



SE101 - RAZVOJ SOFTVERA I INŽENJERA SOFTVERA

Biti dobar inženjer nije dovoljno

Lekcija 15

PRIRUČNIK ZA STUDENTE

SE101 - RAZVOJ SOFTVERA I INŽENJERA SOFTVERA

Lekcija 15

BITI DOBAR INŽENJER NIJE DOVOLJNO

- ✓ Biti dobar inženjer nije dovoljno
- ✓ Poglavlje 1: Zašto nije dovoljno biti dobar inženjer?
- ✓ Poglavlje 2: Komunikacione veštine
- ✓ Poglavlje 3: Pravi stav
- ✓ Poglavlje 4: Tehničko iskustvo
- ✓ Poglavlje 5: Posedovanje „mekih“ veština (soft skills)
- ✓ Poglavlje 6: Organizovanost
- ✓ Poglavlje 7: Timski rad
- ✓ Poglavlje 8: Vežba
- ✓ Zaključak

Copyright © 2017 – UNIVERZITET METROPOLITAN, Beograd. Sva prava zadržana. Bez prethodne pismene dozvole od strane Univerziteta METROPOLITAN zabranjena je reprodukcija, transfer, distribucija ili memorisanje nekog dela ili čitavih sadržaja ovog dokumenta., kopiranjem, snimanjem, elektronskim putem, skeniranjem ili na bilo koji drugi način.

Copyright © 2017 BELGRADE METROPOLITAN UNIVERSITY. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, without the prior written permission of Belgrade Metropolitan University.

▼ Uvod

UVOD

Cilj i sadržaj lekcije

U okviru lekcije saznaćete više informacija o tome zašto nije dovoljno biti samo "dobar inženjer". Oblasti o kojima ćete učiti su:

- Koje su to karakteristike izvanrednih inženjera
- Koji je značaj veštine komunikacije
- Saveti za unapređenje svojim komunikacionih veština
- Primeri pravih stavova izuzetnih inženjera softvera
- Zašto je tehničko iskustvo važno
- Koje su najvažnije meke veštine, a koje organizacione veštine
- Prednosti rada u timu

▼ Poglavlje 1

Zašto nije dovoljno biti dobar inženjer?

ODLIKE IZVANREDNOG INŽENJERA SOFTVERA

Sjajni softverski inženjeri mogu biti tri puta produktivniji od prosečnih i deset puta produktivniji od loših

Kada je u pitanju softversko inženjerstvo, razlika između izvanrednog, lošeg i prosečnog (dobrog) inženjera je ogromna. Sjajni softverski inženjeri mogu biti tri puta produktivniji od prosečnih i deset puta produktivniji od loših. Odlični inženjeri su žila kucavica softverskih kompanija. Različite organizacije prate različite prakse i stilove rada i imaju jedinstvene preferencije u pogledu dobrog kandidata. Međutim, postoji nekoliko kvaliteta odličnih inženjera za koje se svi mogu složiti. Jer, nije dovoljno biti samo dobar inženjer. Cilj je izdvojiti se i biti prepoznatljiv u svojim profesionalnim krugovima! U nastavku se nalaze kvalitete koji, osim tehničkog znanja kodiranja, odlikuju izvanredne softverske inženjere:

- *Komunikacione veštine*

Iako mnogi ljudi mogu zamisliti inženjere softvera kako samo nečujno sede za računarom po ceo dan, u stvarnosti, efikasna komunikacija je jedan od faktora koji razlikuju izvanredne programere softvera. Komunikacija je suštinski deo posla programera softvera, za povezivanje sa klijentima i koordinaciju sa drugim članovima tima.

- *Pravi stav*

To je jedan od onih neizrecivih atributa koji se zataškavaju kao 'prah', ali pravi stav može zaista da definiše performanse inženjera. Sjajan softverski inženjer brine o projektu iznad svega, i ne dozvoljava da njihov ego ili radno opterećenje smetaju da uloži 100% u projekat.

- *Tehničko iskustvo*

Svaki sjajan inženjer koji se izdvaja ima izvanredne tehničke veštine i iskustvo. Iskustvo je važno jer učenje veština zahteva relativno malo vremena u poređenju sa godinama koje su potrebne da se zaista postane stručnjak. Idealni programer ne bi trebalo da ima samo 'veštine' u programskom jeziku, već i duboko iskustvo, jer je koristio jezik u drugim projektima tokom dužeg vremenskog perioda.

OSOBINE IZVANREDNOG INŽENJERA SOFTVERA

Neki od atributa koje treba posedovati navedeni su u okviru sekcije

- *Posedovanje „mekih“ veština (soft skills)*

Ove osobine su podjednako važne kao i tehničke veštine. Sposobnost rada sa timom, upravljanje vremenom, društvene veštine, komunikacija i upravljanje konfliktima su atributi koje treba posedovati.

