



SE101 - RAZVOJ SOFTVERA I INŽENJERA SOFTVERA

Inženjerstvo softverskih proizvoda

Lekcija 04

PRIRUČNIK ZA STUDENTE

SE101 - RAZVOJ SOFTVERA I INŽENJERA SOFTVERA

Lekcija 04

INŽENJERSTVO SOFTVERSKIH PROIZVODA

- ✓ Inženjerstvo softverskih proizvoda
- ✓ Poglavlje 1: Softverski proizvodi
- ✓ Poglavlje 2: Vizija proizvoda
- ✓ Poglavlje 3: Upravljanje softverskim proizvodom
- ✓ Poglavlje 4: Izrada prototipova softverskog proizvoda
- ✓ Poglavlje 5: 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

U ovoj lekciji će fokus biti softverski proizvod a ne softverski projekat, tj ceo softverski proces.

U prethodnim lekcijama smo izučavali metode softverskog inženjerstva koje primenjuju procesni pristup razvoju, koji obuhvata ceo softverski proces sa svim njegovim aktivnostima. Taj pristup razvoju softvera se često naziva i projektni pristup jer projekat obuhvata ceo softverski proces sa svim njegovim aktivnostima. Međutim, ovakav pristup nije uvek pogodan za razvoj softverskih proizvoda. U mnogim slučajevima je efikasnije primeniti tehnike inženjerstva proizvoda koje odgovaraju tipu softverskog proizvoda koji treba razviti. Ti metodi više odgovaraju realizaciji studentskih projekata razvoja proizvoda, mada to nećemo primeniti u ovom predmetu, ali ćemo primenjivati u drugim predmetima softverskog inženjerstva u kasnijim godinama studija.

U ovoj lekciji će fokus biti softverski proizvod a ne softverski projekat, tj ceo softverski proces. Pošto u prvom semestru, studenti još nemaju dovoljno znanja iz programiranja, to će se u ovom predmetu, pa i u ovoj lekciji, obrađivati samo onaj deo softverskog inženjerstva koji se bazira na proizvodu i to samo na konceptijskom nivou, tj. u kojme se još ne primenjuje programiranje. Inženjerstvo softverskih proizvoda koje uključuje programiranje (implementaciju softvera) i evoluciju softvera je u programu ostalih specifičnih predmeta softverskog inženjerstva, koje će studenti izučavati u narednim godinama studija, a u okviru osnovnih akademskih studija (OAS) Softversko inženjerstvo.

▼ Poglavlje 1

Softverski proizvodi

SOFTVERSKI PROIZVODI I PRILAGOĐENI SOFTVERI

Softverski proizvodi su generički softverski sistemi koji se prodaju vladama, preduzećima i potrošačima

U ovoj lekciji, kao što je napomenuto, izučavaju se tehnike softverskog inženjerstva koje se koriste za razvoj softverskih proizvoda. *Softverski proizvodi su generički softverski sistemi koji se prodaju vladama, preduzećima i potrošačima*. Mogu biti projektovani da podrže poslovnu funkciju, kao što je računovodstvo; mogu biti alati za produktivnost, kao što su sistemi za kreiranje dokumenata; ili mogu biti igre ili sistemi ličnih informacija.

Softverski proizvodi se kreću po veličini od miliona linija koda u velikim poslovnim sistemima do nekoliko stotina linija koda u jednostavnoj aplikaciji za mobilne telefone.

Inženjerske tehnike koje se koriste za razvoj softverskog proizvoda evoluirale su od tehnika softverskog inženjerstva razvijenih u 20. veku za podršku razvoju softvera po meri. Kada se softversko inženjerstvo pojavilo kao disciplina 1970-ih, praktično sav profesionalni softver bio je „jednokratni“, prilagođeni softver (softveer «po meri naručioca»). Kompanije i vlade su želele da automatizuju svoje poslovanje i naveli su šta žele da njihov softver radi. Zatim je razvijen interni inženjerski tim ili eksterna softverska kompanija radi razvoja potrebnog softvera.

Primeri prilagođenog softvera koji je razvijen u to vreme uključuju:

- sistem upravljanja vazдушnim saobraćajem;
- računovodstveni sistemi za sve glavne banke;
- sistemi naplate za komunalna preduzeća kao što su snabdevači električnom energijom i gasom;
- sistemi vojne komande i kontrole.

GENERIČKI SOFTVERSKI PROIZVOD

Umesto razvoja softvera po zahtevima određenog naručioca, sve češće počinje razvoj generičkih softverskih proizvoda koji su namenjeni tržištu softverskih proizvoda.

Softverski projekti su postavljeni za razvoj ovih jednokratnih sistema razvijenih po zahtevu kupca, sa softverskim sistemom zasnovanim na skupu softverskih zahteva. Ugovor između kupca softvera i kompanije za razvoj softvera uključivao je dokument sa zahtevima, koji

je predstavljao specifikaciju softvera koji treba da bude isporučen. Kupci su definisali svoje zahteve i radili sa razvojnim timom kako bi detaljno specificirali funkcionalnost softvera i njegove kritične atribute.

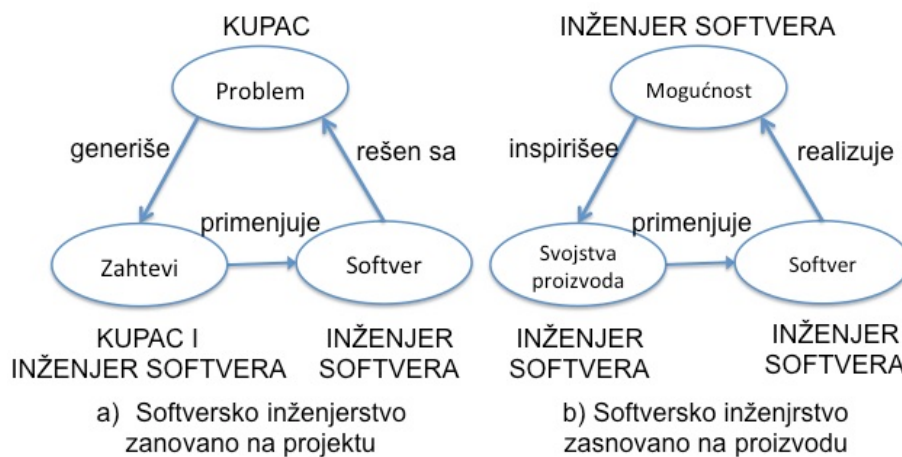
Ovaj pristup zasnovan na projektima dominirao je softverskom industrijom više od 25 godina. Metode i tehnike koje su evoluirale da podrže razvoj zasnovan na projektima definisale su šta se podrazumeva pod „softverskim inženjerstvom“. *Osnovna pretpostavka je bila da je za uspešno softversko inženjerstvo potrebno mnogo pripremnog rada pre nego što se počne sa pisanjem programa.* Na primer, bilo je važno potrošiti vreme na dobijanje „ispravnih“ zahteva i crtanje grafičkih modela softvera. Ovi modeli su kreirani tokom procesa projektovanja softvera i korišćeni za dokumentovanje softvera.

