



## SE322 - INŽENJERSTVO ZAHTEVA

### Modeli procesa

#### Lekcija 02

PRIRUČNIK ZA STUDENTE

# SE322 - INŽENJERSTVO ZAHTEVA

## Lekcija 02

### *MODELI PROCESA*

- ✓ Modeli procesa
- ✓ Poglavlje 1: Dobra praksa u modeliranju procesa zahteva
- ✓ Poglavlje 2: Akteri procesa
- ✓ Poglavlje 3: Podrška projektu i upravljanje
- ✓ Poglavlje 4: Vežba
- ✓ Poglavlje 5: Domaći zadatak
- ✓ Poglavlje 6: Projektni zadatak
- ✓ 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

*Šta ćemo naučiti u ovoj lekciji?*

U ovoj lekciji proučićemo modele procesa koji se koriste u inženjerstvu zahteva.

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## ✓ Poglavlje 1

# Dobra praksa u modeliranju procesa zahteva

## VIDEO PREDAVANJE ZA OBJEKAT "DOBRA PRAKSA U MODELIRANJU PROCESA ZAHTEVA"

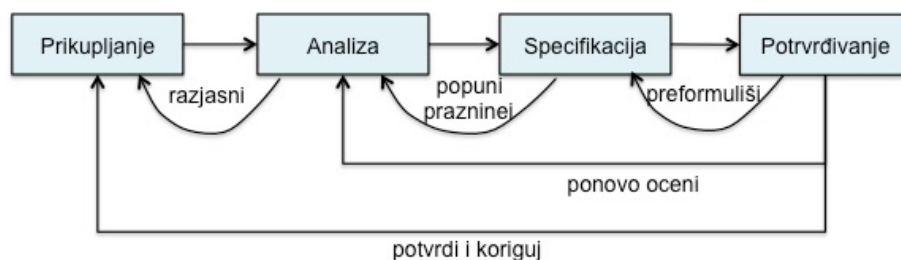
*Trajanje video snimka: 34min 16sek*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## AKTIVNOSTI PROCESA RAZVOJA ZAHTEVA

*Proces nije linearan, već ima i povratne sprege, te se iterativno realizuje.*

Na slici 1 prikazane su aktivnosti procesa razvoja zahteva. Taj proces nije linearan, već ima i povratne sprege, kao što se vidi na slici. To znači da su ove aktivnosti isprepletene, inkrementalne i iterativne. Ako obavljate ulogu analitičara poslovanja, onda ćete razgovarati sa korisnikom da bi utvrdili njegove zahteve. Onda ćete ih malo kasnije analizirati i eventualno zaključiti da treba da neke stvari da razjasnite sa korisnikom, te opet razgovarate sa njim. Sledeće što treba da uradite je da napišete u dokument o zahtevima, novi zahtev koji ste utvrdili. Ako tada utvrdite da je potrebna dodatna analiza, vi ćete je izvršiti. Kada ste spremili dokument sa zahtevima, dostavljate ga odgovarajućem akteru na potvrđivanje. Potvrđivanje može zahtevati da se neki zahtev dodatno razjasni sa korisnikom, ili da se neka analiza doradi ili da se izvrše neke preformulacije u dokumentu sa zahtevima. U slučaju agilnih projekata razvoja softvera, ovaj proces se ponavlja u svakom ciklusu.



Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 1.1 Osnovne aktivnosti procesa razvoja zahteva

Zbog različitosti projekata i načina poslovanja organizacije, tj. organizacione kulture, ne postoji samo jedan model procesa razvoja zahteva. Ipak, na slici 2 prikazan je okvir procesa razvoja zahteva koji se može primeniti u većini slučajeva. On podrazumeva da je prethodno izvršena analiza poslovnih potreba ili mogućnosti tržišta. Prikazani zadaci se uglavnom izvršavaju po navedenom numeričkom redosledu, mada se neki od njih obavljaju i iterativno. Na primer, zadatke 8,9,10 možete raditi u delovima i da idete na recenziju (zadatak 12), posle svake iteracija.

Slika 1.2 Uobičajeni zadaci u procesu razvoja zahteva

## PRIKUPLJANJE ZAHTEVA

*Primenjuju se različiti načina za prikupljanje zahteva od korisnika i drugih aktera.*

Sve vrste zahteva treba prikupiti primenom različitih tehnika. Ovde se daju neke preporuke, na osnovu prakse.

**Definiši viziju projekta i njegov okvir:** Dokument o viziji i okviru projekta sadrži poslovne zahteve za proizvod. Iskaz o viziji daje svim akterima zajedničko razumevanje u proizvodu koji treba razviti. Okvir definiše granice sistema koji treba razviti, tj. šta treba, a šta ne treba. Na osnovu ovog dokumenta se vrši ocena predloženih zahteva. Dok je vizija, po pravilu, stabilna tokom trajanja projekta razvoja, iskaz o okviru je promenljiv u svakoj iteraciji.

**Utvrđi klase korisnika i njihove karakteristike:** Utvrđi grupe korisnike sistema, koje se razlikuju po frekvenciji korišćenja sistema, svojstvima koja koriste, nivou privilegija, ili iskustvu. Opiši njih posao, tj. radne zadatke, stavove, lokaciju, i dr.

**Odaberi "šampiona proizvoda" za svaku klasu korisnika:** Odaberi pojedinaca koji je tipičan za svoju klasu korisnika. On predstavlja potrebe svoje klase i donosi odluke u njeno ime. U slučaju razvoja proizvoda za tržište, odberite pojedince iz firme koje su glavni kupci ili rade testiranje beta verzija proizvoda.

**Vodite fokus grupe s tipičnim korisnicima:** Važne su pri razvoju komercijalnih proizvoda. Najčešće one ne donose nikakve odluke.

**Radi sa predstavnicima korisnika da bi utvrdio zahteve korisnika:** Sa predstavnicima razgovarajte šta očekuju od softvera da bi uspešno obavili svoj posao i koju vrednost očekuju da postignu. Zahtevi korisnika se najčešće daju u formi slučajeva korišćenja, priča korisnika i scenarija. Utvrdite potrebnu interakciju korisnik-sistem da bi obavio svoj posao.

**Utvrđi događaje u sistemu i moguće odgovore sistema:** Sačini listu spoljnih događaja i odgovora koje sistem treba da da na njih. Postoje tri klase događaja: signalni, i poslovni događaji.

**Organizuj intervju sa akterima:** Mogu biti pojedinačni i grupni. Oni su efikasni u prikupljanju zahteva, jer su fokusirani samo na zahteve određenog aktera.

**Organizuj radionice radi prikupljanja zahteva:** Omogućavaju kolaborativni rad analitičara i kupaca, radi utvrđivanja njihovih potreba i zahteva.

**Osmatraj rad ljudi na poslu:** Napravi dijagram procesa radnog mesta.

**Raspodeli upitnike:** Korisni u slučaju velikih i distribuiranih grupa korisnika.

**Prouči raspoloživu dokumentaciju:** Prikupi potrebne informacije.

Prouči prihvatljive postojeće zahteve, kao i izveštaje o problemima.

