

UTJECAJ 37SIGNALS MODELA UPRAVLJANJEM RAZVOJA SOFTVERSKIH PROIZVODA NA KORISNIKOVU PERCEPCIJU KVALITETE

Županović, Ivan

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, Faculty of economics Split / Sveučilište u Splitu, Ekonomski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:124:574504>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-28**

Repository / Repozitorij:

[REFST - Repository of Economics faculty in Split](#)



SVEUČILIŠTE U SPLITU
EKONOMSKI FAKULTET

DIPLOMSKI RAD

**UTJECAJ 37SIGNALS MODELA
UPRAVLJANJEM RAZVOJA SOFTVERSKIH
PROIZVODA NA KORISNIKOVU PERCEPCIJU
KVALITETE**

Mentor:

prof. dr. sc. Neven Šerić

Student:

Ivan Županović

Split, rujan, 2022.

1. UVODNE NAZNAKE.....	4
1.1. Predmet i problem istraživanja	4
1.2. Ciljevi i metodologija istraživanja	10
1.3. Specifičnosti globalnog tržišta softvera.....	12
1.4. Trendovi i inovacije na tržištu softvera	12
1.5. Razvoj tvrtke 37signals	14
2. GLOBALNO TRŽIŠTE SOFTVERA	16
2.1. Odnosi ponude i potražnje	16
2.2. Konkurentno okruženje	18
2.3. Tehnologije na tržištu softvera.....	19
3. KVALITETA I ZADOVOLJSTVO	21
3.1. Definiranje kvalitete	21
3.2. Percipirana kvaliteta	22
3.3. Odnos kvalitete i zadovoljstva	23
4. MODELI I PRISTUPI UPRAVLJANJA RAZVOJEM SOFTVERSKIH PROIZVODA	24
4.1. Vodopadni model upravljanja razvojem softverskog proizvoda	24
4.1.1. Tradicionalni pristup	24
4.1.2. Specifičnosti Vodopadnog modela	26
4.2. <i>Scrum</i> model upravljanja razvojem softverskog proizvoda.....	27
4.2.1. Agilni pristup	27
4.2.2. Specifičnosti <i>Scrum</i> modela	29
5. MODEL UPRAVLJANJA RAZVOJEM SOFTVERSKOG PROIZVODA TVRTKE 37SIGNALS.....	32
5.1. Povijest i razvoj modela	32
5.2. Faze razvoja novog softverskog proizvoda prema 37signals modelu	33
5.3. Tržišna iskustva.....	37

5.4.	Mogućnosti primjene modela	40
6.	PRIMARNO ISTRAŽIVANJE.....	41
6.1.	Uvodne naznake	41
6.2.	Analiza podataka iz strukturiranog intervjua.....	42
6.2.1.	Zaključak strukturiranog intervjua	45
6.3.	Usporedba proizvoda <i>Ruby on Rails</i> i <i>Basecamp</i> s konkurentnim proizvodima	45
6.3.1.	<i>Ruby on Rails</i>	46
6.3.2.	<i>Basecamp</i>	51
6.3.3.	Analiza utjecaja broja zaposlenih	54
6.4.	Rezultati istraživanja	55
7.	ZAKLJUČAK.....	57
LITERATURA	61
	Literatura	61
	Web izvori:	64
	Popis slika	69
	Popis tablica	70
SAŽETAK.....	71
SUMMARY.....	71

1. UVODNE NAZNAKE

1.1. Predmet i problem istraživanja

Softveri, odnosno računalne korisničke aplikacije tijekom posljednjeg desetljeća doživljavaju značajne promjene. Te se promjene odnose kako na industriju softvera, tako i na tehnološka rješenja (Mehta et. al. 2021). Tehnološka rješenja proizvoda novih tehnologija temelj su njihove diferencijacije i konkurentnosti. Alati, iskustva i praksa čimbenici su koji pridonose poboljšanoj automatizaciji, širini primjene i skalabilnosti. Razlog tome je što su isti međusobno usklađeniji, primjerenije integrirani i bolje prilagođeni različitim kategorijama korisnika, kupaca i naručitelja (Mehta et. al. 2021). Razvijaju se nove platforme u *oblaku* koje upravljaju temeljnim hardverskim i nižim razinama softverskih sustava. Poslovni subjekti, odnosno zaposlenici tvrtki koji softvere koriste za upravljanje svim poslovnim aktivnostima još uvijek se sporo i često uz mnoštvo problema prilagođavaju novim tehničkim rješenjima (Mehta et. al. 2021). Nova, bolja tehnička rješenja, odnosno tehnološki savršeniji softveri su imperativ tržišnog opstanka tvrtki kojima je to primarna djelatnost. Zbog toga proizvođači softvera sve više vremena troše na procjene, analize i donošenje odluka tijekom razvoja i upravljanja novim proizvodima (Mehta et. al. 2021).

Iz perspektive korisnika, od iznimne važnosti je brzo ovladati novim softverskim rješenjima i inovacijama postojećih proizvoda. Iz perspektive proizvođača softvera od iznimne je važnosti razumjeti nove potrebe ciljanih kupaca. U tom cilju tvrtke proizvođači softvera trebaju pratiti nove tehnološke trendove ali i marketinške koncepte u osmišljavanju, razvoju i komercijalizaciji novih rješenja. Tehnologija kontinuirano napreduje, pa je potrebno pratiti spoznaje novih istraživanja o relevantnim tržišnim trendovima. Identificiranje novih trendova pomaže premoštenju raskoraka između nalaza istraživanja i vlastite poslovne prakse, te nudi uvide u budućnosti softverske industrije (Mehta et. al. 2021).

Tržište softvera obuhvaća razvoj i upravljanje softverskim aplikacijama, složene

poslovne softvere, softvere u funkciji rasta produktivnost, infrastrukturne i druge softverske sustave. Obzirom na brojnost kategorija različitih proizvoda, ovo tržište se danas smatra jednim od najvećih i najkompleksnijih (Mehta et. al. 2021). Dinamika rasta ovoga tržišta se sagledava rastom ukupne vrijednosti transakcija. Veličina globalnog softverskog tržišta 2016. godine se procjenjivala na 417,63 milijardi američkih dolara, a do danas je dodatno porasla za 200 milijardi američkih dolara. Točnije, za 2022 godinu ovo tržište je procijenjeno na 608,68 milijardi dolara, s tim da se očekuje daljnji rast za dodatnih 200 milijardi dolara tijekom sljedeće četiri godine (Statista, 2022). Izneseni podaci ukazuju da se radi o dinamičnom, brzorastućem tržištu, što je posljedica eksponencijalnog trenda rasta konkurencije, odnosno ukupne ponude (Statista, 2022). O ovome obilježju tržišta treba posebno voditi računa u procesima promišljanja i planiranja razvoja i upravljanja novim proizvodima (Šerić, 2016).

Općepoznato je kako se na ovome tržištu tvrtka *37signals* ističe jedinstvenom poslovnom filozofijom, organizacijskom kulturom i originalnim poslovnim modelima. U ovome istraživanju fokus je na njihovom modelu razvoja i upravljanja novim proizvodima. Tvrtka *37signals* je dokaz da na ovom specifičnom tržištu poslovni subjekt ne mora spadati u kategoriju velikih poduzeća da bi ostvario značajan utjecaj na tržišne odnose i tržišne trendove. Ovu tezu argumentiraju njihovi projekti prvih web-aplikacija komercijaliziranih u obličju softverskih usluga. Posebno treba istaknuti njihov patent *Ruby on Rails*. Tvrtka je ujedno izdavač vlastitih za tržište utjecajnih knjiga o svojim poslovnim iskustvima. Na listi najprodavanijih knjiga *The New York Timesa* o novim tehnologijama dva su njihova djela o organizaciji virtualnog vođenja tvrtke na daljinu (*Basecamp* platforma za *online* poslovanje) i drugim proizvodima. U knjigama predstavljena softverska rješenja komercijalizirana su desetljeće prije nego su slični proizvodi pojavili na tržištu (Basecamp, 2022).

Tvrtka *37signals* je osnovana 1999. godine u Sjedinjenim Američkim Državama. Osnivači su bili Jason Frieda, Carlos Segura i Ernest Kima. Današnji naziv tvrtke je identičan nazivu pod kojim je osnovana premda je 2014. godine bilo promijenjeno ime u *Basecamp*, čime su dodatno promovirali svoj tadašnji temeljni softverski proizvod. Stari naziv tvrtke je vraćen 2022. godine (Basecamp, 2022). Tijekom dva desetljeća postojanja tvrtka je ostavila značajan trag u softverskoj industriji. U početku se bavila web-dizajnom. Krajem devedesetih godina dvadesetog stoljeća o toj se kategoriji proizvoda rijetko

govorilo. Šest godina kasnije patentirali su svoj prvi revolucionarni izum - programsku platformu *Ruby on Rails* koja je unatoč rastućoj konkurenciji do danas ostala jedna od najpopularnijih programskih platformi za web aplikacije. Tvrtka je i danas na tržištu prepoznata kao lider u području organizacije poslovanja na daljinu, što je počela prakticirati godinama prije konkurenata. Poslovni razvoj i asortiman ponude rezultat su originalne poslovne filozofije i oslanjanja na vlastiti model razvoja i upravljanja proizvodom (Basecamp, 2022). Ova činjenica je predstavljala temeljni poticaj za odabir teme rada.

Najznačajniji proizvodi tvrtke *37signals* su:

- *Basecamp* - prvi komercijalni proizvod tvrtke, web-alat za upravljanje projektima, timsku komunikaciju i rad na daljinu;
- *Campfire* - poslovno orijentirana usluga online chata, kasnije integrirana u *Basecamp*;
- *Highrise* - proizvod za upravljanje odnosima s kupcima (CRM);
- *Ruby on Rails* - besplatna programska platforma za web aplikacije;
- *HEY* – originalna vrhunska usluga e-pošte.

Specifičnost tvrtke je svega 65 zaposlenika, što je iznimno malen broj u usporedbi s konkurentima koji razvijaju tehnološka rješenja u istim kategorijama proizvoda najviše klase (Basecamp, 2022). Usporedbe radi u tvrtki *ClickUp* koja je vlasnik glavnog konkurentnog proizvoda *Basecamp*-u zaposleno je 800 ljudi (ClickUp, 2022).

Polazeći od predstavljenog predmeta istraživanja definiran je i problem istraživanja. Konkretno, to je doprinos modela upravljanja razvojem softverskih proizvoda koji prakticira tvrtka *37signals* na korisnikovu percipiranu kvalitetu njihovih softverskih rješenja.

Usprkos neupitnoj tehničkoj kvaliteti suvremenih softverskih proizvoda u korisničkim kontekstu su učestale primjedbe. Praktično, tek jedan od pet projekata u konačnici rezultira potpunim ispunjenjem očekivanja ciljnih kupaca (Bevanda, Sinković, 2009). Jedna od važnih ocjena ispunjenja očekivanja je zadovoljstvo kupaca. Zbog specifičnosti softvera kao proizvoda, jedan od problema upravljanja razvojem softverskog proizvoda je činjenica da često ni kupac ni proizvođač nemaju jasnu predodžbu konačnog rješenja. I u slučajevima kada nema dilema po tom pitanju praksa ukazuje da se predodžbe korisnika, o tome kako

softverski proizvod treba izgledati, tijekom životnog ciklusa proizvoda mijenjaju. I operativne potrebe korisnika softvera se vremenom mijenjaju, pa postojeće rješenje treba omogućavati širu dogradnju i unapređenje. Relativna važnost pojedinih obilježja softverskog proizvoda varira od proizvoda do proizvoda, ali i ovisno o specifičnostima potreba kupca (naručitelja). S druge strane, poslovna iskustva mnogih ukazuju da su niži troškovi uz osiguranje više korisničke kvalitete softverskog proizvoda jedno od strateških poslovnih opredjeljenja u ovoj industriji (Bevanda, Sinković, 2009).

“Protekom vremena i razvojem znanosti o kvaliteti došlo je do pojave velikog broja različitih definicija kvalitete. U početku se kvaliteta jednostavno definirala kao percipirano zadovoljstvo kupca, no razvojem tehnologije i porastom zahtjeva korisnika prema poduzećima kvaliteta se počela doživljavati na različite načine.” (Lazibat i Baković, 2020 navedeno u Buntak et al., 2021, str 9.)

„Ono što jedna osoba percipira kao kvalitetno druga osoba može vidjeti kao prosječno ili ispodprosječno. Naime, elementi koji uključuju navike, stavove, iskustva, set vrijednosti ili društvenu klasu pojedinca, samo su neki čimbenici koji utvrđuju način na koji osoba može percipirati kvalitetu.“ (Marinković, Senić, 2012, str. 14.)

Gronroos (1987) ističe kako se percipirana kvaliteta usluge može definirati kao opći sud ili stav prema usluzi, koji proizlazi iz usporedbe očekivanja s percepcijom klijentele o performansama aktualne usluge. Iako se ova definicija odnosi na percipiranu kvalitetu usluge, primjenjiva je i za softverski proizvod.

Softver kao proizvod je specifičan po mnogim obilježjima. Među tim proizvodima je tek manji broj onih koji direktno doprinose percepciji kvalitete u svijesti korisnika. Svaki softver je apstraktan rezultat mentalnog rada i vizije programera zapisan u obliku koda. Softver je tijekom razvoja neopipljivo dobro. Opipljivim se mogu učiniti neka njegova obilježja temeljem apstraktnih opisa kojima će se predstaviti određene konkretne mogućnosti, podaci, funkcije, kontrolni alati, korisnička sučelja i druge sastavnice ovoga proizvoda koje je moguće vizualno prezentirati na ekranu. Softver kao integrirani proizvod je nematerijalan i neopipljiv, pa je njegova usporedna mjerljivost kompleksna. Kvaliteta konačnog softverskog rješenja za korisnika se ne može uvijek garantirati. Teško ju je kvantificirati, a prvi dojam često razočara. Obzirom na kompleksnost procesa razvoja softverskog proizvoda troškovi i procjene rokova isporuke često su nerealni. Kvaliteta

softvera za korisnika se može sagledati u potpunosti tek kada ga se počne koristiti. U nekim kategorijama proizvoda, npr. onih za nadzor kritičnih sustava, eventualne slabosti mogu generirati iznimne financijske i druge gubitke (gubitak dijela podataka, materijalne štete, a u iznimnim slučajevima i ugrožavanje ljudskih života).

Većina problema vezanih uz softverski proizvod je posljedica činjenice da se ni provjerene programerske prakse ne mogu jednostavno implementirati u razvoju svih novih proizvoda (Stankić, 2015).

Upravljanje proizvodom podrazumijeva kompleksne aktivnosti koje započinju kreiranjem ideje o novom proizvodu, a u kasnijim fazama simulacije i predviđanja. Preduvjet primjerenj provedbi predviđenih aktivnosti su prethodna iskustva u upravljanju razvojem novih proizvoda. Složenost procesa upravljanja proizvodom podrazumijeva angažman odabranih stručnjaka i značajna financijska ulaganja (Šerić, 2016). Slijedom toga svaki projekt razvoja novog proizvoda podrazumijeva financijski rizik. Zbog te činjenice vrijedno je i korisno analizirati različite modele upravljanja razvojem proizvoda kako bi se opredijelilo za optimalan. Tvrtka *37signals* u svojoj poslovnoj praksi odbacuje mnoge tradicionalne paradigme, pa tako i upravljanje razvojem proizvoda provodi na sebi svojstven način. Temeljno pravilo kojeg se pridržavaju u modelu je načelo jednostavnosti. Utemeljeno je na njihovoj poslovnoj filozofiji, predstavljenoj u knjizi *Rework* (2010). U drugoj knjizi, *Getting real: the smarter, faster, easier way* (2006) predstavili su vlastitu praksu upravljanja razvojem softverskih proizvoda. Tako ističu težnju skraćivanja vremena razvoja, preskakanje nekih procedura, fokus na obilježja od posebnog značaja za korisnike, jednostavniju strukturu proizvoda, prilagodljivost nadogradnjama, prilagođenost zaslonima kakve većinom koriste ciljani kupci i dr. Po ovome je evidentno da u upravljanju razvojem proizvoda polaze iz perspektive ciljanog kupca. Stječe se dojam kao da je kasnija faza - testiranje, redosljedom utvrđena za početak procesa. Praktično, ovakav pristup omogućuje jednostavnije preinake sučelja softverskog proizvoda (Fried, Heinemeier Hansson, Linderman, 2006). U njihovoj praksi su učestala i ponavljanja faza u razvoju novog proizvoda koje su posebno važne za konačni koncept (testiranje, dodatne prilagodbe, poboljšanja i sl.). U primarnom fokusu tijekom razvoja novoga proizvoda je sve ono što može imati direktne reperkusije na zadovoljstvo korisnika, te se manja pažnja posvećuje svemu što značajno ne pridonosi percepciji kvalitete u svijesti korisnika. (Fried, Heinemeier Hansson, Linderman, 2006). To su tek neke činjenice spoznate tijekom izviđajnog

istraživanja specifičnosti modela upravljanja razvojem novog proizvoda koji prakticira ova tvrtka (Business Insider, 2022). Premda svaki softverski proizvod omogućava prilagodbe, korisnikova percepcija kvalitete će ovisiti o upoznatosti programera s poslovnim procesima za čiju se potporu isti razvija. U literaturi se ovi softveri nazivaju prilagođenim softverima, odnosno softverima po mjeri klijenta (ITenterprise, 2022). Za razliku od standardnih softvera za širi segment korisnika, softver po narudžbi se dizajnira za konkretne poslovne potrebe ciljanog klijenta. Radi se personaliziranom proizvodu koji bi trebao udovoljiti svim očekivanjima kupca (ITenterprise, 2022).

Kada se govori o tradicionalnom pristupu razvoja novog softverskog proizvoda kao najkorišteniji ističe se vodopani model koji obuhvaća pet faza (Bassil, 2012):

- Analiza zahtjeva - koja daje detaljan opis funkcionalnosti softvera koji se razvija
- Dizajn - proces kojim se planira način rješavanja problema softvera
- Implementacija - podrazumijeva pisanje koda odnosno stvaranje operativne aplikacije
- Testiranje - provjera funkcioniranja softvera kao cjeline
- Održavanje - ispravljanje grešaka i poboljšanje performansi i kvalitete

Nadalje vrijedno je spomenuti pristup agilnog razvoja softverskih proizvoda. Naime 2001. godine skupina softverskih inženjera objavila je „Proglas o metodi agilnog razvoja softvera“ (2001) u kojem navode principe temeljem kojih bi trebalo razvijati softverski proizvod. Proglas se zasniva na dvanaest ključnih načela, a uz njih je navedeno kako još više treba cijeniti (Beck et. al. 2001):

- Ljude i njihove međusobne odnose, nego procese i oruđa.
- Upotrebljiv softver, nego iscrpnu dokumentaciju.
- Suradnju s naručiteljem, nego pregovaranje oko ugovora.
- Reagiranjem na promjenu, nego ustrajanjem na planu.