- *Organizovanost*

Ako posao koji treba da se uradi za projekat ne bude isporučen na vreme, sam rad neće mnogo značiti, bez obzira koliko je programer odličan. Upravljanje vremenom i zadacima su neophodni za najbolje programere jer moraju da upravljaju svojim radom kako bi projekti bili završeni na vreme.

- *Timski rad*

Proces razvoja softvera često uključuje širok i raznolik tim, uključujući druge inženjere softvera i više rukovodstvo. Rad sa timom na upravljanju i izvršavanju zadataka u cilju krajnjeg uspeha projekta je suštinska veština najboljih inženjera softvera.

▼ Poglavlje 2

Komunikacione veštine

ZNAČAJ SPOSOBNOSTI KOMUNIKACIJE

Komunikacija je važna jer je veza između planova, dizajna i vizije kompanije i konačnog softverskog proizvoda koji će se razvijati

U današnje vreme se svet promenio, a IT industrija je postala drugačija. Sada, veštine komunikacije nisu manje važne od veština kodiranja. Ako programeri ne mogu pravilno da komuniciraju, ne mogu efikasno da rade na projektu jer:

- Kupci su se promenili. Većina njih razume industriju. Oni jasno znaju koji proizvod žele da imaju. Štaviše, žele da budu stalno obavešteni o napretku tima i da jasno vide na šta se troši novac.
- Timovi su se promenili. Dok su ranije neki programeri radili u kancelariji, sada se uglavnom radi o distribuiranim timovima. Članovi tima se nalaze u različitim zemljama. Uglavnom tim ima članove iz različitih vremenskih zona. Sastanci u kancelariji kao što su nekada bili su nemogući. Stoga je jasna komunikacija između članova tima neophodna. Danas će programeri moći ne samo da razgovaraju otvoreno i jasno, već i da dokumentuju šta rade i sa kojim problemima se suočavaju.

Ukoliko se traži izvanredan razvoj softvera, u fokusu su verovatno visoko kvalifikovani programeri softvera. Međutim, znanje i veština nisu jedine stvari koje treba tražiti prilikom biranja tima za razvoj softvera. Previše je uobičajeno videti kako projekti za razvoj softvera zaostaju, skreću sa koloseka ili potpuno propadaju jer im nedostaje jedan kritični faktor – komunikacija.

Zašto je komunikacija važna za razvoj softvera?

Komunikacija je važna jer je veza između planova, dizajna i vizije kompanije i konačnog softverskog proizvoda koji će se razvijati. Prilikom razvoja softvera, komunikacija pokreće dizajn, planiranje, vremenske okvire, očekivanja, rešavanje problema, revizije, testiranje i pokretanje. To je način da se ostane u vezi sa vizijom klijenta i da se osigura da ono što se kreira tačno odgovara njihovim potrebama.

Znanje i veštine programiranja nije teško pronaći među inženjerima softvera, ali odličnu komunikaciju jeste.

KAKO BITI DOBAR KOMUNIKATOR U SOFTVERSKOM INŽENJERSTVU?

Saveti za dobrog komunikatora u softverskom inženjerstvu

- **Biti sažet** - Dobra komunikacija ne znači više komunikacije. U mnogim slučajevima, ključ je kvalitet nad kvantitetom. Trebalo bi imati za cilj reći manje stvari, ali bez gubljenja uticajne tačke poruke, a to nije lako. Komunikacija bi trebalo da bude kratka. Koliko ključnih tačaka imamo na umu kada kupujemo automobil? Možda 5 do 6. Ako bi prodavac držao prodajnu prezentaciju bez prekida 20 minuta, verovatno bismo prekinuli razgovor pre donešenja odluke o kupovini. Isti princip važi i za svakodnevni rad. Potrebno je imati na umu i vremenska ograničenja i raspon pažnje publike bilo da je u pitanju tester u vašem timu, drugi programer softvera, kupac ili menadžer.
- **Deljenje informacija i poboljšanje komunikacije** - Jedna od najbitnijih stvari u cilju poboljšanja komunikacije je pisanje. Jednostavno u teoriji, ali malo teže u praksi kada većina ljudi obavlja više zadataka i istovremeno vodi različite razgovore. To zahteva doslednu praksu. Ovo se može uraditi pisanjem svom timu i molbom da daju povratne informacije. Na primer, moguće je napisati beleške o primeni, napomene o instalaciji ili nešto specifično u cilju pomoći klijentu da razume novu funkciju softvera. Zatim zamoliti pojedinca koji koristi ovaj materijal da kaže da li je pomogao ili ne. Ovo će obezbediti odličnu (i besplatnu) vežbu komunikacije.
- **Pažljivo slušanje** - Potrebno je slušati pažljivo šta druga osoba govori, bez njenog prekidanja. Zatim, pre davanja svog komentara, odlična tehnika je preformulisati ono o čemu je osoba pričala i ponoviti svojim rečima u cilju dobijanja potvrde potpunog razumevanja onoga šta druga osoba pokušava da komunicira.
- **Postavljanje pitanja** - Na ovaj način ne samo da se dobija više informacija, već će opet osigurati da jasno razumevanje poruke koja se prenosi.
- **Razumeti zahteve klijenata** - Biti efikasan komunikator takođe se u velikoj meri oslanja na odvajanje vremena za jasno razumevanje zahteva i uputstava kupaca.
- **Napadati problem, ali ne i osobu** - Ponekad je član tima možda uneo grešku u kod koja uzrokuje nazadovanje u projektu. Važno je pri razgovoru sa svojim kolegom pozabaviti se temom ali bez lične kritike. U slučaju kritikovanja ili napadanja osobe, postoji rizik od uticaja na ego i isključivanje svake otvorenosti uma. Ovo eliminiše svaku šansu da se krene napred sa korisnim rešenjem za saradnju.
- **Izabrati odgovarajući kanal komunikacije** - Postoje različiti kanali komunikacije od kojih svaki ima rangiranje efikasnosti i cene. U zavisnosti od toga šta treba reći, kada i koliko je važno izbeći greške u komunikaciji, postoji predloženo sredstvo za najbolji kanal za korišćenje. Primer: Ako treba podeliti kritične, vremenski osetljive informacije, možda je najbolje poslati e-poštu ili snimiti video i podeliti to danas umesto pokušavanja da se zakaže sastanak sa svima koji su uključeni u toku sledeće nedelje.