## SPECIFIKACIJA ZAHTEVA

*Svrha specifikacije zahteva je dokumentovanje zahteva različitog tipa na jedan konzistentan, pristupačan i proverljiv način koji je razumljiv akterima.*

Svrha specifikacije zahteva je dokumentovanje zahteva različitog tipa na jedan konzistentan, pristupačan i proverljiv način koji je razumljiv akterima sistema. Poslovni zahtevi se mogu zapisati u dokumentu o viziji i okviru sistema. Zahtevi korisnika su obično izražene u formi slučajeva korišćenja ili priča korisnika. Detaljniji funkcionalni i nefunkcionalno softverski zahtevi se zapisuju u specifikaciji softverskih zahteva (**SRS-Software Requirements Apecification**) ili u alternativnom repozitorijumu pod kontrolom alata za upravljanje zahtevima. Evo preporuka dobre prakse:

- **Usvoji formular za dokumentovanje zahteva** - dokument o viziji i okviru projekta, obrazac za zapis slučajeva korišćenja i formular za specifikaciju softverskih zahteva (SRS).
- **Utvrđi izvore zahteva** - daju indikaciju o razlozima za postavljanje zahteva. Na primer, to može biti neki slučaj korišćenja ili neko poslovno pravilo. Zapis o akterima na koje se odnose ova ograničenja se koristi za komunikaciju sa njim kada dođe do promene ovih zahteva. Praćenje izvora zahteva može da se realizuje preko linka datog u vidu atributa zapisa.
- **Jedinstveno označavanje zahteva** - u skladu sa konvencijom o oznakama zahteva koju treba da donesete. Označavanje zahteva omogućava njihovo praćenje i zapisivanje njihovih promena.
- **Zapisivanje poslovnih pravila** - ona uključuju pravila poslovanja firme, zakonsku regulativu, standarde, i algoritme računanja. Poslovna pravila važe i posle i pre projekta, i tome se razlikuju od projektnih pravila. Potrebno je zapisati linkove zahteva sa poslovnim pravilima koji su uzroci postavljanja tih zahteva.
- **Specificiraj nefunkcionalne zahteva** - uključuju kvalitativna svojstva, kao što su performanse, pouzdanost, upotrebljivost, promenljivost, i dr. Oni utiču na projektna rešenja softvera, koja na osnovu njih definiše projektant softvera.

## ANALIZA ZAHTEVA

*Analiza zahteva poboljšava zahteve, otklanja greške i nedostatke u njima, a omogućava bolje razumevanje zahteva od strane svih aktera.*

**Analiza zahteva** (**requirements analysis**) poboljšava utvrđene zahteve i obezbeđuje da ih svi akteri razumeju i ukažu na eventualne greške, ispuštene zahteve i druge nedostatke. Analiza uključuje dekompoziciju uopštenih zahteva na odgovarajući nivo detalja, izradu prototipova, ocenjuje ostvarljivost, i usklađuje prioritete zahteva. Cilj je dobijanje zahteva dovoljnog kvaliteta i sa dovoljnom preciznošću koje omogućavaju menadžerima da urade realističnu procenu projekta, a tehničkim članovima razvojnog tima mogućnost projektovanja, konstrukcije i testiranja softvera.

Vrlo je preporučljivo da se pojedini zahtevi prikažu na više načina, u tekstualnoj i u vizualnoj formi, i u vidu zahteva i u vidu testova. Ovi različiti pogledi otkrivaju detalje koje ponekad samo jedan pogled ne može da otkrije. Višestruki pogledi takođe olakšavaju akterima da dođu da zajedničke vizije, tj. razumevanja šta treba da se razvije.

Analiza obuhvata sledeće aktivnosti:

- Određivanje prioriteta zahtevima - daje se prednost onim zahtevima koji omogućavaju najveću vrednost ili hitnu funkcionalnost.
- Kreiranje rečnika podataka - definicija podataka, olakšava komunikaciju
- Modeliranje zahteva - daje grafičko predstavljanje zahteva (model toka podataka, ER dijagrami, dijagrami stanja, mape dijaloga i dr.)
- Analiza interfejsa između sistema i spoljnog okruženja - to omogućava dobru usaglašenost sistema sa okruženjem.
- Dodeljivanje zahteva podsistemima - Složeni sistemi imaju podsisteme i preporučljivo je da im se dodele odgovarajući zahtevi.

## POTVRĐIVANJE ZAHTEVA

*Potvrđivanje (validation) obezbeđuje kontrolu da li su zahtevi korektni, i pokazuju karakteristike kvaliteta i zadovoljenje potreba korisnika*

Potvrđivanje (**validation**) obezbeđuje kontrolu da li su zahtevi korektni, i pokazuju karakteristike kvaliteta i zadovoljenje potreba korisnika. Ponekad zahteve treba korigovati da bi programerima bili jasniji i jednosmisleni. Ova aktivnost podrazumeva sledeće akcije:

- **Recenzija zahteva:** Recenzenti koji predstavljaju različite poglede na softver (analitičar, kupac, programer, tester) pažljivo analiziraju zapisane zahteve, modele analize, i povezane informacije, radi nalaženja grešaka. Od koristi je i preliminarne recenzija, dok su zahtevi još u fazi razvoja. Važno je obučiti članove tima za realizaciju efektivnih recenzija.
- **Testiranje zahteva:** Definisanje testova koji proveravaju da li su određeni zahtevi korisnika ostvareni
- **Definisanje kriterijuma prihvatljivosti:** Pitajte korisnika da opiše kako bi on odredio da li je neko rešenje zadovoljava njegove potrebe i da je spremno za upotrebu. U to svrhu se definišu testovi prihvatljivosti softvera zasnovani na zahtevima korisnika, koji treba da pokažu zadovoljenje i određenih nefunkcionalnih zahteva, vođenja do grešaka, kao i posedovanje odgovarajuće infrastrukture i obučenog personala.

- **Simulacija zahteva:** Postoje komercijalni alati koji omogućavaju razvojnom timu da simuliraju rad razvijenog prototipa sistema. Primenom prototipova, analitičar može, u interakciji sa korisnikom, da utvrdi prihvatljivost pojedinih rešenja. Korisnik može da potvrdi valjanost rešenja ili da izvrši izbor varijantnih rešenje, ako mu se ponude.