Scrum je najpoznatiji agilni model i namijenjen je upravljanju složenim radom, s naglaskom na razvoj softvera (Wykovski i Wykovska, 2018). Provodi se u četiri faze od kojih se posljednja može podijeliti na dvije (Awad, 2005):

- Razvoj općeg modela,
- Lista funkcionalnosti,
- Kreiranje plana,
- Dizajn i razvoj po funkcionalnostima.

U usporedbi s tradicionalnim pristupom model koji prakticira tvrtka *37signals* se temelji isključivo na kako ih oni nazivaju “dojmljivim” idejama što podrazumijeva da ideja mora biti inovativna i doprinositi rješenju određenog problema. Vezano uz odabranu ideju dizajniranje softvera započinje bez preliminarnih istraživanja tržišta. Izrađuje se prva skica koju slijedi HTML kod, odnosno izgled korisničkog sučelja. Potom slijedi pisanje koda, odnosno konkretnije oblikovanje proizvoda. Faza testiranja i evaluacije se provodi testiranjem razine na kojoj novi softver rješava konkretan problem u poslovanju. Kada je proizvod operativno funkcionalan, testira ga se u praksi, nakon čega slijede nadogradnje i unapređenja. Prije komercijalizacije *37signals* prakticira standard skraćivanje na pola. Konkretno, proizvod se pojednostavljuje otklanjanjem obilježja koja ne narušavaju njegovu inicijalnu korisničku svrhu. Ovog su se standarda pridržavali i pri uređivanju knjiga u kojima su predstavili svoju poslovnu praksu. Primjerice prva verzija njihove knjige *Rework* je imala 57.000 riječi, a u konačnici je svedena na 27.000 riječi (Fried, Heinemeier Hansson, 2010).

1.2. Ciljevi i metodologija istraživanja

Primarni aplikativni cilj istraživanja je predstaviti i konkretizirati varijable modela upravljanja razvojem novog proizvoda tvrtke *37signals*.

Sekundarni aplikativni cilj istraživanja je usporediti percipiranu kvalitetu korisnika softverskih proizvoda tvrtke *37signals* s kvalitetom proizvoda konkurenata koji imaju sličnu namjenu.

Tercijarni aplikativni cilj istraživanja je analizirati utjecaj broja zaposlenih softverske tvrtke na percipiranu kvalitetu korisnika novih proizvoda.

U prvom do petog dijela diplomskog rada predočiti će se teorijska razrada predmeta i problema istraživanja slijedom analize prikupljenih sekundarnih informacija i podataka iz dostupnih izvora.

U svrhu obrade prikupljenih informacija i podataka bit će korištena metoda deskripcije za opisivanje činjenica i procesa vezanih uz tržište softvera, kvalitetu i zadovoljstvo te modele upravljanja razvojem softverskog proizvoda.

Nadalje, pri obradi i pisanju navedenih dijelova rada bit će korištene metode analize i klasifikacije u svrhu boljeg objašnjenja kompleksnih pojmova što se posebno odnosi na dio vezan uz modele upravljanja razvojem softverskog proizvoda.

U dijelu koji se odnosi na *37signals* model upravljanja razvojem softverskog proizvoda bit će korištena i metoda kompilacije kojom će se povezivati spoznaje iz knjiga *Getting real: the smarter, faster, easier way* (2006) i *Rework* (2010)

U nastavku će se provesti primarno istraživanje modelom dubinskog strukturiranog intervjua s korisnicima različitih proizvoda tvrtke koja je predmet ovoga istraživanja. Kao sudionici intervjua za programske radne okvire bit će anketirani korisnici programeri razine senior developer, a za intervju o platformi za upravljanje projektima bit će također anketirani programeri razine senior developer, ali koji imaju iskustvo kao voditelji projekta. Konkretno, usporedna analiza bit će provedena na način da će se prvo prikupiti sekundarni podaci te potom biti obrađeni skupa s primarnim podacima dobivenim istraživanjem na terenu. Analiza će biti provedena na proizvodima *Ruby on Rails* i *Basecamp* čija će se korisnička obilježja analizirati u odnosu na konkurentne proizvode slične namjene. Usporedba *Ruby on Railsa* će biti provedena po uzoru na istraživanje provedenom na *Temple University* pod nazivom *Evaluating web development frameworks: Django, Ruby on Rails and CakePHP* (2009). Istom metodom, samo koristeći obilježja proizvoda relevantna za platforme za upravljanje projektima, bit će provedena i usporedba *Basecamp* proizvoda s konkurentskim.

1.3. Specifičnosti globalnog tržišta softvera

Tržište softvera obuhvaća razvoj i upravljanje softverskim aplikacijama, složene poslovne softvere, softvere u funkciji rasta produktivnost, infrastrukturne i druge softverske sustave (Mehta et. al. 2021). Obzirom na brojnost kategorija različitih proizvoda, ovo tržište se danas smatra jednim od najvećih i najkompleksnijih (Mehta et. al. 2021). Dinamika rasta ovoga tržišta se sagledava rastom ukupne vrijednosti transakcija. Veličina globalnog softverskog tržišta 2016. godine se procjenjivala na 417,63 milijardi američkih dolara, a do danas je dodatno porasla za 200 milijardi američkih dolara. Točnije, za 2022 godinu ovo tržište je procijenjeno na 608,68 milijardi dolara, s tim da se očekuje daljnji rast za dodatnih 200 milijardi dolara tijekom sljedeće četiri godine (Statista, 2022). Izneseni podaci ukazuju da se radi o dinamičnom, brzorastućem tržištu, što je posljedica eksponencijalnog trenda rasta konkurencije, odnosno ukupne ponude (Statista, 2022). O ovome obilježju tržišta treba posebno voditi računa u procesima promišljanja i planiranja razvoja i upravljanja novim proizvodima (Šerić, 2016). Više o globalnom tržištu softvera bit će rečeno u drugoj cjelini ovog rada.

1.4. Trendovi i inovacije na tržištu softvera

Industrija softvera direktno je vezana uz razvoj interneta i web tehnologija. Nakon Web 1.0 odnosno statičnog interneta uslijedilo je doba Web 2.0 u kojem se nalazimo i danas. Ono je obilježeno razvojem web aplikacija koje pružaju korisniku ne samo dobivanje informacija s web poslužitelja već i interakciju sa sadržajem i dijeljenje informacija. Uz Web 2.0 tehnologije veže se i koncept računalnih oblaka trend čijeg korištenja je i danas u porastu. (Atchinson, 2022)

Jedan od najizraženijih trendova u sferi poslovnog softvera je korištenje IT resursa kao usluge, a ne kao proizvoda. Takav suvremeni pristup u računarstvu se naziva računalnih oblaka. Tokom proteklih nekoliko godina, koncept računalnih oblaka je sve više prihvaćen na

tržištu. Model računalnih oblaka može biti posebno privlačan za manje kompanije koje imaju ograničen broj tehničkih kadrova. Pogodan je za određene vrste poslovnog softvera, kao što su Upravljanje odnosima sa kupcima (CRM), Upravljanje kadrovima (HR), Upravljanje projektima i slični (Stankić, 2015).

- Samostalno korištenje na zahtjev.
- Širok spektar mogućnosti mrežnog pristupa
- Alokacija resursa
- Elastičnost sistema
- Sve usluge su mjerljive

Postoje tri glavne vrste rješenja u obliku računalne usluge u oblaku: IaaS, PaaS i SaaS (RedHat, 2018):

- IaaS (Infrastructure-as-a-Service) znači da pružatelj usluga u oblaku upravlja infrastrukturom umjesto vas putem internetske veze. Korisnik ima pristup putem API-ja ili nadzorne ploče i u biti iznajmljuje infrastrukturu. Korisnik upravlja stvarima kao što su operativni sustav, aplikacija i *middleware*, dok se davatelj brine o hardveru, umrežavanju, tvrdim diskovima, pohrani podataka i poslužiteljima; i ima odgovornost za brigu o prekidima, popravcima i hardverskim problemima. Ovo je tipičan model implementacije pružatelja usluga pohrane u oblaku.
- PaaS (Platforms-as-a-Service) znači da hardver i aplikacijsko-softversku platformu omogućava vanjski pružatelj usluga u oblaku i njima upravlja, ali korisnik upravlja aplikacijama koje se pokreću na vrhu platforme i podacima na koje se aplikacija oslanja. PaaS korisnicima daje zajedničku platformu u oblaku za razvoj aplikacija i upravljanje bez potrebe za izgradnjom i održavanjem infrastrukture koja je obično povezana s procesom.
- SaaS (Software-as-a-Service) je usluga koja svojim korisnicima isporučuje softversku aplikaciju kojom upravlja pružatelj usluga u oblaku. SaaS aplikacije obično su web aplikacije ili mobilne aplikacije kojima korisnici mogu pristupiti putem web-preglednika. Ažuriranja softvera, ispravci grešaka i drugo općenito održavanje softvera se brinu za korisnika, a oni se povezuju s aplikacijama u oblaku putem nadzorne ploče

ili API-ja. SaaS također eliminira potrebu da se aplikacija instalira lokalno na računalu svakog pojedinog korisnika te dopušta bolje metode grupnog ili timskog pristupa softveru.

Web 3.0 je budućnost interneta i sve tehnologije koje se uz njega vežu značajno će utjecati na softversku industriju (Atchinson, 2022). Web 3.0 temeljen je na *blockchain* tehnologiji. Blockchain je tehnologija za vođenje zapisa osmišljena da onemogući hakiranje sustava ili krivotvorenje podataka pohranjenih na njemu, čineći ga tako sigurnim i nepromjenjivim (Pratt, Gillis, 2021.) To znači da web aplikacije više ne pohranjuju i ne održavaju podatke već je to zadatak *blockchaina*. *Blockchain* nije samo tehnologija, već temeljno novi način razmišljanja o podacima koji će stvoriti novu iteraciju interneta. Ono je jednako temeljno za podatke kao što je internet okosnica za prijenos informacija (Atchinson, 2022).

U budućnosti sve više će se u arhitekturu aplikacija uključivati *blockchain*. Blockchain će biti jednako važan za sljedeću generaciju internetskih aplikacija kao što su računalni oblak, mikroservisne arhitekture i *devops* za sadašnju generaciju (Atchinson, 2022).

1.5. Razvoj tvrtke 37signals

Godine 1999., web dizajner Jason Fried pokrenuo je tvrtku s dva prijatelja u Chicagu. Tvrtka se zvala 37Signals i kao i mnogi drugi u to vrijeme, trojac se bavio redizajnom web stranice. Ali za razliku od njihovih konkurenata, način na koji su pristupali klijentima bio je izrazito čudan. Da ste tada posjetili njihovu web stranicu, barem bi vas malo zbunio njezin običan, jalov izgled i uzalud biste tražili portfelj prijašnjih projekata ili svjedočanstva klijenata ili bilo što čime se hvale. Na njihovom web mjestu našli biste strastvene manifeste i članke u kojima su tri dizajnera govorili o svemu što misle da nije u redu s poslovanjem i kako to treba učiniti na drugi način. Njihovi stavovi bili su bizarni i kontradiktorni. Ljudi koji su ih primjećivali, a s vremenom mnogi su postali klijenti (Shaji, 2020).

Već 2003. dizajneri u *37Signalsu* jedva su stizali odraditi sav posao klijenata. Stvari su počele izmicati kontroli i odlučili su potražiti namjenski alat za upravljanje projektima. Tada

je većina takvih alata bila nezgrapna i složena, gubili su više vremena nego što su uštedjeli, pa je godinu dana nakon toga, 2004., Jason unajmio danskog programera po imenu David Heinemeier Hanson da razvije vlastitu aplikaciju za upravljanje projektima. Kada su klijenti vidjeli gotov proizvod, mnogima se od njih toliko svidio da su tražili licencu. Naravno riječ je o *Basecampu* (Shaji, 2020).

Razvoju *Basecampa* prethodio je drugi veliki izum. „Ovdje je izmišljen Ruby on Rails“ stoji na *37signals*ovim stranicama. To je besplatni programski radni okvir otvorenog koda koji pokreće moćne tvrtke kao što su *Shopify*, *Coinbase*, *GitHub*, *Airbnb*, *Kickstarter*, *Square*, *Twitch*, *Basecamp*, *HEY*. Zapravo, *Basecamp* je prva *Ruby on Rails* aplikacija ikada. Na *Railsu* se danas temelje tisuće programerskih karijera i doveo je tvrtke do milijuna korisnika i milijardi u tržišnim procjenama. (37signals, 2022)

Godine 2014. tvrtka je promijenila naziv u *Basecamp*, kako bi dodatno promovirali *Basecamp* platformu za upravljanje projektima kao svoj temeljni softverski proizvod. Stari naziv tvrtke, *37signals*, je vraćen 2022. godine (Basecamp, 2022). To razdoblje obilježeno je razvojem i rastom *Basecampa* kao proizvoda, ali i cijele tvrtke. U tom razdoblju razvijeni su i drugi proizvodi poput *Campfirea*, poslovno orijentirane usluge *online chata*, kasnije integrirane u *Basecamp* i *Highrisea* koji služi za upravljanje odnosima s kupcima (CRM), ali nije više komercijalno dostupan već samo postojećim korisnicima (Basecamp, 2022).

Od 2020. komercijaliziran je još jedan proizvod tvrtke *37signals*, a to je HEY originalna vrhunska usluga e-pošte (Basecamp, 2022). „E-pošta je godinama bila sranje. Ne više — popravili smo je. HEY nudi svjež pristup koji pretvara e-poštu u nešto što želite koristiti, a ne u nešto s čime ste prisiljeni nositi se.“ (HEY, 2022).

Kao specifičnost ove tvrtke, kako kroz povijest tako i danas, može se istaknuti iznimno malen broj zaposlenih u usporedbi s konkurentima koji razvijaju tehnološka rješenja u istim kategorijama proizvoda najviše klase. Danas tvrtka zapošljava svega 65 zaposlenika, što je ujedno i najveći broj zaposlenih od osnutka (Basecamp, 2022). Usporedbe radi u tvrtci *ClickUp* koja je vlasnik glavnog konkurentnog proizvoda *Basecampu* zaposleno je 800 ljudi (ClickUp, 2022).

2. GLOBALNO TRŽIŠTE SOFTVERA

2.1. Odnosi ponude i potražnje

Ponuda nekog dobra ili usluge je količina koju su prodavači spremni i u mogućnosti prodati, a potražnja je količina nekog dobra ili usluge koju su kupci spremni i koju mogu kupiti (Garača, 2010).

Iako se čini da je postojeći broj programera dostatan potrebama softverske industrije, potrebe za novim, dodatnim programerima rastu. Tisuće novih tvrtki koje omogućuju i rad na daljinu kontinuirano objavljuju oglase za zapošljavanje novih programera. (Mleczo, 2022).

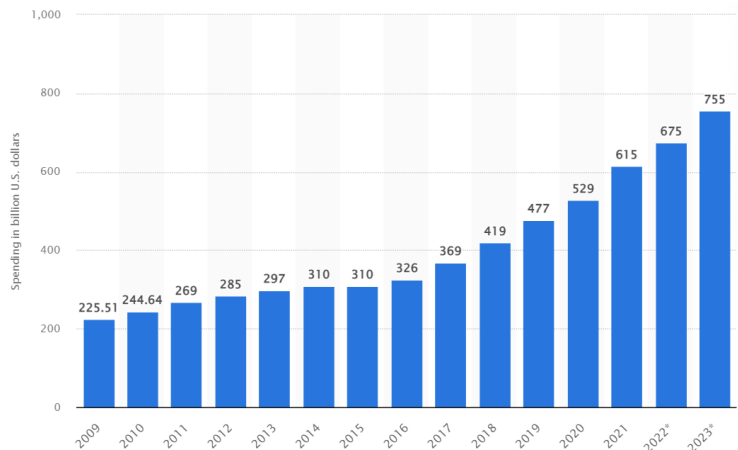
Kao što je rečeno u uvodnom dijelu softver će se analizirati kao usluga. Sukladno tome, kada se govori o ponudi softvera, relevantno je promatrati broj ponuditelja konkurentnih rješenja koja se koriste za iste ili slične namjene. Softver je danas katalizator razvoja novih tehnologija, s procijenjenih 9 milijuna uređaja u aktivnoj upotrebi koji povezuju poslovne subjekte i ljude sa svih strana svijeta. Programeri softvera aktivno sudjeluju u izgradnji suvremenog tehnološkog svijeta. Broj programera softvera kontinuirano raste, 2013. godine se spominjao podatak od 18,2 milijuna osoba, a krajem 2021. godine taj broj je narastao na 26,8 milijuna. Aktualne procjene govore da će broj programera do 2030. godine narasti na 45 milijuna (Mleczo, 2022).

Premda broj angažiranih programera kontinuirano raste, potražnja za novim stalno ukazuje na njihov nedostatak. Razlog tome je povećana potražnja za direktnom i indirektnom potporom za korisnike softvera (Hughes, 2022).

Rast potražnje na tržištu softvera argumentiraju podaci o ukupnim novčanim sredstvima utrošenima za kupnju ovih proizvoda. U ovome radu fokus je na tržištu poslovnih softvera, jer za to postoje relevantni i potpuni podaci koji omogućuju uvid u stvarnu potražnju za pojedinim kategorijama proizvoda. Poslovni softver se koristi za različite svrhe u poslovanju. Važna je to potpora ostvarivanju korporativnih ciljeva na strateškoj razini, ali i potpora u praćenju i unapređivanju lanca opskrbe, upravljanju svim resursima, interakciji s kupcima i za

druge svrhe na taktičkoj razini. Pojam *softver za poduzeća* obuhvaća različite aplikacije - računovodstveni softver, uredski paketi, grupni softver, sustavi za upravljanje ljudskim resursima (HRM), upravljanje odnosima s vanjskim izvorima, upravljanje odnosima s klijentima (CRM), planiranje resursa poduzeća (ERP), upravljanje lancem opskrbe (SCM), poslovna inteligencija (BI), analitički softver, upravljanje poslovnim procesima (BPM) i sigurnosni softver (Statista, 2022).

