



## SE101 - RAZVOJ SOFTVERA I INŽENJERA SOFTVERA

### Razvojni put softverskih inženjera

Lekcija 14

PRIRUČNIK ZA STUDENTE

# SE101 - RAZVOJ SOFTVERA I INŽENJERA SOFTVERA

## Lekcija 14

### *RAZVOJNI PUT SOFTVERSKIH INZENJERA*

- ✓ Razvojni put softverskih inženjera
- ✓ Poglavlje 1: Razlika između inženjera softvera i programera
- ✓ Poglavlje 2: Postajanje inženjera softvera
- ✓ Poglavlje 3: Uloge u softverskom inženjerstvu
- ✓ Poglavlje 4: Put karijere softverskog inženjera
- ✓ Poglavlje 5: Promocija inženjera softvera u više titule
- ✓ Poglavlje 6: Najbolje opcije za karijeru u razvoju softvera
- ✓ Poglavlje 7: Vežba - Pokazni primeri
- ✓ Poglavlje 8: Vežba - zadaci
- ✓ 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 dobićete više informacija o razvoju karijere softverskog inženjera. Oblasti o kojima ćete učiti su:

- Koja je uloga programera, a koja inženjera softvera
- Koje su ključne razlike između uloga programera i inženjera softvera
- Osnovni koraci koje je potrebno proći kako bi osoba postala softverski inženjer
- Najvažnije lične osobine uspešnog inženjera softvera
- Na koji način su definisane uloge u softverskom inženjerstvu
- Kako teče put karijere inženjera softvera
- Korisni saveti u cilju promocije softverskog inženjera
- Sedam trenutno najpoželjnijih opcija za karijeru u softverskom inženjerstvu

## ▼ Poglavlje 1

# Razlika između inženjera softvera i programera

## OBJAŠNJENJE ULOGE PROGRAMERA

*Programer uzima funkcionalnu specifikaciju i isporučuje potreban kod unutar strogih parametara, u suštini završavajući zadatak izolovano*

Uspešni *programeri softvera* imaju ulogu kada su u pitanju računarski sistemi - da pišu kod i kreiraju softver visokog kvaliteta.

Programer obično započinje proces pravljenja softverskih aplikacija tako što razgovara sa klijentima i otkriva šta su njihove softverske potrebe. Sa ovim kriterijumima u ruci, oni nastavljaju da dizajniraju i razvijaju različite komponente u aplikaciji koje će morati da rade zajedno. Da bi posao obavili, veliki deo onoga što rade je upravljanje celokupnim razvojnim procesom od početka do kraja, pazeći da ostanu u stalnom kontaktu sa ključnim zainteresovanim stranama.

Često će programeri koristiti različite veštine uključujući dobru komunikaciju, upravljanje projektima i tehničku ekspertizu specifičnu za domen. U stvari, ova poslednja tačka je ključna za razumevanje razlike između programera softvera i softverskog inženjera. Programeri imaju tendenciju da rade sve što inženjeri rade, ali u ograničenom obimu. To im omogućava da budu kreativniji od inženjera, smišljajući rešenja za probleme kako se pojave.

Razlika između inženjera i programera je u tome što programer uzima funkcionalnu specifikaciju i isporučuje potreban kod unutar strogih parametara, u suštini završavajući zadatak izolovano.

Ključne kompetence programera:

- Matematička sposobnost
- Poznavanje ciklusa razvoja softvera
- Veštine rešavanja problema
- Poznavanje računarskih programa i veštine računarskog programiranja
- Poznavanje programskih jezika i softverskih alata
- Odlične organizacione veštine i veštine upravljanja vremenom
- Timski rad
- Pažnja prema detaljima i odlične analitičke sposobnosti
- Razumevanje struktura podataka
- Interesovanje za najnovije tehnološke trendove

# OBJAŠNJENJE ULOGE INŽENJERA SOFTVERA

*Softverski inženjeri moraju biti više fokusirani na dizajn strukture i eliminisanje tehničkog duga*

Programer treba da bude softverski arhitekta, neko u stanju da pravi softverske aplikacije od početka do kraja. Oni imaju opštije znanje o celom procesu razvoja. Sa druge strane, *softverski inženjer* bi trebalo da bude u stanju da uradi sve što programer softvera radi, ali sa drugačijom filozofijom dizajna.

Inženjeri imaju širu sliku nego programeri. To znači da moraju biti više fokusirani na dizajn strukture i eliminisanje tehničkog duga. Umesto da budu zainteresovani samo za rešavanje trenutnog problema, oni su zainteresovani za rešavanje tog problema dok istovremeno minimiziraju kompromise sa drugim delovima sistema i njegovom arhitekturom.

Često su softverski inženjeri podeljeni u dve kategorije:

- *inženjeri aplikacija* koji su donekle slični programerima - oni imaju tendenciju da dizajniraju, prave, instaliraju i održavaju aplikacije, obično u velikim razmerama;
- *sistem inženjeri*, s druge strane, održavaju IT infrastrukturu tehnoloških kompanija - umesto da rade na određenoj aplikaciji, inženjeri su zainteresovani za dizajn softvera, da izgrade i održavaju sisteme koji podržavaju aplikacije, uzimajući u obzir sve uključene zainteresovane strane.

Verovatnije je da će softverski inženjeri pre nego programeri biti uključeni u rad systemske analize i naprednije systemsko inženjerstvo.

Ključne kompetence inženjera softvera:

- Softverski inženjering uključuje sposobnost analize složenih tehničkih informacija.
- Oni će implementirati inženjerske principe kako bi uticali na postojeću arhitekturu baze podataka.
- Softverski inženjer ima iskustvo korišćenja računarskog softvera i pravljenja softverskih aplikacija.
- Softverski inženjeri imaju iskustvo sa različitim programskim jezicima, uključujući JavaScript, Java, Rubi, PHP i Python.
- Softverski inženjeri primenjuju inženjerske principe i isporučuju složene softverske sisteme.
- Softverski inženjeri su upoznati sa različitim radnim okvirima.
- Obezbeđuju drugim inženjerima podršku za razvoj softvera i testiranje.