## REQUIREMENTS ENGINEERING PROCESS - GEORGIA TECH (VIDEO)

*Trajanje: 1:18 minuta*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**



## ▼ Poglavlje 2

# Akteri procesa

## KORISNICI SOFTVERA

*Ne postoji jedna kategorija korisnika. Korisnike treba razvrstati u više klasa korisnika.*

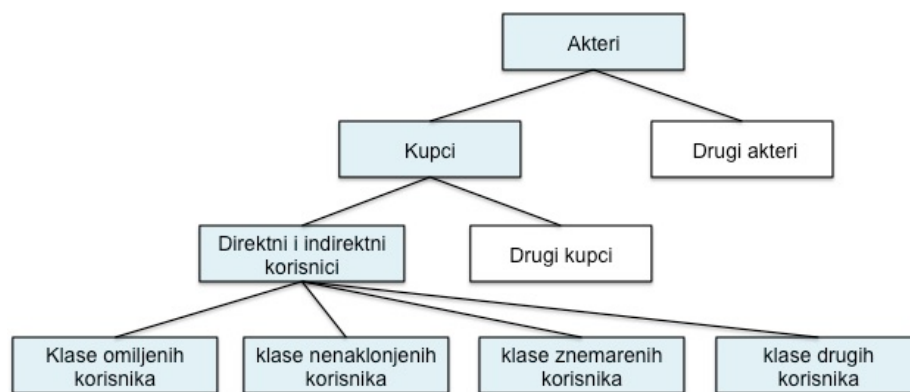
Uspeh u utvrđivanju zahteva softvera, a to znači i uspeh u razvoju softvera, prvenstveno zavisi od uključenja korisnika softvera u proces razvoja zahteva. Zbog toga, potrebno je da uradite sledeće:

- Utvrditi različite klase (kategorije) korisnika softvera koji treba da se razvije
- izaberite i radite sa pojedincima koji predstavljaju pojedine klase korisnika.
- Dogovorite ko će biti donosioci odluka u vezi zahteva u vašem projektu

Međutim, ovaj posao nije tako jednostavan kao što izgleda. Ako bukvalno usvojite sve što su vam predstavnici korisnika rekli i razvijete takav softver, vrlo je verovatno da to neće biti uspešan proizvod. I korisnici ne znaju tačno šta žele, a postoji i mogućnost da ih niste dobro razumeli. To što oni "žele" ne znači automatski da je to funkcionalnost koju softver treba da ima. Analitičar mora da prikupi znatno širi broj predloga korisnika, da ih analizira i razjasni i da ustanovi šta je stvarno potrebno da softver ima da bi korisnici mogli da uspešno rade svoj posao. U tome je vodeća odgovornost analitičara poslovanja i da to uskladi sa ostalim akterima procesa razvoja. To usklađivanje zahteva i definisanje svojstava softvera je jedan iterativan proces koji zahteva određeno vreme. U suprotnom, lako se može desiti da dobijete softver koji mora da se dorađuje, koji kasni, koji je probio budžet i izazvao nezadovoljstvo korisnika.

Ne postoji jedna kategorija korisnika. Nema monolitnog korisnika. Uvek ima više kategorija korisnika koje mi opisujemo posebnima klasama korisnika. Ko što se vidi na slici 1, klasa korisnika je podskup šireg skupa korisnika, a skup korisnika je podskup aktera procesa. Jedan pojedinac može biti deo više ovih podskupova, te i da pripada većem broju klasa korisnika. Kako se korisnici razlikuju, oni se predstavljaju različitim klasama korisnika. Te razlike se ogledaju u sledećem:

- imaju različita prava pristupa ili nivoe bezbednosti
- imaju različite zadatke na poslu u vezi korišćenja softvera
- koriste različita svojstva softvera, različite frekvencije korišćenja softvera, različite domene primene ili IT kompetencije i dr.



Izvor: Karl Wieggers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.1 Hijerarhija aktera, kupaca, korisnika i klasa korisnika softvera

## OMILJENI I NENAKLONJENI KORISNICI

*Svaka klasa korisnika ima svoj skup zahteva u skladu sa zadacima koje članovi klase treba da izvrše.*

Pri klasifikaciji korisnika prvenstveno treba gledati posao koji obavljaju korišćenjem softvera koji treba razviti, a ne zvanja, lokacija rada, nivo u kadrovskoj hijerarhiji i dr. Kada počnete da definišete slučajeve korišćenja, priče korisnika, da radite dijagrame procesa - ukazaće vam se jasnije koje sve kategorije (klase) korisnika imate u projektu.

Neke klase korisnika mogu da budu važnije od drugih. **Klasa omiljenih korisnika** (*favoured user class*) je ona čije zadovoljenje je blisko povezano sa ostvarenjem poslovnih ciljeva firme koja kupuje softver. Ovo su korisnici zbog koji se prvenstveno i razvija novi softver.

**Klasa nenaklonjenih korisnika** (*disfavored user class*) su korisnici koji ne koriste softver iz legalnih, bezbednosnih ili sigurnosnih razloga. To su korisnici koji mogu da budu i bespravni korisnici softvera, ili legalni korisnici, ali koji pokušavaju da koriste funkcije koje nisu njima namenjene i onda se suočavaju sa raznim problemima (pristup, nerazumevanje i dr.). Sa stanovišta softvera, sa korisnicima iz ove kategorije treba biti oprezan u davanju privilegija i pristupa različitim funkcijama sistema.

Koliko različitih klasa korisnika ćete koristiti i razviti, to zavisi od vas, tj. od specifičnosti vašeg softvera.

Svaka klasa korisnika ima svoj skup zahteva u skladu sa zadacima koje članovi klase treba da izvrše. Njihovi zahtevi se mogu preklapati.

Različite klase korisnika mogu imati različita očekivanja kvaliteta, kao što je način korišćenja softvera. To može da utiče na razvoj korisničkih interfejsa. Novi i povremeni korisnici pre svega žele jednostavan i razumljiv interfejs, te znači i dosta grafike. Iskusni i stalni korisnici žele pre svega efikasan interfejs, manje grafike, a više direktnog poziva funkcija koje su im potrebne, kao i manje klikova.

Korisničke klase mogu da predstavljaju i druge sistema, tj. da budu softverski agenti. Softverski agent može da za vaše potrebe obavi neki potreban posao i da vam isporuči rezultat takve obrade podataka.

Pored direktnih i indirektnih klasa korisnika, postoje i drugi mogući korisnici koje ne treba da zaboravite. Svakom od njih treba omogućiti servise koje očekuju od vašeg softvera.

## KAKO IZABRATI KLASU KORISNIKA?

*Primena organizacionih šema i poslovnih procesa može da olakša utvrđivanje klasa korisnika.*

Utvrdjivanje klasa korisnika rano u projektu pomoći će vam da od njihovih predstavnika dobijete njihove specifične zahteve. Da bi utvrdili potrebne klase korisnika, pitajte sponzora projekta ko će sve da koristi softver. Razgovarajte sa svi navedenim mogućim korisnicima softvera. Grupišite ih prema sličnosti zahteva koje imaju. Tako možete doći i do 15-tak klasa korisnika.

Pri kreiranju klasa korisnika može vam pomoći i organizaciona struktura organizacije koje treba da koristi softver. Pri analizi aktera i korisnika, obratite pažnju na sledeće:

- Organizacione jedinice (OJ) koje koriste softver
- Organizacione jedinice koje su zahvaćene poslovnim procesom
- Organizacione jedinice ili nosioci zaduženja (**roles**) koji su direktni ili indirektni korisnici sistema
- Klase korisnika koje obuhvataju više organizacionih jedinica
- Organizacione jedinice koje imaju interfejs sa akterima van organizacije

Moguće je da utvrdite više klas korisnika i u okviru samo jedne organizacione jedinice. Analizom organizacione strukture, pokušajte da utvrdite i vrstu informacija korisnika iz svake OJ.

Dokumentujte utvrđene klase korisnika i njihove karakteristike, odgovornosti i fizičku lokaciju.

