

Jenkins ukratko

Uroš Milenković, Nikola Sojčić, Bojan Nestorović

Recenzija: Goran Vinterhalter

24. april 2016

1 O čemu rad govori?

Rad opisuje osnovne mogućnosti softverskog paketa Jenkins koji automatizuje procese: prevođenja, testiranja, arhiviranja i deployment-a fokusirajući se na sekvencionalni metod izvršavanja.

2 Krupne primedbe i sugestije

Rad potpuno ignoriše "*eng. pipeline*" pristup i fokusira se isključivo na slobodne projekte koji dozvoljavaju samo sekvencijalno izvršavanje zadataka. Tvđenje da slobodni projekat pruža maksimalnu fleksibilnost s'toga nije tačna.

Opis opcije parametrizovane izgradnje (*eng. parametrized build*) treba dopuniti konkretnijim primerima. Neupućenom čitaocu je teško razumeti kakve kakve su to opcije koje se prosleđuju i koji format bi bio pogodniji za koju situaciju.

3 Sitne primedbe

Rad se u nekim trenucima čini kao kratak tutorijal za početnike umesto uopštenog pregleda mogućnosti. Navođenje detalja kao što su opisi instalacije proširenja (git ...) ili specifičnosti kreiranja novog projekta nisu toliko bitne za samu temu.

Trebalo bi izbegavati rečenice kao što su:

- "pokušaćemo da zagrebemo površinu ovog složenog sistema", jer rad čini manje vrednim.

Postoji nekoliko sitnih grešaka pri kucanju.

4 Provera sadržajnosti i forme seminarskog rada

1. Da li rad dobro odgovara na zadatu temu?
Da, rad je za temu imao da opiše mogućnosti i prednosti softvera Jenkins što je i opisano.

2. Da li je nešto važno propušteno?
Pipeline (ili workflow) je postao standard za upravljanje kompleksnijim nelinearnim izvršavanjem. U ovom radu pipeline je potpuno zanemaren. Takođe postoji specijalna terminologija koja se koristi za definisanje jednog pipeline konfiguracionog fajla koju bi trebalo navesti i predstaviti jedan šematski prikaz.
3. Da li ima suštinskih grešaka i propusta?
Nema.
4. Da li je naslov rada dobro izabran?
Da.
5. Da li sažetak sadrži prave podatke o radu?
Da.
6. Da li je rad lak-težak za čitanje?
Rad je pitak i lepo ilustrovan uz par primedbi navedenih gore.
7. Da li je za razumevanje teksta potrebno predznanje i u kolikoj meri?
Jeste.
Neophodno je razumevanje oblasti:
 - (a) Životni ciklus razvoja softvera.
 - (b) Version sistemi, prvenstveno git.
 - (c) Unit testovi.
8. Da li je u radu navedena odgovarajuća literatura?
Da.
9. Da li su u radu reference korektno navedene?
U radu postoji samo jedna referenca. Potrebno je da svako veće tvrđenje bude podržano sa odgovarajućom referencom. Recimo navodi se da je najznačajnija stvar za jenkins integracija sa sistemom za kontrolu verzija. Zašto to tvrđenje nema referencu a tvrđenje za Unit testove ima?
10. Da li je struktura rada adekvatna?
Struktura je adekvatna, ali fale reference.
11. Da li rad sadrži sve elemente propisane uslovom seminarskog rada (slike, tabele, broj strana...)?
Da.
12. Da li su slike i tabele funkcionalne i adekvatne?
Da, jedino bi valjalo da se pojavljuju malo ranije u tekstu radi lakšeg praćenja. Poslednju tabelu bi trebalo Transponovati. Praksa je da se tabele koje predstavljaju razlike između 2 i više softvera prave tako da jedna opcija odgovara jednom redu.

5 Ocenite sebe

d) malo upućeni

Tokom obrazovanja nisam imao kontakt sa rešenjima kao što su Jenkins. Međutim zbog kurseva kao što su: Razvoj softvera 1, 2 i Microsoft kurseva imam dobru ideju o potrebama razvojnih timova.

6 Poverljivi komentari