Međutim, kako je sve više kompanija automatizovalo svoje poslovanje, *postalo je jasno da većini preduzeća zapravo nije potreban prilagođeni softver*, tj. softver razvijen u skladu sa njihovim zahtevima, jer su uvideli da u mnogim projektima, njihovi zahtevi su slični sa zahtevima drugih kompanija. Mogli su da koriste generičke softverske proizvode koji su projektoavani za uobičajene poslovne probleme. Industrija softverskih proizvoda razvijena je da zadovolji ovu potrebu. *Tehnike softverskog inženjeringstva zasnovane na projektima su prilagođene razvoju softverskih proizvoda.* Generički softverski proizvod, koji je razvijen za prodaju na tržištu, a ne razvijen za potrebe određenog naručioca, postao je sve češći cilj projekata softverskog inženjerstva. Serijakom "proizvodnjom" generičkih softverskih proizvoda, softverko inženjerstvo postaje industrija slična industriji koja proizvodi fizičke proizvoda za tržište, kao što su računari, automobili i dr. Naravno, zbog svog nefizičkog svojstva, proizvodnja generičkih softverskih proizvoda se odvija u formi kontinualnog ovih proizvoda, koji se povremenu nude na tržištu u vidu novih izdanja. Taj proces prestaje samo na kraju životnog ciklusa trajanja softverskog proizvoda.

DVA PRISTUPA RAZVOJU SOFTVERA

Softversko inženjerstvo zasnovano na projektu i zasnovano na proizvodu.

Tehnike zasnovane na projektima nisu pogodne za razvoj proizvoda zbog fundamentalnih razlika između softverskog inženjerstva zasnovanog na projektu i softverskog inženjerstva zasnovanog na proizvodu. Ove razlike su ilustrovane na slici 1.



Slika 1.1 Dva pristupa razvoju softvera [Sommerville 2021, Fig.1.1, Fig.1.2]

Softverski projekti, tj. softversko inženjerstvo zasnovano na projektu, odn. procesu (slika 1.a), uključuju eksternog klijenta ili kupca koji odlučuje o funkcionalnosti sistema i sklapa pravni ugovor sa kompanijom za razvoj softvera. Problem klijenta i trenutni procesi se koriste kao osnova za kreiranje softverskih zahteva, koji određuju softver koji će se implementirati. Kako se poslovanje menja, prateći softver mora da se promeni. Kompanija koja koristi softver odlučuje i plaća za promene. Softver često ima dug životni vek, a troškovi promene velikih sistema nakon isporuke obično premašuju početne troškove razvoja softvera.

Softverski proizvodi su specifikirani i razvijeni na drugačiji način. Ne postoji spoljni kupac koji kreira zahteve koji definišu šta softver mora da radi. Programer softvera odlučuje o karakteristikama proizvoda, kada će nova izdanja biti dostupna, platformama na kojima će softver biti implementiran i tako dalje. Očigledno se uzimaju u obzir potrebe potencijalnih kupaca za softverom, ali kupci ne mogu insistirati da softver sadrži određene karakteristike ili attribute. Razvojna kompanija bira kada će se izvršiti promene u softveru i kada će biti puštene korisnicima.

SOFTVER RAZVIJEN KAO PROIZVOD

Softver razvijen kao proizvod je obično jeftiniji za svakog kupca, od prilagođenog softvera.

Kako su troškovi razvoja raspoređeni na mnogo veću bazu korisnika, *softver razvijen kao proizvod je obično jeftiniji*, za svakog kupca, od prilagođenog softvera. Međutim, *kupci softvera moraju da prilagode svoj način rada softveru*, jer on nije razvijen s obzirom na njihove specifične potrebe. Pošto programer, a ne korisnik, kontroliše promene, postoji rizik da će programer prestati da podržava softver. Tada će kupci proizvoda morati da pronađu alternativni proizvod.

Polazna tačka za razvoj proizvoda je prilika koju je kompanija identifikovala da stvori održiv komercijalni proizvod. Ovo može biti originalna ideja, kao što je Airbnb-ova ideja za deljenje smeštaja; poboljšanje u odnosu na postojeće sisteme, kao što je računovodstveni sistem

zasnovan na oblaku; ili generalizacija sistema koji je razvijen za određenog kupca, kao što je sistem upravljanja imovinom.

Pošto je programer proizvoda odgovoran za identifikaciju mogućnosti, oni mogu odlučiti o funkcijama koje će biti uključene u softverski proizvod. Ove funkcije su projektovane da privuku potencijalne kupce tako da postoji održivo tržište za softver.

Pored razlika prikazanih na slikama 1 i 2 postoje i dve druge važne razlike između softverskog inženjeringa zasnovanog na projektu i proizvoda:

1. Kompanije proizvoda mogu odlučiti kada da promene svoj proizvod ili da skinu svoj proizvod sa tržišta. Ako se proizvod ne prodaje dobro, kompanija može smanjiti troškove zaustavljanjem njegovog razvoja. *Prilagođeni softver razvijen u softverskom projektu obično ima dug životni vek i mora biti podržan tokom tog životnog veka.* Kupac plaća podršku i odlučuje kada i da li treba da se završi.
2. Za većinu proizvoda, *brzo dostavljanje proizvoda kupcima je ključno.* Odlični proizvodi često propadaju jer inferiorni proizvod prvi stigne na tržište i kupci kupuju taj proizvod. U praksi, kupci nerado menjaju proizvode nakon što su uložili vreme i novac u svoj prvobitni izbor.

Brzo izvođenje proizvoda na tržište važno je za sve vrste proizvoda, od malih mobilnih aplikacija do poslovnih proizvoda kao što je Microsoft Word. To znači da *se inženjerske tehnike usmerene na brz razvoj softvera (agilne metode) univerzalno koriste za razvoj proizvoda.* Agilne metode i njihovu ulogu u razvoju proizvoda izučaćemo u sledećoj lekciji

LINJE SOFTVERRSKIH PROIZVODA I PLATFORME

Linije softverskih proizvoda su sistemi projektovani da budu prilagodljivi korisniku, a platforme pružaju skup funkcija koje se mogu koristiti za kreiranje nove funkcionalnosti.

Ako čitate o softverskim proizvodima, možda ćete naići na još dva termina: „linije softverskih proizvoda“ i „platforme“ (tabela 1).

Linije softverskih proizvoda su sistemi projektovani da budu prilagodljivi da zadovolje specifične potrebe kupaca promenom delova izvornog koda.

Platforme pružaju skup funkcija koje se mogu koristiti za kreiranje nove funkcionalnosti. Međutim, uvek morate da radite u okviru ograničenja definisanih od strane dobavljača platforme.