Procenite obim transakcija sa sistemom koji svaka klasa korisnika može da ima. Na slici 2 prikazana je tabela klase korisnika u slučaju Chemical Tracking System, koji ćemo koristiti u ovom predmetu da bi na njemu pokazali kako se primenjuje znanje koje vam izlažemo.

Ime	Broj	OPis
Hemičar (omiljeni)	Oko 1000 u 6 zgrada	Zahtevaju hemikalije od isporučioaca i od skladišta hemikalija. Svaki hemičar koristi sistem nekoliko puta dnevno, najčešće zahtevajući info o hemikalijama i o kontejnerima koje sadrže te hemikalije, preko kojih dolaze i odlaze hemikalije iz njihove laboratorije. Hemičari pretražuju kataloge prodavaca za određene hijse strukture preuzete iz alata koji koriste i abi nacrtali te strukture.
Kupci	5	Kupci rade u odeljenju nabavke i obrađuju zahteve za kupovinom. Oni šalju naloge za kupovinu isporučiocima hemikalija. Oni malo znaju o hemiji, i treba im jednostavni način za pretraživanje kataloga. Kupci ne koriste kontejnere sistema za praćenje lekova. Svako od njih koristi sistem oko 25 puta dnevno.
Osoblje skladišta hemikalija	6 tehničara i 1 supervizor	Skladište sadrži oko 500.000 hemijskih kontejnera. Isporučuju kontejnere iz 3 skladišta, zahtevaju nove hemikalije od isporučioaca, i prat edolazak i odlazak kontejnera iz skladišta. Oni su jedini korisnici iyveštaju o stanju skladišta. Zbog velikog obima, funkcije koje oni interno kroiste moraju da budu automatizovane iefikasne.
Osoblje OJ za zdravlje i bezbednost	1 menadžer	Koristi sistem samo za dobijanje kvartalnih izveštaja koji zadovoljavaju propise korišćenja hemikalija. Menadžer povremeno yahteva promene u iyveštajima u skladu sa promenama propisa. Ove promene imaju najviši prioritet i primena mora d abude brza.

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

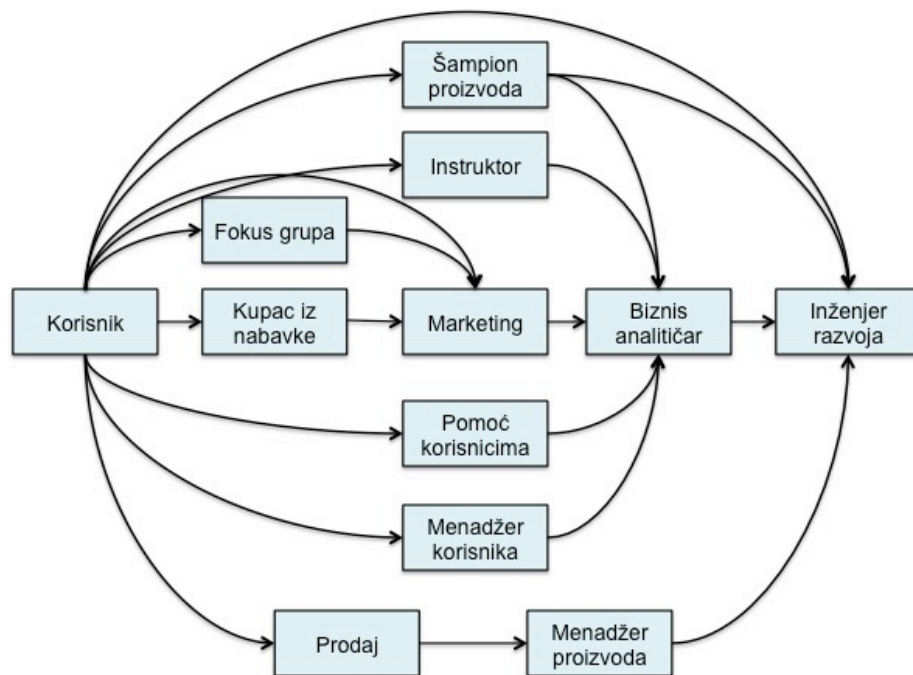
Slika 2.2 Primer klasa korisnika u slučaju ChemicalTracking System-a

## KOMUNIKACIJA SA PREDSTAVNICIMA KLASA KORISNIKA

*Neophodno je koristiti sve moguće vidove komunikacija između korisnika i inženjera razvoja. Treba biti svestan da posredništvo uvodi rizik, ali je nužno*

Neophodno je uključiti predstavnike sadašnjih korisnika sistema, a ako je sistem nov, predstavnike očekivanih prvih korisnika sistema (beta tester). U tom cilju se pozivaju predstavnici različitih klasa korisnika na fokus grupe, radi utvrđivanja njihovih potreba. One treba da uključe i iskusnije, ali i manje iskusne korisnike softvera. Ne treba da uključe samo avangardne korisnike koji traže samo nove sisteme i svojstva, jer time bi zanemarili većinu drugih, običnih korisnika. Najbolje je da inženjer razvoja, ako ima dovoljno veštine analitičara, da razgovara sa krajnjim korisnicima sistema. Ali, to je moguće samo u slučaju manjih projekata. Kod većih to nije moguće, jer ima puno korisnika. Na slici 3 prikazane su neke od mogućih komunikacija sa korisnicima sistema.

Uvođenje posrednika između kupca i inženjera razvoja uvodi rizik nerazumevanja stvarnih potreba kupaca i njegovih zahteva. Veštih biznis analitičara eliminiše taj rizik i dodaje vrednost eliminišući manje važne ili nevažne zahteve korisnika. Rizik najviše unosi aktiviranje marketinga, menadžera proizvoda, stručnjaka za domenska znanja, i drugih koji treba da predstavljaju krajnjeg korisnika. Bez obzira na prepreke, pa i na troškove, treba optimizirati predstavljanje korisnika, jer će u suprotnom, vaš softver trpeti u kvalitetu (ne odgovara zahtevima korisnika) te treba da omogućite razgovor sa ljudima koji mogu da vam obezbede najbolju informaciju.



Izvor: Karl Wiegers, Joy Beatty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.3 Komunikacije između korisnika i inženjera razvoja softvera

## ŠAMPION PROIZVODA

*Šampion proizvoda služi kao primarni interfejs između određene klase korisnika i biznis analitičara.*

Šampion proizvoda (product champion) služi kao primarni interfejs između određene klase korisnika i biznis analitičara. Idealno, oni su i krajnji korisnici, a ne neki posrednici. Šampioni proizvoda prikupljaju zahteve od drugih članova njegove klase korisnika i otklanjaju nekonzistentnosti. Na ovaj način, razvoj zahteva postaje zajednička odgovornost šampiona proizvoda i biznis analitičara, mada ovaj drugi i piše specifikaciju softverskih zahteva.

Najbolji šampioni projekta imaju jasnu viziju novog sistema. Šampion mora da bude dobar komunikator i da uživa poštovanje svojih kolega. Mora dobro da razume domen primene sistema, i radom okruženje sistema.

Šampion proizvoda postaje još efektivniji ako ima ovlašćenja da može da odlučuje u ime svoje klase korisnika. Međutim, on mora da ima podršku svojih kolega i da ne zastupa samo svoje, već i njihove stavove.

