saradnja između programera. Kada postoje napisani testovi za neku aplikaciju, programeri mogu da se oslone na testove prilikom implementacije novih promena ili funkcija. Ovim se obezbeđuje visok nivo pokrivenosti koda i smanjuje verovatnoća grešaka u novom kodu.

Poboljšana skalabilnost

Automatski testovi se mogu koristiti na mnogim uređajima i konfiguracijama, što olakšava testiranje više stvari istovremeno. Na primer, automatski testovi se mogu pisati za merenje performansi aplikacije na različitim uređajima ili browserima. Ovo omogućava lakše testiranje i proveru rada aplikacije na različitim uređajima.



Automatsko testiranje softvera - MANE

- ✓ **Nerealna očekivanja**

Menadžment, ali i članovi razvojnog tima često mogu imati prevelika očekivanja što se tiče automatskog testiranja.

- ✓ **Loša praksa testiranja**

Automatiziranim i kvalitetnim testovima neće se videti efikasnost i kvalitet samog softvera

- ✓ **Automatizovani razvoj testova je obično skuplji proces** od manualnog testiranja (dobro planiranje budžet je neophodno)

- ✓ **Očekivanja da će se pronaći velik broj novih grešaka**

- ✓ **Tehnički problem**

Budući da su promene u softveru česte, mogući su slučajevi kada alati za automatsko testiranje ne mogu pokriti sve te izmene.

- ✓ **Neki problemi u organizacijskog strukturi**

- ✓ **Automatizacija će zahtevati dodatno obučeno osoblje**



ALATI ZA AUTOMATSKO TESTIRANJE SOFTVERA

Softverski alati za automatsko testiranje softvera se mogu podeliti u sledeće tri kategorije:

- ☐ Open Source alat
- ☐ Komercijalni testni alati
- ☐ Prilagođeni alati

Testiranje softvera

cypress



ZAPTEST

Unlimited Software Automation

BlazeMeter



APACHE

JMeterTM



Selenium



Katalon Studio



Unified
Functional
Testing



TestComplete



watir



Ranorex[®]

An Idera, Inc. Company



Selenium

- Prva verzija Selenium-a **nastala je 2004. godine** kada je Džejson Hagins (Jason Huggins), za potrebe testiranja aplikacije u firmi ThoughtWorks. Uspeo je da napiše Javascript biblioteku koju je integrisao sa Web stranama, što mu je omogućilo automatsko izvršavanje testova u različitim browserima.



Selenium

- Kada je Jason Huggins, uvideo potencijal ove ideje da pomogne automatizaciji drugih veb aplikacija, razvijao je kreiranu skriptu i kreirao Selenium Core.
- Prilikom jednog problema u timu sa kompanijom Mercury Interactive, Jason je došao na ideju da ovaj alat dobije naziv Selenium.



Selenium

Selenium je najzastupljenije open-source rešenje automatskog testiranja softvera, koje omogućava upravljanje web browserima, kontrolisanje njegovog ponašanja i pomoću kojeg je moguće inicirati određene akcije nad web aplikacijom. Namenjen je pisanju TestCase, koji su jako bitni u Agilnoj metodologiji rada, (na kojoj se baziraju ili koriste vodeće kompanije u oblasti sofvera)



Selenium

Selenium je skup alata za automatizovano testiranje korisnickog interfejsa Web aplikacija.

Može da se koristi za više programskih jezika (Java, C#, Python, Ruby, Perl ...) za kreiranje Selenium test skripti. Može se koristiti na Windows, Macintosh ili Linux platformama.