## KLJUČNE RAZLIKE IZMEĐU PROGRAMERA I INŽENJERA SOFTVERA

*Programer softvera teži da se fokusira na rešavanje problema jednog korisnika ili grupe korisnika, dok će se inženjer fokusirati na probleme svih korisnika*

U mnogim osnovnim inženjerskim principima, softverski inženjer i programer softvera mogu da rade slične stvari. Ali ono čemu programer softvera teži jeste da se fokusira na rešavanje problema jednog korisnika ili grupe korisnika, dok će se inženjer fokusirati na probleme svih korisnika. Stoga će njihovo stručno znanje obuhvatiti više inženjeringa.

Dobro je pogledati veštine koje kandidat poseduje. Programer softvera će imati tendenciju da ima znanje specifično za domen, dok inženjeri obično imaju mnogo šire iskustvo.

Prilikom zapošljavanja, veštine koje treba tražiti kod programera i softverskog inženjera su različite. Softverski inženjer će biti neko ko zna matematiku na visokom nivou i pristupa problemima na vrlo logičan način. Ne bi trebalo tražiti nekoga ko prihvata dvosmislenost. Inženjer bi trebalo da bude u stanju da vidi kako bi njegove akcije mogle stvoriti probleme u nastavku.

Kada se traži programer, želi se dati veći naglasak kreativnom rešavanju problema. Programer ima tendenciju da bude više fokusiran, tako da se želi neko ko je više zainteresovan za pronalaženje rešenja nego da brine o dugoročnom tehničkom dugu.

#### Ključne razlike:

- Programeri softvera imaju tendenciju da rade sve što softverski inženjeri rade, ali u ograničenom obimu. Ovo omogućava programeru da bude kreativniji od inženjera - ne samo da razvijaju softver već i da pronađu rešenja za probleme kada se pojave.
- Softverski inženjering uključuje mogućnost da uradi sve što programer softvera radi, ali sa drugačijom filozofijom dizajna.
- Programer softvera obično započinje proces razvoja aplikacije tako što razgovara sa klijentima i otkriva koje su njihove softverske potrebe.
- Softverski inženjeri imaju širu sliku od programera softvera, što znači da su više fokusirani na celokupnu strukturu softverskog sistema - njegov dizajn i eliminisanje tehničkog duga.
- Softverski inženjeri su zainteresovani za rešavanje trenutnih problema uz minimiziranje kompromisa u odnosu na druge delove sistema i njegovu arhitekturu.

## ✓ Poglavlje 2

# Postajanje inženjera softvera

## KAKO SE POSTAJE INŽENJER SOFTVERA?

*Postoji sedam osnovnih koraka u cilju postajanja softverskog inženjera*

Tehnologija je pomogla u rešavanju mnogih problema našeg vremena, a softverski inženjeri su pokretačka snaga ove inovacije koja pomaže boljem svetu. Oblast softverskog inženjerstva nudi neke od najplaćenijih poslova danas, međutim, to zahteva mnogo strpljenja, posvećenosti i želje za stalnim usavršavanjem.

U nastavku su navedeni postepeni koraci koje je potrebno proći kako bi osoba postala softverski inženjer:

- **Sticanje diplome iz računarstva** - Tradicionalni način bavljenja softverskim inženjerstvom jeste sticanje obrazovanja u srodnim oblastima računarskih nauka. Na ovaj način studenti detaljno ispituju teorijske aspekte programskih jezika, što im pomaže da razumeju principe softverskog inženjerstva i steknu širu sliku.
- **Sticanje relevantnih veština i pouzdanog znanja o programiranju** - Diploma iz ove oblasti nije dovoljna ukoliko se ne razume u potpunosti programski jezik koji se koristi za kreiranje softvera. Poželjno je izabrati programski jezik na osnovu preferencija i ciljeva i vežbati kreiranje aplikacija na tom jeziku.
- **Detaljno razumevanje struktura podataka i algoritama** - Softverski inženjer bi trebalo da bude vešt i da efikasno ponudi optimizovan odgovor za specifične softverske probleme. Zbog toga mora imati veštine za donošenje razumnih odluka o pitanjima kao što je tačan format podataka koji najviše odgovara datim okolnostima.
- **Poboljšanje i usavršavanje veština programiranja na osnovu novih tehnoloških trendova** - Obrazovanje softverskih inženjera ne prestaje nakon sticanja diplome. Sticanje diplome zaista znači čvrstu osnovu glavnih principa softverskog inženjerstva, međutim, neophodno je nastaviti sa vežbanjem i važno je biti u toku sa promenama i napretkom softvera. Trebalo bi se upoznati sa alatima i tehnologijom koji preovlađuju na terenu.
- **Sticanje praktičnog iskustva kroz rad na stvarnim projektima i kreiranje softverskih programa od nule** - Idealan softverski inženjer razume kako da primeni osnove programiranja i teme koje se proučavaju u udžbenicima da bi napravio neke izvanredne aplikacije. Zbog toga je dobro iskoristiti sve što je naučeno i vežbati kreiranje aplikacija ili projekata.
- **Jačanje znanja u industriji i sticanje praktičnog iskustva kroz stažiranje** - Stažiranje nudi praktičnu obuku i industrijsko izlaganje radu na odgovarajućim programima ili aplikacijama. Stažisti takođe razumeju timski rad, što doprinosi njihovom prelasku u poslovnu kulturu.
- **Potragaza mogućnostima zaposlenja** - Nakon sticanja potrebnih veština i dovoljno praktičnog iskustva, poslednji korak je traženje mogućnosti za zaposlenje.