U slučaju razvoja novog komercijalnog sistema, šampiona proizvoda možete dobiti ili angažovanjem konsultanta ili angažovanjem predstavnika budućeg kupca. Vaša organizacija može i da zaposli ljude koji su te poslove obavljali u nekoj drugoj organizaciji, gde su radili.

Kategorija	Aktivnosti
Planiranje	<p>Poboljšava okvir i granice sistema</p> <p>Utvrdjuje druge sisteme sa kojim sistem treba da bude u interakciji</p> <p>Ocenjuje efekat novog sistema na poslovne operacije kupca</p> <p>Definiše tranzicioni put od sadašnjih aplikacija ili ručnih operacija</p> <p>Utvrdjuje relevantne standarde i zahteve sertifikacije</p>
Zahtevi	<p>Prikuplja zahteve ostalih korisnika u klasi koju predstavlja</p> <p>Razvija scenarije korišćenja, slučajeva korišćenja, i priče korisnika</p> <p>Rešava konflikte između zahteva iz svoje klase korisnika</p> <p>Određuje prioritete primene zahteva</p> <p>Obezbeđuje unos potrebnih performansi i drugih zahteva kvaliteta</p> <p>Ocenjuje prototipve</p> <p>U saradnji sa drugima, rešava konflikte između zahteva različitih klasa korisnika</p> <p>Obezbeđuje specifične algoritme</p>
Potvrđivanje (validation) i proveravanje (verification)	<p>Recenzija specifikacije zahteva</p> <p>Definiše kriterijume prihvatanja</p> <p>Razvija testove prihvatanja korisnika iz scenarija korišćenja</p> <p>Obezbeđuje skup podataka za testiranja od poslovnog dela firme</p> <p>Realizuje beta testiranje ili testiranja prihvatanja od strane korisnika</p>
Pomoć korisnicima	<p>Piše delove teksta namenjen korisnicima sistema</p> <p>Učestvuje u pripremi materijala za obuku ili kurseve</p>

Izvor: Karl Wiegers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

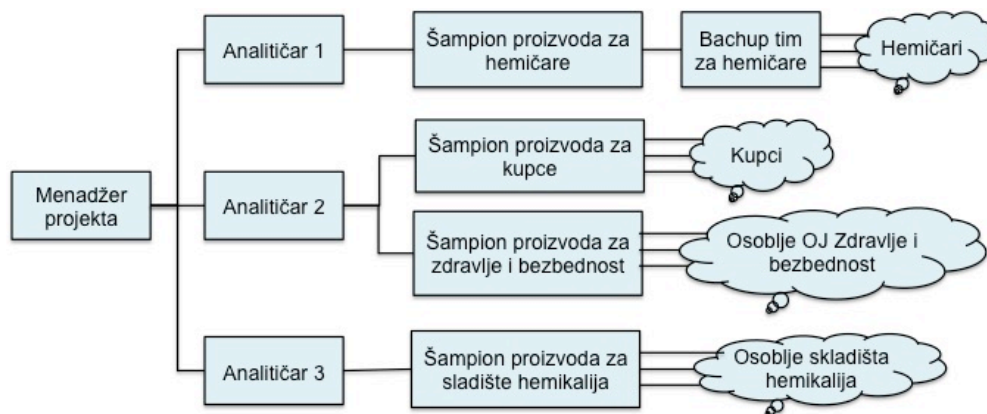
Slika 2.4 Aktivnosti šampiona proizvoda

## PRIMER ŠAMPIONA PROIZVODA - THE CHEMICAL TRACKING SYSTEM

*Broj biznis analitičara i broj šampiona proizvoda se određuju zavisno od vrste i veličine projekta.*

Obično jedna osoba ne može da bude šampion projekta za ceo sistem. U slučaju Chemical Tracking System, koriste se 4 šampiona proizvoda (slika 5). Oni su izabrani kao stručnjaci u kompaniji Contosa Pharmaceuticals, koja je i naručilac softvera. Kao što se vidi na slici, Menadžer projekta radi sa tri biznis analitičara i sa 4 šampiona proizvoda, da bi se prikupili zahtevi na pravi način. Šampioni proizvoda ne rade puno radno vreme za ovaj projekat, ali nedeljno mu posvećuju nekoliko sati. Jedan od biznis analitičara radi sa dva šampiona proizvoda, a ostali sa po jednim. Jedan od biznis menadžera je zadužen da sve prikupljene zahteve spoji i stavi u jedinstven SRS dokument (specifikacija softverskih zahteva).





Izvor: Karl Wieggers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013

Slika 2.5 Model šampiona proizvoda u slučaju CHT

Kako ima mnogo hemičara, šampion proizvoda za hemičare je formirao svoju back-up grupu koja mu pomaže da pokupe sve zahteve. Članovi ove grupe predstavljaju podklase hemičare. Na taj način su izbegnute velike radionice, koje nisu tako efikasne kao individualni razgovori sa predstavnicima podklasa krajnjih korisnika, u ovom slučaju, sa hemičarima.

## STAKEHOLDERS, VIEWPOINTS AND CONCERNS (VIDEO)

*Trajanje: 8:06 minuta*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## ▼ Poglavlje 3

# Podrška projektu i upravljanje

## VIDEO PREDAVANJE ZA OBJEKAT "PODRŠKA PROJEKTU I UPRAVLJANJE"

*Trajanje video snimka: 11min 49sek*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## PRIPREMA TIMA - OBUKE I PRIPREMNE AKTIVNOSTI

*Svi akteri u projektu, a posebno analitičari poslovanja moraju da prođu odgovarajuću obuku za razvoj softvera u određenom domenu primene.*

Obično neki članovi projektnog tima dobiju ulogu analitičara poslovanja na projektu, iako često i nemaju formalno obrazovanje za vršenje te uloge. Analitika poslovanja je specijalizovana i izazovna uloga. Pored neophodne obuke za izvršenje ovakve uloge, potrebno je da analitičar ima dobre komunikacione sposobnosti, da je zainteresovan za ovu ulogu i da u što većoj meri poznaje i oblast primene softvera, te da razume jezik korisnika, da bi razumeo i njihove potrebe. Potrebno je realizovati sledeće akcije:

- **Obuka analitičara poslovanja (business analysts):** U većoj ili manjoj meri, svi članovi tima vrše i ulogu analitičara poslovanja. Analitičar poslovanja trebalo bi da ima bar nekoliko dana obuke u različitim aktivnostima poslovne analize. To je osnova za izgradnju sopstvenog iskustva i dalje usavršavanje. On mora da bude strpljiv, da je dobro organizovan, da ima efektivnu veštinu komunikacije sa drugima, kao i da razume domen primene softvera.
- **Obuka aktera o zahtevima:** Najefektivnija obuka obuhvata više funkcionalnih disciplina, ne samo rad sa zahtevima. Akteri treba da imaju dan-dva obuke za rad sa zahtevima, da bi razumeli terminologiju, ključne koncepte i prakse, da bi shvatili zašto je inženjerstvo zahteva ključno za uspeh projekta razvoja softvera. Ova obuka je dobra i za razvoj sposobnosti tima i povezivanje članova tima.
- **Obuka inženjera razvoja o domenu primene softvera:** Organizujte seminar za inženjere razvoja na kome će se upoznati sa domenom primene softvera, sa aktivnostima kupaca, sa njihovom terminologijom, i sa ciljevima zbog kojih se realizuje projekat. To je i prilika da se upoznaju sa predstavnicima korisnika sa kojima će sarađivati tokom trajanja projekta.