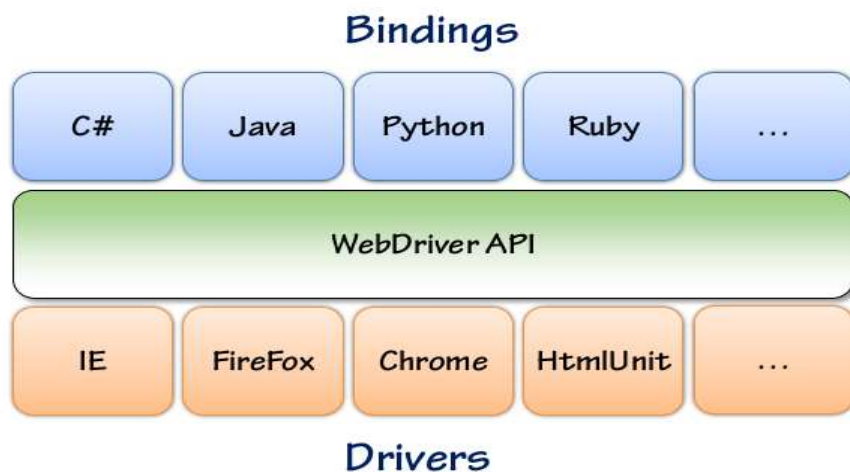
- U pitanju je softver otvorenog koda
- Lako se integriše sa drugim framework platformama i alatima
- Podržava sve zastupljenije operativne sisteme
- Podržava sve bitnije internet pretraživače
- Omogućuje pisanje testova na velikom broju programskih jezika



Selenium

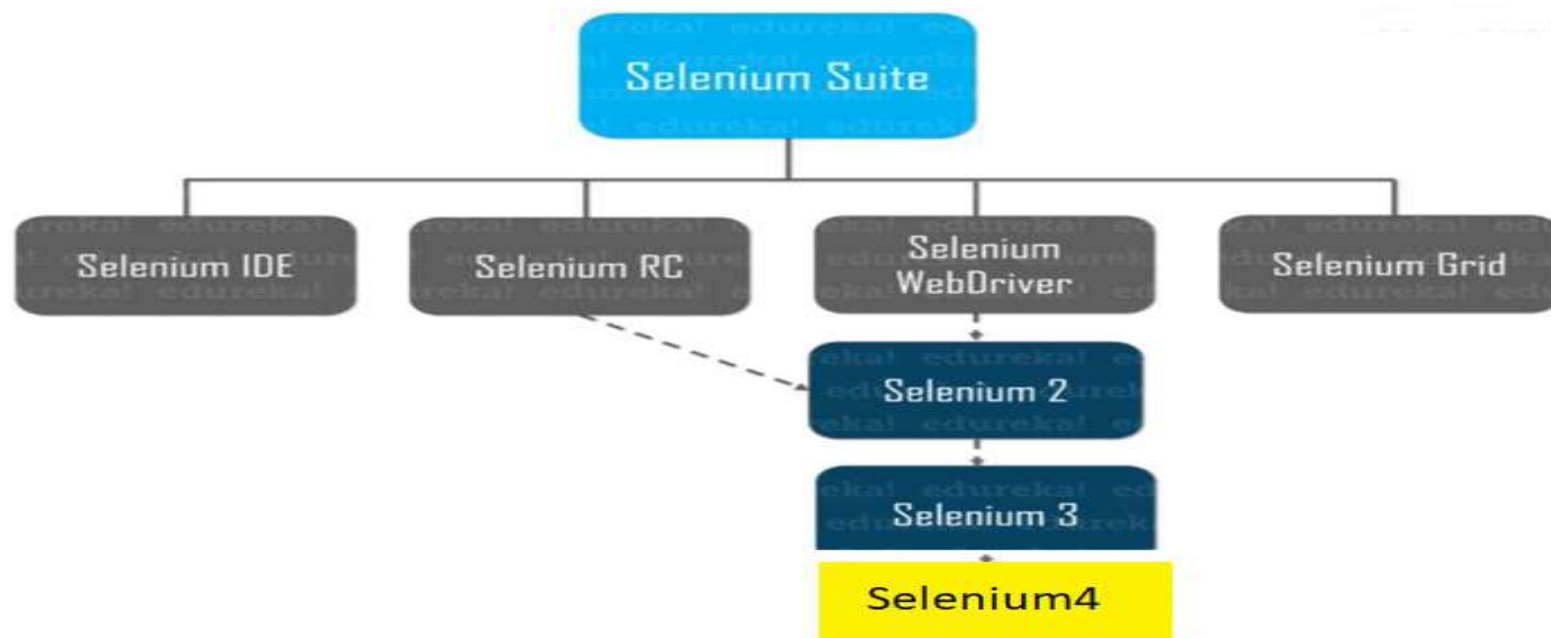
Softver se sastoji iz četiri nezavisne komponente: Selenium IDE, Selenium Remote Control (RC), Selenium Webdriver i Selenium-Grid.