# LIČNE OSOBINE USPEŠNOG INŽENJERA SOFTVERA

*Sve lične osobine podeljene su na unutrašnje i spoljašnje osobine*

Analiza urađena anketiranjem najiskusnijih Microsoft-ovih inženjera sa preko 25 godina radnog iskustva u softverskom inženjerstvu donela je grupu od 53 osobine veoma uspešnih inženjera softvera. Sve ove osobine podeljene su u četiri oblasti: unutrašnje - *osobine ličnosti* softverskog inženjera i *spособnosti da donosi* efektivne *odluke*, kao i spoljašnje - *osobine uticaja* koji uspešni softverski inženjeri imaju *na ljude i proizvode*.

Donošenje efikasnih odluka uključivalo je prepoznavanje situacija, kao i poznavanje alternativnih pravaca delovanja, verovatnih ishoda i vrednosti ishoda. Spoljni atributi su se fokusirali na sjajne inženjere koji primenjuju svoju emocionalnu inteligenciju i modele donošenja odluka na svoj softver, svoje saigrače i potencijalno milione korisnika i zainteresovanih strana kojima služe kroz svoje napore u softverskom inženjeringu.

| Lične osobine                   |                      | Donošenje odluka               | Timski rad                       |                             | Softverski proizvod        |
|---------------------------------|----------------------|--------------------------------|----------------------------------|-----------------------------|----------------------------|
| Ima želju za usavršavanjem      | Preuzimanje rizika   | Zna ljude i organizaciju       | Stvara zajednički kontekst       | Postavlja izazove           | Elegantan                  |
| Strast prema poslu              | Prilagodljivost      | Menja svoje ponašanje          | Stvara zajednički uspeh          | Učestvuje u radu            | Kreativan                  |
| Otvoren um                      | Samopouzdanje        | Sposoban da vidi i drvo i šumu | Stvara bezbedno okruženje        | Ostvaruje očekivanja        | Anticipira potrebe         |
| Odluke su zasnovane na podacima | Samosvesnost         | Upravlja složenostima          | Iskenost                         | Ima dobru reputaciju        | Pravi kompromise           |
| Sistematičnost                  | Usaglašenost         | Ima znanje o tehničkom domenu  | Uključuje različite aspekte      | Realan je                   | Posvećuje pažnju detaljima |
| Produktivnost                   | Izvršavanje zadataka | Ima znanje o kupcima i biznisu | Lepo ponašanje                   | Ima sposobnost pregovaranja | Softver je dobro opremljen |
| Upornost                        | Ponos                | Poznavanje alata               | Dobijanje konteksta              | Šarmantan                   | Spreman je da se razvija   |
| Vredni rad                      | Kreativnost          | Pozna je inženjerske procese   | Ne preuzima zasluge samo za sebe | Pita za pomoć               | Softver je dugoročan       |
| Radoznalost                     | Fokusiranost         | Stanja modela i izlazi         | Mentor kolegama                  |                             | Pažljivo kreiran softver   |

Slika 2.1 Tabela sa osobinama odličnog inženjera softvera [Izvor: autor]



## ▼ Poglavlje 3

# Uloge u softverskom inženjerstvu

## LISTA ULOGA U SOFTVERSKOM INŽENJERSTVU

*Postoji mnogo različitih titula koje softverski inženjer može imati*

Iako je centralna uloga softverskog inženjera pravljenje softvera, postoji mnogo različitih titula koje softverski inženjer može imati. U nastavku su navedene neke od njih.

- Front-End inženjer

**Front-end** inženjer koristi alate i tehnologije kao što su JavaScript, CSS i HTML za izgradnju korisničkih interfejsa (UI). Korisnički interfejsi obuhvataju vizuelne elemente koje korisnik vidi kada poseti veb lokaciju ili otvori aplikaciju. Ovo uključuje font, izgled, dugmad i ukupan izgled bilo kog softvera.

Programeri nazivaju ove komponente i bilo koji grafički interfejs kao klijentsku stranu, jer korisnik vidi softver sa kojim direktno komunicira.

- Back-End inženjer

**Back-end** inženjeri rade na strani servera. Nije iznenađujuće da se ovde inženjeri bave svim zadacima u vezi sa serverom, uključujući upravljanje bazom podataka. Back-end razvoj generalno podrazumeva upotrebu specifičnih pozadinskih programskih jezika kao što su Python, Rubi ili Java.

- Full Stack inženjer

**Full-stack** inženjeri su odgovorni i za **front-end** i **back-end** razvoj. Imaju širok spektar veština iz oba tipa posla, što im daje veliku prednost ukoliko je poslodavcu potreban neko ko može sve to, ili u ovom slučaju, razvoj kompletnog steka. Angažovanje full stack inženjera za poslovanje znači imati tim za razvoj softvera u kome svi na brodu znaju šta se dešava.

- Inženjer igara

Inženjeri igara prave igre. Razvoj iz ovog ugla prevazilazi standardne alate za razvoj softvera. Okviri i alati otvorenog koda kao što su Unity i Blender omogućavaju inženjerima igara da modeliraju fiziku, manipulišu mehanikom igre i kreiraju složenu 2D i 3D grafiku.

## ULOGU U SOFTVERSKOM INŽENJERSTVU

*U zavisnosti od oblasti kojom se bave, inženjeri softvera koriste različite programske jezike*

- Programer mobilnih aplikacija

Programeri mobilnih aplikacija prave softver za mobilne uređaje, u rasponu od pametnih telefona, tableta, pa čak i konzola za igre, e čitača i bilo kog drugog uređaja sa dragocenom prodavnicom aplikacija. Razvoj mobilnih aplikacija zavisi od znanja i parametara specifičnih za domen. Na primer, Android i iOS imaju programske jezike specifične za domen. Android koristi Kotlin i Java, dok iOS koristi Swift. Za optimalne performanse, programeri mobilnih aplikacija moraju da se pridržavaju ovih parametara da bi razvili aplikacije. Programeri mobilnih aplikacija takođe moraju da uzmu u obzir različite faktore oblika kao što su ograničenja veličine ekrana, kao i smanjena memorija i procesorska snaga u mobilnom telefonu u odnosu na računar.

