

# SUVREMENE METODE U FORMULIRANJU I VOĐENJU INFORMATIČKIH PROJEKATA

---

**Blazinić, Matea**

**Master's thesis / Diplomski rad**

**2021**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **Josip Juraj Strossmayer University of Osijek, Faculty of Economics in Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Ekonomski fakultet u Osijeku**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:145:862762>

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

*Download date / Datum preuzimanja:* **2025-02-02**



*Repository / Repozitorij:*

[EFOS REPOSITORY - Repository of the Faculty of Economics in Osijek](#)



Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Diplomski studij, Poslovna informatika

Matea Blazinić

**SUVREMENE METODE U FORMULIRANJU I VOĐENJU  
INFORMATIČKIH PROJEKATA**

Diplomski rad

Osijek, 2021.

Sveučilište Josipa Jurja Strossmayera u Osijeku

Ekonomski fakultet u Osijeku

Diplomski studij, Poslovna informatika

Matea Blazinić

**SUVREMENE METODE U FORMULIRANJU I VOĐENJU  
INFORMATIČKIH PROJEKATA**

Diplomski rad

**Kolegij: Upravljanje projektima**

JMBAG: 0010218823

e-mail: mblazinic@efos.hr

Mentor: prof. dr. sc. Josip Mesarić

Osijek, 2021.

Josip Juraj Strossmayer University of Osijek

Faculty of Economics in Osijek

Graduate Study, Business informatics


Matea Blazinić

**MODERN METHODS IN FORMULATING AND MANAGING  
IT PROJECTS**

Graduate paper

Osijek, 2021.

**IZJAVA**  
**O AKADEMSKOJ ČESTITOSTI,**  
**PRAVU PRIJENOSA INTELEKTUALNOG VLASNIŠTVA,**  
**SUGLASNOSTI ZA OBJAVU U INSTITUCIJSKIM REPOZITORIJIMA**  
**I ISTOVJETNOSTI DIGITALNE I TISKANE VERZIJE RADA**

1. Kojom izjavljujem i svojim potpisom potvrđujem da je \_\_\_\_\_ diplomski (navesti vrstu rada: završni / diplomski / specijalistički / doktorski) rad isključivo rezultat osobnoga rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu. Potvrđujem poštivanje nepovredivosti autorstva te točno citiranje radova drugih autora i referiranje na njih.
2. Kojom izjavljujem da je Ekonomski fakultet u Osijeku, bez naknade u vremenski i teritorijalno neograničenom opsegu, nositelj svih prava intelektualnoga vlasništva u odnosu na navedeni rad pod licencom *Creative Commons Imenovanje – Nekomercijalno – Dijeli pod istim uvjetima 3.0 Hrvatska*. 
3. Kojom izjavljujem da sam suglasan/suglasna da se trajno pohrani i objavi moj rad u institucijskom digitalnom repozitoriju Ekonomskoga fakulteta u Osijeku, repozitoriju Sveučilišta Josipa Jurja Strossmayera u Osijeku te javno dostupnom repozitoriju Nacionalne i sveučilišne knjižnice u Zagrebu (u skladu s odredbama Zakona o znanstvenoj djelatnosti i visokom obrazovanju, NN br. 123/03, 198/03, 105/04, 174/04, 02/07, 46/07, 45/09, 63/11, 94/13, 139/13, 101/14, 60/15).
4. izjavljujem da sam autor/autorica predanog rada i da je sadržaj predane elektroničke datoteke u potpunosti istovjetan sa dovršenom tiskanom verzijom rada predanom u svrhu obrane istog.

**Ime i prezime studenta/studentice:** Matea Blazinić

**JMBAG:** 0010218823

**OIB:** 87522408565

**e-mail za kontakt:** blazinic.matea@gmail.com

**Naziv studija:** Diplomski sveučilišni studij poslovna ekonomija, smjer Poslovna informatika

**Naslov rada:** Suvremene metode u formuliranju i vođenju informatičkih projekata

**Mentor/mentorica diplomskog rada:** prof. dr. sc. Josip Mesarić

U Osijeku, 30. rujna 2021. godine

Potpis Matea Blazinić

## **Suvremene metode u formuliranju i vođenju informatičkih projekata**

### **SAŽETAK**

Projekti i njihovo kvalitetno upravljanje i vođenje dobivaju na značaju u poslovnim organizacijama, neovisno o veličini ili grani industrije. Zbog naglih i brzih tehnoloških promjena u svijetu, organizacije su se morale prilagoditi i biti u koraku s promjenama, a posljedica toga je kompleksnost poslovnih procesa. Tehnološki razvitak donio je i veći broj softverskih rješenja, pa se javlja i veća potreba za kvalitetnijem i efikasnijem upravljanjem. S obzirom na potrebu za informatičkim projektima javljaju se i razne metodologije kojima se definiraju procesi, faze i pravila kako pristupiti vođenju projekata koji u konačnici donose novi kvalitetan proizvod. Javljaju se dva najpopularnija načina upravljanja projektima, a to su tradicionalni i agilni načini upravljanja. Tradicionalni oblici i načini provođenja više nisu mogli pratiti nove i suvremene koncepte upravljanja stoga su razvijene sve popularnije suvremene odnosno agilne metode provođenja projekata. U svijetu softverskih proizvoda najčešće su korištene agilne metode jer su u odnosu na tradicionalne fleksibilnije, što rezultira većom uspješnošću projekata i zadovoljstvom svih dionika. Rad ima za cilj sistematizirati ključne značajke tradicionalnih metoda u odnosu na agilne i uočiti glavne razlike pri odabiru prave metode kod provođenja informatičkih projekata.