Selenium WebDriver arhitektura





Komponente alata Selenium





Selenium

- ✓ **Selenium IDE** je dodatak za Firefox koji se može koristiti samo u stvaranju relativno jednostavnih test slučajeva i programskih paketa.
 - ✓ **Selenium RemoteControl** je takođe poznat kao Selenium 1 jer predstavlja prvi alat Seleniuma koji je korisnicima omogućio da koriste programske jezike u kreiranju složenih testova.
 - ✓ **WebDriver**, predstavlja najnoviji paker koji omogućava test skriptama da komuniciraju direktno sa browserima.
 - ✓ **Selenium Grid** je alat koji koristi sa Selenium RC za izvršavanje paralelnih testova u različitim browserima i operativnim sistemima.
- ❑ **Selenium RC i WebDriver** su objedinjeni da bi se formirao Selenium 2.



Selenium

Kod pisanja automatskih testova najbitniji aspekt je interakcija sa HTML elementima veb stranice.

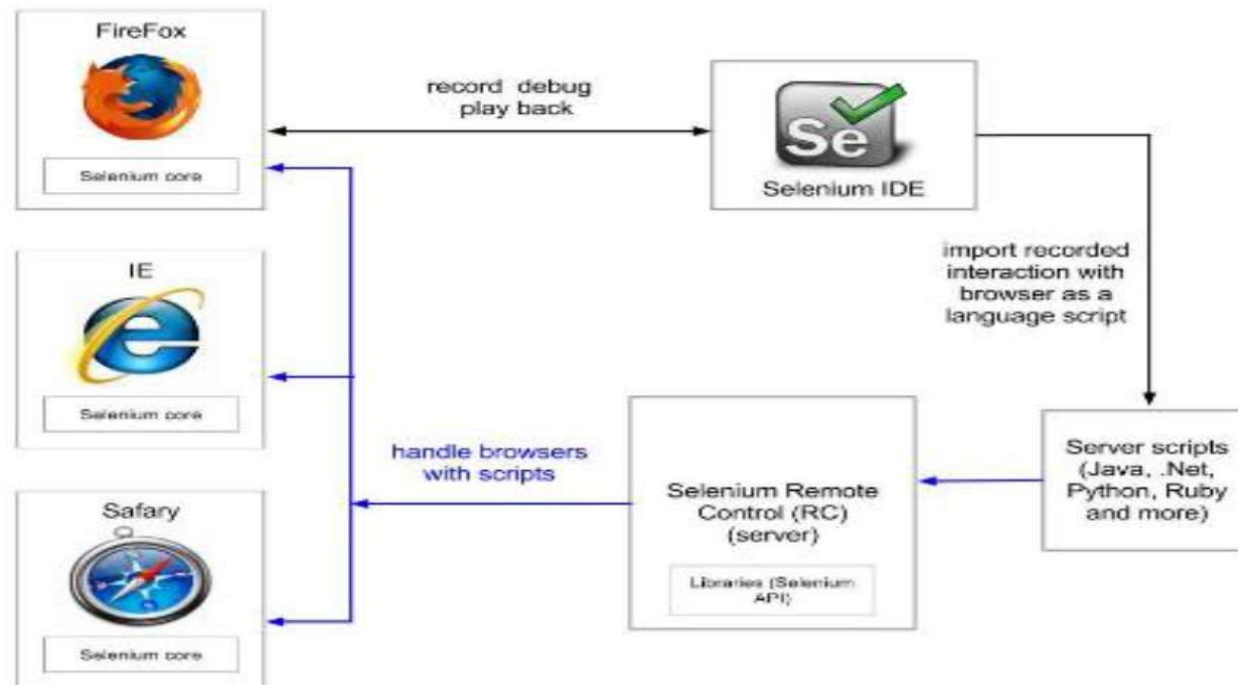
Da bi Selenium test mogao da funkcioniše potrebno je identifikovati elemente HTML stranice sa kojima test treba da komunicira.

Test skripte su niz komandi koje se snimaju automatski ili se pak mogu i ručno unositi.

Rezultate testiranja možemo proveriti u "Logovima". U slučaju da test skripta nije uspešno izvršena tu ćemo moći detaljnije da vidimo i gde se javio problem i da izvršimo detaljno analiziranje.