Kada su softverski proizvodi prvi put razvijeni, oni su isporučeni na disku i instalirani od strane kupaca na svoje računare. Softver je radio na tim računarima i na njima su se čuvali korisnički podaci. Nije bilo komunikacije između računara korisnika i računara dobavljača. **Sada kupci mogu da preuzimaju proizvode iz prodavnice aplikacija ili sa veb lokacije prodavca.**

Tabela T1: Linije softverskog proizvoda i platforme

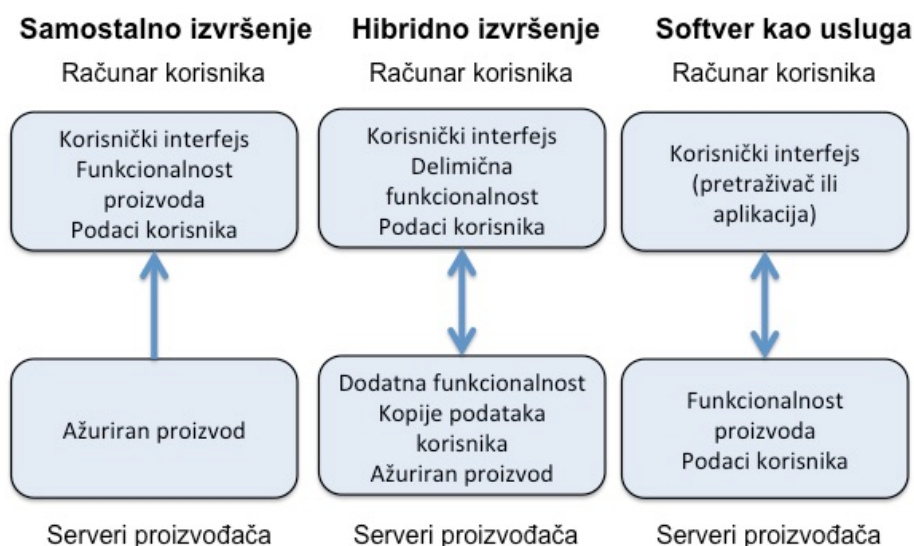
Tehnologija	Opis
Linija softverskog proizvoda	Skup softverskih proizvoda koji dele zajedničko jezgro. Svaki član linije proizvoda uključuje prilagođavanja i dopune specifične za kupca. Linije softverskih proizvoda mogu se koristiti za implementaciju prilagođenog sistema za kupca sa specifičnim potrebama koje se ne mogu zadovoljiti generičkim proizvodom. Na primer, kompanija koja pruža komunikacioni softver službama za hitne slučajeve može imati liniju softverskih proizvoda gde osnovni proizvod uključuje osnovne komunikacione usluge kao što su prijem i evidencija poziva, pokretanje odgovora u hitnim slučajevima, prosleđivanje informacija vozilima itd. Međutim, svaki kupac može koristiti različitu radio opremu i njihova vozila mogu biti opremljena na različite načine. Osnovni proizvod mora biti prilagođen svakom kupcu za rad sa opremom koju koristi.
Platforma	Softverski (ili softver+hardverski) proizvod koji uključuje funkcionalnost tako da se na njemu mogu graditi nove aplikacije. Primer platforme koju verovatno koristite je Facebook. Pruža širok skup funkcionalnosti proizvoda, ali takođe pruža podršku za kreiranje „Facebook aplikacija“. Ovim se dodaju nove funkcije koje mogu da koriste preduzeća ili interesna grupa na Fejsbuku.

Slika 1.2 Linija softverskog proizvoda i platforme [Sommerville 2021, Table 1.1]

NAČINI IZVRŠENJA SOFTVERA

Samostalno izvršenje, hibridno izvršenje i softver kao usluga

Neki proizvodi su i dalje zasnovani na samostalnom modelu izvršenja u kojem se sva računanja vrše na računarima vlasnika proizvoda. Međutim, sve prisutno umrežavanje velike brzine znači da su sada dostupni alternativni modeli izvršenja. U ovim modelima, računari vlasnika proizvoda deluju kao klijenti, sa nekim ili svim izvršenjem i skladištenjem podataka na serverima prodavca (slika 3).



Slika 1.3 Modeli izvršenja softvera [Sommerville 2021, Fig.1.3]

Postoje dve alternative za samostalne softverske proizvode:

1. **Hibridni proizvodi:** Neke funkcionalnosti su implementirane na računaru korisnika, a neke na serverima dobavljača proizvoda kojima se pristupa preko Interneta. Mnoge telefonske aplikacije su hibridni proizvodi sa računarski intenzivnom obradom koja se prenosi na udaljene servere.
2. **Proizvodi zasnovani na uslugama:** Aplikacijama se pristupa preko Interneta iz veb pretraživača ili aplikacije. Možda postoji neka lokalna obrada koristeći Javascript, ali većina izračunavanja se obavlja na udaljenim serverima. Sve više i više proizvodnih kompanija pretvara svoje proizvode u usluge jer to pojednostavljuje ažuriranje proizvoda i čini izvodljivim nove poslovne modele, kao što je pay-as-you-go.

TEHNIKE RAZVOJA SOFTVERSKOG PROIZVODA

Postoje aktivnosti, osim programiranja, koje su neophodne za razvoj visokokvalitetnih softverskih proizvoda.

Ključna karakteristika razvoja proizvoda je to ne postoji eksterni kupac koji generiše softverske zahteve i plaća softver. Ovo važi i za neke druge vrste razvoja softvera:

1. **Studentski projekti:** Kao deo predmeta računarstva ili inženjerstva, studentima se mogu postaviti zadaci u kojima rade u grupama na razvoju softvera. Grupa je odgovorna za odlučivanje o karakteristikama sistema i kako da rade zajedno na implementaciji ovih karakteristika.
2. **Softver za istraživanje:** Softver je razvijen od strane istraživačkog tima da podrži njihov rad. Na primer, istraživanje klime zavisi od velikih klimatskih modela koje dizajniraju istraživači i implementiraju u softver. U manjem obimu, inženjerska grupa može da napravi softver za modeliranje karakteristika materijala koji koristi.
3. **Interni razvoj alata:** Tim za razvoj softvera može odlučiti da su mu potrebni neki specifični alati za podršku njihovom radu. Oni specificiraju i implementiraju ove alate kao „interne“ proizvode.

Tehnike razvoja proizvoda koje ovde objašnjavamo možete koristiti za bilo koji tip razvoja softvera koji nije vođen zahtevima spoljnih kupaca.

Postoji uobičajeno gledište da je inženjerstvo softverskih proizvoda jednostavno napredno programiranje i da je tradicionalno softversko inženjerstvo nebitano. Sve što treba da znate je kako da koristite programski jezik plus okvire i biblioteke za taj jezik. Ovo je pogrešno shvatanje što je detaljnije objašnjeno u [Sommerville 2021] postoje aktivnosti, osim programiranja, koje su neophodne za razvoj visokokvalitetnih softverskih proizvoda.

Ako želite da vaš proizvod bude uspešan, morate razmišljati o drugim pitanjima osim o programiranju.