Globalna potražnja softvera za poduzeća stalno je i ubrzano rasla do pojave pandemije COVID-19, kada je zabilježen blagi pad potražnje za ovim kadrovima. Tijekom 2021. zabilježen je nastavak rasta potražnje za njima. Očekuje se da će do kraja 2022. godine vrijednost prodaje softvera iznositi više od 570 milijardi američkih dolara. Na tržištu softvera za poduzeća očekuje se rast potražnje za softverima za planiranje resursa poduzeća (ERP), softverima za upravljanje odnosima (CRM) i softverima za upravljanje sadržajem. Od ovih podržišta, tržište ERP softvera trenutno je najveće, te se očekuje da će ove godine porasti na više od 95 milijardi američkih dolara. Tržište ERP softvera obuhvaća poseban softver za upravljanje poslovnim procesima koji tvrtkama omogućuje integraciju različitih aspekata poslovanja unutar jedne baze podataka, aplikacije i korisničkog sučelja (Statista, 2022).



Slika 1: Troškovi kupnje poslovnog softvera u svijetu 2009. – 2023.

Izvor: Statista, 2022.

2.2. Konkurentno okruženje

U razdoblju posljednjih deset godina ubrzano raste broj prodavatelja softvera. To je najbrže rastući segment tehnološke industrije koji potiče nastavak digitalne transformacije u svim sferama gospodarstva. Otvorenost tržišta koju omogućuju platforme u oblaku u kombinaciji s novim rješenjima i prihvaćanjem aplikacijskih softvera kao usluga temeljenih na pretplati potiče rast ulaganja u tehnologiju. Tvrtke postižu bolje poslovne rezultate spajanjem više softverskih rješenja unutar jedne aplikacije. Tako se zadovoljavaju potrebe različitih industrija i sektora i geografski udaljenih tvrtki (McBain, 2017). Procjene su da danas posluje više od 100 000 softverskih tvrtki (ISV). Prije desetak godina bilo ih je svega 10 000. Obzirom na kontinuitet specijalizacije koju zahtijevaju današnji kupci, taj broj bi do 2027. godine mogao narasti na milijun tvrtki (McBain, 2017).

Na intenzitet konkurentnosti ukazuje i vrijednost softverskih tvrtki. Slika 2 daje prikaz deset softverskih kompanija s najvećom tržišnom vrijednošću. Tu su globalno najpoznatije *Apple*, *Microsoft*, *Adobe* i *IBM*, ali i manje poznate - *Salesforce* i *SAP*.

Rb.	Naziv	Tržišna vrijednost	Država
1	Apple	\$2.344 T	USA
2	Microsoft	\$1.978 T	USA
3	Alphabet	\$1.529 T	USA
4	Oracle	\$190.72 B	USA
5	Adobe	\$179.78 B	USA
6	Salesforce	\$173.48 B	USA
7	IBM	\$126.38 B	USA
8	Intuit	\$113.55 B	USA
9	SAP	\$104.76 B	Njemačka
10	ServiceNow	\$98.30 B	USA

Slika 2: Najveće softverske kompanije

Izvor: <https://companiesmarketcap.com/software/largest-software-companies-by-market-cap/>
(12.6.2022.)

2.3. Tehnologije na tržištu softvera

Dvije najznačajnije tehnologije koje se danas koriste u razvoju softvera su tehnologija računalnog oblaka i *blockchain* tehnologija. Tehnologija računalstva u oblaku (CCT) predstavlja potpuno novi način korištenja interneta za pružanje softverskih i infrastrukturnih rješenja tvrtkama diljem svijeta. Korištenjem ove tehnologije osnažuje se komunikacija između tvrtki i stvara se platforma za generiranje značajnih financijskih i operativnih koristi i za manja poduzeća (Attaran, Woods, 2018).

Na mnogo načina, računalstvo u oblaku postalo je sastavni dio IT svijeta. Pojam *oblak* se koristi za označavanje različitih vrsti platformi za klaster poslužitelja, mreže, softvera, sučelja i dr. korisnicima potrebnih za izvršavanje konkretnih zadataka. *Računalstvo* se odnosi na isporuku ovoga paketa kao usluge koju korisnici mogu koristiti. Korisnik više ne treba izdvajati značajna sredstva za ulaganja u IT infrastrukturu. Korisnik može koristiti infrastrukturu u vlasništvu treće strane, te plaćati naknadu za korištenje. Ovaj model s naknadom za korištenje omogućuje praktičan pristup mreži na zahtjev, na zajednički skup konfigurabilnih računalnih resursa kao što su poslužitelji, pohrana, aplikacije i usluge. Klijenti se mogu povezati s dostupnim fizičkim ili virtualnim okruženjima koristeći različite ulazne točke. Korisnik pristupa informacijama *on-line* u formatu 24/7 s jednostavnih uređaja – stolnih i prijenosnih računala, tableta i pametnih telefona (Attaran, Woods, 2018).

Oblak, odnosno CCT se temelji na nekoliko IT inovacija, uključujući virtualizaciju, povećanje kapaciteta Interneta i rastuću sofisticiranost tehnologija. *National Institute of Standards and Technology* (NIST) opisuje pet obilježja računalstva u oblaku (Attaran, Woods, 2018):

(1) Samoposluživanje na zahtjev - vrijeme poslužitelja, mrežna pohrana i ostalo se odabire prema potrebama. Nije potrebna osobna interakcija s pružateljem usluga.

(2) Široki pristup mreži - resursi su dostupni preko mreže, a pristupa im se putem standardnih uređaja (mobilnih telefona, tableta, prijenosnih računala i radnih stanica).

(3) Udruživanjem resursa – objedinjuju se za potrebe više korisnika. Fizički i virtualni resursi se dodjeljuju dinamički prema potražnji.

(4) Elastičnost - resursi se povećavaju i smanjuju s potražnjom, elastična opskrba slobodnim resursima.

(5) Mjerljiva usluga - za automatsku optimizaciju korištenja resursa (pohrana, obrada, propusnost i aktivni korisnički računi).

Blockchain je distribuirana baza podataka ili knjiga koja se dijeli među čvorovima računalne mreže. Kao baza podataka, *blockchain* pohranjuje informacije elektronički u digitalnom formatu. *Blockchain* je najpoznatiji po funkciji u sustavima kriptovaluta (*Bitcoin* npr.) za održavanje sigurne i decentralizirane evidencije transakcija. Inovacija *blockchaina* je jamstvo vjernosti i sigurnosti zapisa podataka uz povjerenje bez potrebe za posrednikom (Hayes, 2022).

Ključna razlika tipične baze podataka i *blockchaina* je način strukturiranosti podataka. *Blockchain* prikuplja informacije u grupama (blokovi) koje sadrže skupove informacija. Blokovi imaju određene kapacitete za pohranu. Kada se isti popune, zatvaraju se i povezuju s prethodno popunjenim blokom tvoreći lanac podataka - *blockchain*. Novi podaci se nakon toga bilježe u novom bloku koji će također biti pridodan lancu nakon što se popuni (Hayes, 2022).

Baza podataka strukturira podatke u tablice, dok *blockchain* strukturira podatke u blokove u nizu. Ova struktura podataka inherentno čini nepovratnu vremensku liniju podataka kada se decentralizirano implementira. Kada je blok popunjen postaje dijelom vremenske crte. Svaki blok u lancu ima točnu vremensku oznaku kada je dodan (Hayes, 2022). Cilj *blockchaina* je omogućiti snimanje i distribuciju digitalnih podataka bez njihovog uređivanja. Tako *blockchain* postaje temelj nepromjenjivih knjiga i zapisa o transakcijama koje se ne mogu mijenjati, brisati ili uništiti. Zbog toga je *blockchain* poznat i kao tehnologija distribuirane knjige - DLT (Hayes, 2022).

3. KVALITETA I ZADOVOLJSTVO

3.1. Definiranje kvalitete

Kvaliteta usluge je danas poseban izazov, obzirom da se uvijek teži što višoj kvaliteti uz trošak koji će omogućiti konkurentnu cijenu (Šerić, Luetić, 2016). Kvaliteta usluge može biti apstraktan, višedimenzionalan i vremenom promjenljiv koncept koji je ponekad teško definirati, opisati i kvantitativno mjeriti. Premda je kvalitetu usluge teško definirati i mjeriti, jednostavno je ustvrditi kada je ista nezadovoljavajuća (Šerić et al. 2011). Na tržištu softvera se to prepoznaje po ponašanju korisnika usluga koji mijenjaju pružatelja usluge s čijom kvalitetom zbog nekog razloga nisu zadovoljni (Ozretić, 2010).

Kvaliteta usluge je pretpostavka konkurentske prednosti i obilježje po kojem se pružatelj može diferencirati. Ozretić (2010) konstatira da kvaliteta usluge može:

- zainteresirati korisnika za probnu kupnju,
- potaknuti prodaju usluge,
- doprinijeti dojmu zadovoljstva korisnika pruženom uslugom, te
- graditi i održavati lojalnost korisnika određenom pružatelju usluge.

Poslovni subjekt čije usluge ispunjavaju navedene kriterije percipira se kao poželjan i preporučljiv odabir (Šerić, 2016). Kvalitetnu uslugu je kompleksno kopirati ili imitirati. Slijedom toga korisnik će subjekt koji pruža kvalitetne usluge percipirati *posebnim* i drugačijim od konkurenata. Diferencijacija usluge kvalitetom koliko god predstavljala izazov imati će pozitivne reperkusije na poslovanje pružatelja (Meža, Šerić, 2009).

U procesima mjerenja kvalitete mnoštvo je poteškoća i dilema. Jedinstveni, općeprihvaćeni model mjerenja kvalitete usluge ne postoji. Ozretić (2010) navodi deset temeljnih obilježja koja doprinose dojmu kvalitete usluge kod korisnika:

1. Pouzdanost pružanja usluge (ispunjavanje obećanja danog korisniku, rutina i konzistentnost načina pružanja usluge).
2. Poslovnost i odgovornost (pravodobnost, spremnost i raspoloživost onih koji uslugu pružaju).
3. Kompetentnost (znanje i vještina potrebni za pružanje kvalitetne usluge, stručnost osoblja).
4. Pristupačnost (dostupnost usluge – prikladno radno vrijeme pružanja usluge, mjesto pružanja usluge i kratko vrijeme čekanja za uslugu).
5. Susretljivost (ljubaznost, poštovanje, razumijevanje i srdačnost zaposlenika koji pružaju uslugu u kontaktu s korisnikom).
6. Komunikacija s korisnikom (informiranje korisnika na njemu razumljiv način, prikupljanje i uvažavanje mišljenja, primjedbi i prijedloga korisnika).
7. Kredibilitet (poštenje, profesionalnost, ugled i povjerenje pružatelja usluge).
8. Sigurnost (potrebni sigurnosni uvjeti tijekom pružanja usluge, kontrolirani rizici za korisnika, te njegova fizička i financijska sigurnost - jamstva).
9. Razumijevanje potreba korisnika (personalizacija usluge).
10. Opipljiva obilježja usluge (eksterijer i interijer mjesta pružanja usluge, izgled i dojam zaposlenika, oprema, promotivni materijali i drugo).

3.2. Percipirana kvaliteta

“Protekom vremena i razvojem znanosti o kvaliteti došlo je do pojave velikog broja različitih definicija kvalitete. U početku se kvaliteta jednostavno definirala kao percipirano zadovoljstvo kupca, no razvojem tehnologije i porastom zahtjeva korisnika prema poduzećima kvaliteta se počela doživljavati na različite načine.” (Lazibat i Baković, 2020 - u Buntak et al. 2021, str. 9)

„Ono što jedna osoba percipira kvalitetnim druga osoba može doživjeti prosječnim ili ispodprosječnim. Elementi koji uključuju navike, stavove, iskustva, set vrijednosti ili društvenu klasu pojedinca, samo su neki čimbenici koji utvrđuju način na koji osoba može percipirati kvalitetu.“ (Marinković, Senić, 2012, str. 14)

Gronroos (1987) ističe kako se percipirana kvaliteta usluge može definirati kao opći sud ili stav prema usluzi, koji proizlazi iz usporedbe očekivanja s percepcijom klijentele o performansama aktualne usluge. Ova definicija, koja se odnosi na percipiranu kvalitetu usluge, primjenjiva je i za softverski proizvod.

3.3. Odnos kvalitete i zadovoljstva

Zadovoljstvo i lojalnost su usko vezani pojmovi. Ukoliko isporučena usluga potiče zadovoljstvo korisnika, velika je vjerojatnost ponovnog korištenja usluge. Lojalan korisnik dijeliti će pozitivna iskustva i dojmove s drugima, od kojih će neki također postati korisnici usluga istog subjekta. Lojalnost za posljedicu ima veću otvorenost za suradnju, manju osjetljivost na promjene cijena i iskazivanje interesa za druge usluge u ponudi korisnika (Puška, Šadić, Beganović, 2016).

Kvaliteta percipirana u svijesti korisnika je temelj njegova zadovoljstva, odnosno nezadovoljstva. Percepcija visoke kvalitete pružene usluge pretpostavka je lojalnosti korisnika pružatelju iste. Ukoliko korisnici pozitivno ocjenjuju proces pružanja usluge moguće će biti zadovoljni i pruženom uslugom. Lojalnost pak predstavlja predanost korisnika određenoj marki ili pružatelju usluge temeljem prethodnih pozitivnih iskustava. Posljedica je ponovljena kupnja usluge (Puška, Maksimović, Fazlić, 2015). „Može se zaključiti kako percipirana kvaliteta ima pozitivan utjecaj na zadovoljstvo, a posljedično zadovoljstvo rezultira stvaranjem lojalnih odnosa.“ (Puška, Šadić, Beganović, 2016)

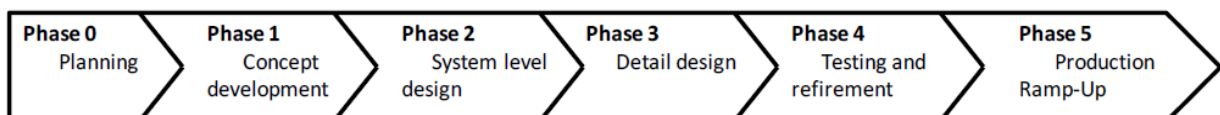
4. MODELI I PRISTUPI UPRAVLJANJA RAZVOJEM SOFTVERSKIH PROIZVODA

4.1. Vodopadni model upravljanja razvojem softverskog proizvoda

Tradicionalni pristup nerijetko se u kolokvijalnom govoru greškom miješa s vodopadnim modelom zbog njegove iznimne popularnosti i učestalosti. Nužno je naglasiti kako pristup i model nisu jedno te isto, te ih se tek iz tog aspekta može u potpunosti razumjeti (Awad, 2005/15).

4.1.1. Tradicionalni pristup

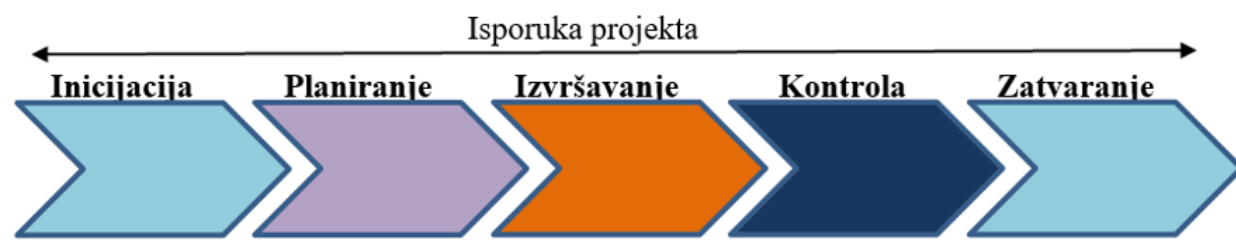
Tradicionalni procesi razvoja softvera temelje se na generičkom procesu razvoja proizvoda koji se uobičajeno koristi u različitim proizvodno poslovnim linijama. Generički proces razvoja proizvoda slijed je koraka ili aktivnosti koje poduzeće koristi da osmisli, dizajnira i komercijalizira proizvod (Šerić, 2016). Mnogi od ovih koraka su intelektualne i organizacijske, a ne fizičke aktivnosti.” Generički proces razvoja proizvoda opisan je kao proces u šest koraka, koji se sastoji od planiranja, razvoja koncepta, projektiranja sustava, projektiranje detalja, testiranja i usavršavanja, te faze početka proizvodnje. (Ulrich & Eppinger, 2008. str. 12).



Slika 3: Generički proces razvoja proizvoda

Izvor: Molin-Juustila, 2001

Tradicionalni pristup upravljanja razvoja softverskog proizvoda ovisi o unaprijed određenim procesima i uzastopnom dokumentiranju što se kontinuirano provodi tijekom trajanja projekta. Uspjeh projekta razvijanog tradicionalnim pristupom ovisi o upoznatosti programera sa svim zahtjevima prije samog razvoja. Posljedično, uvođenje promjena tijekom razvoja može prouzrokovati probleme. Prednost ovog pristupa je olakšano utvrđivanje troškova, rasporeda i ekonomičnije upravljanje resursima (Khong W. L. et al, 2012),



Slika 4: Tradicionalni pristup upravljanjem razvoja softverskog proizvoda

Izvor: Marić I. 2020.

Tradicionalni pristup provodi se kroz slijedeće faze (Khong W. L. et al, 2012):

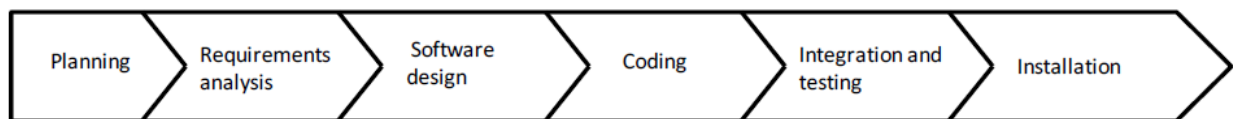
- Inicijacija: procjenjuje se količina sistemskih zahtjeva i na temelju toga se donosi funkcijska specifikacija.
- Planiranje: stvaranjem arhitekture i dizajna izrađuje se izgled korisničkog sučelja te osigurava funkcionalnost.
- Izvršavanje: pisanje koda temeljem prethodnog upoznavanja zahtjeva, dokumentacije i već razvijenog korisničkog sučelja.
- Kontrola: testiranje individualnih komponenti više puta tijekom razvojnog procesa kao i testiranje sustava u cijelosti na kraju razvojnog procesa. Sve se ponovno testira nakon isporuke proizvoda korisniku u cilju utvrđivanja ispunjenosti zahtjeva.