**Ključne riječi:** upravljanje projektima, tradicionalne metode, agilne metode, vodopadni model, SCRUM

## **Modern methods in formulating and managing IT projects**

### **ABSTRACT**

Projects and the quality of their management and leadership are gaining in importance in business organizations, regardless of the size or branch of industry. Due to the sudden and rapid technological changes in the world, organizations have had to adapt and keep up with the changes, the consequence of which is the complexity of business processes. Technological development has brought a larger number of software solutions, hence the greater need for better and more efficient management. Given the need for IT projects, there are various methodologies that define the processes, phases, and rules of how to approach project management that ultimately bring a new quality product. There are two most popular ways to manage projects, and these are the traditional and agile ways of managing. Traditional forms and methods of management could no longer follow new and modern management concepts, therefore increasingly popular modern and agile methods of project implementation were developed. In the world of software products, agile methods are most often used, because they are more flexible compared to traditional ones, which results in greater project success and satisfaction of all stakeholders. This paper aims to systematize the key features of traditional methods in relation to the agile ones and to notice the main differences in choosing the right method in the implementation of IT projects.

**Keywords:** project management, traditional methods, agile methods, waterfall model, SCRUM

# SADRŽAJ

1. Uvod.....	1
2. Metode istraživanja i izvori podataka .....	2
3. Definiranje projekta i projektnog menadžmenta.....	3
3.1. Projekt i njegove odrednice .....	4
3.2. Projektni menadžment .....	6
4. Metodologije pri izradi softverskih rješenja .....	8
4.1. Tradicionalni pristupi .....	8
4.1.1. Faze procesa upravljanja projektima.....	10
4.1.2. Upravljanje projektima kritičnog lanca.....	17
4.1.3. Metodologija lanca događaja .....	20
4.1.4. PRINCE2.....	23
4.1.5. Vodopadna metodologija .....	25
4.2. Agilne metodologije .....	28
4.2.1. SCRUM.....	30
4.2.2. Model brzog razvoja aplikacija .....	34
4.2.3. Kanban .....	36
4.2.4. Ekstremno programiranje .....	39
4.2.5. Crystal .....	43
5. Zaključak.....	48
LITERATURA.....	50
POPIS SLIKA .....	55



## 1. Uvod

Projekti i projektni menadžment, pojmovi su kojima se u današnje vrijeme sve više posvećuje pažnje. Kvalitetno vođenje projekata bilo koje vrste postaje sve značajnije u poduzećima u gotovo svim granama industrije, neovisno o veličinama odnosno, neovisno o tome radi li se o malom, srednjem ili velikom poduzeću. Nagle i intenzivne tehnološke promjene u svijetu odrazile su se i na poslovne organizacije, a posljedica toga je razvoj kompleksnih poslovnih procesa. Razvojne aktivnosti se provode kao zasebni ili integrirani projekti, pa se počela razvijati sve veća potreba za kvalitetnim upravljanjem. Može se reći kako je vođenje projekata i najvažnije zanimanje. Bez uspješnog i kvalitetnog vođenja nema ni kvalitetne realizacije posla i pozitivnih rezultata. Ono što je najvažnije kod projektnog upravljanja jesu: znanje, vještine, tehnike i pravila kojima se omogućava efikasno provođenje projekata. Što je provođenje projekta efikasnije to je uspješnija njegova isporuka u cjelini. Tehnološki razvitak, kako u svijetu tako i u poslovanju, donio je i veći broj informatičkih rješenja. Opseg investiranja u efikasna i kvalitetna informatička rješenja sve više raste iz godine u godinu, što zapravo povećava potrebu za kvalitetnim upravljanjem i informatičkim projektima. Projekti postaju kompleksniji i opsežniji pri čemu vođenje projekata poprima sve veću važnost u IT industriji. S obzirom na veliki tehnološki razvitak, suvremeni informatički projekti odnosno stvaranja softverskih rješenja za različite namjene, pripremaju se korištenjem novih projektnih pristupa kojima je cilj projektno rješenje isporučiti u najkraćem mogućem roku uz ispunjenje svih relevantnih korisničkih zahtjeva. Ubrzanom digitalizacijom, tehnološkim razvojem poslovanja poslovnih organizacija i kompleksnijim projektima, tradicionalni oblici i načini vođenja više nisu mogli pratiti nove i suvremene koncepte upravljanja, stoga su razvijene i sve su popularnije agilne metode upravljanja projektima, pogotovo kada je riječ o informatičkim rješenjima. Agilnost se u modernom vremenu primjenjuje u raznim dijelovima poslovanja, od samog menadžmenta preko projekt menadžmenta pa sve do vođenja timova. Agilne metode za cilj imaju bržu i učinkovitiju suradnju prilikom isporuke softverskih rješenja. Također, agilnim metodama svojstvena je iterativnost i kontinuirani *feedback* (povratne informacije) koji omogućava uspješniju nadogradnju i doradu te u konačnici isporuku softverskog rješenja, planiranje, testiranje, integracija i ostali oblici kontinuirane izrade projekta i samog softvera. Cilj rada je napraviti sistematski pregled s prednostima i nedostacima suvremenih metoda u oblikovanju i vođenju informatičkih projekata, a poglavito agilnih metoda koje se koriste u realizaciji navedenih projekata. Također će se istražiti sve njihove značajke kako bi se mogle

uočiti glavne razlike, prednosti i nedostaci kod odabira prave metode kod vođenja projekata, ali i pojasniti koje su to tradicionalne metode te njihove glavne značajke i razlike u odnosu na agilne metode.

## **2. Metode istraživanja i izvori podataka**

Rad ima karakter sistematskog istraživanja postojećih dostignuća u području upravljanja projektima odnosno informatičkim projektima pa će kao metode istraživanja koristiti analiza, sinteza te djelom klasifikacija uz upotrebu induktivnog i deduktivnog kao sistemskog pristupa. Kao izvori podataka koristit će se literatura o projektnom menadžmentu profesionalnih projektnih udruženja, raspoloživi izvori specifičnih (informatičkih) projekata, stručni i znanstveni radovi iz područja projektnog menadžmenta i podatci tvrtki koje se bave projektnim menadžmentom.