- **Definisanje procesa inženjerstva zahteva:** Dokumentujete sve aktivnosti procesa inženjerstva zahteva. Dajte uputstva kako da se one sprovedu da bi analitičari uspešno odradili svoj posao.
- **Kreiranje rečnika termina:** Rečnik termina sadrži potrebne termine iz oblasti domena primene sistema. Time se minimiziraju nerazumevanja. Uključite sinonime, skraćenice i dr. Takav rečnik postaje stalna vrednost organizacije i služi i u sledećim projektima. Može da posluži novim članovima tima da se upoznaju sa problematikom i terminologijom, što je neophodno da bi se uključili uspešno u projekat.

## UPRAVLJANJE ZAHTEVIMA

*Postoje razlike u načinu definisanja zahteva pri primeni različitih metoda razvoja softvera.*

Upravljanje zahtevima (**requirements management**) je peta osnovna aktivnost u razvoju zahteva. Upravljanje zahtevima obuhvata aktivnosti koje treba da uradite posle utvrđivanja zahteva, kao što je kontrola verzija dokumenata sa zahtevima, održavanje osnovnih zahteva, utvrđivanje statusa zahteva, i prenošenje zahteva drugim elementima sistema. Upravljanje zahtevima traje za celo vreme trajanja projekta, ali sa niskim intenzitetom.

Iako su osnovni zadaci razvoj zahteva uglavnom isti, bez obzira na primenjenu metodologiju razvoja softvera, ipak, postoje neke specifičnosti

Pri primeni čisto *metoda vodopada*, planirate samo jedno glavno izdanje softvera, te se skoro svi zahtevi utvrđuju na početku projekta. Kasnije se ipak vrši dopuna zahteva, ako je potrebno,.

Kod *modela iterativnog razvoja softvera* kao što je RUP (**Rational United Process**) utvrđivanje zahteva se vrši u svakoj iteraciji procesa razvoja softvera, s tima što se najviše zahteva ipak radi u prvoj iteraciji. To se radi i ako softver ima više izdanja.

*Agilne i druge iterativne metode razvoja softvera* proizvodi nova izdanja svakih nekoliko nedelja, te imaju česte aktivnosti razvoja zahteva, ali malog intenziteta. U početku se prikupi malo veći broj zahteva, a onda se oni raspodeljuju po sprintovima, zavisno od određenog prioriteta. Kada dođu na red, ovi zahtevi se mogu više detaljisati.

Bez obzira na primenjeni metod razvoja softvera, morate kod svakog izdanja ili iteracije da se pitate koji će od zadataka prikazanih na slici 2 u poglavlju 1, najviše dodati vrednost softveru, i koji rizik unosite u projekat sa ovim zahtevom.

Posle završetka zadatka br. 17, vi ste spremni za početnu konstrukciju odgovarajućeg dela softvera. Za svaki zahtev korisnike, obično treba da izvršite zadatke od 8 do 17, i tako se stvara osnova sledećeg izdanja softvera.

## PLANIRANJE PROJEKTA U VEZI SA ZAHTEVIMA

*Zahtevi se menjaju tokom trajanja projekta. Neophodno je uspostaviti sistem za upravljanje zahtevima i koristiti odgovarajuće alate za podršku tom sistem*

Posle kreiranja prve verzije zahteva, dolazi do neophodnih izmena zahteva posle kontakta s korisnicima, menadžerima, marketingom, razvojnim timom i drugima koji su postavili neke zahteve. Efektivno upravljanje promenama zahteva obradu predloženih izmena, ocenjujući troškove njihove primene i uticaj na projekat, i obezbeđuje da odgovarajući akteri donesu odgovarajuće odluke o prihvatanju predloženih promena.

Preduslov za primenu efektivnog upravljanja promenama je korišćenje upravljanja konfiguracijom. Alat za kontrolu mora da ima istu verziju kao i vaš kod, da bi mogao da upravlja dokumentima sa zahtevima. Druga i bolja alternativa je da svi zahtevi budu pod kontrolom alata za upravljanje zahtevima. Primenjuju se sledeće akcije:

- **Postavljanje procesa za kontrolu promena zahteva:** Neophodno je izbeći haos pri promeni zahteva. Zato je neophodno postaviti odgovarajući mehanizam kontrole promene zahteva. Upravljanje svim predloženim promenama u ovom procesu. Primena alata za otkrivanje grešaka može da podržava ovaj proces kontrole. Imenuje se mala grupa aktera projekta koji čine Odbor za kontrolu promena ( **Change Control Board - CCB**). On ocenjuje predloge za promenu zahteva, CCB odlučuje koji zahtev prihvatiti i postavlja prioritete primene usvojenih zahteva i određuje izdanje u kome će biti primenjeni.
- **Izvršenje analize uticaja promene zahteva:** Pomaže da CCB donese odluku o predlogu izmene zahteva. Vršiti sa analiza uticaja na projekat za svaku predloženu izmenu zahteva. Koristi se matrica povezanih zahteva da bi se odredili ostali zahtevi, elementi projektovanja, izvorni kod, i testovi koji možda moraju da se promene. Identifikuje zadatke koji treba da primene promene i procenjuje rad koji je neophodan za realizaciju predloženih promena.
- **Postavljanje početne vrednosti zahteva i kontrole verzija skupa zahteva:** Početan skup zahteva (**requirements baseline**) predstavlja dogovoreni skup zahteva za određeno izdanje softvera. Njihova promena je moguća samo preko posebnog sistema za kontrolu promena zahteva, da bi se izbegle zabune pri radu sa prethodnim i novim verzijama zahteva, potrebno je primeniti jedinstveni identifikator za specifikaciju zahteva.
- **Održavanje istorije promene svakog pojedinačnog zahteva:** To je potrebno naročito ako želimo da se vratimo na prethodnu verziju zahteva. ili kada želimo da vidimo kako smo došli do sadašnje verzije zahteva. Zapisuje se datum izmene, opisuje izmena koja je izvršena, osoba koja je izvršila promenu i razlog za promenu.