Uobičajeni modeli koji se danas koriste, a sukladni su sa standardima tradicionalnog pristupa su (Ruparelia, 2010):

- Vodopadni model
- V-model
- Model unificiranog procesa razvoja
- Spiralni model

4.1.2. Specifičnosti Vodopadnog modela

Model vodopada je izvorno osmislio Winston Royce 1970. godine. Model vodopada ilustriran je na slici 5. Uzastopni procesi modela vodopada su: planiranje, analiza zahtjeva, dizajn softvera, kodiranje, integracija i testiranje, te instalacija ili održavanje (Molin-Juustila, 2001).



Slika 5. Model vodopada

Izvor: Molin-Juustila, 2001

Vodopadni model, kao što je rečeno, prepoznatljiv je po strukturiranom napredovanju među definiranim fazama. Svaka faza se sastoji od određenog seta aktivnosti i rezultata koji treba zadovoljiti kao preduvjet iduće faze. Kao nulta faza se prakticira planiranje jer ono prethodi početku razvojnog procesa (Awad, 2005/15). Nakon planiranja slijede:

- Prva faza - analiza zahtjeva kojom se definira funkcionalnost sustava prilagođene sustavnim i softverskim zahtjevima.
- Druga faza – dizajn softvera, definira kako će konačni proizvod izgledati.
- Treća faza – razvoj, odnosno ispisivanje koda.

- Četvrta faza – testiranje radi ispravljanja pogreški sustava.
- Peta faza – održavanje, uključuje procese kao što su edukacija korisnika, izrada dokumentacije i drugo.

4.2. *Scrum* model upravljanja razvojem softverskog proizvoda

Scrum je agilni upravljački okvir namijenjen upravljanju složenim radom, s naglaskom na razvoj softvera, iako se koristi i za druge svrhe, a posebno se istražuju njegove mogućnosti za složene funkcije u naprednim tehnologijama (Wykovski i Wykovska, 2018). *Scrum* model se temelji na agilnom pristupu upravljanju razvojem softverskog proizvoda.

4.2.1. Agilni pristup

Temelj agilnog pristupa je ubrzati vrijeme procesa razvoja novog proizvoda i omogućiti kasnije promjene korisničkih zahtjeva. Agilni pristup predstavlja pokušaj odgovora zahtjevnoj poslovnoj zajednici koja traži brže i slobodnije procese razvoja softvera. Takvi su zahtjevi intenzivirani u industriji web softvera i mobilnih aplikacija (Abrahamsson et al. 2002). U cilju standardizacije definiran je *agilni manifest* koji podrazumijeva opća pravila i principe za agilne metode. Potpisalo ga je sedamnaest utjecajnih osoba u ovoj poslovnoj branši u Snowbirdu u Utahu, SAD, 2001. godine. Potpisnici *Agile Manifesta* su pojedinci koji su objavili zasebne metode razvoja softvera sličnih karakteristika. Njihove metode se temelje na iskustvima najbolje prakse i praksama evolucijskog razvoja s fokusom na ranu isporuku i kvalitetu softvera (Strode, 2006).

Beck K. et al. (2001) u *Agile Manifestu* ističu: tražimo bolje načine razvoja softvera razvijajući softver i pomažući drugima u njegovom razvoju. Takvim radom smo naučili da više cijenimo:

- Ljude i njihove međusobne odnose u odnosu na procese i oruđa;
- Upotrebljivost softvera u odnosu na pisanu dokumentaciju;
- Suradnju s naručiteljem u odnosu na pregovaranje u svezi ugovora;
- Reagiranjem na promjene u odnosu na ustrajanje na onome što je planirano.

Nadalje, manifestom se definira dvanaest ključnih načela na kojima se on temelji:

1. Najvažnije je zadovoljstvo naručitelja koje se postiže ranom i neprekinutom isporukom softvera koji doprinosi vrijednosti.
2. Spremno prihvaćanje promjena zahtjeva i u kasnijoj fazi razvoja. Agilni procesi podrazumijevaju promjene u funkciji kompetitivnih prednosti naručitelja.
3. Isporuca softvera upotrebljivog za korištenje, u razmacima od nekoliko tjedana do nekoliko mjeseci, težeći stalnom skraćivanju vremena isporuke.
4. Suradnja naručitelja softvera i razvojnih inženjera tijekom trajanja projekta razvoja.
5. Realizacija projekta oslanjanjem na motivirane pojedince. Osiguranje potrebnog i poticajnog okruženja i podrške za obavljanje posla.
6. Prakticirati direktne kontakte i komunikaciju u cilju potpunog prijenosa informacija razvojnom timu i unutar tima.
7. Upotrebljivost softvera je temeljna mjera napretka.
8. Korištenje agilnih procesa u funkciji održivoga razvoja. Pokrovitelji, razvojni inženjeri i korisnici trebaju održavati usklađenu dinamiku rada tijekom razvoja.
9. Naglasak na tehničkoj izvrsnosti i dobrom dizajnu u funkciji agilnosti.
10. Jednostavnost – smanjivanje nepotrebnih radnih aktivnosti je od suštinske važnosti.
11. Samo organizirani timovi pretpostavka su najbolje arhitekture, projektnih zahtjeva i dizajna.
12. Radni tim kontinuirano razmatra načine povećanja učinkovitosti, te slijedom novih spoznaja prilagođava svoje aktivnosti.

Većina ovih načela je poznata prije nastanka manifesta, ali njegovom su objavom povezana u primarnoj funkciji izrade softvera. Najpoznatiji modeli koji se temelje na agilnom pristupu su:

- Ekstremno programiranje;
- *Scrum* i
- Razvoj vođen funkcionalnošću.

4.2.2. Specifičnosti *Scrum* modela

Inicijatori *Scrum frameworka*, Schwaber i Sutherland (2010) su istaknuli da *Scrum* ima dugu povijest u razvoju softvera. Korišten je za razvoj složenih proizvoda tijekom ranih devedesetih godina XX stoljeća. "*Scrum* se temelji na najboljoj praksi prihvaćenoj u industriji, koja se koristi i dokazala kroz desetljeća." (Schwaber i Sutherland, 2010)

Schwaber i Sutherland (2010) ističu da *Scrum* nije tehnika izgradnje procesa ili proizvoda, već okvir koji omogućuje korištenje različitih procesa i tehnika. "Uloga *Scruma* je iznijeti relativnu učinkovitost vaših razvojnih praksi kako biste ih mogli poboljšati, istovremeno pružajući okvir unutar kojeg se mogu razvijati složeni proizvodi." Isti autori nadalje naglašavaju da je *Scrum* pristup razvijen za upravljanje procesom razvoja softverskih sustava: „Empirijski pristup koji primjenjuje ideje teorije upravljanja industrijskim procesima na razvoj softverskih sustava što rezultira pristupom koji ponovno uvodi ideje fleksibilnosti, prilagodljivosti. i produktivnost.”

Implementacija empirijske kontrole procesa ima tri strane, a to su transparentnost, pregled i prilagodba. Schwaber i Sutherland (2010) definiraju tri uporišta kontrole *Scrum* procesa na sljedeći način:

- Transparentnost osigurava da aspekti procesa koji utječu na ishod budu vidljivi onima koji upravljaju ishodima. Osim transparentnosti treba se znati ono što je vidljivo (prilikom pregledavanja procesa treba se konstatirati da je isti dovršen).
- Različiti aspekti procesa se trebaju pregledavati dovoljno često kako bi se otkrile neprihvatljive varijacije u procesu. Učestalost pregleda se određuje temeljem činjenice da se svi procesi mijenjaju nakon provjera.
- *Scrum* pretpostavlja tri točke za pregled i prilagodbu. Dnevni *Scrum* sastanak se prakticira radi provjere napretka prema cilju i za prilagodbe kojima će se optimizirati sljedeći radni dan. *Sprint pregled* i planski sastanci se koriste za provjeru napretka prema cilju izdavanja i za prilagodbe koje optimiziraju vrijednost sljedećeg sprints. Konačno, *Retrospektiva sprinta* se koristi za pregled prethodnog sprints i određivanje koje će prilagodbe sljedeći sprint učiniti produktivnijim, potpunijim i radno ugodnijim.

Scrum pristup uključuje tri faze: faza prije igre, faza razvoja i faza nakon igre. Faza prije igre podijeljena je u dvije pod faze: planiranje te arhitektura i dizajn visoke razine. U fazi planiranja se kreira lista zaostalih proizvoda koja sadrži sve u tom trenutku poznate zahtjeve. Zahtjevi su prioritetni te se procjenjuje njihova provedba. U fazi planiranja definiranje resursa i potreba projekta odobrava vrhovni menadžment. U fazi arhitekture planira se dizajn visoke razine sustava i arhitekture na temelju trenutnih uzroka kašnjenja. U fazi razvoja, različite ekološke i tehničke varijable (npr. vremenski okvir, zahtjevi, resursi i metode razvoja) identificirane su u *Scrumu*. Mogu se mijenjati tijekom procesa, promatraju se i kontroliraju posebnim testovima tijekom sprints. Svaki *sprint* uključuje tradicionalne faze razvoja softvera: zahtjeve, analizu, dizajn, evoluciju i isporuku. Vremenski okvir za jedan sprint može biti od jednog tjedna do jednog mjeseca, a može postojati tri do osam *sprintova* u jednom procesu razvoja sustava prije nego što sustav bude spreman za distribuciju. U fazi nakon igre, svi zahtjevi su ispunjeni, te su dovršeni svi zadaci koji uključujući integraciju, testiranje sustava i dokumentaciju (Abrahamsson et al. 2002).

Postoje tri različite uloge u *Scrum timu*: *Scrum Master*, *Product Owner* i tim. *Scrum Master* je odgovoran za osiguravanje da *Scrum* tim slijedi *Scrum* vrijednosti, prakse i pravila. *Scrum Master* je također neka vrsta osobe za podršku timu tako što pomaže, podučava i vodi. Vlasnik proizvoda je osoba odgovorna za kašnjenje i radni učinak tima. On/ona je također

odgovoran za prioritete zadataka. Tim se sastoji od članova s među funkcionalnim vještinama. Vještina koju članovi tima dijele je vještina rješavanja zahtjeva i pretvaranja istog u upotrebljiv proizvod (Schwaber i Sutherland, 2010)

Abrahamsson et al. (2002) uvode još dvije uloge *Scrum timu* - kupac i uprava. Kupac sudjeluje u zadacima koji se odnose na takozvani *Product Backlog* (zaostale stavke proizvoda) koji definira sve što je potrebno u konačnom proizvodu na temelju trenutnih saznanja za sustav koji se razvija ili poboljšava. Uprava je zadužena za donošenje konačnih odluka, prava na proizvod, standarde i konvencije koje treba slijediti u projektu. U postavljanju ciljeva i zahtjeva sudjeluje i menadžment. Menadžment je uključen u odabir vlasnika proizvoda, mjerenje napretka i smanjenje zaostatka proizvoda sa *Scrum Masterom* (Abrahamsson et al. 2002).

5. MODEL UPRAVLJANJA RAZVOJEM SOFTVERSKOG PROIZVODA TVRTKE 37SIGNALS

5.1. Povijest i razvoj modela

Razvoj 37signals modela upravljanja razvojem softverskih proizvoda tekao je usporedno s organizacijskim razvojem istoimene tvrtke (Shah, 2020). Jason Fried, nakon što je diplomirao, 1999. godine osnovao je konzultantsku tvrtku *37signals* za web dizajn. Prvenstveno su bili zainteresirani za *UX dizajn* (dizajn korisničkog iskustva) te su izradili nekoliko vlastitih suvremenih softverskih proizvoda slijedom kojih su počeli prodavati savjetničke usluge. Po osnivanju tvrtke Fried je objavio 37 načela "internetske mudrosti" vezana uz jednostavnost softvera. Temeljna među tim načelima su slijedeća (Shah, 2020):

- *Internet bi trebao osnaživati, a ne frustrirati;*
- *Ne treba povećavati opcijske mogućnosti za korisnika iznad stvarno potrebnih koje će koristiti. [Occamova britva];*
- *Tvrtke koje tvrde da mogu sve zapravo ne briljiraju ni u čemu. Treba težiti jednoj stvari i odraditi je kako treba;*
- *Web stranice ne koriste korporacije, već ljudi. Zbog toga ih treba dizajnirati na način da one rade za korisnika.*

Načelima kojih se pridržavao u poslovanju Fried je privukao pažnju internetske zajednice. Učestalo se počeo družiti i razmatrati nove ideje o programiranju i web dizajnu s David Heinemeier Hanssonom. Spoznali su da o mnogim stvarima slično razmišljaju, poštujući načelo *što jednostavnije to bolje*. Heinemeier je napisao kod za radni programski okvir *Ruby on Rails* u cilju razvoja alata za upravljanje projektima za internu upotrebu. Slijedom toga nastaje *Basecamp* 2005. godine. Izvorno to nije trebao biti komercijalni proizvod, već samo interni alat za unapređenje vlastitog poslovanja. Težilo se da taj alat bude korisnički jednostavniji od svih postojećih alata za upravljanje projektima. U razvoju i ovoga alata

pridržavalo se načela *što jednostavnije to bolje*. Takav pristup razvoju je doprinio privlačnosti ovoga alata svih koji su ga isprobali. *Basecamp* tako nije ostao nezamijećen od strane klijenata za koje je tvrtka radila. Klijenti su po pravilu bili zainteresirani za ovu platformu, slijedom koje je *Basecamp* komercijaliziran za širu komercijalizaciju (Shah, 2020).

U knjizi *Getting real: the smarter, faster, easier way* (2006) Fried i Heinemeier su detaljno opisali iskustva, filozofiju, pristup i model upravljanja razvojem softverskog proizvoda koji su koristili za *basecamp*. Nekoliko godina kasnije objavljuju knjigu *Rework* (2010) u kojoj je dodatno razrađena njihova poslovna filozofija (Shah, 2020).

5.2. Faze razvoja novog softverskog proizvoda prema 37signals modelu

Model upravljanja razvojem softverskih proizvoda *37signals* predstavljen je u knjizi *Getting real: the smarter, faster, easier way* (2006). U toj knjizi je taj model prvi put nazvan *Getting Real*, pa će se takav naziv koristiti u poglavljima koja slijede.

Getting Real se temelji na lateralnom poslovnom promišljanju. Temeljen je na prethodnim iskustvima tvrtke u razvoju i komercijalizaciji softverskih aplikacija. Premda načelno jasan, teško ga je detaljno definirati (Sajra, 2011). Platforma ovoga modela jasna je u njegovim snagama (Fried i Heinemeier Hansson, 2006). *Getting Real* omogućuje dobar rezultat jer potiče na bavljenje stvarnim problemima tijekom razvoja novog proizvoda, umjesto da je fokus na idejama vezano uz uzroke tih problema. Model tako prisiljava da tijekom razvoja novoga proizvoda primarni fokus bude na stvarnosti. *Getting Real* daje prednost dizajnu korisničkog zaslona pred specifikacijama i popratnoj dokumentaciji. Fried i Heinemeier Hansson smatraju specifikaciju funkcionalnosti iluzijom kao dijelom prodaje, dok je web stranica jedina stvarnost te prodaje. Svoj koncept argumentiraju težnjom da *Getting Real* brže vodi kreatora novog softverskog proizvoda postavljenom cilju. Drugim riječima odluke tijekom razvoja novog softverskog proizvoda temeljene su na stvarnosti, umjesto na apstraktnim aspektima.

Getting Real tako predstavlja pristup utemeljen i na agilnom manifestu, odnosno radi se o varijaciji agilnog pristupa temeljenog na šest načela. U knjizi *Getting real: the smarter, faster, easier way* (2006) u modelu upravljanja razvojem softverskog proizvoda prepoznaju se četiri faze (poglavlje: „Od ideje do implementacije“). Prije konkretnije analize treba istaknuti načela *Getting Real* pristupa. Ta opća načela su (Fried i Heinemeier Hansson, 2006):

1. *Getting Real* znači preskočiti sve ono što nije konkretna stvarnost (ljestvice, grafikoni, okviri, strelice, sheme, žičani okviri i dr.), te se fokusirati na ono što je konkretno – stvaranje prave stvari.
2. *Getting Real* je manje – manji softver, manje mogućnosti, manje dokumentacije, eliminaciju svega što nije od ključne važnosti za novi proizvod.
3. *Getting Real* znači ostati malen i biti agiln.
4. *Getting Real* počinje s korisničkim sučeljem, pravim prikazima onoga što će korisniku služiti. Počinje s onim što kupac stvarno doživljava i gradi se *unatrag*. To omogućuje da se stvori sučelje neposredno prije nego se pojavi mogućnost da softver ne bude prilagođen korisniku.
5. *Getting Real* upućuje na ponavljanja uz smanjenje troškova izmjena aplikacije. U *Getting Realu* sve se temelji na pokretanju, dotjerivanju i stalnim poboljšanjima što ga čini savršenim za web softver.
6. *Getting Real* pruža upravo ono što klijentima treba i eliminira sve što im ne treba.

Prvo načelo *Getting Real* je slično standardnim definicijama agilnosti u kontekstu dokumentacije. Opća pravila za sve agilne metode (Agile Manifesto, 2001), preporučuju fokusiranje na funkcionalan softver umjesto na dokumentaciju (*Radni softver iznad dokumentacije*). *Agile Manifest* podrazumijeva 12 eksplicitnih načela. Jedan od njih je fokusiranje na razvoj mjerenja procesa - *Softver koji radi primarna je mjera napretka* (Shah, 2020). Prvo načelo *Getting Real* je detaljnije i opsežnije definirano, podržano stručnom literaturom i spoznajama iz istraživanja (Sajra, 2011).

