

MOGUĆNOSTI UNAPREĐENJA PROCESA NASTAVE UPOTREBOM SOFTVERSKOG PAKETA GITHUB EDUCATION

Miroslav Stefanović¹, Aleksandra Kolak², Ivana Spasojević³, Danilo Nikolić⁴

^{1,2,3,4}Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Novi Sad, Srbija

¹mstef@uns.ac.rs, ²kolaka@uns.ac.rs, ³ivana.spasojevic@uns.ac.rs, ⁴nikolic.danilo@uns.ac.rs

Kratak sadržaj: Izvođenje nastave na različitim studijskim programima, za osposobljavanja kadrova za potrebe IT industrije, podrazumeva izvođenje računarskih vežbi na kojima studenti stiču potrebna znanja vezana za osnovne, a potom i napredne koncepte programiranja. Čest problem u toku izvođenja nastave predstavlja neophodnost razmene programskog kôda između studenata i izvođača nastave. U ovom radu predstavljena je mogućnost upotrebe sistema za upravljanje verzijama, konkretno softverskog paketa GitHub Education za rešenje ovog problema i unapređenje procesa izvođenja nastave.

Ključne reči: Unapređenje/Nastava/Git/GitHub Education

POSSIBILITIES FOR IMPROVEMENT OF EDUCATION PROCESS BY USING GITHUB EDUCATION SOFTWARE PACKAGE

Abstract: Education process in different study programs, that are preparing students for jobs in IT sector, are conducted through computer exercises where students are starting with courses that are teaching them about basic concepts of software development, that are then followed by courses that include some more advanced concepts of software development. Common problem in these courses is the need for students and teachers to exchange programming code. In this paper, possibilities of using version control systems, especially GitHub Education, is presented as possible solution for this problem and improvement of education process.

Key Words: Improvment/Education/GitHub Education

1. UVOD

Obrazovanje kadrova za potrebe IT industrije odvija se u različitim institucijama i po različitim studijskim programima. Ono što je zajedničko svim tim studijskim programima je da se na velikom broju predmeta deo nastave izvodi u obliku računarskih vežbi. Nastava koja se izvodi u okviru računarski vežbi najčešće podrazumeva upoznavanje studenata sa osnovnim konceptima programiranja, savladavanje različitih programskih jezika, te upoznavanje sa nekim od naprednih koncepata koji će im biti od značaja u profesionalnom radu. Sam proces izvođenja nastave je takav da je neophodno da izvođači nastave i studenti redovno razmenjuju programske kôdove, odnosno, nakon završetka računarskih vežbi, izvođač nastave programski kôd koji je raden na vežbama prosleđuje studentima, a najčešće, nakon odslušanih računarskih vežbi, studenti kôd svojih predmetnih projekata prosleđuju izvođaču nastave na ocenjivanje.

Razmena kôda može se izvršiti na različite načine, od slanja kôda korišćenjem elektronske pošte, što u određenim slučajevima nije moguće iz bezbednosnih razloga, odnosno ograničenja koja postavljaju pružaoci usluge elektronske pošte, do postavljanja materijala na nekim od sajtova za deljenje sadržaja. Pored već pomenutog nedostatak, glavni nedostatak ovakve razmene programskog kôda je što je jako teško pratiti promene koje se dešavaju na pojedinačnim terminima izvođenja nastave. Ukoliko se nastava i promene kôda izvode nad istim datotekama, student nije u mogućnosti da na jednostavan način prati tok izvođenja nastave.

Značajno unapređenje procesa izvođenja nastave donosi primena sistema za upravljanje verzijama (eng. *Version Control System*). Na ovaj način rešavaju se i pominjani problemi ograničenja zbog bezbednosti u slučaju elektronske pošte i praćenja toka izvođenja nastave. Jedan od najčešćih sistema za upravljanje verzijama koji se koristi u procesu nastave je *Git*, odnosno platforma *GitHub* koji predstavlja vodeću platformu za razvoj softvera.

Nedostatak ovog rešenja bio je što su se u procesu nastave i izvođači nastave i studenti koristili svojim privatnim nalozima i što su delili svoje, do skora isključivo javne repozitorijume i što rad u okviru jednog predmeta nije bio integrisan između izvođača nastave i studenata. Tokom 2016. godine kompanije *GitHub* kreirala je softverski paket *GitHub Education* koji je nastao sa namerom da reši pominjane nedostatke i da bude alat koji će olakšati izvođenje nastave. U ovom radu biće dat kratak pregled osnovnih funkcionalnosti *Git Education* paketa i opis prednosti koje one donose u procesu izvođenja nastave.