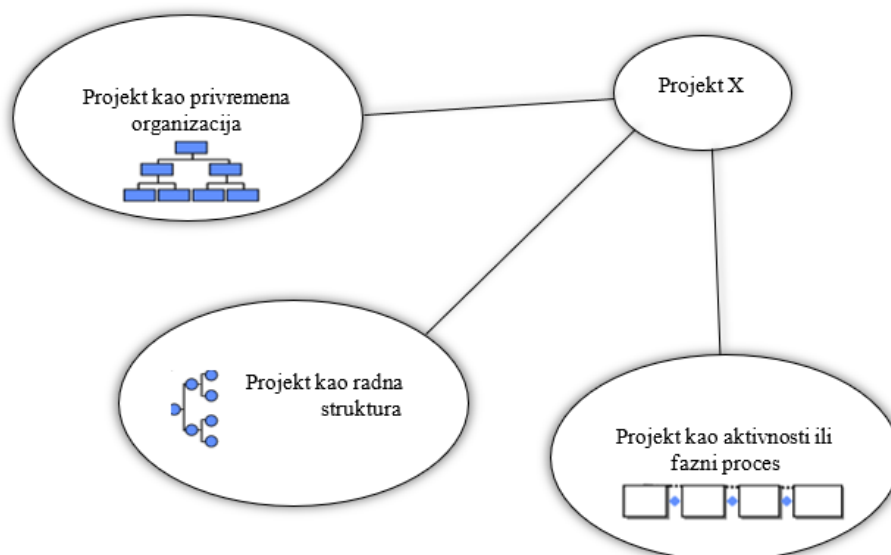
### 3. Definiranje projekta i projektnog menadžmenta

Povijest različitih postignuća koje imaju karakter onoga što danas nazivamo projekt je vrlo duga i ta su postignuća izvedena na različite načine i s više ili manje uspjeha. Krajem prošlog stoljeća počele su se formirati profesionalne udruge prvenstveno s ciljem da prikupe i sistematiziraju stečena znanja i iskustva o izvođenju i upravljanju različitim manje ili više složenim skupovima aktivnosti u kojima su angažirani različiti resursi (materijalni, ljudski, financijski...) namijenjenih realizaciji planiranih ciljeva odnosno postignuća. Upravljanje projektima kao područje odnosno skup općih i specifičnih znanja, vještina, metoda, modela i alata sistematiziranih kroz brojna iskustva u upravljanju projektima iz svih područja ljudske prakse, nastoji se standardizirati od strane profesionalnih udruženja za projektni menadžment kao što su PMI, IPMA, APM i dr. PMI (*Project Management Institute*) je najveća zasebna međunarodna udruga osnovana 2004. godine za upravljanje projektima, pružanje potpore i razne edukacije o projektnom menadžmentu. IAPM (*Internacional Association of Project Managers*) jest međunarodna organizacija, a APM (*Association Project Management*) je udruga iznimne važnosti u stvaranju i širenju znanja o upravljanju projektima sa sjedištem u Velikoj Britaniji. Nude razne aktivnosti i zanimljive programe konferencija te mogućnosti susreta, podrške i učenja od strane stručnjaka, a uz sve to akreditiraju se i organizacije za osposobljavanje i akademske institucije za osposobljavanje prema svom nastavnom planu i programu. IAPM se od ostalih udruženja razlikuje po tome što ono ne nudi izravnu obuku, već nudi *IAPM Network University*, a to je virtualna mreža na Facebook-u koja pomaže studentima kako bi se povezali i stupili u kontakt sa drugim studentima, dijelili informacije o svojim radovima i tezama te razgovarali o raznim pitanjima i temama vezanim za upravljanje projektima. Također, IAPM studente dovodi u kontakt sa iskusnim projekt menadžerima IAPM-a kako bi lakše pronašli željeno radno mjesto. Prema *Project Management Institute Zagreb Croatia (2021)*, ono što je zajedničko svim udruženjima su ciljevi: njegovati stručnost u upravljanju projektima; doprinijeti kakvoći i djelokrug struke; poticati odgovarajuće globalne promjene upravljanja projektima radi dobrobiti šire javnosti; osigurati prepoznatljiv forum za razmjenu ideja, praksi, aplikacija i rješenja te prepoznati i promicati osnove upravljanja projektima te unaprijediti saznanja o uspješnom upravljanju.

Kako bi se mogao definirati projekti menadžment, u nastavku će se definirati pojam projekta i njegove odrednice te razvoj i vođenje.

### 3.1. Projekt i njegove odrednice

Projekt (*engl. Project*), je prema Thackeray-u, „organizirani skup aktivnosti koji treba dovesti do željene promjene u određenom vremenskom razdoblju“ (Thackeray, 2002). Može se također reći kako je projekt „unaprijed planirani skup međusobno povezanih zadataka za koje postoji redoslijed (faze) izvršavanja unutar određenog razdoblja, troškova i drugih ograničenja koje projekt ima“ (Business Dictionary, navedeno u Jenjić, 2018). Projekti mogu uključivati jedan zadatak zajednički za nekoliko stranaka, privremenu organizaciju, ciljnu orijentiranost i definirani kontinuitet aktivnosti ili problem koji se planira riješiti. Važno je definirati završetak prije nego što projekt uopće započne, a potrebno ga je definirati tako da bude mjerljiv ili pokaziv, a dobro bi bilo i oboje. Također, jako je važno jasno definirati projekt iz tri razloga: planiranje svakog koraka, motiviranost tima i raspodjela odgovornosti (jasno razlučivanje vođenja i odgovornosti koja proizlazi iz uloga koje ima svaki dionik u projektu). Projekti i njihovi ishodi ovise o planiranom budžetu, veličini projektnog tima koji radi na projektu te očekivanim ishodima. Neke definicije projekta ističu jedinstveni zadatak s obzirom na zahtjeve odnosno opseg, vrijeme, trošak i kvalitetu, dok neke druge definicije naglašavaju kako, za razliku od ostalih zadataka, projekti imaju zasebno organiziranu i definiranu projektnu organizaciju. Postoji uistinu mnogo definicija projekta pa se projekt može promatrati iz mnogo različitih perspektiva, a slika 1 prikazuje 3 paralelne perspektive projekta.