Drugo načelo *Getting Real* općenito je i opsežno. Potiče da se planirane aktivnosti provedu na jednostavan način na svim razinama. Pojam *manja masa* odnosi se na sve aktivnosti softverske tvrtke, uključujući posao na dnevnoj razini, planiranje, dokumentaciju i

aktivnosti dizajna proizvoda. Fraza *manje papirologije* također spada pod filozofiju *manje mase* samo je konkretna. Filozofija *manje papirologije* u skladu je s agilnim pristupom, u *Agilnom manifestu* definirana pravilom - *Radni softver ispred dokumentacije*. Filozofija *manje mase* je u skladu s načelom *Agilnog manifesta - Jednostavnost – smanjivanje nepotrebnih radnih aktivnosti od suštinske je važnosti*. Razmišljanje o manje značajki je blisko razmišljanju o *manjem softveru* jer je u većini slučajeva manje softvera posljedica smanjenja mogućnosti (koje nisu od posebnog značaja korisniku). Do zaključka o korelaciji između broja značajki i uspjeha proizvoda nije moguće doći zbog različitih korisničkih segmenata s različitim željama i potrebama. Inovatori i vizionari teže većem broju značajki i obilježja softverskog proizvoda nego što je korisnicima potrebno. S druge strane, korisnici žele proizvod jednostavan za korištenje, što često podrazumijeva manje značajki (Sajra, 2011).

Treće načelo je vezano uz agilnost kao rezultat veličine tvrtke. Konkretno se navodi - “Sav novac, sav marketing, svi ljudi na svijetu ne mogu osigurati agilnost malog poduzeća” (Fried i Heinemeier Hansson, 2006). *Agilni manifest* ne uvjetuje veličinu tvrtke, ali upućuje na značaj brze reakcije na promjene zahtjeva naručitelja i u kasnoj fazi razvoja novog softverskog proizvoda. Agilnost je u funkciji konkurentske prednosti za. Pri tome treba napomenuti da se u *Agilnom manifestu* u cilju više agilnosti ne uvjetuje da su za razvoj novog softverskog proizvoda angažirani timovi s manjim brojem djelatnika (Sajra, 2011).

Četvrto načelo *Getting Real* se može shvatiti dvosmisleno. Sugerira da se na početku dizajnira aplikacija iz korisničkog sučelja, odnosno da se na početku dizajniraju budući korisnički zasloni. Ovo načelo posljedica je dotadašnje dobre prakse tvrtke. U literaturi se o ovoj dilemi ne govori. Tek se spominje kako je iskustvo korisnika povezano s ovim načelom. Međutim u tvrtci *37signals* korisničkom iskustvu daju iznimnu važnost, pa mu pridaju i poseban značaj u modelu upravljanja razvojem softverskog proizvoda (Sajra, 2011).

Peto *Getting Real* načelo ukazuje na značaj ponavljanja čime se doprinosi snižavanju ukupnih troškova razvoja softverskog proizvoda. *Agilni Manifest* ukazuje na važnost fleksibilnosti prema potrebnim promjenama, pa se stječe dojam da je na tom postulatu ovo načelo definirano (Sajra, 2011).

Šesto načelo *Getting Real* je univerzalnog tipa. Radi se o načelu tzv. više razine u usporedbi s ostalim načelima jer sažima više načela u strateško promišljanje tvrtke. U skladu

je s načelom *Agilnog Manifesta* “Jednostavnost – smanjivanje nepotrebnih radnih aktivnosti je od suštinske važnosti” (Sajra, 2011).

Uvažavajući principe šest predstavljenih modela *Getting real* model upravljanja razvojem softverskog proizvoda tvrtke *37signals* razvija se kroz četiri faze

1. Generiranje ideja

U ovoj se fazi razmatraju smisao i svrha novog softverskog proizvoda. Često su fokusirani na vlastite potrebe. Tako su pri kreiranju *Basecampa* tražili rješenje objavljivanja ažuriranja projekata. Sugeriraju da klijent sudjeluje već u fazi generiranja ideja. Klijentima na uvid daju centraliziranu arhivu kako bi mogli vidjeti postojeća rješenja i proizvode koje su razvili. Tako generiranje ideja u ovome modelu podrazumijeva kompleksnije aktivnosti, razrješavanje mnogih opcijskih dilema - *što aplikacija treba učiniti / kako prepoznati kakva verzija aplikacije može biti doista korisna / što točno poduzeti i napraviti*. U ovome modelu razvoja softverskog proizvoda u fazi generiranja ideja iste se razrađuju na visokoj razini dovršenosti, bez bavljenja s trivijalnim dilemama (Fried i Heinemeier Hansson, 2006).

2. Skiciranje na papiru

Sljedeća faza, skiciranje sugerira korištenje skica jer one nastaju brzo, njihova izrada ne predstavlja značajan trošak. Korištenjem različitih oblika (pravokutnih tijela, krugova, linija) djelatnike angažirane na razvoju proizvoda se potiče da ideje slikovito predočavaju skicama. Tako se koncept iz faze generiranja ideja transformira u okvirni dizajn sučelja. Ova faza se smatra eksperimentiranjem pri čemu se ništa ne smije smatrati pogrešnim odgovorom (Fried i Heinemeier Hansson, 2006).

3. Stvaranje HTML oblička

Izrada HTML oblička i definiranje njegovih obilježja (odjeljak, tijekovi i sl.). U ovoj fazi se oblikuje nešto stvarno, kako bi svi (i kreatori i korisnici) mogli vidjeti kako to izgleda na ekranu. Za *Basecamp* je prvo bio oblikovan ekran *objavi poruku*, a zatim ekran *uredi poruku*. Tek nakon toga se nastavilo s daljnjim razvojem. Dakle, tek nakon toga na red dolazi pisanje programskog koda, nakon što je izrađen HTML i CSS model (Fried i Heinemeier Hansson, 2006).

4. Kodiranje

Kada se konstatira da prototip zadovoljava potrebe budućeg korisnika, te omogućuje dovoljno potrebnih funkcija, na redu je izrada programskog koda. Pri tome se tijekom i ove faze razvoja treba pridržavati načela fleksibilnosti i omogućiti što više ponavljanja. Sugerira se, ako je to potrebno odbaciti spoznaje i rezultate ranijih faza razvoja i započeti proces ponovno. Sugerira se ovaj ciklus/provjeru provesti više puta (Fried i Heinemeier Hansson, 2006).

Navedene četiri faze razvoja predstavljaju okvir modela koji prakticira tvrtka *37signals*. Iščitavajući knjigu *Getting real: the smarter, faster, easier way* (2006) između redaka uočava se postojanje još nekih procesa koji nisu jasno definirani u ove četiri faze, pa je evidentno da vlasnici tvrtke, odnosno kreatori modela isto drže poslovnom tajnom. Slijedom tih spoznaja evidentno je da:

- Postoji i faza testiranja koja se provodi kroz dva koraka. Prvo se testira na razini gdje novi softver rješava konkretan problem u poslovanju, a nakon toga slijedi testiranje softvera u praksi. Ovdje je fokus na primarnom cilju, a ne na formi.
- Postoje i faze nadogradnje i unapređenja softvera u kojima se otklanjaju svi uočeni nedostaci tijekom faze testiranja.
- Prije stvarne komercijalizacije prakticira se standard *skraćivanje na pola* kojim se konačni proizvod pojednostavljuje otklanjanjem svih obilježja koja ne narušavaju njegovu inicijalnu korisničku svrhu.

Sukladno petom načelo *Getting Real*a temeljenom na ponavljanju proces razvoja se ponavlja više puta, što se napominje i u četvrtoj fazi razvoja (*prirodno je kroz cijeli proces proći više puta*).

5.3. Tržišna iskustva

Šire uvriježeni standard informatičke industrije *kreirajte proizvod koji rješava potrebe vaše glavne publike* sugerira osiguranje sredstava potrebnih za izradu novih proizvoda koje je moguće prodati ciljanim kupcima/korisnicima. Nadalje se preporučuje ulaženje i širenje na

rastućim tržištima sve dok mala tvrtka ne naraste u veliko poduzeće. Ovo su tek neke od paradigmi kako bi se softverska tvrtka trebala razvijati ako želi biti i ostati tržišno uspješna. *37signals* je jedna od najstarijih uspješnih Internet tvrtki koja je ovakve paradigme potpuno transformirala u nove postulate tržišnog ponašanja (Shah, 2020). Da bi se njihov pristup jasno predstavio predočiti će se iskustva njihovih softverskih proizvoda *Ruby on Rails*, *Basecamp* i *HEY* od pojavljivanja na tržištu. Svi ti proizvodi razvijeni su temeljem načela i modela razvoja *37signals*.

Ruby on Rails je rezultirao revolucijom u web programiranju. Kada je predstavljen 2005. Godine predstavio je novi pristup izradi web aplikacija. Jedna od novih stvari je bila paradigma dizajna softvera kojom se konvencija postavlja ispred konfiguracije olakšavajući tako rad *developer* na mnogim razinama. Konvencija umjesto konfiguracije, također poznata kao kodiranje po konvenciji, je paradigma dizajna softvera koju koriste softverski radni okviri koja pokušava smanjiti broj odluka koje programer mora donijeti bez nužnih gubitka fleksibilnosti i bez ponavljanja. Zajedno s *Djangom*, *Pythonovim* najpopularnijim radnim programskim okvirom objavljenim kasnije iste godine, *Rails* je propagirao korištenje MVC (model-view-controller)¹ obrasca i dobre razvojne prakse, kao što je *DRY* (*ne ponavljaj se*) princip. *Rails* koncept web razvoja ne predviđa developerima zamorna kodiranja. Umjesto toga se mogu fokusirati na konkretne poslovne značajke i logiku aplikacije koju razvijaju. U praksi je to rezultiralo većom produktivnošću *developer* i bržim dovršavanjem MVP-ova (Minimum Viable Product)² i aplikacije (Nowak, 2022). O tržišnoj uspješnosti *Ruby on Rails* govori visina prosječne godišnje plaće *developer* koji koriste taj radni programski okvir. *Ruby on Rails* je u 2022. drugi najplaćeniji radni programski okvir (prosječna godišnja plaća *developer* je cca 90.000\$, u usporedbi s godinu dana ranije 77.556\$). Slijedom rasta potražnje za *developerima* koji rade u *Ruby on Rails-u* porasla je prosječna mjesečna plaća svih kategorija *developer*. Ove godine je *Ruby on Rails* u visini plaće *developer* prestigao radni programski okvir *Phoenixa* (prosječna godišnja plaća 94.644\$) (StackOwerflow, 2022). *Phoenix* je programski radni okvir čija je nova verzija, puštena u rad u siječnju 2022. godine pokazala izričito dobrom za izradu modernih web aplikacija. Što je uvelike što uvelike utjecalo na visinu plaće tih programera korisnika *Phoenix* radnog programskog okvira. (Soppa, 2022). U svakom slučaju ovi podaci ukazuju izniman uspjeh *Ruby on Railsa* ukupno

¹ MVC - model-view-controller ([Izvor](#))

² MVP - razvojna je tehnika u kojoj se novi proizvod ili web-mjesto razvija s dovoljno značajki da zadovolji rane korisnike u svrhu dobivanja povratnih informacija za daljnji razvoj proizvoda ([Izvor](#)).

među svim programskim radnim okvirima, a *Phoenix* se nameće kao konkurent koji se još mora dokazati u budućnosti.

Većina informatičkih tvrtki je bila usredotočena na "sljedeću veliku stvar", kao što je to na primjer bilo *strojno učenje*, besplatni prvoklasni alati, *chatbotovi*³, *Basecamp* je bio fokusiran na stalno poboljšavanje osnovnog proizvoda poslovanja tvrtke. Ovo je bilo razmišljanje Frieda temeljem savjeta Jeffa Bezosa kada je započeo poslovanje tvrtke - "*Pronađite ono nešto što se neće promijeniti u vašem poslovanju i uložite velika sredstva u to.*" Upravo je to nagnalo *37signals* da donese odluku o prelasku natrag u tvrtku s jednim proizvodom. Tada je Fried je obavio: "Nikad nećemo zaboraviti što je *Basecamp* uopće učinilo tako popularnim, a to je jednostavnost rada. Jednostavan je, jednostavan za korištenje, jednostavan za razumijevanje, jasan, vjerodostojan i pouzdan. Nastavit ćemo razvijati proizvod u tom smjeru." To je cilj *Basecampa* učiniti ga jasnijim, sigurnijim i pouzdanijim proizvodom, ne vezanim za određeno vrijeme u evoluciji SaaS industrije. Definitivno postoji sloboda u ovom pristupu koje su se mnoge druge zrele SaaS tvrtke odrekle u zamjenu za prihode i brzi rast. Uspjeh *Basecampa* nije oblikovan onim što druge tvrtke rade ili onim što se u Silicijskoj dolini najavljivalo kao sljedeća velika stvar. Potpuno se oslanja na *Basecampov* tim, proizvod i kupce. (Shah, 2020).

HEY Email aplikacija je za uslugu e-pošte koja košta 99\$ godišnje s @hey.com adreseom koje imaju za cilj popraviti ono što nije u redu s e-poštom općenito. Omogućuje vam brzo pregledavanje novih e-poruka, blokira kodove za praćenje⁴ i optimizira tijek rada kako bi bili više produktivni i kao krajnji rezultat omogućuje užitek u korištenja e-pošte čitanjem samo ono što želite. Projekt su najavili Fried i Heinemeier u veljače 2020. *Basecampov* tim je radio godinama na vlastitoj usluzi e-pošte kako bi se prilagodili tržištu. Prema riječima kreatorima, željeli su slanje e-pošte na brži i jednostavniji način za korisnike. O uspješnosti *HEYa* govori podatak da čak i prije lansiranja, *HEY Emaila* lista čekanja brojala je 50.000 adresa, a tjedan dana kasnije je već dvostruko veći broj. Da, u samo tjedan dana postojanja *HEY Email* je prikupio 100.000 korisnika. Što se tiče njihove konkurencije, slobodno se može reći da je *HEY Email* veliki uspjeh. Prema proračunima Owlara (2021) procijenjeno je da *HEY Emailov* godišnji prihod iznosi gotovo milijun dolara. (Kickstarz Side Hustle, 2022)

³ Chatbot je softver ili računalni program koji simulira ljudski razgovor ili "čavrljanje" putem tekstualne ili glasovne interakcije. ([Izvor](#))

⁴ Kod za praćenje - ili takozvani piksel za praćenje e-pošte kvadratna je slika veličine 1x1 piksel stvorena linijom koda koja je umetnuta u poruku e-pošte u svrhu dobivanja analitičkih podataka. ([Izvor](#))

5.4. Mogućnosti primjene modela

Getting real: the smarter, faster, easier way (2006.) je svojevrstan vodič, ispunjen zdravorazumskim pravilima, kako na pametniji, brži i bolji način razviti uspješnu web aplikaciju. Lako je čitljiv s dobrim uvidom u to kako su u *37signalsu* razvijali svoje proizvode i savjetima kako se to može primijeniti na radnu metodologiju ne samo programera softvera već bilo čiju poput dizajnera, voditelj projekta, poduzetnika ili trgovaca. *Getting Real* nije udžbenik, više je napisan u stilu manifesta odnosno filozofije tima i tijeka rada tvrtke *37signals*. Napisan je laičkim jezikom koji ne-developeri mogu razumjeti i pun je čvrstih stavova. Ova je knjiga antiteza ogromnim štivima, kao što je *Designing Interactions* Billa Moggridgea iz 1981. godine. *Getting Real* izaziva paradigme i razbija stare predrasude u web developmentu. (Busse, 2022)

Iako nema toliko novih aspekata u načelima *Getting Real*, može se reći da je mala tvrtka pokupila prave principe iz raznih poslovnih praksi te razvojna pravila i načela za jačanje njihovih aktivnosti. Iz praktičnog gledišta, može se zaključiti da bi istraživana tvrtka mogla biti dobra studija slučaja za usporedne analize i za druge vlasnike malih poduzeća. S teorijskog gledišta, može se zaključiti da ne postoje sukobi između dosadašnjih istraživanja i *Getting Real* pristupa (Sajra, 2011).

Na temelju svih dosad prikupljenih podataka iz raznih izvora literature u ovom radu može se zaključiti kako postoji mogućnost primjene *Getting Real* modela upravljanjem razvoja softverskih proizvoda od strane drugih softverskih tvrtki naročito, ako je riječ o malim i srednjim poduzećima. Također, model ne može biti primijenjen za druge vrste proizvoda, ali, ako se ne gleda model striktno vezano za softver, filozofije vezane uz model mogu biti od pomoći svima koji se bave razvojem proizvoda bilo koje vrste.

6. PRIMARNO ISTRAŽIVANJE

6.1. Uvodne naznake

Primarno istraživanje prvotno će biti provedeno koristeći model dubinskog strukturiranog intervjua s korisnicima različitih proizvoda tvrtke koja je predmet ovoga istraživanja.

Intervju će za cilj imati utvrditi percipiranu kvalitetu proizvoda *Ruby on Rails*, *Basecamp* i *HEY*. Intervju se sastoji od sljedećih pitanja:

- Koju od navedenih poslovnih pozicija obnašate ili ste obnašali? (Ponuđeni odgovori: Senior developer i Projektni manager)
- Koje od proizvoda tvrtke *37signals* koristite ili ste koristili? (Ponuđeni odgovori: proizvoda *Ruby on Rails*, *Basecamp* i *HEY*)
- Zašto ste se odlučili za korištenje tog/tih proizvoda?
- Kakva su vaša dosadašnja korisnička iskustva vezana uz navedeni proizvod/e?
- Kako ocjenjujete funkcionalnost proizvoda i zašto?
- Koje bi značajke funkcionalnosti posebno istaknuli u pozitivnom i/ili negativnom smislu?
- Kakva su vaša dosadašnja iskustva vezana uz korisničku podršku? Koliko brzo i lako ste stvarno dolazili do rješenja problema?
- Prokomentirajte lakoću upotrebe proizvoda.
- Kakva je po vama vrijednost za novac proizvoda? (Ne odgovara se za *Ruby on Rails*)

Uz zadnje pitanje stoji napomena kako se na pitanje „Ne odgovara se za *Ruby on Rails*“. Razlog tome je što je *Ruby on Rails* dostupan besplatno.