## PRAĆENJE REALIZACIJE PROJEKTA U VEZI SA ZAHTEVIMA

*Praćenje statusa zahteva, praćenje pitanja vezanih za zahteve, održavanje matrice povezivanja zahteva, upotreba alata za upravljanje zahtevima.*

- **Praćenje statusa svakog zahteva:** Postavite bazu čiji svaki slog nosi informaciju o jednom zahtevu. Odredi atribut za ključeve nalaženja svakog atribut, uključujući atribut o statusu zahteva (predložen, odobren, primenjen, verifikovan) tako da možete da kontrolišete sve zahteve u istom statusu. Praćenje statusa svakog zahteva tokom trajanja projekta odražava i status projekta tokom svog trajanja.
- **Praćenje svih pitanja u vezi zahteva:** Alat za praćenje otvorenih pitanja vezanih za zahteve pomaže da se ta pitanja ne zaborave i da se na njih reaguje. Potrebno je da se neko imenuje za ovu aktivnost. Posmatra se status zahteva da bi se pružila ocena o stanju zahteva u projektu.
- **Održavanje matrice povezivanja zahteva:** Ponekad je neophodno pogledati vezu nekog zahteva sa elementima projektnog rešenja, koda koji primenjuje zahtev i testa koji ga verifikuje. Matrica povezivanja zahteva (**requirement tracability matrix**) pomaže da se utvrdi da li su svi zahtevi primenjeni i verifikovani. Koriste se i u održavanju softvera kada treba da dođe do promene zahteva. Matrica povezivanja zahteva može da pomogne da se poveže neki funkcionalni zahtev, sa zahtevima višeg nivoa apstrakcije (npr. poslovna pravila), da bi se videlo odakle zahtev potiče. Matricu treba popunjavati za vreme razvoja, a ne tek na njegovom kraju. Podrška odgovarajućeg alata je vrlo preporučljiva, sem u najmanjim projektima.
- **Upotreba alata za upravljanje zahtevima:** Komercijalni alati za upravljanje zahtevima omogućuju vam da sve tipove zahteva stavite u jednu bazu podataka. Ovakav alat vam omogućava da automatizujete najveći deo operacija nad zahtevima koji je ovde opisan.

## UPRAVLJANJE PROJEKTOM

*Na osnovu prikupljenih i prihvaćenih zahteva, potrebno je planirati njihovu primenu. Potrebne akcije: izbor vrste projekta, plan razvoja zahteva, procena potrebnog rada, i plana projekta.*

Upravljanje softverskim projektom je blisko povezano sa procesima projektnih zahteva. Menadžer projekta treba da podesi plan projekta, resurse, i zaduženja članovima tima na osnovu zahteva koje treba da se primene.

Alternativna strategija je primena vremenske kutije i ciklusa. Projektni tim treba da proceni rad koji se treba uložiti u jednu iteraciju fiksnog trajanja, što je slučaj kod alternativnih metoda razvoja softvera. Opseg funkcija se pregovara, a u skladu sa fiksnim trajanjem ciklusa ("Šta se može uraditi u te periodu?"). Vlasnik projekta može da zahteva šta hoće, ali mora da

odredi prioritete po kome će tim razvijate softver u iterativnim ciklusima. Evo akcija koje treba realizovati:

- **Izbor odgovarajućeg životnog ciklus razvoja proizvoda:** Svaki menadžer projekta treba da izabere životni ciklus projekta koji najviše odovara specifičnostima njegovog projekta. Pri tome treba da uključi definiciju zahteva u definicije životnog ciklusa projekta. Ako je moguće, treba da specifikira i primeni skup funkcionalnosti inkrementalno kako bi isporučio prve verzije softvera što pre.
- **Planiranje zahteva:** Svaki projektni tim treba da planira kako će razvijati zahteve i aktivnosti upravljanja. Plan prikupljanja zahteva treba da omogući da dobijete predloge različitih aktera u pravim fazama projekta upotrebom tehnika koje najviše odgovaraju projektu. Poslovni analitičar (BA) i menadžer projekta treba da rade zajedno radi obezbeđenja da se zadaci i rezultati koji odgovaraju inženjerstvu zahteva nađu zajedno u planu upravljanja projektom.
- **Procena potrebnog rada:** Akteri obično žele da znaju u koje vreme se mogu dobiti zahtevi i koji procenat rada na projektu se treba dodeliti inženjeringu zahteva. To zavisi od mnogih faktora. Pretpostavite vreme malo veće od prosečnog, kako bi imali dovoljno vremena da obezbedite zahteve koji treba da budu solidan temelj za razvoj .
- **Izrada plana projekta u skladu sa zahtevima:** Razvijajte plan i termine iterativno, kako opseg i detaljni zahtevi postaju jasniji. Počnite se procenom troškova i vremenskih rokova u skladu sa početnom vizijom i opsegom projekta. U slučaju agilnih projekta iteracije u formi vremenske kutije omogućuje planiranje u okviru fiksnog trajanja jedne iteracije i ograničenih resursa.

## UPRAVLJANJE PROJEKTOM (NASTAVAK)

*Potrebne akcije: određivanje donosioca odluka o zahtevima, promena zahteva, upravljanje rizicima u vezi sa zahtevima, praćenje uloženog rada, i analiza stečenog iskustva.*

- **Utvrđite donosioca odluka u vezi zahteva:** Treba razrešiti konfliktne zahteve aktera, izvršiti izbor komercijalnih komponenti, Zato je važno na početku odrediti ko su oni koji mogu da donose odluke u vezi zahteva.
- **Ponovo pregovarajte raspodelu zadataka kada dođe do promene zahteva:** Zadaci projektnog tima, i njihov kapacitet, određen je na osnovu početnog plana i budžeta. Sa unošenjem novih zahteva u projekat, izvršite ponovnu procenu da li sa raspoloživim resursima možete da ispunite početni plan i budžet. Ako ne možete, upoznajte sa ovim menadžment i dogovorite novi, realističan raspored zadataka i broj ljudi. To se isto radi kada počnete procene koje su bile zasnovane na nedovoljno jasnim zahtevima, postanu problem.
- **Analizirajte, dokumentujte i upravljajte rizicima vezanih za zahteve:** Iznenadni događaji i uslovi, mogu da dovedu do opustošenosti u nepripremljenom projektu. Zato, utvrdite i dokumentujte rizike povezane sa zahtevima kao deo sistem za upravljanje rizicima u projektu. Razmatrajte pristupe za sprečavanje ili izbegavanje rizika, primenite akcije ublažavanja, i pratite njihovog dejstvo i napredak, kao i efektivnost.

- **Pratite uloženi rad na zahtevima:** Da bi poboljšali vašu sposobnost procenjivanja potrebnog rada na zahtevima u budućim projektima, pratite troškove rada na zahtevima i za upravljanje aktivnostima u ovom projektu, Osmatrajte efekte koje ostvaruju vaše aktivnosti sa zahtevima na projekat i na ocenu povraćaja investicija u inženjerstvo zahteva.
- **Analizirajte stečeno iskustvo vezano za zahteve i u drugim projektima:** Organizacija koja uči vrši periodične retrospektive radi prikupljanja stečenog iskustva i "naučenih lekcija" ("Lesson learnt") iz završenih projekata ili iz ranijih iteracija sadašnjeg projekta.