Slika 1. Tri perspektive projekta

Izvor: Prevedeno i prilagođeno od strane autora ovog rada prema Artto, K. A., & Wikström, K. (2005). What is project business?. *International Journal of Project Management*

Privremena organizacija se odnosi na projektnu organizaciju koja je osnovana isključivo i jedino u svrhu izvršenja određenog zadatka, a raspušta se kada je zadatak dovršen. Stoga, prema toj perspektivi za svaki pojedini zadatak formira se tim ljudi odnosno projektni tim koji je najprikladniji za obavljanje tog posla. Struktura proizvoda ili rada odnosi se na projekt koji se može vidjeti kroz proizvod ili dovršen posao kao rezultat projekta. Finalni proizvod može se podijeliti i na podprodukte tako što se formira struktura proizvoda. Također, posao se može i hijerarhijski podijeliti na opis manjih i jednostavnijih jedinica za upravljanje, a to je tzv. struktura raščlambe rada koja se odnosi na upravljanje opsegom projekta. Aktivnosti kroz faze se odnose na projekt koji ima ovisnosti vezane s aktivnostima i određenim fazama, redosljed između kojih se omogućuje upravljanje rasporedom projekta. Životni ciklus projekta (koji će biti detaljnije prikazan pod točkom 3.1.1.) pruža faznu perspektivu projekta, raspored i raspodjelu resursa te naglašava karakter projekta uz planirane aktivnosti i njihove međuovisnosti.

Projekti često imaju unaprijed definirani cilj, a on se odnosi na svrhu zbog koje je projekt i izvorno utemeljen. Prema Artto i Wikström (2005), cilj projekta je željeno buduće stanje koje se očekuje da se postigne završetkom projekta. Cilj projekta je promijeniti status quo. Također, definicija cilja projekta uključuje i tumačenje poslovno orijentiranog ili strateškog cilja postavljenog projekta. Ono se može opisati na nekoj općenitoj razini ili pak kao misija, a interpretacija se izoštrava tek kada se pomno definiraju zahtjevi za rezultirajućim proizvodom, radom, resursima, troškovima i vremenom. Kako je i ranije navedeno, važna karakteristika projekta je jedinstvenost strukture aktivnosti koje obuhvaća, a ni jedan projekt nije identičan "sestrinski" projekt. Jedinstvenost uzrokuju mnogi čimbenici kao što su cilj i rezultirajući proizvod; način na koji se projekt realizira ili sama organizacija projekta koja je različita od ranijih projekata; sličan projekt nije realiziran u istim okolnostima; finalni proizvod koji proizlazi iz projekta je izrađen prema specifičnim zahtjevima kupaca/klijenta te se za svaki projekt koriste novi dobavljači ili se dionici razlikuju od prethodnih projekata. S obzirom na definicije projekta može se zaključiti kako za sve projekte uvijek vrijede zajedničke odrednice, a one su sljedeće (Prevedeno od strane autora prema Artto i Wikström, 2005):

- Projekt je *složen* jer njegove aktivnosti obično nisu predvidive niti se mogu ponoviti. Ponavljanje jednostavnih aktivnosti karakteristično je za serijsku proizvodnju, ali složene aktivnosti, kao što se događa u projektom okruženju, često su nove, a za

njihovo učinkovito provođenje potrebne su posebne vještine, kreativnost, stalna evaluacija i jasno donošenje odluka.

- Projekt ***formiran iz međusobno povezanih aktivnosti*** zahtijeva da se projektne aktivnosti provode određenim redoslijedom. Aktivnost se može definirati radom koji čini aktivnost. Aktivnosti mogu biti poredane u logički slijed na temelju odnosa prvenstva i sukcesije. Uz to, ovisnosti i odnosi među aktivnostima čine bitan dio složenosti projekta.
- Projekt je ***vremenski ograničen*** (određenim vremenom), jer ima unaprijed definirani raspored, koji uključuje početnu i završnu točku u kojoj proizvod koji proizlazi iz projekta mora biti spreman i dostupan za upotrebu od strane kupaca.
- ***Trošak projekta uvijek je ograničen*** (u okviru proračuna), što znači da u bilo kojem projektu postoji ograničena mogućnost trošenja resursa poput rada, materijalnih resursa, vremena i novca; projekt mora biti dovršen u okviru unaprijed definiranog proračuna.
- ***Zahtjev prema specifikaciji*** ukazuje da proizvod proizveden kao rezultat projekta mora imati određeni opseg. Proizvod koji proizlazi iz projekta mora ispunjavati tehničke i operativne specifikacije koje su za njega postavljene. Takve se specifikacije temelje na potrebama i očekivanjima povezanim s projektom i prethodno su dogovorene između dobavljača projekta i kupca projekta. Opseg se, dakle, odnosi na unaprijed definirani proizvod koji, pak, uzrokuje promjenu koja postiže cilj projekta.