<https://selenium.dev/>

Selenium



Lociranje elemenata HTML stranice

- Putem
 - imena taga
 - teksta u linku
 - XPath upita
 - vrednosti atributa id
 - vrednosti atributa name
 - dela teksta u linku
 - imena CSS klase
 - CSS selektora

ClassName

CssSelector

Id

LinkText

Name

PartialLinkText

TagName

XPath



Apache JMeter

Apache JMeter je softver otvorenog koda. Java aplikacija dizajnirana za učitavanje test funkcionalnog ponašanja i merenje performansi.

U prvo vreme je bila implementirana za testiranje web aplikacija, ali se vremenom razvijala i na druge funkcije testiranja softvera.

Dobre karakteristike su na primer:

- ✓ Licenca otvorenog koda
- ✓ Odličan GUI
- ✓ Može da se koristi na više različitih operativnih Sistema
- ✓ Odlična simulacija
- ✓ Podržava više protokola kao što su FTP, HTTP, JDBC
- ✓ Vizuelizacija testa

<http://jmeter.apache.org/>



Cypress

- Cypress predstavlja jedan od mnogobrojnih softvera za testiranje. Tačnije, on je framework koji izvršava testove nad nekom aplikacijom u isto vreme komunicirajući i sa *front-end* i *back-end* delom softvera.
- Cypress predstavlja framework koji se sastoji od **Node.js** koji radi u pozadini i izvršava serverske procese i **JavaScript** koja radi sa svim elementima aplikacije kojima pristupa iz veb browsera.



Cypress

- Većina alata za testiranje deluje tako što se pokreće izvan veb browser i izvršava daljinske komande širom mreže, Cypress je upravo suprotno od toga. Cypress testove izvršava u istoj petlji kao i aplikacija koja se testira. U pozadini Cypress-a je Node.js serverski proces, oni neprestano međusobno komuniciraju, sinhronizuju se i izvršavaju zadatke uporedo jedan sa drugim.
- Pristup u oba dela softvera (*front and back*) pruža nam mogućnost da testiramo i odgovaramo na događaje aplikacije u realnom vremenu dok istovremeno radimo van veb browser na zadacima koji zahtevaju više pažnje.



Ranorex

- Ranorex se smatra alatom za automatsko testiranje softvera i on koristi Selenium WebDriver.

Koristi se kod web, desktop i mobilnih aplikacija.

Dosta je jednostavan za korištenje UI (*User interface*)

Osim tehničkih plus bodova, Ranorex nudi dobru korisničku podršku putem online foruma i putem e-pošte

<https://www.ranorex.com/>



Watir

- To je alat za automatsko testiranje softvera - softver otvorenog koda.
- Ruby
- Automatsko testiranje softvera web aplikacija u Rubyju.
- WatirCraft framework koji je razvijen kako bi se olakšalo i ubrzalo pisanje Watir testova
- Watir podržava vašu web aplikaciju bez obzira koju ste tehnologiju koristili za vašu aplikaciju.
- Neke kompanije koje koriste ovaj alatu su SAP, Oracle, Facebook
- <http://watir.com/>



Lambda test

- Pomoću LambdaTest-a može se izvesti i ručno i automatsko testiranje softvera.
- Neke od dobrih osobina LambdaTest-a:
- LambdaTest daje mogućnost da izvedete interaktivno i automatizovano testiranje unakrsnih browsera (na velikom broju 2000)
- ✓ Podržava paralelno izvršavanje
- ✓ Cloud alat za testiranje unakrsnih browsera i za sve aktivnosti daje akcenat na sam kvalitetet softvera
- ✓ Neograničena veb automatizacija sa Selenom
- ✓ Testiranje geolokacije
- ✓ Integrisano otklanjanje grešaka
- <https://www.lambdatest.com/>



TestComplete

- To je sofverski alat za automatsko testiranje. Vlanik je softvera kompanija SmartBear.
- TestComplete nudi sveobuhvatno testiranje mobilnih, i web aplikacija.
- JavaScript, Python, VBScript i C ++
- Može obaviti testiranje ključnih reči i podataka.
- <https://smartbear.com/product/testcomplete/overview/>



Subject7

Subject7 je komercijalna platforma koja je na Cloudu.