- Morate pokušati da razumete *šta je potrebno vašim klijentima* i kako potencijalni korisnici mogu da rade sa vašim softverom.
- Morate da *projektujete ukupnu strukturu vašeg softvera* (arhitekturu softvera) i znate o *tehnologijama kao što su računarstvo u oblaku i bezbednosno inženjerstvo*.

- Morate da *koristite profesionalne tehnike za verifikaciju i testiranje* vašeg softvera i sistema za upravljanje kodom da biste pratili promenu kodne baze.
- Morate da *razmislite o poslovnom slučaju vašeg proizvoda*, što uključuje istraživanje tržišta, analizu konkurenata i razumevanje načina na koji ciljani klijenti žive i rade.

ENGINEERING SOFTWARE PRODUCTS INTRO (VIDEO)

2.24 min. Ian Sommerville o softverskom proizvodu i svojoj novoj knjizi

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

SOFTWARE PRODUCT LINES FUNDAMENTALS: BASIC IDEAS AND TERMS (VIDEO)

45,19 min. Software Enngineering Institute, Carnegie Mellon Univeryity

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

▼ Poglavlje 2

Vizija proizvoda

UPOTREBA VIZIJE SOFTVERA

Vizija proizvoda je jednostavna i sažeta izjava koja definiše suštinu proizvoda koji se razvija

Vaša polazna tačka za razvoj proizvoda treba da bude neformalna „vizija proizvoda“. Vizija proizvoda je jednostavna i sažeta izjava koja definiše suštinu proizvoda koji se razvija. Objašnjava po čemu se proizvod razlikuje od drugih konkurentskih proizvoda. *Ova vizija proizvoda se koristi kao osnova za razvoj detaljnijeg opisa karakteristika i atributa proizvoda.* Kako se predlažu nove funkcije, trebalo bi da ih proverite u odnosu na viziju da biste bili sigurni da doprinose tome

Vizija proizvoda treba da odgovori na tri osnovna pitanja:

1. Koji je proizvod koji predlažete da razvijete? Po čemu se ovaj proizvod razlikuje od konkurentskih proizvoda?
2. Ko su ciljni korisnici i kupci proizvoda?
3. Zašto bi kupci trebali kupiti ovaj proizvod?

Potreba za prvim pitanjem je očigledna — pre nego što počnete, morate da znate šta ciljate. Ostala pitanja se tiču komercijalne održivosti proizvoda. Većina proizvoda je namenjena za upotrebu od strane kupaca van razvojnog tima. Morate razumeti njihovu pozadinu da biste kreirali održiv proizvod koji će ovi kupci smatrati privlačnim i koji će biti voljni da kupe.

Ako na internetu pretražujete „viziju proizvoda“, naći ćete nekoliko varijanti ovih pitanja i šablona za izražavanje vizije proizvoda. Bilo koji od ovih šablona se može koristiti. Jedan od šablona za izradu vizije (Džefrij Mur) koristi strukturisan pristupa pisanju vizije proizvoda zasnovanog na ključnim rečima:

- **ZA** (ciljni kupac)
- **KO** (izjava o potrebi ili mogućnosti)
- **NAZIV PROIZVODA** je (kategorija proizvoda)
- **KOJI** (ključna prednost, ubedljiv razlog za kupovinu)
- **ZA RAZLIKU** (primarna konkurentna alternativa)
- **NAŠ PROIZVOD** (izjava o primarnoj diferencijaciji)

PRIMER KREIRANJE VIZIJE

Početna ideja se usavršava u fazama kako se prikuplja više informacija i razvojni tim raspravlja o praktičnosti implementacije proizvoda

Evo Primera korišćenja ovog šablona za formiranje sledeće vizije proizvoda:

ZA odeljenje marketinga i prodaje srednje veličine kompanije KOJA ima potrebu da njen proizvod ima osnovnu CRM funkcionalnost. CRM-Inovator je veb-bazirana usluga **KOJA** obezbeđuje praćenje prodaje, generisanje potencijalnih klijenata i funkcije podrške prodajnim predstavnicima koje poboljšavaju odnose sa kupcima na kritičnim tačkama dodira. **ZA** razliku od drugih usluga ili paketnih softverskih proizvoda, **NAS** proizvod pruža veoma sposobne usluge po umerenoj ceni.

Možete videti kako ova vizija odgovara na ključna pitanja koja sam gore identifikovao:

1. **ŠTA:** Usluga zasnovana na vebu koja pruža funkcije praćenja prodaje, generisanja potencijalnih klijenata i podrške prodajnim predstavnicima. Informacije se mogu koristiti za poboljšanje odnosa sa kupcima.
2. **KO:** Proizvod je namenjen srednjim preduzećima kojima je potreban standardni softver za upravljanje odnosima sa klijentima.
3. **ZAŠTO:** Najvažnija razlika proizvoda je u tome što pruža sposobne usluge po umerenoj ceni. Biće jeftinije od alternativnih proizvoda.

Veliki deo mitologije okružuje vizije softverskih proizvoda. Za uspešne softverske proizvode za potrošače, mediji vole da predstavljaju vizije kao da proizilaze iz „Eureka trenutka“ kada osnivači kompanije imaju „odličnu ideju“ koja menja svet. Ovaj pogled previše pojednostavljuje trud i eksperimentisanje koji su uključeni u prečišćavanje ideje o proizvodu.

Vizije proizvoda za uspešne proizvode obično se pojavljuju nakon mnogo rada i diskusija. Početna ideja se usavršava u fazama kako se prikuplja više informacija i razvojni tim raspravlja o praktičnosti implementacije proizvoda. Nekoliko različitih izvora informacija doprinosi viziji proizvoda (Tabela 2.1).

INFORMACIONI IZVORI ZA RAZVOJ VIZIJE PROIZVODA

Izvori: Iskustvo u domenu, iskustvo proizvoda, iskustvo kupaca i izgradnja prototipa

Tabela 2.1 Informacioni izvori za razvoj vizije proizvoda

Informacioni izbor	Objašnjenje
Iskustvo u domenu	Programeri proizvoda mogu raditi u određenoj oblasti (recimo marketing i prodaja) i razumeti softversku podršku koja im je potrebna. Oni mogu biti frustrirani nedostacima u softveru koji koriste i vide mogućnosti za poboljšani sistem
Iskustvo proizvoda	Korisnici postojećeg softvera (kao što je softver za obradu teksta) mogu da vide jednostavnije i bolje načine pružanja uporedivih funkcionalnosti i da predlože novi sistem koji to implementira. Novi proizvodi mogu iskoristiti prednosti nedavnog tehnološkog razvoja kao što su govorni interfejsi.
Iskustvo kupaca	Programeri softvera mogu imati opsežne diskusije sa potencijalnim kupcima proizvoda kako bi razumeli probleme sa kojima se suočavaju; ograničenja, kao što je interoperabilnost, koja ograničavaju njihovu fleksibilnost za kupovinu novog softvera; i kritične attribute softvera koji su im potrebni.
Izrada prototipa i „igranje“	Programeri mogu imati ideju za softver, ali treba da razviju bolje razumevanje te ideje i šta bi moglo biti uključeno u njeno razvijanje u proizvod. Oni mogu razviti prototip sistema kao eksperiment i „igrati se“ sa idejama i varijacijama koristeći taj sistem prototipa kao platformu.