### **3.2. Projektni menadžment**

Upravljanje projektima može se pronaći gotovo u svakom sektoru globalne ekonomije. *Project Management Institute* definira tzv. projektno intenzivne industrije kao one „u kojima profesionalno zapošljavanje ima visoku razinu projektno orijentiranog rada“ (*Project Management Institute*, 2013:2, navedeno u Horváth, 2019). Takve industrije uključuju

proizvodnju, poslovne usluge, financije i osiguranje, naftu i plin, informacijske usluge, građevinarstvo, komunalne usluge, a osim ovih sektora, upravljanje projektima dobiva sve veći značaj i u drugim industrijama kao što su zdravstvo, izdavaštvo i profesionalne uslužne djelatnosti. „U moderno doba njegovo se podrijetlo može pratiti u upravljanju velikim ad hoc pothvatima, poput projekta Manhattan, te u manjim razmjerima do praktičnih modela koje pruža projektno inženjerstvo“ (Cleland, 1990:2). „Upravljanje projektima je primjena kolekcije alata i tehnika (kao što su CPM i organizacija matrice) za usmjeravanje upotrebe različitih resursa prema postizanju jedinstvenog, složenog, jednokratnog zadatka u vremenskim ograničenjima, troškovima i kvaliteti“ (Atkinson, 1999:337). Tako svaki zadatak zahtijeva određenu kombinaciju teza, alata i tehnika, strukturiranih za okruženje zadatka i životni ciklus (od početka do završetka) zadatka. Atkinson (1999) navodi kako je Britansko udruženje za upravljanje projektima (APM) izradilo Britansko tijelo znanja (BoK) koje također daje definiciju upravljanja projektima kao: planiranje, organizacija, nadzor i kontrola svih aspekata projekta i motivacija svih uključenih za postizanje ciljeva projekta sigurno i u dogovorenom roku, troškovima i kriterijima izvedbe. Dionici projekta su zapravo interesni dionici, a to su članovi projektnog tima kao i svi ostali dionici koji su uključeni u proces razvoja projekta. Najčešći dionici projekta i projektnog menadžmenta su: sponzor kao pojedinac ili grupa koja osigurava (financijske) resurse; projektni tim; potporno osoblje; naručitelj (klijent ili krajnji korisnik) projekta; dobavljači odnosno poslovni partneri te drugi dionici. S obzirom da projekt može imati veliki broj dionika u procesu realizacije, sadržavati mnogo faza, trajati čak i do nekoliko godina, važnu ulogu ima voditelj projekta da svojim pravilnim vođenjem prema svim aspektima projekta izvrši projekt na zadovoljstvo svih dionika. Kada su projekti veliki i kompleksni javlja se potreba i za nekolicinom voditelja projekata i njihovim asistentima koji zajedno formiraju projektni tim. Kada je riječ o informatičkim projektima u npr. izgradnji informacijskih sustava, voditelj tima, ali i ostali članovi moraju poznavati sve aspekte informacijskih sustava i minimalno osnovne tehnologije koje će se koristiti. Osoba koja je voditelj informatičkog projekta odnosno projekt menadžer, mora držati korak s razvojem novih suvremenih tehnologija što zahtjeva kontinuirani rad, učenje, usavršavanje i sposobnost brze prilagodbe. S obzirom da se danas u većini projekata koriste razni softveri za praćenje i analiziranje procesa projekta, znanje i sposobnost korištenja alata obavezan su dio spektra vještina koje projekt menadžer mora imati. Ono čime projekt menadžer još upravlja su ljudski resursi, obuhvat projekta, rizici, troškovi, razmjena informacija te nabava.

## 4. Metodologije pri izradi softverskih rješenja

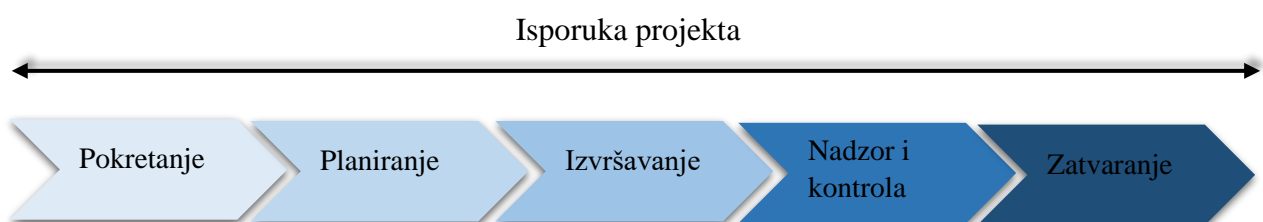
Postoji niz pretpostavki za uspješno vođenje projekta razvoja informatičkih rješenja kao što su korištenje raznih metodologija projektiranja informacijskih sustava, metodologija upravljanja projektima, alata za razvoj programske podrške, formiranje i upravljanje razvojnim timom i dr. Gotovi proizvodi koje izrađuju poduzeća koja su u IT industriji i timovi informatičkih stručnjaka jedinstveni su jer imaju posebne inačice koje naručitelj zahtijeva, te su u potpunosti prilagođeni zahtjevima naručitelja projekta. Kako bi realizacija takvih ciljeva kroz projekte bila moguća i što učinkovitija, u određenom vremenskom periodu uz dostupne resurse i unutar određenog proračuna, potrebno je projekte voditi na pravilan način. U tome značajno pomaže odabir prikladnih metoda i tehnika. U podjeli odnosno klasifikaciji metodologija za upravljanje projektima vlada i dosta velika raznolikost, ponekad se nešto zove metodologija, nešto okvir za rad (*Fremework*), nešto metoda ili tehnika, a nešto i proces. Načelno se metodologije mogu podijeliti na dva osnovna pristupa, a to su tradicionalni i moderni ili agilni pristup. Oba pristupa imaju svoje prednosti i nedostatke, a razlikuju se i okolnosti u kojima su se određeni pristupi pokazali kao bolji odabir. Tradicionalni pristup je generalno bolja opcija kada se radi o većim projektima koje uključuju jasno definirane aktivnosti i faze, a to mogu na primjer klasični informacijski sustav ili implementacija novog operacijskog sustav. Kada je riječ o agilnom pristupu bolji je za projekte koji nemaju jasno definirane zahtjeve i ne zna se koji su to rezultati projekta a projekti kao takvi kraće traju. Takvi projekti mogu biti na primjer razvoj nekog novog uređaja za komunikaciju ili pak razvoj nove platforme za društvene mreže. U ovom se slučaju kao i u drugim granama poslovanja, često se odbacuju neke tradicionalne metode, a odabiru moderne jer su moderne a ne moraju nužno imati prednosti pred klasičnim metodama.