U nastavku istraživanja bit će provedena usporedna analiza proizvoda *Ruby on Rails* i *Basecamp* koristeći kombinirano podatke dobivene primarnim i sekundarnim istraživanjem.. Analiza će biti provedena na gore navedenim proizvodima čija će se korisnička obilježja analizirati u odnosu na konkurentne proizvode slične namjene. Usporedba će biti provedena po uzoru na istraživanje provedenom na *Temple University* pod nazivom *Evaluating web development frameworks: Django, Ruby on Rails and CakePHP* (2009).

6.2. Analiza podataka iz strukturiranog intervjua

U ovom dijelu rada će bit analizirani podaci dobiveni putem provedenog strukturiranog intervjua. Strukturirani intervju proveden je razgovorom lice u lice s tri ispitanika slijedom pitanja navedenih u uvodnom dijelu ovog poglavlja.

Svo troje ispitanih su senior developeri koji su bar nekada obnašali funkciju voditelja projekta. Dvoje od njih koristili su *Ruby on Rails* i *Basecamp* dok je jedan koristio samo *Basecamp* platformu. Nitko od ispitanih ne koristi HEY Email platformu.

Od dvoje ispitanika jedan je koristi *Ruby on Rails* tri godine, a drugi pet godina dok su isti koriste *Basecamp* jedan tri godine, a drugi tek jednu godinu. Treći ispitanik koristi *Basecamp* divje godine.

Na sljedeća pitanja komentari će biti poredani ispitanik 1, ispitanik 2 i ispitanik 3 od kojih ispitanici 1 i 2 odgovaraju na pitanja vezano za *Ruby on Rails* i *Basecamp*, a ispitanik 3 odgovara samo za *Basecamp*.

Kao odgovor na pitanje „Zašto ste se odlučili za korištenje tog/tih proizvoda?“ za *Ruby on Rails* dani su sljedeći odgovori:

Ispitanik 1: “*Ruby on Rails* sam počeo koristiti kao pripravnik za potrebe tvrtke, a *Basecamp* je druga tvrtka koristila za project management.”

Ispitanik 2: "Pratim rad Davida Heinemeiera Hanssona, kreatora *Ruby on Railsa* i vjerujem u njegov način razvoja proizvoda te sam ga zato i počeo koristiti. *Basecamp* koristi tvrtka za koju radim."

Ispitanik 3: "Jednostavna dostupnost platforme, prihvatljiva cijena, jednostavno korisničko sučelje, olakšano rješavanje strukturiranih i nestrukturiranih problema i procesa."

Na pitanje o dosadašnjim korisničkim iskustvima ključne izjave u odgovorima su izgledale ovako:

Ispitanik 1: „*Ruby on Rails* i općenito *Ruby* kao jezik je jako intuitivan i jako ga je lagano naučiti, ali ima problema pri skaliranju većih aplikacija ukoliko kod nije dobro napisan. *Basecamp* je jako dobar za projekt management i taskove, ali kako kad ima previše svojstava“

Ispitanik 2: „Zadovoljan sam, rješenja uvijek imaju fokus na otklanjanju nepotrebnih stvari i omogućuju interakciju na što više načina kako bi korisnik imao što bolje iskustvo.“

Ispitanik 3: „Bez zamjerke osim puno efikasnijeg korištenja alata na webu nego u aplikaciji.“

Kako ocjenjujete funkcionalnost proizvoda i zašto?

Ispitanik 1: „Petica“ za *Ruby on Rails*. Dokumentacija jako dobra i jako lagano se kodira. Kod je samo objašnjavajući. Nisam imao nikakvih većih problema ni s kvalitetom koda pa ne mogu reći da je skaliranje problem. *Basecamp* je dobar, mada kao programeru mi baš mi ne znači to jer samo želim dobiti zadatak.“

Ispitanik 2: „Odlično - Oba rješenja koja sam koristio olakšali su moju svakodnevicu.“

Ispitanik 3: „Sasvim solidno na skali od jedan do pet dao bi ocjenu četiri. Automatizirano poslovanje i puno jednostavnije obavljanje svakodnevnih aktivnosti.“

Koje bi značajke funkcionalnosti posebno istaknuli u pozitivnom i/ili negativnom smislu?

Ispitanik 1: „(*Ruby on Rails*) Brzo učenje i lako izrađivanje srednjih aplikacija. Kod *Basecampa* istaknuo bi *schedule* (raspored) u pozitivnom smislu“

Ispitanik 2: „Raznovrsnost pristupa. Korisniku je omogućena interakcija sa proizvodom na puno načina i s time on ne mora učiti alat nego njegova intuicija je najčešće podržana. Za *Basecamp* nemam što istaknuti“

Ispitanik 3: „Nestabilnost mobilne aplikacije.“

Kakva su vaša dosadašnja iskustva vezana uz korisničku podršku? Koliko brzo i lako ste stvarno dolazili do rješenja problema?

Ispitanik 1: „(*Basecamp*) Nisam imao kontakta i ne mogu ništa reći. *Ruby* ima jako dobar community i dokumentaciju pa nije ni bilo potrebno.“

Ispitanik 2: „Nisam ju koristio.“

Ispitanik 3: „Nisam imao kontakta s korisničkom podrškom.“

Prokomentirajte lakoću upotrebe proizvoda.

Ispitanik 1: „Oba jako lagana. Barem kao programeru.“

Ispitanik 2: „Jedan i drugi proizvod su vrlo intuitivni.“

Ispitanik 3: „Vrlo jednostavno korištenje uz minimalno vrijeme za učenje rada u programu.“

Kakva je po vama vrijednost za novac proizvoda?

Ispitanik 1: „Za *Basecamp* cijena je jako dobra jer se dobije beskonačno usera (korisnika unutar tvrtke) i sve mogućnosti. Problem prema drugima je po meni da nisam vidio free plan. Bolje od drugih je da se ne placa per user (po korisniku), a dobiju se skoro isti featuri (svojstva) i integracije.“

Ispitanik 2: „Nudi više od konkurencije u smislu da ima značajke većine drugih proizvoda a ne naplaćuje nepotrebne stvari.“

Ispitanik 3: „Nemam što puno reći. Ok je.“

6.2.1. Zaključak strukturiranog intervjua

Već tijekom provođenja ovog intervjua, a i kasnijom analizom podataka primijećeno je kako svi ispitanici imaju generalno pozitivno mišljenje o proizvodima tvrtke *37signals*. Nije bilo značajnijih negativnih komentara osim primjerice pritužbe na nestabilnost *Basecampove* mobilne aplikacije i netipičnog komentara drugog ispitanika kako je *Basecamp* ima previše svojstava. Svi ostali komentari bili su iznimno pozitivni ili u najmanju ruku zadovoljavajući. To zadovoljstvo naravno proizlazi iz kvalitete *37signalsovih* proizvoda *Basecamp* i *Ruby on Rails*. Rezultat toga je pozitivna percepcija kvalitete *37signalsovih* proizvoda kod njihovih korisnika. Nadalje, spomenuta kvaliteta *37signalsovih* proizvoda posljedica je tvrtkinog modela upravljanja razvojem softverskog proizvoda. Iz svega toga proizlazi zaključak kako korištenje modela upravljanja razvojem softverskog tvrtke *37signals* ima pozitivan doprinos na percipiranu kvalitetu njihovih proizvoda u svijesti potrošača.

6.3. Usporedba proizvoda *Ruby on Rails* i *Basecamp* s konkurentnim proizvodima

U ovom dijelu rad bit će istraženi *37signalsovi* proizvodi *Ruby on Rails* i *Basecamp* kao i njihovi konkurentski proizvodi iste kategorije u svrhu usporedbe njihove kvalitete koja reprezentira kvalitetu tvrtkinog modela upravljanja razvojem softverskog proizvoda.

Koristit će se metoda Instituta za poslovnu ekonomiju i informacijske tehnologije (IBIT) sveučilišta Temple iz 2009. godine, istom metodom ponoviti će se istraživanje te tako dobiti aktualne podatke za 2022. Cilj IBIT-a je proizvesti vrhunsko znanje za stvaranje i održavanje izvrsnosti u informacijskih tehnologija (Plekhanova, 2009).

Ruby on Rails bit će uspoređen s *Djangom* i *CakePHP* programskim radnim okvirima, a *Basecamp* s platformama *Asana* i *ClickUp*.

6.3.1. Ruby on Rails

Ruby on Rails programski radni okvir prvotno je osmišljen za internu upotrebu, a tek kasnije postao svima dostupan kao besplatni okvir otvorenog koda. Svojedobna inovacija kasnije je kopirana od strane svih velikih konkurenata.

Kao što je već rečeno u ovom dijelu istraživačkog rada *Ruby on Rails* bit će uspoređen s programskim radnim okvirima *Django* i *CakePHP* na način da će se analizirati istraživanje Instituta za poslovnu ekonomiju i informacijske tehnologije sveučilišta Temple iz 2009. godine, istom metodom ponoviti istraživanje te tako dobiti aktualne podatke za 2022. godinu i na kraju usporediti rezultate.

Kao napomena uz IBITovo istraživanje stoji (Plekhanova, 2009): “Ovo se izvješće fokusira na tri vodeća okvira za web razvoj otvorenog koda: *Django*, *Ruby on Rails* i *CakePHP*. Sva tri okvira imaju sličnu arhitekturu i tvrde da imaju slične karakteristike, kao što su znatno poboljšana produktivnost i ponovna upotreba koda. Ovo izvješće pruža metodologiju za ocjenu svakog okvira. Metodologija, kriteriji i ponderi navedeni u ovom izvješću su generički i sveobuhvatni. Svaka organizacija treba prilagoditi metodologiju ovog izvješća svom jedinstvenom kontekstu.”

Pri izboru konkurentskih programskih okvira kao kriterij uzete su sljedeće karakteristike koje posjeduje *Ruby on Rails*:

- Licenca otvorenog koda,
- MVC (model-view-controller)⁵ arhitektura, iako *Django* svoju arhitekturu naziva MVT (model-view-template) vrlo je slična onoj kod *Ruby on Rails* te se može uzeti u obzir,
- Mogućnost pojednostavnjenja razvojnih procesa automatizacijom neki od dijelova,
- Dodavanje strukture kodu i čitljiviji kod
- Ponovna upotreba komponenata za brži razvojni proces

⁵ MVT - je obrazac za dizajn softvera. To je zbirka od tri važne komponente prikaza, modela i predloška. Model pomaže u rukovanju bazom podataka. To je sloj pristupa podacima koji rukuje podacima. ([Izvor](#))

- Podržava istodobno kreiranje/ažuriranje sadržaja i razvoj.

U tablici 1 obrađeni su podaci vezani za podršku pri razvoju korisničkog sučelja koju pojedini programski radni okviri pružaju. Procjenjuju se dva parametra upotrebljivost predložaka i podrška za JavaScript knjižnicu. Predlošci formiraju korisničko sučelje programskog radnog okvira te su stoga bitan element pri procjeni njihove kvalitete. JavaScript skriptni jezik široko je korišten u izradi web stranica i aplikacija koji omogućava interakciju istih s korisnikom. Iz toga proizlazi važnost podrške njegove knjižnice.

	Django	Ruby on Rails	CakePHP
Upotrebljivost od predložaka	Jednostavna sintaksa; uključuje predloške jezika	Složenija sintaksa; uključuje isječke koda	Složena sintaksa; uključuje isječke koda
Podrška za JavaScript knjižnicu	Zahtijeva JavaScript, više slobode	Jednostavan za upotrebu, Manje slobode Nenametljiv JavaScript pipeline	Jednostavan za korištenje, manje slobode
Ocjena	4	4	3

Tablica 1: Razvoj korisničkog sučelja

Izvor: Istraživanje autora, 2022.

Tablica 2 odnosno održavanje se u ovom slučaju odnosi na mogućnosti održavanja stranica koje izrađenih uz pomoć programskog radnog okvira. Zbog velikog broja promjena i dinamičnosti rada na web stranicama i aplikacijama, na kojima učestalo radi po više programera dolazi do izraženije potrebe korisnika da sami upravljaju sadržajem. Stoga iste moraju imati razvijeno korisničko sučelje koje kojima korisnik može lako upravljati i mijenjati sadržaj.

	Django	Ruby on Rails	CakePHP
Upravljanje sadržajem	Automatski generiran	Automatski generiran	Dobar
Autentifikacija korisnika	Dobar	Dobar	Ok
Korisničke sesije	Podržava	Podržava	Podržava
Ocjena	4	4	3

Tablica 2: Održavanje

Izvor: Istraživanje autora, 2022.

Tablica 3, upravljanje podacima i migracija odnosi se na kvalitetu povijesti podataka i mogućnosti vraćanje unatrag u svrhu smanjenja rizika od gubitka važnih podataka. Lakoća povratka u slučaju poteškoća ili greške ključna je za olakšavanje razvojnog procesa jer su one neizbježne. Također dobro upravljanje podacima daje programeru osjećaj slobode jer se može lakše priuštiti pravo na pogrešku što rezultira i većom kreativnošću koja se smatra poželjnom pri pisanju koda.

	Django	Ruby on Rails	CakePHP
Upravljanje podacima	Ručna migracija	Snažno	Loše
Relacijsko preslikavanje objekata	Da	Da, najrazvijenije	Da
Ocjena	4	5	3

Tablica 3: Upravljanje podacima i migracija

Izvor: Istraživanje autora, 2022.

Tablica 4 daje podatke o mogućnosti testiranja. Otklanjanja pogrešaka i alati za testiranje kao i njihova jednostavnost i intuitivnost pri rješavanju razvojnih probleme u svakom okviru od velike su važnosti za proces razvoja. To se najbolje očituje kroz doprinos smanjenju vremena potrošenog na testiranje i otklanjanje pogrešaka što uvelike ubrzava i olakšava cjelokupan proces.

	Django	Ruby on Rails	CakePHP
Otklanjanje pogrešaka	Vrlo Dobro	Vrlo Dobro	Dobro
Alati za testiranje	Nekoliko	Razni	Nekoliko
Okvir za testiranje	Da	Da	Da
Ocjena	4	5	4

Tablica 4: Mogućnost testiranja

Izvor: Istraživanje autora, 2022.

U tablici 5 prikazana je popularnost programskih jezika koju najrelevantnije pokazuje TIOBE indeks. TIOBE je tvrtka specijalizirana za procjenu i praćenje kvalitete softvera te već dugi niz godina svakog mjeseca kreira popis najpopularnijih programskih jezika. Ocjene se dobivaju uz pomoć brojnih kvalificiranih inženjera diljem svijeta, tečajeva i nepristranih prodavača (Tiboe, 2022).

	Django	Ruby on Rails	CakePHP
Popularnost jezika	#1	#15	#10
Ocjena	5	2	3

Tablica 5: Popularnost

Izvor: Istraživanje autora, 2022.

Rezultati tablice 6 daju nam ocjenu zajednice i dostupne podrške za svaki programski radni okvir i lakoću dobivanja odgovora na potencijalna pitanja koja bi programera. Drugim riječima, pokazuje koliko razvijenost programskog radnog okvira u smislu podrške, zajednice i dokumentacije.

	Django	Ruby on Rails	CakePHP
API Referenca	Dobra	Dobra	Dobra
Knjige	51	53	10
Uputstva za korisnike	Vrlo dobra	Vrlo dobra	Dobra
Blogovi korisnika	Nekoliko	Puno	Nekoliko
Zajednica	Dobra	Razvijena	Mala
Ocjena	5	5	3

Tablica 6: Zajednica i razvijenost

Izvor: Istraživanje autora, 2022.

Tablica 7, tržišni potencijal odnosno sposobnost prodaje na tržištu očituje se kroz potražnju za programerima koji rade u pojedinom programskom radnom okviru, kao i ukupnu potražnja po programskom jeziku koji radni okviri koriste. Podaci po ponudi poslova uzeti su s LinkedIn platforme koja trenutno pruža jednu od najvećih baza podataka ponude i potražnje poslova u IT sektoru. Za programske radne okvire na spomenutoj platformi trenutno broj dostupnih radnih mjesta veći je od: 32000 za *Django*, 74000 za *Ruby on Rails* i 9000 za *CakePHP* (LinkedIn, 2022).

	Django	Ruby on Rails	CakePHP
Trenutni poslovi i budući trendovi (okvir)	Visok porast	Iznimno visok porast	Nizak porast
Trenutni poslovi i budući trendovi (programski jezik)	Visoka potražnja	Visoka potražnja	Visoka potražnja
Ocjena	4	5	3

Tablica 7: Tržišni potencijal

Izvor: Istraživanje autora, 2022.

U tablici 8 izračunate su ukupne ocjene za svaki pojedini proizvod na način da je za svaki parametar dan ponder koji se množi s dodijeljenom ocjenom te se tako dobije ponderirana vrijednost. Zbroj ponderiranih vrijednosti daje nam ukupnu ocjenu za pojedini proizvod.

		Django		Ruby on Rails		CakePHP	
Kriteriji evaluacije	Ponder	Ocjena	Ponderirana vrijednost	Ocjena	Ponderirana vrijednost	Ocjena	Ponderirana vrijednost
Razvoj korisničkog sučelja	0.2	4	0.8	3	0.6	3	0.6
Održavanje	0.15	4	0.6	3	0.45	3	0.45
Upravljanje podacima i migracija	0.2	4	0.8	3	1	3	0.4
Testabilnost	0.15	4	0.45	4	0.6	4	0.3
Popularnost	0.1	5	0.5	3	0.3	3	0.5
Zajednica i zrelost	0.1	5	0.5	3	0.5	3	0.3
Tržišni potencijal	0.1	4	0.4	3	0.4	3	0.4
Ukupno:	1	-	4,2	-	4,35	-	3,15

Tablica 8: Ukupna ocjena

Izvor: Istraživanje autora, 2022.

Usporedbom dobivenih ocjena u istraživanju Plekhanove (2009) i sadašnjeg istraživanja može se ustvrditi kako su svi proizvodi, kroz unapređenja vođena modelom upravljanja razvojem softverskog proizvoda tvrtke vlasnika pojedinog proizvoda, postali bar malo kvalitetniji. Najveći napredak ostvario je *Ruby on Rails* čija je ocjena porasla s 3,85 na 4,35 dok je *Django* ostvario rast s 4,05 na 4,2, a *CakePHP* sa 2,95 na 3,15. Uzimajući u obzir dane rezultate može se zaključiti kako je *Rubi on Rails* postao najkvalitetniji među konkurentima.