Razvoj softvera je umetnost, a ne samo nauka. Dobar inženjer može naučiti sve tehničke detalje razvoja softvera, ali mora biti apsolutno strastven oko kodiranja i doživljavati ga kao umetnost da bi bio izuzetan u ovom poslu. Cilj svakog velikog programera jeste da svoju umetnost učini što lepšom i najboljom.

Neki od stavova koje bi veoma uspešni programeri trebalo da imaju, osim opštih očekivanja da budu tehnički i analitički ispravni, da detaljno razumeju zahteve, da imaju dobre dizajnerske veštine itd. su:

- Greška je pitanje sposobnosti za pisanje dobrog koda

Ispravljanje grešaka je sastavni deo aktivnosti inženjera softvera. Greška je očigledno najgori neprijatelj softverskih inženjera, ali pitanje je koliko se njih zaista zapita kako da spreče ponovnu pojavu greške u budućnosti. Osoba koja je voljna da bude odličan programer treba da smatra grešku kao pretnju svojoj poziciji, kao pretnju svom kredibilitetu i veštinama programiranja. To je stav koji doprinosi stvaranju odličnog inženjera softvera.

- Izazivanje testera dapronađu greške u kodu

Sjajan inženjer softvera ili osoba koja je voljna da to postane uvek treba da pozove/izazove testera da pronađe greške u njenom kodu. Trebalo bi da ima toliko poverenja u svoj kod da bude spremna da izazove na takav način.

- Samouverenost, ali ne arogancija

Odličan softverski inženjer treba da bude potpuno uveren u svoje sposobnosti, ali ne bi trebalo da bude arogantan prema kolegama i testerima. Uvek treba zapamtiti da je osoba deo tima koji radi na zajedničkom cilju isporučivanja projekta na vreme i dobrog kvaliteta.

▼ Poglavlje 3

Pravi stav

VAŽNOST PRAVOG STAVA

Dobar inženjer može naučiti sve tehničke detalje razvoja softvera, ali mora biti apsolutno strastven oko kodiranja i doživljavati ga kao umetnost da bi bio izuzetan u ovom poslu

Razvoj softvera je umetnost, a ne samo nauka. Dobar inženjer može naučiti sve tehničke detalje razvoja softvera, ali mora biti apsolutno strastven oko kodiranja i doživljavati ga kao umetnost da bi bio izuzetan u ovom poslu. Cilj svakog velikog programera jeste da svoju umetnost učini što lepšom i najboljom.

Neki od stavova koje bi veoma uspešni programeri trebalo da imaju, osim opštih očekivanja da budu tehnički i analitički ispravni, da detaljno razumeju zahteve, da imaju dobre dizajnerske veštine itd. su:

- Greška je pitanje sposobnosti za pisanje dobrog koda

Ispravljanje grešaka je sastavni deo aktivnosti inženjera softvera. Greška je očigledno najgori neprijatelj softverskih inženjera, ali pitanje je koliko se njih zaista zapita kako da spreče ponovnu pojavu greške u budućnosti. Osoba koja je voljna da bude odličan programer treba da smatra grešku kao pretnju svojoj poziciji, kao pretnju svom kredibilitetu i veštinama programiranja. To je stav koji doprinosi stvaranju odličnog inženjera softvera.

- Izazivanje testera da pronađu greške u kodu

Sjajan inženjer softvera ili osoba koja je voljna da to postane uvek treba da pozove/izazove testera da pronađe greške u njenom kodu. Trebalo bi da ima toliko poverenja u svoj kod da bude spremna da izazove na takav način.