- Inženjer podataka

Inženjeri podataka ne samo da se staraju za prikupljanje podataka za poslovanje, već i da su podaci tačni i pouzdani za upotrebu. Podaci su ključni deo svakog poslovanja. Oni se mogu koristiti kako bi se pronašle najbolje prakse za optimizaciju prihoda, poboljšao životni ciklus razvoja softvera i poboljšali bezbednosni protokoli. Inženjeri podataka verovatno imaju jedan ili dva pozadinska programska jezika u svom arsenalu kao što su Python ili Java.

- DevOps inženjer

„DevOps“ proizilazi iz razvoja i operacija. DevOps inženjeri su specijalizovani za isporuku softvera velikom brzinom. Oni su krstaši u agilnim razvojnim timovima gde je brza isporuka softvera uvek krajnji cilj. Da bi postigli takav cilj, DevOps inženjeri moraju da kombinuju svoje znanje o razvoju softvera sa kulturnim strategijama i sistemskim metodologijama u cilju brzog izdavanja i primene koda.

- Inženjer ugrađenih sistema i inženjer bezbednosti

Inženjeri ugrađenih sistema moraju da se nose sa jedinstvenim izazovima kao što su ograničenja čvrste memorije za uređaj. Inženjeri bezbednosti štite vaše mreže i sisteme od sajber napada i gubitka podataka.

## ✓ Poglavlje 4

# Put karijere softverskog inženjera

## KARIJERA SOFTVERSKOG INŽENJERA

### *Napredovanje u karijeri softverskih inženjera je uglavnom linearno*

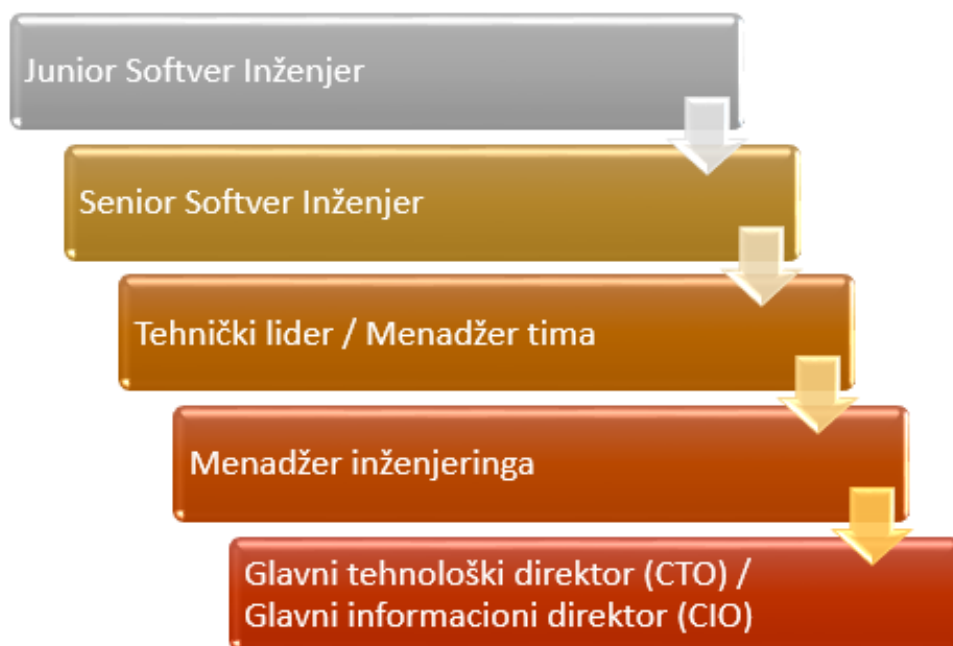
Napredovanje u karijeri softverskih inženjera je uglavnom linearno. Što više iskustva softverski inženjer ima, više je sposoban da napreduje u karijeri. Međutim, neki softverski inženjeri odlučuju da skrenu sa putanje od dna ka vrhu. Kada steknu dovoljno iskustva i počnu da veruju u kvalitet svog rada mogu da odluče da postanu slobodni. U ovom slučaju, oni mogu da rade po sopstvenom rasporedu i imaju određenu fleksibilnost u projektima na kojima odluče da rade. S druge strane, softverski inženjeri koji odluče da rade u okviru strukture kompanije, slede put koji odražava sledeće:

- Junior softver inženjer - ima osnovne veštine bilo kog softverskog inženjera za pravljenje, pokretanje i otklanjanje grešaka u aplikacijama. Njegov skup veština uključuje praktično znanje i iskustvo sa programskim jezicima, operativnim sistemima, algoritmima i bazama podataka. Junior softver inženjer ima između nula i tri godine iskustva.
- Senior softver inženjer - Nakon tri do šest godina iskustva u softverskoj industriji, inženjer može evoluirati u senior softver inženjera. Na ovom nivou, oni imaju napredno znanje o kodu i mogu bez problema da obučavaju druge softverske inženjere. Njihove odgovornosti mogu uključivati nadgledanje projekata razvoja softvera i obučavanje inženjera.
- Tehnički lider / menadžer tima - Neko sa više od šest godina iskustva u softverskom inženjerstvu ima mnoštvo mogućnosti za posao. Sa ovim iskustvom, softverski inženjer može postati tehnički lider ili menadžer tima. U ovim ulogama, softverski inženjer upravlja timom u izgradnji složenih softverskih rešenja. Tehnički lider, odnosno menadžer tima direktno izveštava zainteresovane strane kompanije sa ažuriranjima projekta i traži doprinos u donošenju odluka. Veštine za ovu ulogu uključuju projektovanje sistema, upravljanje projektima i naprednu arhitekturu softvera.
- Direktor inženjeringa - Nakon devet ili deset godina softverski inženjeri su u stanju da preuzmu vodeću ulogu. Ovo može doći u obliku menadžera inženjeringa ili potpredsednika inženjeringa. Sa veštinama napredne softverske arhitekture i projektovanja sistema, menadžeri su zaduženi za veći deo tehničkih procesa koji stoje iza razvoja proizvoda. Razmišljanje o proizvodu i upravljanje procesima su primarne odgovornosti ove tehničke liderske uloge.
- Glavni tehnološki direktor (CTO) / Glavni informacioni direktor (CIO) - CTO i CIO imaju najmanje 13 godina iskustva u oblasti softverskog inženjeringa. Obe uloge su izvršne po prirodi i podrazumevaju upravljanje tehnološkim potrebama cele organizacije. CTO i CIO su opremljeni i strateškim veštinama i veštinama za nadgledanje istraživanja i razvoja

zajedno sa upotrebom i unapređenjem tehnologije i softverskih proizvoda za organizaciju i njene klijente.