6.3.2. Basecamp

Drugi 37signalsov proizvod uzet za ovo istraživanje je *Basecamp*. *Basecamp* je platforma za upravljanje projektima, usklađivanje rada na daljinu i timsku komunikaciju. Sve to omogućuju i konkurentske platforme *Asana* i *ClickUp* koje kao i *Basecamp* spadaju među najpopularnije platforme ovog tipa. Pri izboru konkurentskih platformi morali su biti zadovoljeni i sljedeći kriteriji:

- web-bazirana platforma,
- imati iPhone i Android aplikaciju i
- prilagođenost svim skupinama korisnika po veličini od pojedinačnih korisnika do velikih poduzeća.

	<i>Basecamp</i>	<i>Asana</i>	<i>ClickUp</i>
Vrijednost za novac	Dobra	Vrlo dobra	Vrlo dobra
Jednostavnost korištenja	Vrlo jednostavno	Jednostavno	Jednostavno
Značajke	59	210	225
Integracije	228	427	100
Korisnička podrška	Vrlo dobra	Vrlo dobra	Odlična, dostupna telefonska korisnička služba
Broj pretplatnika	>120.000	>100.000	>100.000
Trend rasta	Blagi rast	Usporeni rast	Ubrzan rast

Tablica 9: Usporedba Basecampa, Asanae i ClickUpa

Izvor: Istraživanje autora, 2022.

Na temelju provedenog istraživanja u tablici 9 dane su opisne i brojčane vrijednosti za navedene parametre. S obzirom važnost za korisnika najznačajniji su vrijednost za novac i jednostavnost korištenja te se uz njih kao najrelevantniji pokazatelj uzima broj korisnika. Iz tog razloga u tablici 10 tim parametrima dodijeljen je ponder od 0,2 dok za ostale iznosi 0,1.

Kriteriji evaluacije	Ponder	Basecamp		Asana		ClickUp	
		Ocjena	Ponderirana vrijednost	Ocjena	Ponderirana vrijednost	Ocjena	Ponderirana vrijednost
Vrijednost za novac	0,2	4	0,8	5	1	5	1
Jednostavnost korištenja	0,2	5	1	4	0,8	4	0,8
Značajke	0,1	4	0,4	5	0,5	5	0,5
Integracije	0,1	4	0,4	5	0,5	3	0,3
Korisnička podrška	0,1	4	0,4	4	0,4	5	0,5
Broj pretplatnika	0,2	5	1	4	0,8	4	0,8
Trendovi	0,1	4	0,4	3	0,3	5	0,5
Ukupno:	1	-	4,4	-	4,3	-	4,4

Tablica 10: Ocjena Basecampa, Asanae i ClickUpa

Izvor: Istraživanje autora, 2022.

Gledajući vrijednost za novac vidljivo je kako *Basecamp* malo zaostaje no to je zato što je njihov cjenik prilagođen srednjim poduzećima koja i jesu njihova primarna ciljna skupina dok u većini istraživanja brojčano se nalazi najviše malih poduzeća. Naime *Basecamp* nudi besplatnu verziju ili 99\$ mjesečno za neograničen broj korisnika i projekata dok ostale dvije platforme naplaćuju po korisniku odnosno od 13,49\$/korisnik do 30,49\$/korisnik za *Asana* i od 5\$ do 19\$ za *ClickUp* uz dostupnu besplatnu verziju. Iz navedenog je vidljivo kako s porastom broja korisnika *Basecamp*ov plan naplate postaje isplativiji.

Jednostavnost korištenja je definitivno najveća *Basecamp*ova prednost. Sve tri platforme su osmišljene tako da rad na njima bude što jednostavniji i intuitivan no *Basecamp* ipak prednjači. Tome u prilog ide i činjenica da *Basecamp* posjeduje daleko manji broj značajki. To može biti i ograničavajući faktor zbog čega i parametar značajke i je niže ocjenjen za *Basecamp*, ali te iste značajke često znaju biti zbunjujuće i napravljene tako da ih korisnik mora nepotrebno koristiti u svakodnevnom radu što usporava proces rada.

Broj pretplatnika ukazuje na broj kompanija pretplaćenih na platformi što stvaran broj pojedinaca koji koriste određenu platformu čini višestruko većim. Ipak se relevantnijim smatra broj onih koji su se odlučili na kupnju pretplate na platformu nego broj onih koji je koriste jer je njihov poslodavac tako odredio. Brojčano gledano tu *Basecamp* prednjači za oko 20.000 korisnika ispred *Asana*e i *ClickUp*a. Uz tu brojku usko se vezuju i trendovi rasta broja korisnika. *Basecamp* je stavljen na tržište 2004., *Asana* 2008., a *ClickUp* tek 2017. godine. Za *Basecamp* i *Asanu* se može reći da su dosegli zrelost no *Basecamp* i dalje ima blag, ali

konstantan rast dok je kod *Asanae* on nešto sporiji. *ClickUp* je još uvijek relativno nov proizvod na tržištu koji je nakon faze uvođenja doživio nagli rast koji još uvijek traje. Gledajući na trenutne trendove može se očekivati da će *ClickUp* uskoro sustići *Basecampov* broj pretplatnika.

Zaključno za *Basecamp*, očito je da se među iznimno jakom konkurencijom svojom kvalitetom zadržava u samom vrhu kroz dugi niz godina. Unatoč tome što su u ovom istraživanju *Basecamp* jednako ocjenjeni sa 4,4 prema komentarima korisnika malu prednost ipak preuzima *ClickUp* pogotovo, ako se u obzir uzme perspektiva u budućnosti.

6.3.3. Analiza utjecaja broja zaposlenih

U ovom dijelu bit će analiziran i uspoređen broj zaposlenih u tvrtkama vlasnicima proizvoda *Basecamp*, *ClickUp* i *Asana* u odnosu s ostvarenim prihodima kako bi se utvrdio utjecaj broja malog broja zaposlenika tvrtke *37signals* na efektivnost procesa upravljanja razvojem softverskog proizvoda. Također, bit će prokomentiran i broj korisnika proizvoda i usluga, način na koji se ostvaruju prihodi i odnos s kvalitetom.

Naziv tvrtke	Broj zaposlenih	Broj korisnika	Prihod u milijunima dolara	Prihod po zaposleniku u dolarima
37signals	66	120.000	25.000.000	378.787,88
ClickUp	877	100.000	185.000.000	212.643,68
Asana	2101	100.000	334.000.000	159.047,62

Tablica 11: Analiza utjecaja broja zaposlenih

Izvor: Istraživanje autora, 2022.

Prvo treba naglasiti kako su *ClickUp* i *Asana* tvrtke s jednim proizvodom dok *37signals* ima odnedavno dva proizvoda te sve tvrtke pružaju dodatne usluge vezane uz proizvod. Sve tri kompanije imaju vlastitu SaaS platformu za upravljanje projektima, a *37signals* ima dodatno HEY Email platformu za upravljanje mailovima koja za sad ostvaruje, kako je već

rečeno u radu, 1.000.000\$ godišnje što kao takvo značajno ne utječe na rezultate ove usporedbe.

Dakle *37signals* ima 66 zaposlenih što je gotovo dvostruko više u odnosu na početak 2021. kada ih je imala tek 34. No ipak 66 je višestruko manje od *ClickUp*ovih 877, a pogotovo *Asana*ovih 2101 zaposlenog. Nadalje prihod koji tvrtke ostvaruju po zaposlenom zamjetno je veći u *37signalsu* naspram ova dva konkurenta. *37signals* prednjači sa iznosom od 378.787,88\$ po zaposlenom, zatim slijedi *ClickUp* s gotovo dvostruko manjim iznosom od 212.643,68\$ po zaposlenom i zadnja slijedi *Asana* s 159.047,62\$.

Logično se nameće i pitanje kako to da *ClickUp* i *Asana* ostvaruju višestruko više ukupne prihode u odnosu na *37signals*. Odgovor leži u tome da te dvije tvrtke naplaćuju korištenje njihovog proizvoda po pojedinačnom korisniku unutar tvrtke korisnika, a *Basecamp* je dostupan po cijeni od 99\$ mjesečno za neograničen broj korisnika. Nadalje *ClickUp* i *Asana* pružaju programerske usluge nadogradnje dodatnih funkcionalnosti u svrhu što veće personalizacije proizvoda potrebama tvrtke korisnika što se dodatno naplaćuje. U *Basecampu* ne rade takve stvari zbog filozofije o jednostavnosti već spomenute u ovom radu.

S obzirom da je prethodno u ovom radu utvrđeno kako nema značajne razlike u kvaliteti kada je riječ o *Basecampu*, *ClickUpu* i *Asanai* može se zaključiti kako manji broj zaposlenika ne utječe negativno na kvalitetu proizvoda nastalih *37signalsovim* modelom upravljanja razvojem softverskih proizvoda.

6.4. Rezultati istraživanja

Provedbom ovog istraživanja dobiven je odgovor na dva od tri istraživačka pitanja. „Kakva je percepcija kvalitete korisnika *37signals* proizvoda u usporedbi s konkurentnim proizvodima slične namjene?“ i „Utječe li malen broj zaposlenika tvrtke na efektivnost procesa upravljanja razvojem softverskog proizvoda?“

Kao što je već rečeno tijekom samog provođenja strukturiranog intervjua primijećeno je ispitanici imaju generalno pozitivno mišljenje o proizvodima tvrtke *37signals*. Rezultati

analize podataka potvrdili su ta zapažanja te se iz toga može zaključiti kako korisnici imaju pozitivnu percepciju kvalitete *37signalsovih* proizvoda.

U nastavku istraživanja proizvodi *Ruby on Rails* i *Basecamp* uspoređeni su s konkurentskim proizvodima iste kategorije. Rezultati tog dijela istraživanja pokazali su kako je *Ruby on Rails* u blagoj prednosti nad *Djangom* i značajnoj prednosti nad *CakePHPom* kada je riječ o kvaliteti. *Basecamp* pak u jakoj konkurenciji jako dobro kotira i iako rezultati ovog istraživanja pokazuju neznatan zaostatak *Asanae* za *Basecampom* i *ClickUpom* može se ustvrditi da su sa tri proizvoda iznimno kvalitetna.

Na pitanje kako utječe li malen broj zaposlenika tvrtke na efektivnost procesa upravljanja razvojem softverskog proizvoda dobiven je vrlo jasan odgovor. Može se zaključiti kako manji broj zaposlenika ne utječe negativno na kvalitetu proizvoda nastalih *37signalsovim* modelom upravljanja razvojem softverskih proizvoda.

7. ZAKLJUČAK

Tržište softvera iz godine u godinu doživljava kontinuirani rast. Na to upućuju statistički podaci svih istraživanja pregledanih u ovom radu. Gledano na tržište poslovnih softvera, nakon razdoblja blagog rasta od 2010. do 2016. godine počelo je razdoblje povećanog rasta potrošnje sredstava utrošenih u kupnju softvera. Primjerice samo je s 529 milijardi američkih dolara u 2020. godini količina utrošenih sredstava skočila na 615 milijardi američkih dolara u 2021. godini. Također procjenjuje se nastavak trenda rasta te se za 2023. godinu predviđa potrošnja od 755 milijardi američkih dolara. Nadalje, broj programera softvera kontinuirano raste. 2013. godine ta je brojka iznosila 18,2 milijuna osoba, a krajem 2021. godine taj broj je narastao na 26,8 milijuna. Premda broj angažiranih programera kontinuirano raste, potražnja za istima također raste. Razlog tome leži u prethodno navedenim podacima o povećanom rastu potrošnje sredstava utrošenih u kupnju softvera. Svi ovi podaci ukazuju na to da je tržište softvera tržište na kojem vlada rastuća ponuda, ali i još veća rastuća potražnja. Kao rezultat toga dolazi i do rasta cijena programerskih usluga i cijena samih softvera. Uzrok tome dijelom leži i u činjenici da se kontinuirano razvijaju nove tehnologije. Dvije najznačajnije nove tehnologije koje se danas koriste u razvoju softvera su tehnologija računalnog oblaka i *blockchain* tehnologija. Njihov razvoj otvara nove mogućnosti na tržištu softvera počevši od razvoja potpuno novih proizvoda, ali i unapređenja postojećih proizvoda. Naravno razvoj novih proizvoda sa sobom donosi novu potražnju kao i unaprjeđenje postojećih proizvoda što posljedično rezultira povećanjem ukupne potražnje za programerima.

Kvaliteta igra važnu ulogu kada je riječ o softverskom proizvodu. Softverski proizvod ima više karakteristika usluge nego klasičnog proizvoda stoga i njegova kvaliteta može biti promatrana kao kvaliteta usluge. Kvaliteta usluge poseban je izazov, obzirom da se uvijek teži što višoj kvaliteti uz što niži trošak kako bi bila postignuta konkurentna cijena. Iako je kvalitetu zbog svoje apstraktnosti, višedimenzionosti i promjenjivosti u ovom slučaju teško definirati, lako je ustvrditi kada se radi o nezadovoljavajućoj kvaliteti. Ona za posljedicu ima promjenu pružatelja usluga od strane korisnika zbog određenog nezadovoljstva. Ipak isti proizvod ne mora biti percipiran jednako od strane različitih kupaca. Ono što jedna osoba percipira kvalitetnim druga osoba može doživjeti prosječnim ili ispodprosječnim. Tada je riječ o percipiranoj kvaliteti u svijesti korisnika koja je iznimno važna za softverski proizvod koji konkurira na tržištu. Važnost percipirane kvalitete softverskog proizvoda leži u sklonosti

korisnika da sud o kupnji često donose na temelju tuđe percepcije i ocjene softverskog proizvoda. Iz percipirane kvalitete proizlazi zadovoljstvo korisnika, odnosno njegovo nezadovoljstvo. Percepcija o proizvodu kao visoko kvalitetnom izvor je lojalnosti korisnika. Može se zaključiti kako percipirana kvaliteta ima pozitivan utjecaj na zadovoljstvo, a posljedično zadovoljstvo rezultira stvaranjem lojalnih odnosa. Da bi softverski proizvod bio percipiran kao kvalitetan on mora pružiti određenu razinu kvalitete uz zadovoljavanje specifičnih potreba korisnika. Nadalje, kako bi softverski proizvod u praksi ostvario zadovoljavajuću razinu kvalitete te zadovoljio sve potrebe korisnika tvrtka proizvođač mora imati kvalitetan model upravljanja razvojem softverskog proizvoda.

37signalsov model upravljanja razvojem softverskih proizvoda, kako ga oni nazivaju *Getting Real* model, s pripadajućim pristupom, filozofijama i načelima definiran je u ovom radu čime je dan odgovor na prvo istraživačko pitanje. Važnost ovoga ogleđa se u tome da postoji malo radova na ovu temu, a i oni koji su napisani ne razrađuju *Getting Real* pristup i model dovoljno široko, bar ne u literaturi pregledanoj u svrhu ovog istraživanja. U ovom radu definiran je *Getting Real* pristup kroz šest načela, kao i u ostalim istraživanjima, a i samoj knjizi *Getting real: the smarter, faster, easier way* (2006.). Dodatno u ovom radu predstavljen je i *Getting Real* model upravljanja razvojem softverskih proizvoda koji je također jasno definiran u originalnoj knjizi, ali ga ne nazivaju modelom već samo procesom od ideje do implementacije. Također u radu su predstavljene dvije dodatne faze modela koje su logične, ali nisu stavljene zajedno s ostatkom modela u originalnoj knjizi. Nadalje kao dodatak modelu opisano je i načelo skraćivanja na pola koje je specifično za *Getting Real* model upravljanja razvojem softverskih proizvoda.

Nadalje dobiveni su jasni rezultati provedbom primarnog istraživanja. Analizom podataka dobivenih provedbom strukturiranog intervjua utvrđeno je da korisnici percipiraju *37signalsove* proizvode kao kvalitetne. To je vidljivo iz komentara sudionika koji su za bili pozitivni za *Ruby on Rails* i *Basecamp* uz neke manje zamjerke kao iznimku. Ono što također govori puno o percipiranoj kvaliteti je način ponašanja sudionika intervjua pri postavljanju pitanja. Svi sudionici bili su opušteni, imali pozitivan govor tijela i ton glasa tijekom cijelog razgovora. To se ne može kvantificirati, ali daje do znanja da je izraženo mišljenje sudionika o proizvodima pozitivno i iskreno.

Usporedba *37signalsovih* proizvoda s konkurentskim proizvodima iste kategorije također je pokazala dobre rezultate za tvrtku i njen model upravljanja razvojem softverskih proizvoda.

Ruby on Rails nadmašio je konkurenciju. Za *Django* radni programski okvir može se reći da je na istoj razini kvalitete kao i *Ruby on Rails* dok je *CakePHP* ipak jednu skalu niže kvalitete. *Basecamp* se definitivno pokazao kao proizvod iznimne kvalitete u izričito jakoj konkurenciji. Zanimljiv je komentar jednog od sudionika intervjua koji se požalio kako *Basecamp* ima previše svojstava, misleći na broj značajki. Komentar se ističe zato što *Basecamp* ima zamjetno manji broj značajki od oba konkurenta kako bi proizvod u cijelosti ostao što jednostavniji.

Također u ovom radu utvrđeno je kako iznimno malen broj zaposlenih, s obzirom na posao koji se obavlja u *37signalu* i rezultate koje postižu njihovi proizvodi, sigurno ne utječe negativno na kvalitetu njihovih proizvoda. Uzrok tome može se potražiti možda u motiviranosti zaposlenika čemu u prilog ide minimalna godišnja plaća od 70.000\$ i činjenica da tvrtka tijekom ljetnog kvartala posluje po principu četverodnevnog radnog tjedna već više od desetljeća. No to je ipak tema za neko novo i daljnje istraživanje.

37signalsov model upravljanja razvojem softverskog proizvoda prema nalazima ovog rada pokazao se uspješnim, korisnim i pouzdanim. Ipak treba uzeti u obzir činjenicu da su ovo samo rezultati istraživanja dobiveni iz dostupnih sekundarnih izvora te iz provedenog strukturiranog intervjua s korisnicima proizvoda razvijenih korištenjem tog modela dok praktičan primjer korištenja *Getting Real* pristupa i modela te svi njihovi detalji nisu javno dostupni. O kvaliteti tvrtkinog modela govori kvaliteta njihovih proizvoda koja je, kako je i više puta spomenuto u ovom radu, na iznimno visokoj razini. Visoka kvaliteta proizvoda temelj je zadovoljstva korisnika. Zadovoljni korisnici postaju lojalni, a tvrtka *37signals* ih ima na pretek, što je vidljivo na primjeru puštanja u rad *HEY Email* platforme koja je u prvih tjedan dana dosegla 100.000 korisnika od kojih je polovica bila na listi čekanja već prije puštanja platforme u rad. Naravno tu se radi o većinski već postojećim korisnicima drugih *37signalsovih* proizvoda. Sve to rezultat je percepcije o kvaliteti tvrtkinih proizvoda u svijesti korisnika koja je, kako rezultati istraživanja u ovom radu pokazuju, iznimno pozitivna. Uzročno posljedično može se donijeti zaključak kako *37signalsov* model upravljanja razvojem softverskog proizvoda ima pozitivan utjecaj na percepciju o kvaliteti *37signalsovih* proizvoda u svijesti korisnika.