### 4.1. Tradicionalni pristupi

Tradicionalni ili fazni pristup odnosno tradicionalno upravljanje projektima „predstavlja primjenu alata, vještina, tehnika i znanja u aktivnostima projekta kako bi se ispunili ciljevi projekta odnosno zahtjevi naručitelja i njihov zadani opseg“ (Vresk i dr., 2021). Rad na području tradicionalnog upravljanja projektima pripada području takozvanog „*push*“ upravljanja i obično uključuje naručitelja projekta, izvođača projekta, sponzora projekta, rukovoditelja projekta, upravljanja projektom i projektneog tima. Prema Kütz (2012), kako je navedeno u radu Linkea (2019:2), njih kontrolira menadžment za kontrolu portfelja. Ciljevi



projekta dolaze metodički kako se projekt realizirao, a to dovodi do velikog napora u planiranju i malo prostora za pojedine izvršitelje unutar projekta. "Uska grla" u kapacitetu javljaju se na višoj i srednjoj razini upravljanja te prevladava visoka razina napora u koordinaciji, a prilagodbe promijenjenim okvirnim uvjetima odvijaju se vrlo sporo. „Na području tradicionalnog upravljanja projektima uspostavili su se različiti standardi. Najutvrđeniji standard je Tijelo za upravljanje projektima (prema PMBOK) Instituta za upravljanje projektima (PMI)“ (Linke, 2019: 3). Vodič kroz znanje o upravljanju projektima (*engl. PMBOK Guide*) se uzima kao vodeći alat za upravljanje projektima i neophodna literatura svakog voditelja projekta, a danas „se prepoznaje kao globalni standard za upravljanje projektima i jedan je od najboljih i najraznovrsnijih resursa dostupnih svima koji se bave upravljanjem projektima. Vodič kroz PMBOK sadrži temeljne postupke koji su potrebni svim voditeljima projekata kako bi dosegli visoke standarde izvrsnosti projekta“ (Info Novitas, 2021). PMI razlikuje pozicije voditelja projekta, zaposlenika u projektu, sponzora projekta s proračunskom odgovornošću i kupca dok pristup PMBOK-a razlikuje procese za čistu provedbu projekata i specifične procese za različite industrije i proizvode. Uz to, PMI razmatra međuljudske vještine potrebne za uspješno dovršenje projekta te uzima u obzir i agilne metode, ali ih ne uključuje u potpunosti, jer nije orijentiran samo na softverske projekte. Bez obzira na vrstu projekta, proces upravljanja i kontrolinga (*engl. Controlling*), od njegovog početka do završetka, može se podijeliti unutar pet temeljnih faza upravljanja projektima. Slikom 2 prikazano je kako tradicionalni pristupi vođenja projekata uključuju pet faza, a to su: pokretanje, planiranje, izvršavanje, zatvaranje projekta te nadzor i kontrola kao aktivnost prisutna u svim fazama projekta.



Slika 2. Pet faza tradicionalnog pristupa u vođenju projekata

Izvor: Autor rada

Svih pet faza izvršava se pod vodstvom voditelja tima (*Project manager*) i projektnog tima. Kako Vresk i drugi (2021.) navode, značajke tradicionalnih metoda vidljive su u tome da ih

provode organizacije u kojima se provode projekti, koji su u potpunosti određeni i predvidljive su njihove promjene. Samo upravljanje temelji se na usmjeravanju i kontrolama, a komunikacija je formalna te se razvojni model temelji na modelu životnog ciklusa. Ove faze se uglavnom koriste sekvencijalno, ali se mogu i preklapati. Različite industrije koriste različite varijante ovih projektnih faza. Kada se govori o fazama i vremenu, one su međuovisne te se preklapaju, osim kada je riječ o fazi pokretanja i zatvaranja. Faze izvršenja, planiranja i nadzora i kontrole provode se za vrijeme čitavog procesa projekta odnosno njegovog trajanja jer su to aktivnosti koje se kontinuirano prilagođavaju, ponavljaju i nadograđuju.

#### **4.1.1. Faze procesa upravljanja projektima**

Pojedinačne faze procesa upravljanja projekta čine životni ciklus projekta. „Općenito, životni ciklus projekta je niz slijednih i (ponekad) preklapajućih projektnih faza čije ime, broj i način kontrole utvrđuje uprava poduzeća koja pokreće projekt“ (Beram i dr., 2016:6). Ciklus se može formirati i prilagoditi jedinstvenim aspektima organizacije, industrije, tehnologije ili samog projektnog tima i time se omogućava osnovni okvir upravljanja projektom. Za svaku pojedinu fazu postoji i određen broj aktivnosti koje se izvršavaju u toj fazi, a one se dokumentirane nizom izvještaja. Realizaciju svake pojedine faze karakteriziraju informacije odnosno inputi, određeni alati i tehnike te izlazne informacije odnosno outputi kao rezultat primjene alata i tehnika. U tom slučaju „**Ulazi** (inputi) su informacije i podaci koje prikuplja projektni menadžer, ali mogu biti i nove informacije prikupljene u toku izvršenja određenih aktivnosti. **Alati i tehnike** su metode, obrasci ili pristupi koje projektni menadžer primjenjuje u procesu obrade inputa. **Izlazi** (outputi) su ishodišni rezultati postignuti tijekom izvršenja projektnih aktivnosti“ (PMP Certification materials, 2008, navedeno u Beram i dr. 2016: 6).