## PRIKAZ PUTA KARIJERE INŽENJERA SOFTVERA

*Grafički prikaz puta karijere inženjera softvera*



Slika 4.1 Put karijere inženjera softvera [Izvor: autor]

## ▼ Poglavlje 5

# Promocija inženjera softvera u više titule

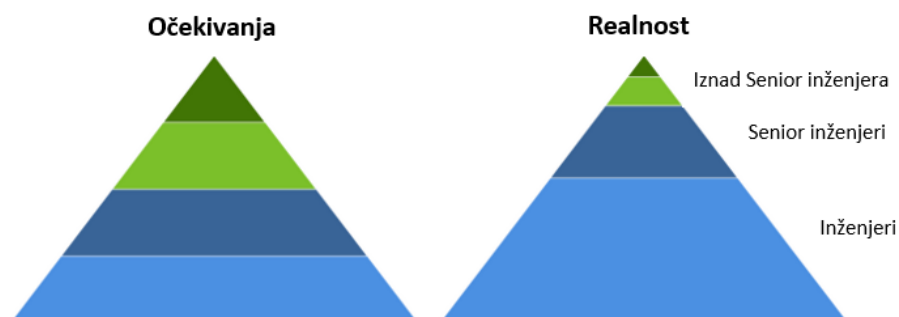
## KAKO DO PROMOCIJE INŽENJERA SOFTVERA?

*Kako vreme prolazi promocija postaje glavni cilj za mnoge inženjere softvera*

Pre ili kasnije promocije postaju osetljiva tema za softverske inženjere. Kada se pridružuju kompaniji, mali broj programera ima ovo na umu - s pravom, jer se fokusiraju na to da dobiju posao. Ali kako vreme prolazi, i kako sve više i više ljudi napreduje oko njih, promocije postaju glavni cilj za mnoge. U nastavku se nalaze saveti o aktivnostima i ponašanjima koji vode do napredovanja u karijeri.

- **Pripremiti se** - Kada softverski inženjer shvati da je zainteresovan za unapređenje, trebalo bi da počne sa prikupljanjem informacija o mogućnostima i procenom koliko je promocija realna za njega. Koraci koje bi trebalo preduzeti u ovoj fazi su:
  - shvatiti proces promocije u kompaniji,
  - proceniti sebe,
  - pobrinuti se o tome da menadžer bude saglasan sa željenom promenom,
  - biti realan vezano za promocije iznad nivoa senior.

Kako se kreće na lestvici karijere, postaje sve teže i teže doći do sledećeg nivoa. Izazov na nivou programera je sličan kao i sa menadžmentom. Prelazak od menadžera tima do direktora obično je slično težak kao što je i prelazak od senior inženjera do tehničkog lidera.



Slika 5.1 Očekivanja vs Realnost u napredovanju inženjera softvera [Izvor: <https://blog.pragmaticengineer.com/software-engineering-promotions/>]

# SAVETI ZA PROMOCIJU INŽENJERA SOFTVERA

*U sekciji su navedeni saveti koji vode do napredovanja u karijeri*

- Postaviti fokus na promociju - Nakon razumevanja procesa promocije, procene sebe i pridobijanja svog menadžera, vreme je za fokusiranje. Potrebno je:
  - postaviti ciljeve tako da se postignu kompetence koje nedostaju za rad na višoj poziciji,
  - ponašati se i preduzeti odgovornosti kao da je već preuzeta viša pozicija,
  - voditi evidenciju svojih dostignuća i uticaja.
- Tražiti pomoć i česte povratne informacije - rad na unapređenju nije lak, ponekad može trajati i predugo tako da je potrebno s vremena na vreme proveriti povratne informacije od nadređenih i da li je promocija i dalje zajednički cilj. Aktivnosti u ovom koraku su:
  - naći mentora,
  - tražiti redovne i jasne povratne informacije.
- Ući u posao - Ukoliko je i menadžeru i mentoru jasno dato do znanja da se radi na unapređenju, potrebno je uključiti se u posao. U mesecima koji prethode nominaciji, potrebno je udvostručiti rad na višem nivou, obavljati zadatke i pomagati drugima. Glavni saveti:
  - Ne raditi na svom napredovanju kroz sputavanje svojih kolega jer to može biti kratkoročno i imati vrlo negativan uticaj na buduću karijeru.
  - Ne ostavljati aktivnosti iz trenutnog posla olako iako je promocija već na vidiku - završavanje zaostalog posla je profesionalno ponašanje i veoma pozitivno utiče na imidž zaposlenog.
- Ostati prizeman
  - Ne verovati nikome ko obećava „sigurno unapređenje“.
  - Ne dozvoliti da promocija bude jedini cilj na poslu.
  - Shvatiti da promocija nije jedini način za dobijanje pozitivnih povratnih informacija.
  - Biti strpljiv i pozitivan jer je proces unapređenja dugotrajan.
- Pomagati drugima - Promovisanje u kompaniji nikada nije lako dostignuće. Kada se uspe u tome, lepo je razmisliti o vraćanju i uzimanju nekoga pod svoje okrilje, pomažući mu da raste. Čak i ako još uvek nije došlo do unapređenja, ukoliko postoji mogućnost trebalo bi pomoći drugima da rastu i dobiju priznanje za njihov rad. Kroz pristupačnost i uslužnost drugima nauči se više, stižu se saveznici, a karijera je na ovaj način daleko ugodnije putovanje.