Rad je strukturiran tako da je nakon uvoda, u drugom poglavlju dat kratak opis sistema za upravljanje verzijama, u trećem poglavlju predstavljeni su *Git*-a i *GitHub*-a, kao i njihove osnovne funkcionalnosti. U četvrtom poglavlju opisane su osnovne karakteristike *GitHub Education*, dok se zaključak nalazi u petom poglavlju.

2. SISTEMI ZA UPRAVLJANJE VERZIJAMA

Osnovni zadatak sistema za upravljanje verzijama je da softverski sistem koji ima različite verzije održava organizovanim. Iako su ovi sistemi inicijalno namenjeni za rad sa softverom, mogu se koristiti i za organizovanje dokumenata, specifikacija, članaka ili drugih elektronskih datoteka [1]. Namena sistema za upravljanje verzijama je da čuva prethodne verzije programskog kôda i da omogući da se prelazak između različitih verzija [2]. Sistemi za upravljanje verzijama najčešće se klasifikuju na centralizovane i decentralizovane sisteme.

Centralizovani sistemi se prepoznaju po tome što imaju jedan centralni repozitorijum iz kog se kôd preuzima (eng. *checkout*) i praktično predstavljaju „snimak“ repozitorijuma u datom trenutku. Pravo upisa u repozitorijum je obično ograničeno, a u slučaju *open source* projekata, potrebno ga je „zaslužiti“ što se postiže redovnim slanjem kvalitetnog kôda. Najpoznatiji predstavnici centralizovanog sistema za upravljanje verzijama su CSV¹⁹ (eng. *Concurrent Versions Systems*) i *Subversion*²⁰.

U slučaju decentralizovanih sistema za upravljanje verzijama, ne postoji jedan centralni repozitorijum, odnosno, svakim preuzimanjem iz određenog repozitorijuma preuzima se i kompletan istorijat promena, odnosno sve prethodne verzije. Takođe, svako ko preuzme određeni repozitorijum može slobodno da nastavi rad na njemu, bez potrebe da dobije posebna prava pisanja u repozitorijumu [3]. Najpoznatiji predstavnici decentralizovanih sistema za upravljanje verzijama su *Git*²¹, *Mercurial*²², *Bazaar*²³ i *Darcs*²⁴.

Nepostojanje jedne tačke otkaza (eng. *single point of failure*), brži rad, jer nema potrebe da se stalno komunicira sa serverom, mogućnost da se radi u lokalnom repozitorijumu i manji broj konflikata prilikom spajanja (eng. *merge conflicts*) su osnovni razlozi zašto su danas primarno u upotrebi decentralizovani sistemi za upravljanje verzijama.

3. GIT I GITHUB

Git je distribuirani sistem za upravljanje verzijama koje je 2005. godine razvio Linus Torvalds. Linus Torvalds je poznat kao tvorac *Linux* operativnog sistema. Nastanak *Git*-a tesno je vezan sa razvojem *Linux*-a. Naime od 1991. godine, kada je objavljena prva verzija *Linux*-a, pa sve do 2002. godine, Linus Torvalds nije koristio ni jedan sistem za upravljanje verzijama. Developeri koji su radili na ovom projektu svoj kôd postavljali su prvo na usenet grupe, a potom su ih razmenjivali preko mailing liste, da bi potom sam Linus Torvalds taj kôd integrisao sa već postojećim kôdom. Nepostojanje sistema za upravljanje verzijama predstavljalo je značajan problem za ovako veliki projekat, te se 2002. godine Linus Torvalds opredelio za korišćenje *BitKeeper* sistema za upravljanje verzijama. Za razliku od *Linux* operativnog sistema koji je besplatan i otvorenog kôda (eng. *open source*), *BitKeeper* to nije bio, što je određenom delu ljudi uključenih u *Linux* projekat predstavljalo problem. *BitKeeper* koji je inicijalno omogućio besplatno korišćenje usluga u slučaju *Linux*-a je nakon jednog nesporazuma, odlučio da tu mogućnost ukine. Taj događaj nagnao je Linus Torvalds da po prvi put od nastanka prestane sa radom na *Linux* operativnom sistemu i da se posveti razvoju novog sistema za upravljanje verzijama, *Git*.