- Samouverenost, ali ne arogancija

Odličan softverski inženjer treba da bude potpuno uveren u svoje sposobnosti, ali ne bi trebalo da bude arogantan prema kolegama i testerima. Uvek treba zapamtiti da je osoba deo tima koji radi na zajedničkom cilju isporučivanja projekta na vreme i dobrog kvaliteta.

PRIMERI STAVOVA IZUZETNOG SOFTVERSKOG INŽENJERA

Bez kompromisa po pitanju kvaliteta koda i Potvrda testera su neki od stavova uspešnih inženjera softvera

- Bez kompromisa po pitanju kvaliteta koda

Kvalitet koda treba da bude od primarne važnosti za inženjera. To podrazumeva praćenje ispravnih standarda kodiranja, omogućavanje lakšeg održavanja koda korišćenjem odgovarajućeg dizajna i refaktorisiranja koda, itd. Sjajan softverski inženjer nikada ne bi trebalo da pravi kompromise u pogledu kvaliteta koda, bez obzira na sve.

- Potvrda testera

Može se desiti da uprkos teškom radu i naporima koje je uložio veoma uspešan inženjer, odličan tester i dalje može da pronađe nedostatke u kodu. U tim slučajevima, trebalo bi odati priznanje testeru. Veoma dobar inženjer bi trebalo uvek da prizna grešku koju je tester pronašao.

▼ Poglavlje 4

Tehničko iskustvo

VAŽNOST TEHNIČKOG ISKUSTVA

Lista razloga koji dokazuju da je tehničko iskustvo od izuzetnog značaja

U nastavku se nalazi lista razloga zbog kojih je tehničko iskustvo u softverskom inženjerstvu od izuzetnog značaja za razvoj izvanredne karijere.

- **Lična nagrada**

Lično zadovoljstvo radom na stvarnom projektu koji koristi preduzeću ili organizaciji ne može se potceniti. Takva iskustva mogu biti pravi podsticaj za samopouzdanje i motivacija ka ciljevima u karijeri.

- **Poboljšanje "mekih" veština**

Komunikacija sa drugima, rad u timu, rešavanje problema, sve su to veštine koje se usavršavaju kroz iskustvo. Takođe, stiče se i navika rada pod pritiskom kroz svakodnevne ciljeve i ispunjavanje zadataka ili projekata. Takve kompetence su prenosive čak i ako se radi u oblasti koja nije povezana sa ulogom u novoj kompaniji.

- **Razvoj tehničkih kompetenci**

Ukoliko osoba poseduje veštine u određenoj oblasti, radno iskustvo joj omogućava da oduševi poslodavca svojim sposobnostima, shvati zanatske trikove i obezbedi sebi unapređenje ili potencijalne nove poslove.

- **Sticanje iskustva u industriji**

Osim izgradnje tehničkih veština, kroz rad zaposleni saznaje kako kompanija funkcioniše, kuda industrija ide, načine napredovanja zaposlenih u karijeri i mnoge druge aspekte posla. Takvo znanje može biti ključno u fazi intervju za posao.

ZNAČAJ TEHNIČKOG ISKUSTVA

Tehničko iskustvo ima značajan uticaj na razvoj karijere veoma uspešnog inženjera softvera

- **Mogućnost razmatranja dalje karijere**

Ako osoba nije sigurna vezano za svoju karijeru, kroz rad može da proceni da li se odabrani put dobro uklapa u njene oblasti interesovanja. Na poslu može steći i uvid u poslove koje možda do tada nije razumela. Ovo je dobra prilika koja može navesti zaposlenog na istraživanje i drugih puteva karijere.

- **Dokazivanje motivisanosti za rad**

Neke karijere zahtevaju sticanje niza godina radnog iskustva kako bi bila dokazana ozbiljnost i spremnost za rad. Pronalaženje niza radnih prilika, bilo plaćenih, dobrovoljnih ili kao pripravnik, pokazuje potencijalnom poslodavcu posvećenost datoj profesiji.

- **Izgradnja sopstvene mrežesaradnika**

Radno iskustvo može otvoriti priliku za umrežavanje kontakata na koje se može naići u okviru industrije i uz određenu polugu može se produžiti položaj u odabranoj oblasti ili steći dodatno iskustvo negde drugde.

- **Dobijanje novog posla**

Mnoge kompanije interno oglašavaju poslove ili zapošljavaju one koje poznaju drugi zaposleni. Sticanje radnog iskustva je put do dobijanja novog posla. U nekim industrijama konkurencija za poslove je toliko žestoka da je radno iskustvo često jedini put.

- **Impresioniranje poslodavca**

Za mnoge poslodavce ništa nije impresivnije od kandidata sa relevantnim radnim iskustvom. Ukoliko kandidat poseduje ovo u svom CV-u, svrlo verovatno će biti razmatran ispred ostalih kandidata.