##### **4.1.1.1. Faza pokretanja (inicijalizacije)**

„Inicijalizacija projekta je prva faza i procesna grupa u životnom ciklusu projekta u kojoj se definiraju ciljevi, očekivanja, članovi tima i stvaraju bitni dokumenti koji će se pratiti kroz sve ostale faze projekta. Unutar faze inicijalizacije projekta definira se početni opseg projekta i okvirni projektni resursi“ (PMBoK, 2008, navedno u Beram i dr, 2016: 16). Jedan od početnih koraka i zadataka projekt menadžera je definiranje posla koji treba obaviti, a ujedno i razvoj

poslovnog slučaja. Taj zadatak se i odnosi na upravljanje ljudima što je vrlo važno u ovoj fazi životnog ciklusa. Podnositelj zahtjeva (poznat i kao kupac) i voditelj projekta dogovaraju se o nekoliko važnih aspekata projekta. Prema Wysocki (2011), bez obzira na format koji se koristi, svaka dobra faza definiranja odgovara na pet osnovnih pitanja:

1. Koji je problem ili prilika kojoj se treba obratiti?
2. Koji je cilj projekta?
3. Koji ciljevi moraju biti ispunjeni da bi se postigao cilj?
4. Kako ćemo utvrditi je li projekt bio uspješan?
5. Postoje li pretpostavke, rizici ili prepreke koje mogu utjecati na uspjeh projekta?

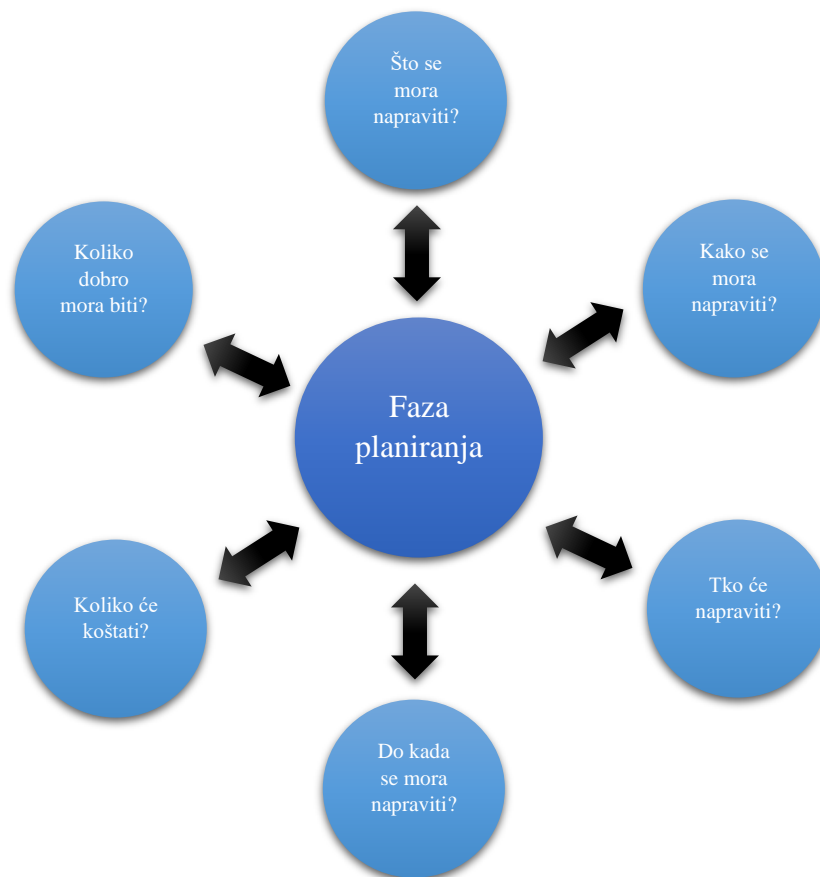
Također, u ovoj fazi su neki od glavnih procesa koji se izvode prije donošenja konačne odluke o odabiru projekta, a oni su: razvoj projektnih prijedloga, ocjena projektnih prijedloga i odabir jednog ili više projekta. Prema Beram i dr. (2016), kriteriji za odabir projekata su najčešće standardi ili mjere koje organizacija koristi za ocjenu važnosti pojedinih zadataka. Uz standarde i mjere, mogu se koristiti i modeli/metode kao podrška pri odlučivanju, kao što su: modeli mjerenja dobiti, matematički modeli, kapitalno budžetiranje te metode za rangiranje i ocjenjivanje projektnih alternativa. Prvi korak pri inicijalizaciji projekta je razvoj poslovnog slučaja. To je najčešće dokument kojim se opravdava inicijalizacija projekta, te sadrži (Westland, 2007, navedeno u Beram i dr. 2016: 17):

- Opis problema ili prilike koje postoji u poslovnom okruženju
- Listu mogućih alternativa za rješenje problema
- Listu troškova i koristi povezanih sa svakom opcijom rješenja
- Preporučeno rješenje za odobrenje

Tehnike koje su najčešće korištene u ovoj fazi životnog ciklusa su **Analiza izvodljivosti** (engl. *Feasibility study*) te **Analiza troška i dobiti** (engl. *Cost – benefit analysis / CBA*). Dokumenti koji proizlaze iz ove faze su **Projektna povelja** (engl. Project Charter) koja predstavlja formalni početak projekta unutar neke organizacije kojom se odobrava korištenje planiranih projektnih resursa, a izdaje ga inicijator odnosno sponzor projekta, te **Izjava o radu projekta**, a dio projektne povelje, koja opisuje proizvod, uslugu ili rezultat tog istog projekta koji je i odobren.

#### 4.1.1.2. Faza planiranja

Kada se govori o planiranju, projektni plan je dinamičan i može se očekivati kako će se on mijenjati. „Proces (faza) projektnog planiranja obuhvaća planiranje glavnih zadataka projekta, procjenu vremena i svih resursa potrebnih za realizaciju projekta, efikasniji nadzor i kontrolu, te uspješniji ishod projekta“ (PMBok, 2008, navedeno u Beram i dr. 2016: 27). Na koja pitanja faza planiranja treba detaljno odgovoriti, prikazano je slikom 3.