Osnovni ciljevi koje je Linus Torvalds postavio pred početak rada na *Git*-u su bili:

- brzina,
- jednostavan dizajn,
- jaka podrška za nelinearan razvoj (hiljade paralelnih grana)
- potpuno distribuiran
- mogućnost da efikasno podrži velike projekte poput *Linux* kernela [4].

Osnovna karakteristika *Git*-a je da svaka verzija predstavlja kompletan prikaz projekta, za razliku od do tada uobičajenih centralizovanih sistema za upravljanje verzijama u kojima je svaka verzija nakon prve verzije sadržavala samo nove promene, odnosno razlike u odnosu na prethodnu verziju. Pored toga, *Git* koristi SHA-1 (eng. *Secure Hash Algorithm 1*) kako bi kreirao kontrolnu sumu (eng. *checksum*) podataka pre njihovog smeštanja u repozitorijum kao nove verzije. Različitim verzijama se pristupa korišćenjem upravo ove kontrolne sume. Na ova način onemogućeno je da dođe do bilo kakve promene podataka, a da to ne bude detektovano od strane *Git*-a. *Git* projekat sastoji se od tri glavne sekcije i to su radno stablo, stejdž (eng. *stage*) i repozitorijum. Radno stablo predstavlja verziju projekta nad kojim se radi. Stejdž predstavlja datoteku u kojoj se nalaze informacije o tome šta će naredno biti poslato u repozitorijum.

Prva verzija *Git*-a objavljena je 2005. godine nakon čega se Linus Torvalds vratio razvoju *Linux*-a, dok su razvoj *Git*-a nastavili drugi developeri.

¹⁹ <https://www.nongnu.org/cvs/>

²⁰ <https://subversion.apache.org/>

²¹ <https://git-scm.com>

²² <https://www.mercurial-scm.org>

²³ <https://bazaar.canonical.com/>

²⁴ <http://darcs.net/>

Tri godine nakon nastanka *Git*-a, odnosno 2008. godine, kreirana je kompanija *GitHub* koja razvija istoimeni proizvod. *GitHub* je sajt za saradnju u razvoju kôda koji je zasnovan na *Git*-u [5]. U prvoj deceniji od nastanka, *GitHub* postaje primer radnog okruženja za razvoj znanja [6] i danas predstavlja najveći hosting sajt za *Git* repozitorijume. U oktobru 2019. godine, na *GitHub*-u se nalazilo preko 44 miliona repozitorijuma na kojima je radilo preko 40 miliona korisnika. Više od 2.9 miliona organizacija iz preko 70 država koristilo je *GitHub*, a od 50 najbogatijih kompanija na svetu (eng. *Global Fortune 50*) čak 29 je doprinelo razvoju nekog od projekata otvorenog kôda čiji repozitorijum se nalazi na *GitHub*-u [7].

Značaj *GitHub*-a kao platforme za razvoj softvera, ali i deljenje znanja, utoliko je veći kada se ima u vidu da je u periodu od oktobra 2018. godine do septembra 2019. godine više od 1.7 miliona studenata učilo da razvija softver koristeći *GitHub*, a da ja preko 31.000 predavača koristilo ovu platformu u izvođenju nastave. Više od 761.000 studenata je u procesu učenja koristilo *GitHub Student Developer Pack*, a preko 19.800 obrazovnih institucije različitih nivoa je *GitHub* koristilo u realizaciji svojih studijskih programa. Kada se govori o obrazovanju, trend u periodu oktobar 2018. – septembar 2019. godine pokazuje rast od 33% u korišćenju *GitHub*-a od strane izvođača nastave, i rast od 50% kada su u pitanju studenti [7].

GitHub je proteklih nekoliko godina posebnu pažnju posvetio korišćenju *GitHub*-a u svrhu obrazovanja, a ranije navedeni statistički podaci pokazuju koliko su u tome bili uspešni. Posebnu ulogu u ostvarivanju ovih rezultata imao je i paket *GitHub Education*.

4. GITHUB EDUCATION

GitHub Education predstavlja skup alata kreiranih tako da olakšaju rad kako predavačima i istraživačima, tako i studentima i da ujedno povećaju efikasnost izvođenja nastave.

Preduslovi da bi se koristio *GitHub Education* je registracija korisnika, bilo kao predavača ili studenta, što se potvrđuje odgovarajućim dokumentom. Bitno je napomenuti da ova registracija nije trajna i da je *GitHub* povremeno zahteva da se ponovo potvrdi status predavača ili studenta.