Slika 2.1 Informacioni izvori za razvoj vizije proizvoda [Sommerville 2021, Table 2.1]

PRIMER: VIZIJA ZA ILEARNING SISTEM

VLE sistemi su više usmereni na podršku administraciji učenja, a ne na samo učenje.

Virtuelna okruženja za učenje (VLE - Virtual Learning Environment), kao što je naš LAMS (Learning Activity Management System) koriste nastavnicima za distribuciju nastavnog materijala i zadataka. Studenti mogu i mogu preuzeti nastavne materijale i otpremiti urađene zadatke. Iako ime sugerise da su VLE fokusirani na učenje, VLE sistemi su više usmereni na podršku administraciji učenja, a ne na samo učenje. Oni pružaju neke funkcije za studente, ali nisu otvorena okruženja za učenje koja se mogu prilagoditi i prilagoditi potrebama određenog nastavnika.

Ovde ćemo izložiti kreiranje vizije za usavršeniji VLE sistem, koji bi imao za cilj da pruži fleksibilnu podršku procesu učenja. Razvojni tim je pogledao postojeće VLE-ove i razgovarao sa nastavnicima i studentima koji su ih koristili. Posetili su različite tipove škola od vrtića do fakulteta da bi ispitali kako koriste okruženja za učenje i kako nastavnici eksperimentišu sa softverom van ovih okruženja. Imali su opsežne razgovore sa nastavnicima o tome šta bi želeli da rade sa digitalnim okruženjem za učenje. Konačno su došli do izjave o viziji prikazane u tabeli 2.2.

U obrazovanju, nastavnici i studenti koji koriste okruženja za učenje nisu odgovorni za kupovinu softvera. Kupac je univerzitet. Službenik za nabavku treba da zna koje su koristi za organizaciju. Stoga je dodat poslednji paragraf izjavi o viziji u tabeli 3 da bismo jasno pokazali da postoje koristi za organizacije, kao i za pojedinačne studente.

Tabela 2.2 Izjava o viziji za iLearning sistem

ZA nastavnike i edukatore KOJI trebaju način da pomognu učenicima da koriste resurse i aplikacije za učenje zasnovane na vebu, sistem iLearn je otvoreno okruženje za učenje **KOJE** omogućava da skup resursa koje koriste odeljenja i učenici lako konfigurišu za ove učenike i odeljenja od strane nastavnika sebe.

ZA razliku od virtuelnih okruženja za učenje, kao što je Moodle, iLearn je u fokusu procesa učenja, a ne administracije i upravljanja materijalima, procenama i kursevima. **NAŠ** proizvod omogućava nastavnicima da kreiraju okruženja specifična za predmet i uzrast za svoje učenike koristeći bilo koje veb-bazirane resurse, kao što su video snimci, simulacije i pisani materijali koji su odgovarajući.

Škole i univerziteti su ciljni korisnici iLearn sistema jer će značajno poboljšati iskustvo učenja učenika po relativno niskoj ceni. On će prikupljati i obraditi analitiku učenika koja će smanjiti troškove praćenja napretka i izveštavanja.

Slika 2.2 Izjava o viziji za iLearning sistem

PRODUCT VISION BOARD INTRODUCTION (VIDEO)

6,68 min.

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

▼ Poglavlje 3

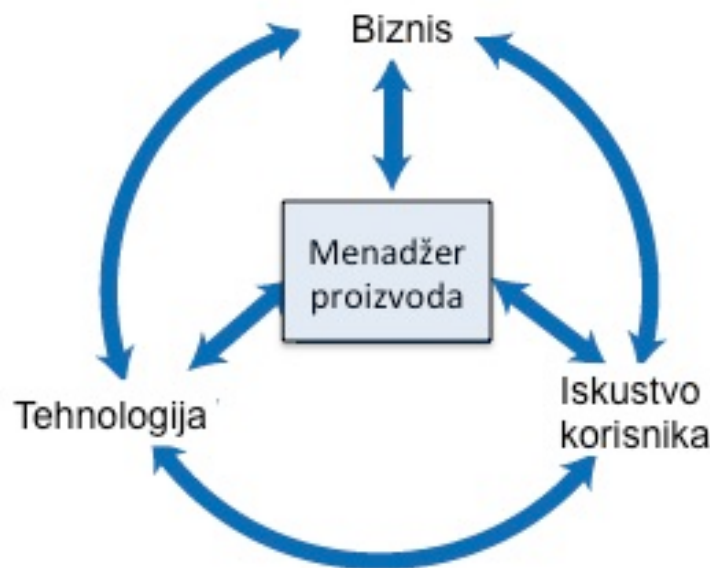
Upravljanje softverskim proizvodom

MENADŽER PROIZVODA

Menadžeri softverskih proizvoda uključeni su u realizaciju svih faza života proizvoda – od inicijalne koncepcije, preko razvoja vizije i implementacije do marketinga

Upravljanje softverskim proizvodima je poslovna aktivnost koja se fokusira na softverske proizvode koje kompanija razvija i prodaje. Menadžeri proizvoda (PM) preuzimaju ukupnu odgovornost za proizvod i uključeni su u planiranje, razvoj i marketing. Oni su interfejs, tj. veza između tima za razvoj softvera, šire organizacije i kupaca proizvoda. PM bi trebalo da budu punopravni članovi razvojnog tima kako bi mogli da komuniciraju poslovne zahteve i zahteve klijenata programerima softvera.

Menadžeri softverskih proizvoda uključeni su u realizaciju svih faza života proizvoda – od inicijalne koncepcije, preko razvoja vizije i implementacije do marketinga. Na kraju, oni donose odluke o tome kada proizvod treba da se povuče sa tržišta. Srednje i velike softverske kompanije mogu imati namenske PM; u manjim softverskim kompanijama, uloga menadžera proizvoda se obično deli sa drugim tehničkim ili poslovnim ulogama, tj. radnim mestima. Posao PM-a je da gleda prema spolja, tj. da bude orijentisan ka kupcima i potencijalnim kupcima proizvoda, a ne da se fokusira na softver koji se razvija. Previše je lako za razvojni tim da se uhvati u koštac sa detaljima „kul karakteristika“ softvera, do kojih veći kupaca verovatno nije ni stalo. Da bi proizvod bio uspešan, PM mora da obezbedi da razvojni tim primenjuje karakteristike koje pružaju stvarnu vrednost kupcima, a ne samo karakteristike koje su tehnički zanimljive.