## ▼ Poglavlje 6

# Najbolje opcije za karijeru u razvoju softvera

## TRENUTNO NAJPOŽELJNIJE OPCIJE ZA KARIJERU SOFTVERSKOG INŽENJERA

*Prema američkom Birou za statistiku rada, potreba za poslovima u okviru razvoja softvera će rasti za 22% od 2020. do 2030. godine*

Skoro svaki aspekt našeg života je protkan tehnologijom, što znači da će se eksponencijalni rast potražnje za IT poslovima nastaviti i u ovoj deceniji. Usred ekspanzije svetske mreže, računarstva u oblaku, velikog skupa podataka, softvera, mobilnih aplikacija, društvenih medija i igara, i velike i male kompanije moraju da se oslone na kvalifikovane i visoko specijalizovane IT stručnjake.

Prema američkom Birou za statistiku rada, potreba za poslovima u okviru razvoja softvera će rasti za 22% od 2020. do 2030. godine, mnogo brže od proseka za druge profesije. S tim u vezi, u nastavku se nalazi lista od sedam trenutno najpoželjnijih opcija za karijeru softverskog inženjera.

- Inženjer mašinskog učenja

Inženjeri mašinskog učenja (ML) su odgovorni za izgradnju i održavanje algoritama koji napajaju sisteme veštačke inteligencije. Ovi inženjeri često rade sa inženjerima podataka i statističarima kako bi razvili algoritme i strukture modela koji mogu da uče iz velikih količina podataka. ML inženjeri takođe koriste ove algoritme za poboljšanje postojećih proizvoda i usluga predviđanjem ponašanja kupaca. Inženjerstvo mašinskog učenja je jedan od najplaćenijih poslova softverskog inženjeringa jer na tržištu postoji veliki nedostatak talenata, a rad može da stvori vredne AI sisteme koji će pokrenuti neverovatnu transformaciju u budućnosti.

- DevOps inženjer

DevOps inženjeri razvijaju i implementiraju procese koji obezbeđuju nesmetan protok informacija između programera softvera i administratora sistema. DevOps inženjering ima veliku potražnju i malu ponudu. DevOps inženjeri imaju iskustvo u programiranju i jezicima za pisanje skripti i platformama zasnovanim na oblaku kao što su Amazon Veb Servisi ili Azure i imaju snažne analitičke veštine i veštine rešavanja problema.

# NAJPOŽELJNIJE OPCIJE ZA KARIJERU SOFTVERSKOG INŽENJERA

## *Inženjer podataka, android inženjer i inženjer računarstva u oblaku*

- Inženjer podataka

Sposobnost inženjera podataka da analiziraju složene skupove podataka kako bi identifikovali trendove i obrasce čini ovu profesiju jednim od najcenjenijih poslova softverskog inženjeringa. Inženjeri podataka koriste informacije iz analiziranih podataka kako bi pomogli organizacijama da donesu bolje odluke, reše probleme i poboljšaju poslovanje. Kako se podaci o ponašanju potrošača brzo razvijaju, potražnja za inženjerima podataka će samo rasti. Pored toga, kompanije će postati svesnije da su im potrebni pouzdani podaci da bi razumeli kako klijenti komuniciraju sa njihovim poslovanjem. Nauka o podacima je jedan od najplaćenijih poslova softverskog inženjeringa jer zahteva jaku pozadinu u matematici i statistici. Takođe je potrebno iskustvo sa programskim jezicima kao što su Python i R.

- Android inženjer

Android inženjeri dizajniraju i razvijaju softver za Android mobilne uređaje. Ovi inženjeri takođe mogu pomoći u otklanjanju grešaka ili rešavanju problema koda. Android je najpopularniji mobilni operativni sistem na svetu, tako da postoji velika potražnja za veštım Android inženjerima.

- Inženjer računarstva u oblaku

Inženjeri računarstva u oblaku su odgovorni za razvoj, održavanje i proširenje infrastrukture i usluga u oblaku. Oni rade sa korisničkim servisom i tehničkim timovima kako bi isporučili visokokvalitetna rešenja u oblaku. Ovaj posao zahteva širok spektar tehničkih veština, uključujući Linux i razvoj softvera otvorenog koda, bezbednosni inženjering, administraciju baze podataka, računarstvo i mrežnu infrastrukturu.

# POŽELJNE OPCIJE ZA KARIJERU SOFTVERSKOG INŽENJERA

## *Full stack i veb inženjeri takođe spadaju u trenutno najpoželjnije poslove vezane za softversko inženjerstvo*

- Full stack inženjer

Full stack inženjer može da radi i **front-end** i **back-end** delove softverskog projekta. Da bi softverski inženjer bio kvalifikovan za ovu ulogu mora imati iskustvo sa **front-end** razvojnim tehnologijama kao što su HTML, CSS i JavaScript. Takođe, potrebno je i razumevanju pozadinskih razvojnih jezika, kao što su PHP, Ruby on Rails i Node.js. Full stack inženjeri moraju imati iskustvo sa više programskih jezika, okvira i platformi.

- Veb inženjer



Veb razvoj je visoko plaćen posao softverskog inženjeringa zbog stalno rastuće internet ekonomije. Preduzeća sada shvataju važnost prisustva na mreži zbog čega su im potrebni vešti inženjeri za kreiranje inovativnih i vizuelno privlačnih veb lokacija. Veb inženjer odgovoran je za kreiranje i upravljanje veb lokacijama i veb aplikacijama za preduzeća. Posao zahteva jake veštine kodiranja i programiranja i iskustva sa različitim okvirima za veb razvoj, kao što su React, Vue.js, Angular, JKueri i sl.