▼ Poglavlje 5

Posedovanje „mekih“ veština (soft skills)

VAŽNOST MEKIH VEŠTINA U SOFTVERSKOM INŽENJERSTVU

Radna etika, stav, komunikacione veštine, emocionalna inteligencija i čitav niz drugih ličnih atributa su meke veštine koje su ključne za uspeh u karijeri

Da biste dobili i zadržali posao, obično je potreban repertoar tehničkih veština. Osim tehničkih veština, na koji način birate kod kog zubara idete? Idete kod onoga ko je prijatan i ima vremena da odgovori na vaša pitanja; ili onoga koji vas tretira kao broj u dugom nizu pacijenata? U ovim situacijama, kao i u svim ostalim poput njih, važne su meke veštine, odnosno **soft skills**.

Radna etika, stav, komunikacione veštine, emocionalna inteligencija i čitav niz drugih ličnih atributa su meke veštine koje su ključne za uspeh u karijeri. Sa ovim mekim veštinama moguće je istaći se kao lider. Rešavanje problema, delegiranje, motivisanje i izgradnja tima su mnogo lakši ukoliko osoba ima dobre meke veštine. Znati kako se slagati sa ljudima – i pokazati pozitivan stav – ključno je za uspeh.

Problem je u tome što se važnost ovih mekih veština često potcenjuje i za njih je obezbeđeno mnogo manje obuke nego za tehničke veštine. Iz nekog razloga, čini se da organizacije očekuju da ljudi znaju kako da se ponašaju na poslu. Oni imaju tendenciju da pretpostave da svi znaju i razumeju koliko je važno stići na vreme, preuzimati inicijativu, biti prijateljski raspoloženi i proizvoditi visokokvalitetan posao. Pretpostavka da su meke veštine univerzalne dovodi do mnogo frustracija. Zato je važno fokusirati se na obuku i razvoj mekih veština koliko i na tradicionalne tehničke veštine.

Meke veštine su ono što prati tehničke veštine i pomaže organizaciji da u potpunosti iskoristi svoju stručnost. Ukoliko je osoba zaista dobra u pridobijanju klijenata, a nije baš dobra u njihovom zadržavanju, velike su šanse da postoji jaz u mekim veštinama. U stvari, kad god ne postoji mogućnost da se iskoristi bogatstvo znanja, iskustva i stručnosti unutar tima, onda bi trebalo proceniti nivo komunikacije i međuljudskih veština koje su prisutne u organizaciji.

NAJVAŽNIJE MEKE VEŠTINE

Odnosi među ljudima omogućavaju im da u potpunosti učestvuju u timskim projektima, pokazuju poštovanje prema drugima i traže podršku za svoje projekte

Odnosi na radnom mestu razvili su međuljudsku dinamiku koja se ne može zanemariti. Delovi slušanja, predstavljanja ideja, rešavanja sukoba i negovanje otvorenog i poštenog radnog okruženja svode se na znanje kako da se izgrade i održavaju odnosi sa ljudima. Upravo ti odnosi omogućavaju ljudima da u potpunosti učestvuju u timskim projektima, pokazuju poštovanje prema drugima i traže podršku za svoje projekte.

Važno je prepoznati **vitalnu ulogu mekih veština** u okviru tima i ne samo raditi na njihovom razvoju u sebi, već i podsticati njihov razvoj u celoj organizaciji. Oblasti koje treba ispitati i proceniti su:

- Lična odgovornost.
- Stepen saradnje.
- Veštine interpersonalnog pregovaranja.
- Rešavanje konflikata.
- Prilagodljivost i fleksibilnost ljudi.
- Jasnoća komunikacije.
- Kreativno razmišljanje.
- Inkluzija.
- Koučing i mentorstvo.

Što se veći broj navedenih elemenata prepoznaje, veća je verovatnoća da su meke veštine zaposlenih razvijenije. Sve ovo ima značajan uticaj na stav koji osoba ima u interakciji sa klijentima, kupcima, kolegama, supervizorima i drugim zainteresovanim stranama. Što je nečiji stav pozitivniji, to će odnosi te osobe biti bolji. To je ono što podstiče odlične performanse tima i navodi ljude da snažno doprinose viziji i strategiji organizacije.

▼ Poglavlje 6

Organizovanost

ORGANIZACIONE VEŠTINE

Organizacione veštine su meke veštine koje doprinose kod upravljanja očekivanjima, ostajanja u toku sa zadacima i blagovremenog davanja rezultata

Organizacione veštine su meke veštine koje doprinose kod upravljanja očekivanjima, ostajanja u toku sa zadacima i blagovremenog davanja rezultata. Jake organizacione veštine takođe pomažu kod podele potrebnog rada na zadatke kojima se može upravljati. Dobrom organizacijom se definišu rokovi za svaki zadatak dajući prioritet hitnim i važnim zadacima. Takođe, označavaju se prekretnice koje zahtevaju komunikaciju i saradnju.

U svrhu dobre organizacije moguće je koristiti alatke za upravljanje projektima, beležnicu ili kalendar.