Slika 3.1 Aspekti proizvoda na koje menadžer proizvoda mora da vodi računa

Zbog toga, menadžeri proizvoda moraju da se bave pitanjima poslovanja, tehnologije i korisničkog iskustva. Slika 4, ilustruje ove višestruke aspekte poslova premadžera proizvoda

TEHNIČKE INTERAKCIJE MENADŽERA PROIZVODA

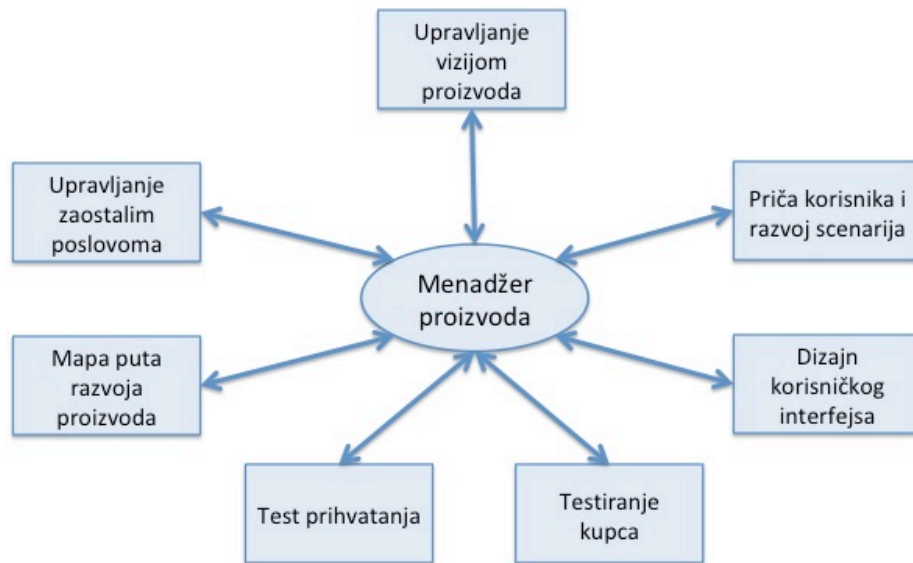
Menadžeri proizvoda moraju biti generalisti, sa tehničkim i komunikacijskim veštinama

Menadžeri proizvoda moraju biti generalisti, sa tehničkim i komunikacijskim veštinama. Pitanja poslovanja, tehnologije i kupaca su međusobno zavisna i PM-ovi moraju da razmotre sve:

1. **Poslovne potrebe:** Menadžeri moraju da osiguraju da softver koji se razvija ispunjava poslovne ciljeve i ciljeve kompanije softverskih proizvoda i njenih kupaca. Oni moraju da prenesu brige i potrebe kupaca i razvojnog tima menadžerima koji su odgovorni za poslovanja sa softverskim proizvodima. Oni rade sa višim menadžerima i sa marketinškim osobljem na planiranju rasporeda izdavanja proizvoda.
2. **Tehnološka ograničenja:** Menadžeri moraju da obaveste programere o tehnološkim problemima koji su važni za kupce. Ovo može uticati na raspored, cenu i funkcionalnost proizvoda koji se razvija.
3. **Iskustvo korisnika:** Menadžeri korisničkog iskustva treba da budu u redovnoj komunikaciji sa kupcima kako bi razumeli šta traže od proizvoda, tipove korisnika i njihovo iskustvo, kao i načine na koje se proizvod može koristiti. Njihovo iskustvo o mogućnostima kupaca je kritičan doprinos dizajnu korisničkog interfejsa proizvoda. PM takođe mogu da uključe kupce u alfa i beta testiranje proizvoda.

Zbog inženjerskog fokusa predmeta SE101, ne ulazimo u detalje o poslovnoj ulozi menadžera proizvoda ili njihovoj ulozi u oblastima kao što su istraživanje tržišta i finansijsko planiranje.

Umesto toga, koncentrišemo se samo na njihovu interakciju sa razvojnim timom. Menadžeri proizvoda mogu da komuniciraju sa razvojnim timom u sedam ključnih oblasti (slika 2).



Slika 3.2 Tehničke interakcije menadžera proizvoda

KLJUČNA ULOGA PM-A JE DA UPRAVLJAJU VIZIJOM PROIZVODA

Menadžeri proizvoda preuzmu vodeću ulogu u razvoju vizije proizvoda

Neki smatraju da bi menadžer proizvoda trebalo da bude odgovoran za razvoj vizije proizvoda. Velike kompanije mogu usvojiti ovaj pristup, ali je on često nepraktičan u malim softverskim kompanijama. U startapima, izvor vizije proizvoda često je originalna ideja osnivača kompanije. Ova vizija se često razvija mnogo pre nego što neko pomisli na imenovanje menadžera proizvoda. Očigledno, ima smisla da menadžeri proizvoda preuzmu vodeću ulogu u razvoju vizije proizvoda. Trebalo bi da budu u stanju da unesu informacije o tržištu i klijentima u proces. Međutim, potrebno je i da svi članovi tima budu uključeni u razvoj vizije proizvoda tako da svaki član tima može da podrži ono što je konačno dogovoreno. Kada tim „posедуje“ viziju, veća je verovatnoća da će svi raditi koherentno na ostvarenju te vizije. Ključna uloga PM-a je da upravlja vizijom proizvoda. Tokom procesa razvoja, promene neizbežno predlažu ljudi iz razvojnog tima i izvan njega. Menadžeri moraju da analiziraju i ocene predloge za ove promene u odnosu na viziju proizvoda. Moraju da provere da promene nisu u suprotnosti sa idejama oličenim u viziji proizvoda. Menadžeri proizvoda takođe moraju da obezbede da ne dođe do „pomeranja vizije“, u kome se vizija postepeno proširuje kako bi postala šira i manje fokusirana.

Razvoj mape puta proizvoda

Mapa puta proizvoda je plan za razvoj, izdavanje i marketing softverskog proizvoda. On postavlja važne ciljeve proizvoda i prekretnice, kao što su završetak kritičnih funkcija, završetak prve verzije za testiranje korisnika itd. Uključuje datume kada ove prekretnice treba da budu dostignute i kriterijume uspeha koji pomažu u proceni da li su ciljevi projekta postignuti. Mapa puta treba da sadrži raspored izdanja koji pokazuje kada će različita izdanja

softvera biti dostupna i ključne karakteristike koje će biti uključeno u svako izdanje softverskog proizvoda.

Razvoj mape puta proizvoda treba da vodi menadžer proizvoda, ali mora uključiti i razvojni tim, kao i menadžere kompanije i marketinško osoblje. U zavisnosti od vrste proizvoda, možda će morati da se ispoštuju važni rokovi da bi proizvod bio uspešan. Na primer, mnoge velike kompanije moraju doneti odluke o nabavkama pred kraj svoje finansijske godine. Ako želite da prodate novi proizvod takvim kompanijama, morate ga učiniti dostupnim pre toga.

KORISNIČKE PRIČE I UPRAVLJANJE ZAOSTALIM POSLOVIMA

Korisničke priče i scenariji se široko koriste za preciziranje vizije proizvoda kako bi se identifikovale karakteristike proizvoda. Zaostali poslovi određuju dalji proces razvoja.