## ▼ Poglavlje 7

### Vežba - Pokazni primeri

#### ZADUŽENJA JUNIOR SOFTVERSKOG INŽENJERA

##### *Lista zaduženja junior softverskog inženjera kroz primer izrade softvera za rad biblioteke*

U okviru dva časa vežbi u trajanju od 90 minuta studentima treba prikazati put razvoja karijere softverskog inženjera i koja zaduženja ima kroz svaku stavku u napredovanju u karijeri. Pokazni primer sa detaljnim zaduženjima dat je za junior softverskog inženjera u izradi softvera za rad biblioteke, dok su za ostale uloge prikazana samo zaduženja.

Informacioni sistem za rad biblioteke sastoji se iz dva dela (web aplikacije za upravljanje bibliotekom i mobilne aplikacije za klijente) pa junior softverski inženjer ima sledeće zadatke:

##### 1. Frontend:

- 1.1. Kreiranje stranica na web aplikaciji i pisanje logike njihovog povezivanja.
- 1.2. Kreiranje ekrana na mobilnoj aplikaciji i pisanje logike njihovog povezivanja.
- 1.3. Kreiranje zahteva i pozivanje krajnjih tačaka web servisa za dobijanje podataka.
- 1.4. Pisanje logike obrade podataka dobijenih od web servisa i njihovo prikazivanje na odgovarajuće forme.
- 1.5. Jedinično testiranje napisanih funkcija.
- 1.6. Pisanje dokumentacije za napisani kod.
- 1.7. Refaktorisanje koda nakon pregleda od strane senior inženjera.
- 1.8. Praćenje integracije delova sistema u svrhu usavršavanja i dobijanja iskustva.

##### 2. Backend

- 2.1. Pregled šeme baze podataka i upoznavanje za arhitekturom softvera
- 2.2. Pisanje CRUD operacija za izdvojene resurse.
- 2.3. Pisanje logike obrade ulaznih podataka i njihove validacije.
- 2.4. Kreiranje DTO (*data transfer object*) objekata za odgovore web servisa.
- 2.5. Jedinično testiranje napisanih metoda iz CRUD-a za sve resurse
- 2.6. Pisanje dokumentacije za napisani kod.
- 2.7. Refaktorisanje koda nakon pregleda od strane senior inženjera.
- 2.8. Praćenje integracije delova sistema u svrhu usavršavanja i dobijanja iskustva.

# JUNIOR I SENIOR SOFTVER INŽENJER

## *Detaljno objašnjenje zaduženja junior i senior softver inženjera*

**Junior softver inženjer** (0-3 godine iskustva) zaduženja:

- Pisanje i otklanjanje grešaka koda
- Rešavanje problema sa softverom
- Bliska saradnja sa programerima na poboljšanju funkcionalnosti proizvoda
- Razvijanje inovativnih rešenja
- Prisustvovanje sastancima programera
- Učestvovanje u pregledima koda i aktivnostima obezbeđenja kvaliteta
- Učestvovanje u diskusijama o proceni sa timom proizvoda
- Neprestano usavršavanje veština kodiranja

**Senior softver inženjer** (6+ godina iskustva) - zaduženja:

- Razvoj softvera tokom celog životnog ciklusa
- Testiranje i procena tehničkih aspekata svake odluke
- Pregled kodova i procena kvaliteta koda
- Obezbeđivanje pravovremenog rešenja za probleme
- Razvijanje koda na strani servera za veb-bazirane aplikacije velikog obima
- Kreiranje i razvijanje veb korisničkog interfejsa
- Implementacija analitike i rutina praćenja
- Izvršavanje analitike i A/B testiranja

# MENADŽER IT TIMA I IT DIREKTOR

## *Detaljno objašnjenje zaduženja menadžera i direktora IT timova*

- **Menadžer IT tima** (8+ godina iskustva) - zaduženja:
  - Koordinacija i nadgledanje IT zaposlenih
  - Rukovanje godišnjim budžetom i obezbeđivanje isplativosti IT odeljenja
  - Upravljanje i koordinacija svih pitanja vezanih za IT, uključujući podatke, bežične komunikacije, mobilnu tehnologiju, telefoniju, bezbednost i računarski hardver i softver
  - Rešavanje problema vezanih za hardver, softver i probleme sa mrežom
  - Istraživanje i preporuka prilikom nabavke mrežnog hardvera i softvera
  - Nadgledanje korišćenja i performansi mreže, sprovođenje procedure za optimizaciju mreže, pouzdanost i dostupnost
  - Identifikacija problematičnih područja i pravovremena primena strateških rešenja
  - Izgradnja dugoročnih odnosa sa spoljnim dobavljačima za IT proizvode i usluge
- **IT direktor** (10+ godina iskustva) - zaduženja:
  - Nadgledati sve tehnološke operacije i funkcije i proceniti ih u skladu sa postavljenim ciljevima

- Osmisliti i uspostaviti IT politike i sisteme za podršku implementaciji strategija postavljenih od strane višeg menadžmenta
- Analiza poslovnih zahteva svih odeljenja u cilju utvrđivanja njihove tehnološke potrebe
- Nabavka efikasne i isplative tehnološke opreme i softvera
- Provera upotrebe tehnološke opreme i softvera kako bi se obezbedila funkcionalnost i efikasnost
- Identifikacija potrebe za nadogradnjom, konfiguracijom ili novim sistemima i prijava višem rukovodstvu
- Koordinacija tima IT menadžera i nadgledanje drugih IT profesionalaca
- Kontrola budžeta i izveštaj o rashodima
- Izgradnja dugoročnih odnosa sa prodavcima