Alati namenjeni predavačima su *GitHub Classroom*, *GitHub Education Community*, plan i program za podučavanje *Git*-a i *GitHub*-a i besplatan pristup *GitHub Team* alatu. Već pominjan besplatni pristup *GitHub Team* dostupan je i istraživačima.

GitHub Classroom je alat koji omogućava predavačima da kreiraju *GitHub* Organizacije. *GitHub* Organizacija predstavlja grupu ljudi koji rade zajedno, bilo unutar kompanije ili na nekom projektu. U okviru *GitHub Education*, organizacija predstavlja jedan predmet/kurs koji se izučava u okviru studijskog programa. U okviru organizacije moguće je kreirati listu studenata (eng. *roaster*) koji pohađaju određeni predmet/kurs, a potom i jedan ili više zadataka (eng. *assignment*) na kojima studenti treba da rade. Sa studentima se deli Internet adresa zadatka i pristupom adresi svaki od studenata dobija mogućnost da kreira sopstveni repozitorijum u kojem će rešavati zadatak. Predavač može koristeći *GitHub Classroom* da na jednom mestu prati rad i napredak svih studenata koji slušaju dati predmet/kurs i da po potrebi na jednostavan način preuzme sve studentske radove. Repozitorijumi koje koriste studenti su privatni.

Education Community predstavlja zajednicu predavača i studenata gde je omogućena razmena ideja i znanja vezanih za obrazovni proces.

Plan i program za podučavanje *Git*-a i *GitHub*-a predstavlja pripremljena predavanja vezana za sledeće teme: Uvod u *GitHub* (eng. *Intro to GitHub*), *GitHub* za sve (eng. *GitHub for Everyone*), *Git* osnove (eng. *Git Essentials*) i *GitHub* za developere (eng. *GitHub for developers*). Uvod u *GitHub* predstavlja kratko predavanje od oko 90 minuta gde se opisuju osnovne funkcionalnosti *GitHub*-a, *GitHub* za sve je nešto opširnije predavanje u kojem se obrađuju i problemi upravljanja konfliktima (eng. *merge conflicts*), dok *Git* osnove predstavljaju predavanja koje se bavi samo funkcionisanjem *Git*-a. *GitHub* za developere predstavlja predavanje u koje se studenti osposobljavaju za upotrebu *Git*-a i *GitHub*-a korišćenjem komandne linije.

Besplatan *GitHub Team* omogućava da se unutar *GitHub* Organizacije kreira neograničen broj javnih i privatnih repozitorijuma, da neograničen broj saradnika može da radi na javnim repozitorijuma, dvofaktornu autentifikaciju, diskusije unutar tima, kontrolu pristupa u okviru tima koja omogućava kreiranje i upravljanje grupama, ali i druge alate poput *GitHub* stranica, *Wiki* stranice, zaštićene grane, vlasništvo kôda i veliki broj statističkih podataka vezanih za repozitorijum.

Predavači takođe mogu pohađati kurs *Git*-a i *GitHub*-a i nakon uspešnog savladavanja kursa mogu postati *GitHub Campus Advisors* što podrazumeva spremnost da kolegama i studentima pomažu svojim iskustvom u rešavanju problema vezanih za *Git* i *GitHub*.

Studentima je omogućen pristup *GitHub Education Community*, a mogu se registrovati i za *GitHub Student Developer Pack* koji će im pored neograničenog broja privatnih i javnih repozitorijuma na *GitHub*-u omogućiti korišćenje alata i servisa koje koriste i profesionalni developeri, a koje razvijaju partnerske kompanije poput *Name.com* (jedan besplatan Internet domen sa naprednim bezbednosnim alatima), *AWS Educate* (besplatan početni nalog sa \$100 koji se mogu potrošiti ne usluge ove kompanije), *JetBrains* (besplatna godišnja pretplata za korišćenje IntelliJ IDE), *Bootstrap Studio* (besplatna licenca za korišćenje *Bootstrap Studio*), *Educative* (šest meseci besplatnog pristupa različitim računarskim kursevima), *DigitalOcean* (\$50 za usluge hostinga ove kompanije) i mnogi drugi [8] [9] [10].