Organizacione veštine pomažu da se vreme, trud i resursi usmere na posao koji je važan. One takođe pomažu da sve zainteresovane strane budu na pravom putu da ostvare zajedničke ciljeve. Čineći to, ne sprečavaju se članovi tima da napreduju u svojim rezultatima.

Efikasno organizovanje pomaže:

- **Kod uštede vremena (i novca) kompanije:** zaposleni provode manje vremena tražeći stvari jer već znaju gde da ih pronađu. Umesto toga, to uštedeno vreme mogu potrošiti na druge važne zadatke na poslu.
- **Prilikom rada visokokvalitetnog posla:** Organizacija smanjuje šanse za dostizanje krajnjeg roka i izvršavanja lošijeg posla.
- **Da se postane zaposleni visoke vrednosti:** Ljudi prepoznaju organizovanu osobu kao onu koja je pouzdana i pametna. Na osnovu ovoga moguće je dobiti preporuke i promocije. Izgradnja ličnog brenda kao osobe sa visokim učinkom dovodi do toga da ljudi računaju na tu osobu da će ona stvari obaviti.
- **U izgradnji mentalne kondicije:** Kada je osoba organizovana da pobedi, u boljoj je poziciji da poboljša svoje mentalne sposobnosti.

Organizovani ljudi izgledaju pouzdanije i imaju kontrolu, i mogu brzo uspostaviti osećaj poverenja i profesionalizma.

OSNOVNE ORGANIZACIONE VEŠTINE ZA USPEŠNOG INŽENJERA SOFTVERA

Organizacione sposobnosti su neophodne za svakog softverskog inženjera koji želi da bude iznad proseka i napreduje u karijeri

Organizacione sposobnosti su neophodne za svakog softverskog inženjera koji želi da bude iznad proseka i napreduje u karijeri. U nastavku se nalazi lista organizacionih veština koje mogu pomoći u ostvarenju svog punog potencijala.

- *Fizička organizacija*

Podrazumeva dobar raspored fascikli na računaru, stolu i u kancelarijskom prostoru, kao i pravilno grupisanje i obeležavanje pomenutih fascikli koje se koriste. Ova veština omogućava trošenje manje vremena na traženje stvari jer se već zna gde se nalaze.

- *Postavljanje ciljeva*

Znači imati akcioni plan koji će voditi i motivisati osobu ka postizanju cilja. To uključuje razbijanje ciljeva na manje jedinice i postavljanje rokova za njih. Poslodavci vole veštine postavljanja ciljeva jer poboljšavaju učinak i optimizam na radnom mestu.

- *Veštine određivanja prioriteta*

Određivanje prioriteta pomaže u proceni svakog zadatka u cilju definisanja nivoa važnosti i posvećenosti koji su im potrebni. Na taj način osoba zna kojim zadatkom prvo treba da se pozabavi. Pošto nikada nema dovoljno vremena da se sav posao obavi, poslodavci će ceniti ljude koji znaju kako da daju prioritete.

- *Samoupravljanje*

Opisuje način planiranja, sprovođenja i poštovanja svog rada. Menadžment ne mora brinuti da li će posao biti završen ako osoba poseduje ovu veštinu.

OSNOVNE ORGANIZACIONE VEŠTINE ZA USPEŠNOG SOFTVERSKOG INŽENJERA

Organizacione veštine su neophodne za svakog softverskog inženjera koji želi da bude iznad proseka i napreduje u karijeri

- *Donošenje odluka*

Ukoliko osoba često predviđa ishode za različite pravce delovanja na osnovu činjenica, a zatim bira bolji izbor, to znači da ima veštinu donošenja odluka. Ovladavanje ovom veštinom ograničava šanse za posledice loših izbora za kompaniju.

- *Strateško razmišljanje*

To je veština rešavanja problema, odnosno način na koji se analizira situacija kako bi se došlo do rešenja za problem u organizaciji. Strateško razmišljanje je važna prenosiva veština koja se može koristiti na većini poslova. Potrebno je od jednostavnog izvršavanja zadataka do razumevanja kako su oni povezani sa poslovnim ciljem.

- *Upravljanje vremenom*

Način na koji osoba deli svoje vreme između zadataka govori mnogo o njenim veštinama upravljanja vremenom. Upravljanje vremenom znači svest o tome koliko dugo zadaci traju i prilagođavanje rasporeda trajanju zadataka. Upravljanje vremenom može pomoći efikasnijem radu jer dodatni minuti ne propadaju nesvesno. Neguje ravnotežu između posla i privatnog života i zadovoljstva na poslu.

- *Samomotivacija*

To je ona unutrašnja sila koja gura ljude da nastave protiv svih izgleda. To je unutrašnja želja da se postigne cilj, ma koliko to izgledalo nemoguće. Menadžeri nemaju vremena da jure zaposlene, pa je sjajno ako oni može pronaći motivaciju da rade sami.