## SOFTWARE PROJECT MANAGEMENT - WHY IT'S DIFFERENT! (VIDEO)

*Trajanje: 4:38 minuta*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## ▼ Poglavlje 4

### Vežba

## VIDEO PREDAVANJE ZA OBJEKAT "VEŽBA"

*Trajanje video snimka: 34min 16sek*

**Ova lekcija sadrži video materijal. Ukoliko želite da pogledate ovaj video morate da otvorite LAMS lekciju.**

## PITANJA ZA DISKUSIJU

*Tekst pitanja za diskusiju*

### **PITANJE 1.**

Zapišite probleme u vezi sa zahtevima sa kojima ste se susreli na svojim prethodnim projektima. Identifikujte svaki deo problema razvoja projekta ili problema upravljanja zahtevima. Opišite osnovni uzrok svakog problema i njegov uticaj na projekat. Imate li ideju koje promene u trenutnom načinu upravljanja zahtevima bi mogle da odgovore na ove probleme. (15 min)

### **PITANJE 2.**

Uzmite 10 do 15 minuta i osmislite proces inženjerstva zahteva kakav smatrate da bi bilo idealno primenjivati u jednoj kompaniji koja se bavi razvojem softvera. Proces predstavite sistematično, u koracima i uz čvrste stavove. Pre nego što krenete u razrađivanje ideje, razmislite o nekoliko ključnih pitanja (30 min):

- Koje je veličine firma?
- Koju metodologiju razvoja softvera primenjuje?
- Na kakvom tipu projekata najčešće radi: interni projekti, outsourcing...
- Kakav je sistem upravljanja?
- Kakva je radna struktura?
- Da li postoji sistem za praćenje procesa rada i dokumentacije?

## ZADACI ZA VEŽBU

*Tekst zadatka za vežbu*

### **ZADATAK 1.**

Uzmite projektnu dokumentaciju koju ste radili za potrebe nekog od predmeta sa prethodne godine studija, poput SE201, SE211 ili CS324. Razmenite međusobno sa kolegama dokumentaciju. Svako treba da pročita tuđi rad i obavi jednostavnu procenu dokumenta, a posebno onog dela koji se odnosi na zahteve. Svaki revizor treba da iznese svoja zapažanja i, ako ih ima, jasne predloge za poboljšanje. (15 min)

## **ZADATAK 2.**

Pretpostavite da ste sistem analitičar koji treba da razvije novi modul u okviru Informacionog sistema univerziteta Metropolitan (ISUM) koji će koristiti samo studenti i koji će imati naziv E-Student. Razmislite i zatim odgovorite na sledeća pitanja:

- Šta biste izdvojili kao sistemske zahteve iz ugla korisnika budućeg sistema?
- Na osnovu sistemskih, koji bi bili softverski zahtevi?
- Šta biste naveli kao zahteve projekta?
- Koji bi bili zahtevi proizvođača?
- Šta biste identifikovali kao procesne zahteve?
- Da li postojali neki eksterni zahtevi?

Obratite posebnu pažnju na način izražavanja zahteva koje ste postavili. Podsetite se prvog predavanja da biste mogli adekvatno da odgovorite na sva pitanja. (30 min)

## ▼ Poglavlje 5

# Domaći zadatak

## DOMAĆI ZADATAK 2

### *Tekst domaćeg zadatka*

Za isti sistem koji ste dobili da analizirate za DZ01, identifikujte i navedite sledeće klase zahteva:

1. Zahteve projekta
2. Zahteve proizvoda
3. Procesne zahteve
4. Eksterne zahteva

Za svaku klasu zahteva navedite bar 2 primera zahteva koja nisu navedena u lekcijama.

*Napomene:*

Zadatak se rešava opisno i šalje kao .docx fajl.

Rešenje zadatka pošaljite na mejl adresu predmetnog asistenta. Rok za izradu je definisan Plan i programom predmeta.



## ▼ Poglavlje 6

### Projektni zadatak

#### ZADATAK ZA RAD NA PROJEKTU

##### *Tekst zadatka za rad na projektu*

Pošaljite asistentu na mejl **predlog teme za projekat** koji ćete raditi na predmetu. Projekat koji asistent usvoji u drugoj nedelji nastave se izrađuje kroz naredne nedelje nastave, paralelno sa domaćim zadacima.

## ▼ Zaključak

### ZAKLJUČAK

#### *Šta smo naučili u ovoj lekciji?*

1. **Proces razvoja zahteva** nije linearan, već ima i povratne sprege, te se iterativno realizuje.
2. Primenuju se različiti načini za **prikupljanje zahteva** od korisnika i drugih aktera.
3. **Analiza zahteva** poboljšava zahteve, otklanja greške i nedostatke u njima, a omogućava bolje razumevanje zahteva od strane svih aktera.
4. Svrha **specifikacije zahteva** je dokumentovanje zahteva različitog tipa na jedan konzistentan, pristupačan i proverljiv način koji je razumljiv akterima sistema
5. **Potvrđivanje** (validacija) **zahteva** obezbeđuje kontrolu da li su zahtevi korektni, i pokazuju karakteristike kvaliteta i zadovoljenje potreba korisnika
6. Ne postoji jedna **kategorija korisnika**. Korisnike treba razvrstati u više klasa korisnika. Svaka **klasa korisnika** ima svoj skup zahteva u skladu sa zadacima koje članovi klase treba da izvrše.
7. Primena organizacionih šema i poslovnih procesa može da olakša **utvrđivanje klase korisnika**.
8. Neophodno je koristiti sve moguće **vidove komunikacija između korisnika i inženjera razvoja**. Treba biti svestan da posredništvo uvodi rizik, ali je nužno kod većih projekata.
9. **Šampion proizvoda** služi kao primarni interfejs između određene klase korisnika i biznis analitičara. Broj biznis analitičara i broj šampiona proizvoda se određuju zavisno od vrste i veličine projekta.
10. Svi akteri u projektu, a posebno analitičari poslovanja moraju da prođu odgovarajuću **obuku za razvoj softvera** u određenom domenu primene.
11. Postoje razlike u **načinu definisanja zahteva** pri primeni različitih metoda razvoja softvera.
12. Zahtevi se menjaju tokom trajanja projekta. Neophodno je **uspostaviti sistem za upravljanje zahtevima** i koristiti odgovarajuće alate za podršku tom sistemu.
13. **Praćenje statusa zahteva**, praćenje pitanja vezanih za zahteve, održavanje matrice povezivanja zahteva, upotreba alata za upravljanje zahtevima.
14. Na osnovu prikupljenih i prihvaćenih zahteva, potrebno je **planirati njihovu primenu**. Izbor vrste projekta, plan razvoja zahteva, procena potrebnog rada, uskladiti plan projekta sa zahtevima.
15. **Određivanje donosioca odluka o zahtevima**, promena zahteva, upravljanje rizicima u vezi sa zahtevima, praćenje uloženog rada, analiza stečenog iskustva

### REFERENCE

Nastavni materijal pripremljen za studente se pravi s namerom da im omogući brži i skraćeni uvid u program lekcije, a na bazi jedne ili više referentnih udžbenika i drugih izvora. Nastavni

materijal nije zamena za ove udžbenike, koje treba koristiti ako student želi da se detaljnije upozna sa nastavnom materijom. Očekuje se od studenta da poseduje bar jedan od navedenih udžbenika u Planu i programu predmeta.

Ova lekcija je urađena na bazi teksta datom **u poglavlju 3 knjige: Karl Wieggers, Joy Beaty, Software Requirements, 3rd ed., Microsoft, 2013.** Za detaljnije proučavanje i primere, studentima se preporučuje da pročitaju ovo poglavlje. Manji uticaj na sadržaj lekcije imaju ostale reference navedene u Planu i programu predmeta.