5. ZAKLJUČAK

Kao što je predstavljeno u uvodu ovog rada jedan od osnovnih problema koji su postojali čak i u slučaju korišćenja sistema za upravljanje verzijama, poput *Git*-a, u procesu nastave je što repozitorijumi koje bi trebalo da dele predavači i studenti nisu bili međusobno integrisani. Takođe, ograničenje koje je do skora postojalo u kreiranju privatnih repozitorijuma predstavljalo je dodatan problem, jer su studenti koji se nalaze na početku svoj školovanja želeli više privatnosti. Upotrebom *GitHub Education*, konkretno *GitHub Classroom*-a, ovi problemi su rešeni. Studenti dobijaju privatne repozitorijume nad kojima rade svoje zadatke. Ti repozitorijumi su vezani za *GitHub* Organizacije, odnosno predmete/kurseve, i pored samih studenata dostupni su i administratorima organizacija, odnosno predavačima. Na ovaj način predavači na jednom mestu imaju uvid u sve radove studenata. Takođe upotrebom sistema za upravljanje verzijama mogu pratiti i tok rešavanja zadatka od strane svakog pojedinačnog studenta. Na ovaj način smanjuje se mogućnost zloupotreba, jer predavač pored konačnog rezultata projekta može da proveriti i proces rada, odnosno postupak kako je student došao do rešenja.

Pored toga u slučajevima kada se isti zadatak ponavlja svake školske godine, predavači sada imaju uvid u sva prethodna rešenja, jer studentima nije dozvoljeno brisanje repozitorijuma, te lakše mogu primetiti ukoliko se ista rešenja ponavljaju u različitim školskim godinama, odnosno otkriti slučajeve plagijarizma.

Predavači mogu deo kôda koji se radi na računarskim vežbama i dalje deliti sa studentima koristeći po potrebi javne ili privatne repozitorijume koji nisu deo *GitHub Education*.

Jasno je da je proces razmene kôda upotrebom *GitHub Education* na ovaj način značajno pojednostavljen, a samim tim unapređen i proces nastave. Imajući u vidu da je, kao što je to ranije navedeno, *GitHub* dominantna platforma za rad softverskim projektima, samo korišćenje *GitHub*-a u procesu nastave predstavlja dodatnu prednost jer priprema studenata za rad na projektima nakon završetka formalnog obrazovanja.

U ovom radu predstavljene su neke od mogućnosti *GitHub Education* paketa alata, kako za predavače i istraživače, tako i za studente, odnosno fokus je bio na mogućnostima koje zadovoljavaju potrebe pojedinačnih predmeta/kurseva. Pored toga, *GitHub Education* pruža mogućnosti unapređenja nastave i za čitave kampuse kroz *GitHub Campus program*, kao i *GitHub Campus Expert* trening program koji se pored razvoja softvera fokusira i na razvoj veština javnog govorništva i liderstva u društvenoj zajednici.

6. LITERATURA

- [1] W. F. Tichy, „RCS — a system for version control,“ *Software: Practice and Experience*, t. 15, br. 7, pp. 637-654, 1985.
- [2] T. Bell, J.-M. Kim, A. Porter i H. Siy, „If Your Version Control System Could Talk...,“ u *ICSE Workshop on Process Modelling and Empirical Studies of Software Engineering*, 1997.
- [3] B. de Alvis i J. Sillito, „Why Are Software Projects Moving From Centralized to Decentralized Version Control Systems?,“ u *ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, 2009.
- [4] S. Chacon i B. Straub, *Pro Git*, Mountain View, 2019.
- [5] E. Kalliamvakou, G. Gousios i K. Blincoe, „The Promises and Perils of Mining GitHub,“ u *11th working conference on mining software repositories*, Hyderabad, 2014.
- [6] L. Dabbish, C. Stuart, J. Tsay i J. Herbsleb, „Social Coding in GitHub: Transparency and Collaboration in an Open Software Repository,“ u *ACM 2012 conference on computer supported cooperative work*, Seattle, 2012.
- [7] GitHub, „The State of the Octoverse,“ GitHub, 30 09 2019. [Na mreži]. Available: <https://octoverse.github.com/>. [Poslednji pristup 26 12 2019].
- [8] GitHub, „Engaged students are the results of using real-world tools,“ GitHub, [Na mreži]. Available: <https://education.github.com/>. [Poslednji pristup 26 12 2019].
- [9] GitHub, „About GitHub Education for educators and researchers,“ GitHub, [Na mreži]. Available: <https://help.github.com/en/github/teaching-and-learning-with-github-education/about-github-education-for-educators-and-researchers>. [Poslednji pristup 26 12 2019].
- [10] GitHub, „About GitHub Education for students,“ GitHub, [Na mreži]. Available: <https://help.github.com/en/github/teaching-and-learning-with-github-education/about-github-education-for-students>. [Poslednji pristup 26 12 2019].