Svakako za kraj, iz provedenog istraživanja za potrebe ovog rada, može se zaključiti kako je *37signals* jedna zanimljiva i nadasve fascinantna tvrtka, a njihov *Getting Real* pristup i

model upravljanja razvojem softverskih proizvoda skupa s cijelom filozofijom poslovanja nešto posebno.

LITERATURA

Literatura

- [1] Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile Software Development Methods. Review And Analysis. VTT Publications 478. Espoo: Otamedia Oy.
- [2] Awad, M. A. (2005), *A comparison between agile and traditional software development methodologies*, University of Western Australia
- [3] Attaran, M., Woods, J. (2018), Cloud computing technology: improving small business performance using the Internet, *Journal of Small Business & Entrepreneurship*
- [4] Bassil, Y. (2012), *A simulation model for the waterfall software development life cycle*, *International Journal of Engineering & Technology*
- [5] Bevanda V., Sinković G. (2009), *Sustavi Znanja U Potpori Upravljanju Kvalitetom Softverskog Proizvoda*, Sveučilište Jurja Dobrile u Puli, Pula
- [6] Buntak K. et al. (2021), *Kvaliteta i sustavi upravljanja kvalitetom*, Hrvatska gospodarska komora, Zagreb
- [7] Fried, J., Heinemeier Hansson, D. (2010), *Rework*, Random House, Inc., New York
- [8] Fried, J., Heinemeier Hansson, D., Linderman, M. (2006), *Getting Real: The Smarter, Faster, Easier Way to Build a Successful Web Application*, *Basecamp* e-book
- [9] Garača, N. (2010), *Osnove Ekonomije 3. Ponuda i Potražnja*, Visoka Škola za Menadžment u Turizmu i Informatici u Virovitici, Virovitica
- [10] Gorchels L. (2000), *Manager's Handbook, The Complete Product Management Resource*, NTC Business Books, Illinois
- [11] Gronroos C. (1987), *Strategic Management and Marketing in the Service Sector*, Chartwell-Bratt
- [12] Lazibat, T., Baković, T., Štulec, I., Damić, M., Dužević, I., Buntić, L. (2020), *Međunarodno poslovanje*, Ekonomski fakultet Zagreb
- [13] Luka Marić, (2020.), *Usporedba tradicionalnog i agilnog načina vođenja projekata u razvoju softverskih proizvoda*, Sveučilište u Zagrebu, Zagreb
- [14] Marinković V., Senić V. (2012), *Analiza elemenata kvaliteta usluga u korporativnom bankarstvu*, Ekonomski horizonti, Kragujevac

- [15] Meža, P., Šerić, N. (2009), The role of product management in the global crisis – case study in domestic appliance industry, *Proceedings of 3rd Marketing Theory challenge in transitional societies 24th-25th of September 2009*. Maribor, Slovenia, str. 189-196
- [16] Moggridge, B. (2007): *Designing Interactions*, MIT Press, Cambridge, Massachusetts
- [17] Molin-Juustila, T. (2001). Market Centered Model of the Early Phases of SW Product Development, *Proceedings of the 24th Information Systems Research Seminar in Scandinavia*, 1–14.
- [18] Ozretić Došen, Đ. (2010), *Osnove marketinga usluga*, Sveučilište u Zagrebu, Zagreb
- [19] Plekhanova, J. (2009), *The IBIT Report, Evaluating web development frameworks: Django, Ruby on Rails and CakePHP*, Temple University, Philadelphia
- [20] Puška A., Šadić S., Beganović I. A., (2016), *Utjecaj kvalitete na zadovoljstvo i lojalnost studenata*, Univerzitet za poslovne studije Banja Luka, Banja Luka
- [21] Puška A., Maksimović A., Fazlić S., (2015), *Utjecaj Percipirane Kvalitete na Zadovoljstvo i Lojalnost Klijenata Banaka*, Visoka škola eMPIRICA, Brčko
- [22] Rocco S., Selinšek A. (2020), *Managing Design: the Impact of Design Orientation on Market Success*, Zagreb School of Business, Faculty of Economics and Business
- [23] Ruparelia, N. B. (2010). Software development lifecycle models. *ACM SIGSOFT Software Engineering Notes*
- [24] Sajra J., (2011.), *A Scientific Review of the “Getting Real” Software Development Approach*, Lapland University of Applied Sciences
- [25] Stankić R. (2015), *Trendovi u Industriji Softvera*, Zbornik radova Ekonomskog fakulteta Brčko, Brčko
- [26] Strode, D. (2006). Agile methods: A comparative analysis, *Proceedings of the 19th Annual Conference of the National Advisory Committee on Computing Qualifications*,
- [27] Šerić, N. (2016), *Upravljanje proizvodom*, Redak Split
- [28] Šerić, N. Luetić, A. (2016), *Suvremena logistika*, Redak Split
- [29] Šerić, N. Žilić, I., Kalinić, S. (2011), The testing of strategic model of positioning of a new product from the country in transition on the global market, *Proceedings of REDETE: Researching Economic Development and Entrepreneurship in Transition Economies*, Faculty of Economics, University of Banja Luka, Bosnia and Herzegovina, 27/29 October 2011. str.322-327

- [30] Ulrich, K.T., Eppinger, S.D. (2008) Product Design and Development. 4th Edition, McGraw-Hill, New York.

Web izvori:

- [31] 37signals, (2022) 37signals Official Website [Internet], raspoloživo na: <https://37signals.com/> [20.5.2022]
- [32] Aglowid (2021), Ruby on Rails vs Django Comparasion – Check Key Difference in 2022, [Internet], raspoloživo na: <https://aglowiditsolutions.com/blog/ruby-on-rails-vs-django/> [14.5.2022]
- [33] Airship (2022): [Internet], raspoloživo na: What Is An MVP In Software Development? <https://blog.teamairship.com/what-is-mvp-in-software-development> [26.8.2022]
- [34] Asana (2022), Asana Official Website [Internet], raspoloživo na: <https://asana.com/> [12.5.2022]
- [35] Atchinson, L. (2022 Why blockchain is the future of the Internet [Internet], raspoloživo na: <https://www.infoworld.com/article/3657635/why-blockchain-is-the-future-of-the-internet.html> [12.5.2022]
- [36] Basecamp (2022), Basecamp Official Website [Internet], raspoloživo na: <https://basecamp.com/> [20.5.2022]
- [37] Beck K., et. al. (2001), Proglas o metodi agilnog razvoja softvera [Internet], raspoloživo na: <http://agilemanifesto.org/iso/hr/manifesto.html> [14.6.2022]
- [38] Brush K. (2021): Chatbot [Internet], raspoloživo na: <https://www.techtarget.com/searchcustomerexperience/definition/chatbot> [23.8.2022]
- [39] Business Insider (2022), Basecamp CEO Jason Fried isn't afraid to write his own rules on leadership. He explains why political talk is banned at work and why he implemented a 4-day workweek in the summer [Internet], raspoloživo na: <https://www.businessinsider.com/basecamp-jason-fried-political-talk-summer-four-day-workweek-2022-4> [23.5.2022]

- [40] Busse M., (2022), Getting Real: 37signals on Web App Development Processes, [Internet], raspoloživo na: <https://industrialbrand.com/getting-real-37signals> [10.7.2022]
- [41] CakePHP (2022), CakePHP Official Website [Internet], raspoloživo na: <https://cakephp.org/> [16.6.2022]
- [42] ClickUp (2022), ClickUp Official Website [Internet], raspoloživo na: <https://clickup.com/> [12.6.2022]
- [43] Developer Mozilla (2022): MVC [Internet], raspoloživo na: <https://developer.mozilla.org/en-US/docs/Glossary/MVC> [28.8.2022]
- [44] Django (2022), Django Official Website [Internet], raspoloživo na: <https://www.djangoproject.com/> [8.5.2022]
- [45] GitHub (2022), GitHub Official Website [Internet], raspoloživo na: <https://github.com/> [9.7.2022]
- [46] Grojwo, (2022.), Revenue and Competitors, [Internet], raspoloživo na: <https://growjo.com/company/ClickUp> [9.7.2022]
- [47] Hayes, A. (2022), What Is a Blockchain?, [Internet], raspoloživo na <https://www.investopedia.com/terms/b/blockchain.asp> [9.7.2022]
- [48] HEY Email, (2022) HEY Email Official Website [Internet], raspoloživo na: <https://www.hey.com/> [20.6.2022]
- [49] Hughes, O. (2022) Developers are in short supply. Here are the skills and programming languages employers need [Internet], raspoloživo na: [Developers are in short supply. Here are the skills and programming languages employers need | ZDNet](#) [20.6.2022]
- [50] ITenterprise (2022), Making Bespoke Software Development Great Again, [Internet], raspoloživo na: <https://itenterprise.co.uk/bespoke-software-development/> [11.6.2022]

- [51] JavaTpoint (2022.): Django MVT, [Internet], raspoloživo na: <https://www.javatpoint.com/django-mvt> [27.8.2022]
- [52] Khong W. L, et al, (2012), Software Development Life Cycle AGILE vs Traditional Approaches, [Internet], raspoloživo na: https://www.researchgate.net/publication/268334807_Software_Development_Life_Cycle_AGILE_vs_Traditional_Approaches [1.7.2022]
- [53] Kickstarz Side Hustle, (2022), How HEY email got 100k users in the first week, [Internet], raspoloživo na: <https://kickstartsidehustle.com/how-hey-email-got-100k-users-in-the-first-week/> [7.7.2022]
- [54] LinkedIn, (2022) LinkedIn Official Website [Internet], raspoloživo na: <https://www.linkedin.com/> [10.6. 2022]
- [55] McBain, J. (2017), The Channel Software Stack – A Comprehensive (And Critical) Look At The Future Of The Industry, [Internet], raspoloživo na: <https://www.forrester.com/blogs/the-channel-software-stack-a-comprehensive-and-critical-look-at-the-future-of-the-industry/> [11.7.2022]
- [56] Mehta V., et. al. (2021), Software Market View, 2020-2021, [Internet], raspoloživo na: <https://www.gartner.com/en/documents/4004846> [4.7.2022]
- [57] Mleczek, A. (2022), How many developers are there in the world in 2022?, [Internet], raspoloživo na: [How Many Developers Are There In The World In 2021? | Blog Future Processing \(future-processing.com\)](https://future-processing.com/blog/how-many-developers-are-there-in-the-world-in-2021/) [4.7.2022]
- [58] Nowak M., (2022), Why Ruby on Rails Is Still a Good Choice in 2022, [Internet], raspoloživo na: <https://www.monterail.com/blog/why-ruby-on-rails-development-ror-revolutionized-web> [10.7.2022]
- [59] Owler, (2022.), HEY Email statistics [Internet], raspoloživo na: <https://www.owler.com/company/heycom> [12.7.2022]
- [60] Pratt, M. K., Gillis, A. S. (2021), Blockchain for businesses: The ultimate enterprise guide, [Internet], raspoloživo na: <https://www.techtarget.com/searchcio/definition/blockchain> [4.7.2022]

- [61] RedHat (2018), Types of cloud computing, [Internet], raspoloživo na: <https://www.redhat.com/en/topics/cloud-computing/public-cloud-vs-private-cloud-and-hybrid-cloud> [4.7.2022]
- [62] Shah H., (2020.), How an Anti-Growth Mentality Helped Basecamp Grow to Over 2 Million Customers [Internet], raspoloživo na: <https://nira.com/basecamp-history/> [24.6.2022]
- [63] Schwaber, K., & Sutherland, J. (2010). The Scrum Guide. [Internet], raspoloživo na: <http://www.scrum.org/scrumguides/> [14.6.2022]
- [64] Sentosh (2021), Django vs. Ruby on rails – Which one is the best?, [Internet], raspoloživo na: <https://sentosh.com/django-vs-ruby-on-rails/> [14.6.2022]
- [65] Shaji, M. (2020), The simple Billionaire Story : Basecamp, [Internet], raspoloživo na: <https://techstory.in/323390-2/> [14.7.2022]
- [66] SimilarTech (2022), Technologies Market Share Framework, [Internet], raspoloživo na: <https://www.similartech.com/categories/framework> [13.5.2022]
- [67] Soppa, S. (2022), Why Elixir & Phoenix is a great choice for your web app in 2022 [Internet], raspoloživo na: <https://curiosum.com/blog/why-elixir-phoenix-great-choice-for-modern-web-app> [23.8.2022]
- [68] StackOwerflow (2022.) Developer Survey, [Internet], raspoloživo na: <https://survey.stackoverflow.co/2022/#top-paying-technologies-web-frameworks> [12.7.2022]
- [69] Statista (2022), Software [Internet], raspoloživo na: <https://www.statista.com/outlook/tmo/software/worldwide> [17.5.2022]
- [70] Ruby on Rails (2022), Ruby on Rails Official Website [Internet], raspoloživo na: <https://rubyonrails.org/> [8.5.2022]
- [71] The Busines Research Company (2022), The Busines Research Company, Information Technology Global Market report 2022, [Internet], raspoloživo na:

<https://www.thebusinessresearchcompany.com/report-preview1.aspx?Rid=information%20technology%20global%20market%20report>
[10.5.2022]

- [72] Tiboe (2022), TIOBE Index for May 2022, [Internet], raspoloživo na: <https://www.tiobe.com/tiobe-index/> [15.5.2022]
- [73] Wykowski, T., Wykowska, J. (2018). Lessons learned: Using Scrum in nontechnical teams. Agile Alliance [Internet], raspoloživo na: <https://www.agilealliance.org/resources/experience-reports/lessons-learned-usingscrum-in-non-technical-teams/> [14.6.2022]

Popis slika

Slika 1: Potrošnja na poslovni softver u cijelom svijetu, od 2009 to 2023, Izvor Statista 2022.

Slika 2: Najveće softverske kompanije, Izvor: <https://companiesmarketcap.com/software/largest-software-companies-by-market-cap/>

Slika 3: Generički proces razvoja proizvoda Izvor: Molin-Juustila, 2001.

Slika 4: Tradicionalni pristup upravljanjem razvoja softverskog proizvoda Izvor: Marić L. 2020.

Slika 5. Model vodopada Izvor: Molin-Juustila, 2001.

Popis tablica

Ruby on Rails

Tablica 1: Razvoj korisničkog sučelja, Izvor: Istraživanje autora, 2022.

Tablica 2: Održavanje, Izvor: Istraživanje autora, 2022.

Tablica 3: Upravljanje podacima i migracija, Izvor: Istraživanje autora, 2022.

Tablica 4: Mogućnost testiranja, Izvor: Istraživanje autora, 2022.

Tablica 5: Popularnost, Izvor: Istraživanje autora, 2022.

Tablica 6: Zajednica i razvijenost, Izvor: Istraživanje autora, 2022.

Tablica 7: Tržišni potencijal, Izvor: Istraživanje autora, 2022.

Tablica 8: Ukupna ocjena, Izvor: Istraživanje autora, 2022.

Tablica 9: Usporedba Basecampa, Asanae i ClickUpa, Izvor: Istraživanje autora, 2022.

Tablica 10: Ocjena Basecampa, Asanae i ClickUpa, Izvor: Istraživanje autora, 2022.

SAŽETAK

Softverski proizvodi danas se razvijaju brže nego ikada te imaju značajan utjecaj na život svakog pojedinca. Ipak, u današnjem poslovnom svijetu, kvaliteta proizvoda i brzina isporuke od iznimne su važnosti za ostvarivanje boljih poslovnih rezultata. Proces upravljanja razvojem softverskog proizvoda osigurava uštedu vremena potrebnog za isporuku i visoku kvalitetu samog softverskog proizvoda. Softverski proizvodi nekada su se razvijali vrlo slično kao i svaki drugi proizvod kroz tradicionalan pristup no zbog spomenute potrebe za bržim razvojem uz što višu razinu kvalitete, nastao je agilni pristup koji za cilj ima upravo zadovoljavanje tih potreba. Nadalje, agilni pristup stavlja naglasak i na fleksibilnost te zadovoljavanje potreba korisnika kako bi percipirana kvaliteta u svijesti istih bila što pozitivnija. Temeljen na agilnom pristupu nastao je i *37signalsov* model upravljanja razvojem softverskog proizvoda ili kako ga oni sami nazivaju *Getting Real*. Korištenjem tog modela tvrtka je razvila proizvode kao što su *Ruby on Rails* programski radni okvir, *Basecamp* platforma za upravljanjem proizvoda i *HEY Email* platforma koje odlikuje iznimna kvaliteta. Uspješni poslovni rezultati ostvareni korištenjem *Getting real* modela ukazuju na potrebu za daljnjim istraživanjem tog modela, što je razlog zašto je jedno takvo provedeno u ovom radu.

Ključne riječi: razvoj, softver, proizvod, model, upravljanje

SUMMARY

Today, software products are developing faster than ever and have a significant impact on the life of every individual. However, in today's business world, product quality and deployment speed are extremely important for achieving better business results. The software product development management process ensures the saving of time required for deployment and high quality of the software product itself. Software products used to be developed similarly to any other product through a traditional approach, but due to the aforementioned need for faster deployment with the highest possible level of quality, an agile approach was created that aims to meet these needs. Furthermore, the agile approach emphasizes flexibility and meeting the needs of users so that the perceived quality is as positive as possible. Based on an agile approach, 37signals' software product development management model, or as they call it, *Getting Real* was created. Using this model the company has developed products such as the *Ruby on Rails* programming framework, the *Basecamp* product management platform, and the *HEY Email* platform which are characterized by exceptional quality. Successful business results achieved using the *Getting real* model indicate the need for further research into that model, which is the reason why one such research was implemented in this paper.

Keywords: development, software, product, model, management