Razvoj korisničkih priča i scenarija

Korisničke priče i scenariji se široko koriste za preciziranje vizije proizvoda kako bi se identifikovale karakteristike proizvoda. Oni su opisi na prirodnom jeziku svojstava proizvoda i koje bi korisnici možda želeli da proizvod poseduje. Koristeći ih, tim može da odluči koje funkcije treba da budu uključene i kako ove funkcije treba da rade. Mi nećemo se u ovom predmetu ovim baviti, jer je to oblast kojim će se baviti predmet SE202 Inženjersstvo zahteva koji ja planira za drugo godinu vaših studija.

Posao menadžera proizvoda je da razume kupce proizvoda i potencijalne kupce. Menadžeri proizvoda bi stoga trebalo da vode razvoj korisničkih scenarija i priča, koji bi trebalo da se zasnivaju na poznavanju oblasti i poslovanja korisnika. PM bi takođe trebalo da scenarije i priče koje predlože drugi članovi tima proslede kupcima kako bi proverili da li odražavaju ono šta bi ciljni korisnici proizvoda zaista želeli da rade sa softverskim proizvodom.

Upravljanje zaostalim poslovima razvoja proizvoda

U razvoju proizvoda, važno je da proces bude vođen „zaostalim poslovima razvoja proizvoda“ (backlog management). *Zaostali poslovi razvoja proizvoda je lista obaveza koja navodi šta treba da se uradi da bi se završio razvoj proizvoda.* Zaostatak u razvoju se dodaje i usavršava postepeno tokom procesa razvoja. O ovome će biti više reči u sledećoj lekciji koja se bavi agilnim metodima razvoja softvera.

Menadžer proizvoda igra ključnu ulogu kao autoritet za stavke zaostalih aktivnosti razvoja proizvoda koje treba da imaju prioritet za razvoj. PM takođe pomažu da se detaljnije preciziraju stavke zaostalih zadataka, kao što je, na primer, „implementacija automatskog čuvanja“, pri svakoj iteraciji projekta. Ako se daju predlozi za izmene, na menadžeru proizvoda je da odluči da li zaostatak proizvoda treba preurediti da bi se dali prioritet predloženim promenama.

TESTIRANJE PROIZVODA I DIZAJN KORISNIČKOG INTERFEJSA

Testiranje prihvatanja, testiranje proizvoda od strane kupca i dizajn korisničkog iskustva

Testiranje prihvatanja

Testiranje prihvatanja je proces provere da li izdanje softvera ispunjava ciljeve postavljene u mapi puta proizvoda i da je proizvod efikasan i pouzdan. Menadžeri proizvoda treba da budu uključeni u razvoj testova svojstava proizvoda koji odražavaju način na koji kupci koriste proizvod. Oni mogu da rade kroz scenarije korišćenja kako bi proverili da li je proizvod spreman za puštanje kupcima.

Testovi prihvatanja se usavršavaju kako se proizvod razvija, a proizvodi moraju proći ove testove pre nego što budu pušteni kupcima.

Testiranje proizvoda od strane kupaca

Testiranje proizvoda od strane kupaca uključuje objavljivanje proizvoda postojećim i potencijalnim kupcima i dobijanje povratnih informacija od njih o svojstvima proizvoda, njegovoj upotrebljivosti i uklapanju proizvoda u njihovo poslovanje. Menadžeri proizvoda su uključeni u odabir kupaca koji bi mogli biti zainteresovani da učestvuju u procesu testiranja novog izdanja softverskog proizvoda i rade sa njima tokom tog procesa. Oni moraju da obezbede da kupac može da koristi proizvod i da proces testiranja koje sprovodi korisnik prikuplja korisne informacije za razvojni tim.

Dizajn korisničkog interfejsa

Korisnički interfejs (User Interface - UI) proizvoda je kritičan u komercijalnom prihvatanju softverskog proizvoda. Tehnički odlični proizvodi verovatno neće biti komercijalno uspešni ako je korisnicima teško da ih koriste ili ako njihov korisnički interfejs nije kompatibilan sa drugim softverom koji koriste. Dizajn korisničkog interfejsa predstavlja izazov za male razvojne timove jer je većina korisnika manje tehnički obučena od programera softvera. Programerima je često teško da predvide probleme koje korisnici mogu imati sa softverskim proizvodom.

Menadžeri proizvoda treba da razumeju korisnička ograničenja i da se ponašaju kao surogat korisnici u svojim interakcijama sa razvojnim timom. Trebalo bi da procene karakteristike korisničkog interfejsa dok su razvijene da bi proverile da li ove funkcije nisu nepotrebno složene ili da primoravaju korisnike da rade na neprirodan način. PM mogu da organizuju potencijalne korisnike da isprobaju softver, komentarišu njegov korisnički interfejs i pomognu u dizajniranju poruka o grešci i sistema pomoći.

THE DIFFERENCE BETWEEN PRODUCT, PROGRAM AND PROJECT MANAGEMENT (VIDEO)

8,36 min.

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

▼ Poglavlje 4

Izrada prototipova softverskog proizvoda

SVRHA PRIMENE PROTOTIPA SOFTVERSKOG PROIZVODA

Vaš cilj bi trebalo da bude da imate radnu verziju vašeg softvera koja se može koristiti za demonstriranje njegovih ključnih karakteristika.

Izrada prototipa proizvoda je proces razvoja rane verzije proizvoda kako biste testirali vaše ideje i ubedili sebe i finansijere kompanije da vaš proizvod ima stvarni tržišni potencijal. Koristite prototip proizvoda da biste proverili da li je ono što želite da uradite izvodljivo i da biste demonstrirali svoj softver potencijalnim klijentima i finansijerima. Prototipovi vam takođe mogu pomoći da razumete kako da organizujete i strukturirate konačnu verziju vašeg proizvoda.

Možda ćete moći da napišete inspirativnu viziju proizvoda, ali vaši potencijalni korisnici mogu da stvarno prihvate na vaš proizvod samo kada vide funkcionalnu verziju vašeg softvera. Oni mogu da istaknu šta im se sviđa, a šta im se ne dopada i da daju predloge za nove funkcije. Investitori raizičnog kapitala (venture capitalists), kojima se možete obratiti za finansiranje, obično insistiraju da vide prototip proizvoda pre nego što se obavežu da će podržati startup kompaniju. Prototip igra ključnu ulogu u ubeđivanje investitora da vaš proizvod ima komercijalni potencijal.

Prototip takođe može pomoći u identifikaciji osnovnih softverskih komponenti ili usluga i testiranju tehnologije. Možda ćete otkriti da je tehnologija koju ste planirali da koristite neadekvatna i da morate revidirati svoje ideje o tome kako da implementirate softver. Na primer, možete otkriti da projektno rešenje arhitekture softverskog proizvoda, a koje ste izabrali za prototip ne može da podnese očekivano opterećenje sistema, tako da morate ponovo projektovati celokupnu arhitekturu proizvoda.