NAČINI ZA UNAPREĐENJE ORGANIZACIONIH VEŠTINA

Saveti koji mogu pomoći u poboljšanju organizacionih veština

Nakon dobijanja zaposlenja mnoge kompanije rade organizacionu obuku da bi podučavale svoje zaposlene ovim veštinama. Međutim, postoji određeni saveti koji mogu pomoći u njihovom poboljšanju, a prikazani su u nastavku.

- Proceniti svoje radne procedure u cilju samoprocene svoje organizovanosti i pronalaženja slabih tačaka.
- Očistiti i održavati urednim fizički i digitalni radni prostor.
- Dokumentovati uputstva i dragocene informacije koje su od pomoći za poštovanje rokova.
- Grupisati svoje dokumente u fascikle radi lakšeg pristupa.
- Imati lični kalendar i blokirati rokove za svoj rad.
- Prioritizacija zadataka prema važnosti i rokovima.
- Odvojiti najmanje 10 minuta za planiranje svog dana i zakazivanje aktivnosti.

▼ Poglavlje 7

Timski rad

PREDNOSTI RADA U TIMU

Ovladavanje fluktuacijom samo je jedna od prednosti timskog rada u razvoju softvera

Postoji opipljiva razlika u nivou motivacije dobrih timova u poređenju sa lošim timovima. Razlika se može videti u prometu sa tehnološkim startapima koji često prelazi 25% u poređenju sa već visokim godišnjim prosekom u industriji za programere softvera od 12,3%. Stopa odustajanja za programere raste sa 1,25% mesečno u 2014. na 1,75% u 2021. Ovladavanje fluktuacijom samo je jedna od prednosti timskog rada u razvoju softvera. Dobar timski rad donosi mnoge druge prednosti, iako neke može biti teško izmeriti samostalno.

- Timski rad pomaže u poboljšanju veštine softverskih inženjera

Svaki član tima ima svoje obrazovanje, iskustvo i veštine. Kada rade zajedno, prirodno je da se znanje deli. Razmena znanja je pojačana kada se uparuju programeri, tokom pregleda koda i vodiča, retrospektiva, mentorstva i programa kontinuiranog obrazovanja.

- Timski rad povećava kvalitet koda

Ne samo da se znanje deli, već se i kvalitet kodiranja poboljšava jer se održava i sprovodi standard kodiranja. Ključno je definisati „dovoljno dobar kod“ i dosledno sprovoditi upotrebu stilskih vodiča sa linterima i pregledima koda. Čak i od samog početka, trebalo bi da prođe samo nekoliko nedelja da svi u timu počnu savesno da primenjuju standard. Kod će biti lakši za čitanje, biće lakše pronaći greške za život projekta – koji može trajati deceniju ili više.

- Povećava produktivnost

Timski rad ima kaskadni efekat - pod uslovom da se sastav tima tokom vremena ne naruši previše. Što je tim bolje u stanju da radi zajedno, timu postaje lakše da radi zajedno. Saigrači dobijaju osećaj za prednosti i slabosti jedni drugih, znaju kome da se obrate za pomoć u konkretnim problemima i kako im najbolje pristupiti.

RAZLOZI ZA RAD U TIMU

Objašnjenje pozitivnih uticaja rada u timu

- Timski rad inspiriše kreativnost i inovacije

Kada postoji problem koji stalno muči tim, oni će razmišljati o tome. Ako nađu efikasno rešenje, oslobađaju se glavobolje i osvajaju poen sa svojim kolegama. Ovo može dovesti do njihovog istraživanja proizvoda i tehnika ili izazvati bljesak sjaja koji ih dovodi do otkrivanja nekog skrivenog uzroka.

- Retrospektive kontinuirano podižu efikasnost

Retrospektive daju programerima priliku da bezbedno saopšte probleme koji bi im mogli olakšati rad ili omogućiti da teče lakše. Svi članovi tima mogu da čuju o radnim procesima koji mogu biti uzvodno ili nizvodno od onoga što rade. Retrospektive pomažu saigračima da nauče da urade nešto kako bi rad svojih kolega učinili efikasnijim.

- Timski rad olakšava transparentnost i uliva odgovornost

Problemi poput igre okrivljavanja nastaju kada postoji nedostatak transparentnosti. Između pregleda koda i retrospektiva, nema mnogo prostora za skrivanje. Postoji prilično jasan pogled na skoro svaki red koda. Sve što svaki član tima radi je dostupno za prikaz. Ovaj način rada podstiče inženjere softvera da daju sve od sebe.

- Timski rad povećava poslovni potencijal

Poboljšane veštine softverskog inženjera inherentno se prevode u poboljšani kvalitet kodiranja, efikasnost, produktivnost, širinu i dubinu razvojnih mogućnosti. Ovo se, zauzvrat, prevodi u kapacitet tima da obavljaju više posla i efikasno rukovode većim brojem softverskih projekata. Poboljšanja kvaliteta kodiranja i vremena ciklusa vode do srećnijih i uspešnijih kupaca. Profitabilni razvojni tim ima više mogućnosti za napredovanje od onog koji nije.