Koristi otvorene izvore kao što su Selenium i Appium
Jednostavan za upotrebu za ne-tehničke ljude

<https://www.subject-7.com/>



ZAPTEST

- Besplatni i poslovni alat za automatizaciju bilo koje Windows aplikacije.
- Testove možete izvoditi brzo i jednostavno bez dugog čekanja na izvođenje.
- Dosta je jednostavan za korišćenje UI
- (*User interface*)
- <https://www.zaptest.com>



BlazeMeter

To je platforma za testiranje opterećenja za simulaciju bilo kog korisničkog scenarija za web aplikacije, web stranice, mobilne aplikacije ili web usluge

Dobar je za testiranje testova opterećenja i performansi.

Kompatibilan je sa alatom JMeter. Bilo koji JMeter test dobro funkcioniše i na BlazeMeteru.

To je licencirani alat.

Na raspolaganju je i njegovo free probno testiranje koje omogućuje 50 istovremenih korisnika, i 10 besplatnih testova.

<https://www.blazemeter.com/>



AUTOMATSKO ILI MANUELNO TESTIRANJE ?

Integracija automatskog i manualnog testiranja je najbolja opcija (ako je to moguće uraditi)

Često se ne pravi poređenje između konačne specifikacije i specifikacija dizajna. Ovaj problem potiče od ljudi koji postavljaju zahteve.

- ❑ Ispravnost softvera je moguće proveravati i bez njegovog izvršavanja, samo na osnovu analize izvornog koda, korišćenjem tehnika statičke analize.
- ❑ Prednost automatskog testiranja je u povećanoj preciznost, jer automatski test će dati precizan rezultat bez obzira na broj ponavljanja testa, dok kod manualnog testiranja postoji faktor zamora i ljudske greške.
- ❑ Statička analiza koda može biti manualna, i podrazumeva ručne provere i preglede koda, ili može biti automatizovana.
- ❑ Automatizacija procesa generisanja test primera i provera rezultata testiranja posebno je važna jer olakšava i ubrzava proces testiranja.



AUTOMATSKO ILI MANUELNO TESTIRANJE ?

▪ **Manuelno testiranje:**

- Proces nalaženja defekata unutar samog softvera
- Tester ponaša kao krajnji korisnik i manuelno izvršava test slučajeve bez korišćenja automatizovanih alata
- Rezultat izvršavanja koji nije očekivan je zapisan i analiziran u formi test izveštaja

▪ **Automatsko testiranje:**

- Proces u kojem se izvršavaju pripremljeni automatski test scenariji kako bi se pronašli defekti
- Cilj: smanjenje ukupne cene manualnog testiranja, sprečavanje ljudske greške



AUTOMATSKO ILI MANUELNO TESTIRANJE ?

- Svaki automatski test se sastoji iz nekoliko procesa:
- Priprema testova,
- Pisanje testova,
- Izvršavanje testova,
- Procena,
- Komunikacija sa developerima,
- Refaktorisanje testova,
- Ponavljanje testiranja.



AUTOMATSKO ILI MANUELNO TESTIRANJE ?

Oba pristupa imaju svoje prednosti i mane.

Testiranje je neizežan proces ukoliko želimo implementirati uspešan i kvalitetan softver.

Softver možemo testirati ručno (manuelno) ili automatski pomoću specijalnih alata za testiranje.

Ako se može napraviti kombinacije automatskog i ručnog (manuelnog) testiranja ona može biti uspešna za neki softver, dok za neki drugi ne mora.



AUTOMATSKO ILI MANUELNO TESTIRANJE ?

- ☐ Kod manualnog testiranja softvera tester ručno obavlja sve potrebne korake, dok kod automatskog testiranja, testovi se pokreću automatski i izvršavaju od strane programa.
- ☐ Program za testiranje piše osoba koja se bavi automatskim testiranjem.
- ☐ Programi za testiranje se pišu u određenim alatima,
- ☐ Na taj način se pronalaze greške u softveru koje će biti naknadno ispravljene od strane programerskog tima.



AUTOMATSKO ILI MANUELNO TESTIRANJE?

Automatizacija testova može da se izvrši imitirajući prave akcije klijenta tokom testiranja razvijene aplikacije ili kroz testiranje ne grafičkih nivoa aplikacije (API funkcionalno testiranje).

Otežavajući faktor za automatizaciju je činjenica da se pojedine funkcionalnosti menjaju često i samim tim nisu pogodne za automatizaciju, a održavanje takvih test scenarija bi oduzimalo previše vremena i resursa.