Izgradnja prototipa bi trebalo da bude prva stvar koju radite prilikom razvoja softverskog proizvoda. Vaš cilj bi trebalo da bude da imate radnu verziju vašeg softvera koja se može koristiti za demonstriranje njegovih ključnih karakteristika. Kratak razvojni ciklus je kritičan. Trebalo bi da imate za cilj da uspostavite pokreneti i dokazljiv sistem za četiri do šest nedelja. Naravno, morate da se potrudite da biste to uradili, tako da možete izabrati da ignorišete probleme kao što su pouzdanost i performanse i da radite sa rudimentarnim korisničkim interfejsom.

IZRADA PROTOTIPA U DVE FAZA

Prototip koristite radi demonstracije izvodljivosti softvera ili radi demonstracije klijentima.

Ponekad je izrada prototipa proces u dve faze:

1. **Demonstracija izvodljivosti:** Kreirate izvršni sistem koji demonstrira nove ideje u vašem proizvodu. Ciljevi u ovoj fazi su da vidite da li vaše ideje zaista funkcionišu i da pokažete finansijerima i menadžmentu kompanije da su karakteristike vašeg proizvoda bolje od onih kod konkurenata.
2. **Demonstracija klijentima:** Uzimate postojeći prototip kreiran da pokažete izvodljivost i proširite ga svojim idejama za specifične karakteristike korisnika i kako se one mogu realizovati. Pre nego što razvijete prototip za kupca, potrebno je da uradite neke studije o korisnicima i da imate jasnu predstavu o vašim potencijalnim korisnicima i scenarijima korišćenja softverskog proizvoda koji razvijate. O ovome ćete se detaljno upoznati i izučavati u predmetu SE202 Inženjerstvo zahteva u drugoj godini studija.

Uvek treba da koristite tehnologiju koju poznajete i razumete da biste razvili prototip kako ne biste morali da trošite vreme na učenje novog jezika ili okvira. Ne morate da projektujete robusnu softversku arhitekturu. Možete da izostavite bezbednosne funkcije i kod za proveru da biste obezbedili pouzdanost softvera. Međutim, **preporučuje se da za prototipove uvek koristite automatizovano testiranje i upravljanje kodom**. Oni su obrađeni u predmetu SE321 Obezbeđenje kvaliteta, testiranje i evolucija softvera koji je u programu treće godine vaših studija.

Ako razvijate softver bez eksternog kupca, kao što je softver za neku grupu istraživača, može se desiti da je dovoljno da samo razvijete prototip sistema, koji može biti prihvatljiv za ovakve specifične kupce. Možete razviti i poboljšati prototip u skladu sa razvojem vašeg razumevanja problema. Međutim, *čim imate eksterne korisnike svog softvera, uvek treba da razmišljate o svom prototipu kao o sistemu koji se „odbacuje“*. Neizbežni kompromisi i prečice koje pravite da biste ubrzali razvoj rezultiraju prototipovima koji se sve sve teže menjaju i evoluiraju kako bi uključili nove funkcije. Dodavanje sigurnosti i pouzdanosti zbog toga može da postane praktično nemoguće.

▼ Poglavlje 5

Vežba

ZADATAK ZA GRUPNI RAD - 1. ČAS

Ove zadatke radite zajedno na vežbi i diskutujte rezultate.

U prvih 20 minuta, svaki student treba da pretraživanjem preko Interneta nađe iskaz o viziji nekog softverskog proizvoda i da pripremi svoj komentar i ocenu uspešnosti iskaza.

U drugom delu časa, svaki student izlaže viziju proizvoda koju je našao ostalim studentima, i na kraju studenti se izjašnjavaju koja vizija im se najviše sviđa, a u skladu sa preporukama datim na predavanju za izradu iskaza o viziji.

ZADACI ZA INDIVIDUALNI RAD - 2. ČAS

Zadaci za individualni rad

U okviru drugog časa vežbi, a i u okviru vašeg kućnog studiranja, uradite sledeće zadatke:

1. Ukratko opišite fundamentalne razlike između softverskog inženjerstva zasnovanog na projektu i na proizvodu.
2. Koje su tri bitne razlike između softverskih proizvoda i linija softverskih proizvoda.
3. Na osnovu primera vizije projekta za sistem iLearn, identifikujte delove vizije na koje e odnose ključne reči: ŠTA, KO i ZAŠTO za taj softverski proizvod.
4. Zašto menadžeri softverskih proizvoda moraju biti generalisti, sa nizom veština, a ne samo tehnički stručnjaci?
5. Vi ste menadžer softverskih proizvoda za kompaniju koja razvija obrazovne softverske proizvode zasnovane na naučnim simulacijama. Objasnite zašto je važno da se razvije mapa puta proizvoda kako bi finalni proizvodi bili dostupni u prva tri meseca u godini.
6. Zašto bi trebalo da implementirate prototip pre nego što počnete da razvijate novi softverski proizvo

▼ Poglavlje 6

Zaključak

ZAKLJUČAK

Ključne poruke ove lekcije

1. Softverski proizvodi su softverski sistemi koji uključuju opštu funkcionalnost koja će verovatno biti korisna širokom spektru kupaca.
2. U softverskom inženjerstvu zasnovanom na proizvodu, ista kompanija je odgovorna za odlučivanje i o karakteristikama koje bi trebalo da budu deo proizvoda i o implementaciji ovih funkcija.
3. Softverski proizvodi mogu biti isporučeni kao samostalni proizvodi koji rade na računarima korisnika, hibridni proizvodi ili proizvodi zasnovani na uslugama. U hibridnim proizvodima, neke funkcije se implementiraju lokalno, a drugima se pristupa sa Interneta. Svim funkcijama se pristupa sa daljine u proizvodima zasnovanim na uslugama.
4. Vizija proizvoda sažeto opisuje šta treba da se razvije, ko su ciljni kupci proizvoda i zašto bi kupci trebalo da kupe proizvod koji razvijate.
5. Iskustvo u domenu, iskustvo proizvoda, korisničko iskustvo i eksperimentalni softverski prototip mogu doprineti razvoju vizije proizvoda.
6. Ključne odgovornosti menadžera proizvoda su da poseduju viziju proizvoda, razviju mapu puta proizvoda, kreiraju korisničke priče i scenarije, upravljaju zaostatom proizvoda, sprovode testiranje kupaca i prihvatanja i dizajniraju korisnički interfejs.
7. Menadžeri proizvoda rade na interfejsu između preduzeća, tima za razvoj softvera i kupaca proizvoda. Oni olakšavaju komunikaciju između ovih grupa.
8. Uvek treba da razvijete prototip proizvoda da biste poboljšali sopstvene ideje i da biste potencijalnim kupcima demonstrirali planirane karakteristike proizvoda.

LITERATURA

Ova lekcija je pripremljena korišćenjem sledećeg udžbenika:

1. [Sommerville 2021] Ian Sommerville, Engineering Software Products - An Introduction to Modern Software Engineering, Chapter 1, Global Edition, Pearson (2021)