RAD U TIMU

Moral i kohezija održavaju timove duže zajedno

- Moral i kohezija održavaju timove duže zajedno

Tim koji dobro radi zajedno obično ima mnogo veći nivo poverenja u svoj rad od onih koji to ne rade. Srećni klijenti i uspešni softverski projekti (na vreme, u okviru budžeta i ljudi koji ga koriste dajući dobre kritike) pružaju pozitivne povratne informacije. Tim koji dosledno proizvodi uspešne projekte može razviti osećaj ponosa i lojalnosti. Pored toga, kada programeri osećaju da ih poštuju njihovi menadžeri i da se dobro uklapaju u tim – ostaće duže.

▼ Poglavlje 8

Vežba

PRAKTIČNA VEŽBA

Na vežbama će biti rezimirano gradivo iz prethodnih lekcija u cilju priprema za testove i završni ispit

Zbog specifičnosti sadržaja ove lekcije, na vežbama se ne rade posebni zadaci, već asistent daje detaljnija objašnjenja mekih veština prikazanih na predavanjima, a u skladu sa pitanjima studenata.

Kako je 15. nedelja poslednja nedelja nastave u kojoj je, u okviru vežbi, planirana realizacija testova za proveru stečenog znanja iz prethodnih lekcija, poželjno je da studenti pripreme pitanja iz dela gradiva koja im nisu bila jasna, kako bi im saradnici na vežbama to objasnili. Ova retrospektiva i pregled svih prethodnih lekcija takođe se može posmatrati i kao priprema i pomoć studentima za pripremu završnog ispita iz predmeta.

Trajanje vežbi je 2 časa, odnosno 90 minuta.

✓ Poglavlje 9

Zaključak

ZAKLJUČAK

Sumiranje stečenih znanja

- Sjajni softverski inženjeri mogu biti tri puta produktivniji od prosečnih i deset puta produktivniji od loših.
- Kvaliteti koji odlikuju sjajne softverske inženjere, pored tehničkog znanja i karakterizacije "dobar inženjer" su: komunikacione veštine, pravi stav, tehničko iskustvo, posedovanje mekih veština, organizovanost i timski rad.
- Prilikom razvoja softvera, komunikacija pokreće dizajn, planiranje, vremenske okvire, očekivanja, rešavanje problema, revizije, testiranje i pokretanje. To je način da se ostane u vezi sa vizijom klijenta i da se osigura da ono što se kreira tačno odgovara njihovim potrebama.
- Postoji sedam osnovnih saveta kako biti dobar komunikator u softverskom inženjerstvu.
- Dobar inženjer može naučiti sve tehničke detalje razvoja softvera, ali mora biti apsolutno strastven oko kodiranja i doživljavati ga kao umetnost da bi bio izuzetan u ovom poslu.
- Radna etika, stav, komunikacione veštine, emocionalna inteligencija i čitav niz drugih ličnih atributa su meke veštine koje su ključne za uspeh u karijeri. Rešavanje problema, delegiranje, motivisanje i izgradnja tima su mnogo lakši ukoliko osoba ima dobre meke veštine. Znati kako se slagati sa ljudima – i pokazati pozitivan stav – ključno je za uspeh.
- Važnost mekih veština često se potcenjuje i za njih je obezbeđeno mnogo manje obuke nego za tehničke veštine.
- Važno je prepoznati vitalnu ulogu mekih veština u okviru tima i ne samo raditi na njihovom razvoju u sebi, već i podsticati njihov razvoj u celoj organizaciji.
- Organizacione veštine su meke veštine koje doprinose kod upravljanja očekivanjima, ostajanja u toku sa zadacima i blagovremenog davanja rezultata. Organizacione veštine pomažu da se vreme, trud i resursi usmere na posao koji je važan. One takođe pomažu da sve zainteresovane strane budu na pravom putu da ostvare zajedničke ciljeve.

LITERATURA

Pogodna literatura za učenje

Obavezna literatura:

- Nastavni materijal za e-učenje na predmetu SE101 Razvoj softvera i inženjera softvera, Univeziitet Metropolitan, školska 2022/23. godina
Nastavni materijal je pripremljen korišćenjem referenci 2-8.

Dodatna literatura:

- <https://blog.bydrec.com/8-signs-that-can-differentiate-a-great-software-developer-from-a-good-one>
- <https://blog.bydrec.com/how-great-communication-can-lead-to-great-software-development>
- <https://santexgroup.com/blog/communication-skills-software-engineering/>
- <https://dzone.com/articles/attitudes-great-software>
- <https://www.careerguidancecharts.com/career-poster-blog/10-benefits-of-work-experience>
- <https://www.mindtools.com/a0uur35/why-soft-skills-matter>
- <https://www.betterup.com/blog/organizational-skills>
- <https://gitential.com/teamwork-in-software-development-8-benefits-of-building-up-a-collaborative-team/>