## GLAVNI TEHNOLOŠKI DIREKTOR I GLAVNI INFORMACIONI DIREKTOR

### *Detaljno objašnjenje zaduženja CTO-a i CIO-a*

- **Glavni tehnološki direktor (CTO)** (10+ godina iskustva) - zaduženja:
  - Vođenje strategije za tehnološke platforme, partnerstva i spoljne odnose
  - Predstavljanje tehnološke agende na sastancima osoblja i prilikom donošenja odluka o zapošljavanju
  - Identifikacija, upoređivanje, biranje i implementacija tehnoloških rešenja kako bi se zadovoljile trenutne i buduće potrebe
  - Razvijanje, praćenje i kontrolisanje razvoja godišnjih operativnih i kapitalnih budžeta za nabavku, zapošljavanje i poslovanje
  - Upravljanje i optimizacija infrastrukturnih sredstava u cilju zadovoljenja internih finansijskih ciljeva
  - Kreiranje sveobuhvatnih tehnoloških standarda i praksi i obezbeđivanje poštovanja
  - Objedinjavanje tehnoloških platformi i kreiranje planova za svaku
  - Praćenje, analiza i nadgledanje metrike performansi tehnologija
  - Nadgledanje dizajna svih sistema i promena u arhitekturi sistema
  - Biti u toku sa novim trendovima i najboljim praksama u tehnološkom okruženju
  - Preuzimanje inicijative u misaonom liderstvu, inovacijama i kreativnosti
  - Bliska saradnja sa marketingom, razvojem poslovanja i operacijama u cilju definisanja i isporučivanja novih proizvoda i poboljšanja
- **Glavni informacioni direktor (CIO)** (10+ godina iskustva)- zaduženja:
  - Razvijanje strategija IT sektora
  - Sprovođenje novih strategija
  - Konstantan rad na optimizaciji postojećih strategija
  - Postavljanje KPI-a za IT sektor
  - Redovno merenje KPI-a
  - Zapošljavanje novih ključnih uloga za svoj sektor
  - Pregled i odobravanje kupovina sektora
  - Određivanje budžeta sektora
  - Planiranje i nadgledanje projekata sektora

## ▼ Poglavlje 8

### Vežba - zadaci

#### ZADACI ZA INDIVIDUALNI RAD STUDENATA

##### *Zadaci za samostalno rešavanje na vežbi i kod kuće*

Posle predavanja a pre časova vežbanja (održavaju se dva dana kasnije), poželjno je da pokušate da rešite neke od ovih zadataka radom kod kuće i da rezultate pošaljete mejlom, preko Zimbre, saradniku koji drži vežbe, najkasnije jedan sat pre održavanja vežbi. Ako neki zadatak nije urađen pre ili za vreme vežbi, preporučuje se studentu da ih uradi posle vežbi, kod kuće.

Obim zadatka može da zahteva najviše 30 do 40 minuta rada studenta. Svaki student mora da ima poseban zadatak, a na temu koja se ovde definiše. Studenti koji pošalju identična ili vrlo bliska rešenja, neće im ta rešenjabiti prihvaćena, tj. dobiće 0 poena.

Tekst domaćeg zadatka:

Iz liste uloga u softverskom inženjerstvu (Objekat učenja 3) izaberite, svako za sebe, svoju željenu ulogu nakon završenih studija i objasnite zbog čega sebe vidite upravo u toj oblasti softverskog inženjerstva. Takođe, navedite ciljani nivo do koga želite da stignete - senior softver inženjer, tehnički lider ili viši menadžment, i objasnite korake u napredovanju karijere na sopstvenom primeru.

## ▼ Poglavlje 9

# Zaključak

## ZAKLJUČAK

### *Sumiranje stečenih znanja*

- Programer obično započinje proces pravljenja softverskih aplikacija tako što razgovara sa klijentima i otkriva šta su njihove softverske potrebe.
- Programer uzima funkcionalnu specifikaciju i isporučuje potreban kod unutar strogih parametara, u suštini završavajući zadatak izolovano.
- Softverski inženjeri moraju biti više fokusirani na dizajn strukture i eliminisanje tehničkog duga.
- Programer softvera teži da se fokusira na rešavanje problema jednog korisnika ili grupe korisnika, dok će se inženjer fokusirati na probleme svih korisnika.
- Koraci koje je potrebno proći kako bi osoba postala softverski inženjer su: Sticanje diplome iz računarstva, Sticanje relevantnih veština i pouzdanog znanja o programiranju, Detaljno razumevanje struktura podataka i algoritama, Poboljšanje i usavršavanje veština programiranja na osnovu novih tehnoloških trendova, Sticanje praktičnog iskustva kroz rad na stvarnim projektima i kreiranje softverskih programa od nule, Jačanje znanja u industriji i sticanje praktičnog iskustva kroz stažiranje i Potraga za mogućnostima zaposlenja.
- Napredovanje u karijeri softverskih inženjera je uglavnom linearno i kreće se sledećim putem: Junior softver inženjer - Senior Softver inženjer - Tehnički lider / menadžer tima - Direktor inženjeringa - Glavni tehnološki direktor (CTO) / Glavni informacioni direktor (CIO).
- Neki od saveta za dostizanje promocije u softverskom inženjerstvu su: Pripremiti se, Postaviti fokus na promociju, Tražiti pomoć i česte povratne informacije, Ući u posao, Ostati prizeman, Pomagati drugima.
- Prema američkom Birou za statistiku rada, potreba za poslovima u okviru razvoja softvera će rasti za 22% od 2020. do 2030. godine, mnogo brže od proseka za druge profesije.
- Sedam trenutno najpoželjnijih opcija za karijeru softverskih inženjera su: Inženjer mašinskog učenja, DevOps inženjer, Inženjer podataka, Android inženjer, Inženjer računarstva u oblaku, Full stack inženjer i Veb inženjer.

## LITERATURA

### *Pogodna literatura za učenje*

#### **Obavezna literatura:**

1. 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 2-7.

**Dodatna literatura:**

- <https://devskiller.com/software-engineer-vs-software-developer/#software-developer-vs-software-engineer>
- <https://www.boardinfinity.com/blog/how-to-become-a-software-engineer/>
- Paul Luo Li, Amy J. Ko, Jiamin Zhu, What Makes A Great Software Engineer?, 37th IEEE International Conference on Software Engineering, IEEE/ACM, 2015
- <https://www.trio.dev/blog/software-engineer-career-path>
- <https://blog.pragmaticengineer.com/software-engineering-promotions/>
- <https://www.turing.com/blog/highest-paying-software-engineering-jobs-in-2022/>