*Slika 3. Faza planiranja projektnog menadžmenta*

Izvor: Izrada autora prema Lewis (2007) navedeno u Beram i dr. (2016)

Trajanje faze planiranja uvelike ovisi o samoj prirodi projekta i zrelosti projektne organizacije. U ovoj fazi vrlo je važno jasno i jednoznačno definirati ciljeve projekta da bi outputi bili zapravo rezultat pojedinačnih planova, a odnosi se na jedno 9 područja projektnog

menadžmenta – **Plan upravljanja projektom**. Wysocki (2011) navodi kako postoje tri koristi izrade projektnog plana:

1. **Planiranjem se smanjuje nesigurnost.** Nikada se ne bi trebalo očekivati da će se projekt odvijati točno onako kako je planirano, planiranje posla omogućuje razmatranje vjerojatnih ishoda i potrebnih korektivnih mjera.
2. **Planiranje povećava razumijevanje.** Sami čin planiranja daje bolje razumijevanje ciljeva i zadataka projekta. Čak i kada bi se plan odbacio, i dalje bi postajala korist od izvođenja vježbe.
3. **Planiranjem se poboljšava učinkovitost.** Nakon što se definira plan projekta i potrebni resursi za izvršenje plana, može se planirati posao kako bi se iskoristila dostupnost resursa. Također, zadaci se mogu planirati paralelno; to jest, zadaci se mogu raditi istodobno, umjesto u nizu. Istodobno obavljanje zadataka može skratiti ukupno trajanje projekta. Resursi se mogu maksimalno iskoristiti i izvršiti posao za manje vremena.

Planiranje na projektu ujedno i započinje izradom „Plan upravljanja projektom – krovnog dokumenta koji sadrži detaljne opise i načine izvođenja projekta u skladu s postavljenim ciljevima“ (PMP Certification materials, 2008, navedeno u Beram i dr. 2016: 28). Kvalitetan plan projekta sadrži sljedeće komponente:

- Listu procesa koji će se izvršiti i nivo implementacije za svaki
- Opis alata i tehnika koje će se koristiti
- Opširni opis zadataka za postizanje postavljenih ciljeva
- Plan nadzora i kontrole promjena
- Način mjerenja performansi
- Tehnike komuniciranja u okviru projekta
- Plan identificiranja i dokumentiranja zadataka, te djelovanja u problematičnim situacijama

Između ostalog, projektni plan također pruža i osnovu za mjerenje planiranog posla u odnosu na obavljeni.

#### **4.1.1.3. Faza izvršavanja**

Izvršenje projektnog plana je ekvivalentno ovlaštenju članova projektnog tima za izvršenje svih zadataka koji definiraju njihove odgovarajuće poslove. Svaki član projektnog tima zna što se

od njega očekuje, kako izvršiti neki zadatak odnosno posao te kada je krajnji rok za njegovo izvršenje. Kako bi se ostvarili postavljeni ciljevi potrebno je izvršiti sve aktivnosti koje su već ranije definirane tijekom faze planiranja, stoga izvršavanje projekta podrazumijeva koordinaciju ljudi, svih resursa te niza aktivnosti koje je potrebno započeti, izvesti, nagledati i kontrolirati. U ovoj fazi važno je obratiti pažnju na vremenske rokove i troškovne okvire koji su ostavljeni, razmatrati zahtjeve za promjenama, primjenjivati korektivne akcije te redovito ažurirati projektni plan. Prema Wisocky (2011) izvršenje projektnog plana uključuje četiri koraka. Uz organizaciju ljudi koji će raditi na projektu, voditelj projekta mora činiti sljedeće:

1. Utvrditi specifične resurse (moć osobe, materijale i novac) koji će biti potrebni za izvršavanje posla definiranog u planu
2. Dodijeliti radnike aktivnostima
3. Zakazati aktivnosti s određenim datumima početka i završetka
4. Pokrenuti plan.

Prethodno razvijeni projektni plan služi prije svega kao referentna osnova za sve daljnje aktivnosti vezane uz projekt, a tijekom njegove provedbe kontinuirano se uspoređuju provedene odnosno ostvarene aktivnosti s onima koje su u planu definirane. Nadalje, unutar faze izvršavanja projekta moguće je razlikovati nekoliko odvojenih procesa, svaki od kojih značajno doprinosi uspješnom dovršenju projekta, (Phillips, 2012 navedeno u Beram i dr., 2016: 109):

- izvršenje projektnih aktivnosti,
- primjena koncepta osiguranja kvalitete,
- formiranje projektnog tima,
- razvoj projektnog tima,
- upravljanje projektnim timom,
- distribucija projektnih informacija,
- upravljanje očekivanjima dionika,
- upravljanje nabavom.

Kada govorimo o informatičkim projektima i njihovoj provedbi, primjerice razvoju softvera, važno je naglasiti kako se proces izvedbe projektnih aktivnosti može podijeliti na nekoliko specifičnih podfaza, kao što su: faza definiranja korisničkih zahtjeva, dizajna, izrade, testiranja, implementacije, procjene, obuke, podrške i zatvaranja.

#### 4.1.1.4. Faza nadzora i kontrole

Kontrola kao jedna od ključnih funkcija životnog ciklusa projekta uvelike olakšava postizanje definiranih ciljeva, a preduvjet za to su jasno, realno i precizno definirani početni ciljevi. Bez obzira koliko se pažljivo stvarao projektni plan, sami rad neće ići prema planu. Projekt menadžer mora imati uspostavljen sustav nadzora i kontrole odnosno kontinuirano praćenje i nadziranje napretka projekta ili njegova nedostatka. Menadžer se koriste visokom tehnologijom kako bi bili u korak sa globalnim okruženjem i zahtjevima. Sustav praćenja sažima izvršen posao mjeren u odnosu na plan, a također gleda i unaprijed kako bi upozorio na potencijalne probleme kako bi se u prvo vrijeme upotrijebile korektivne mjere. Također, svaki projekt ne mora nužno uspješno završiti te rezultira neuspjehom ukoliko projekt nije pravilno planiran, kontroliran i upravljan. Nadzor i kontrola projekta je jedina faza koja ne slijedi prethodnu fazu već se provodi tijekom cijelog projekta odnosno njegova izvršavanja, kako informatičkih projekata tako i projekata drugih vrsta. Ukoliko dođe do odstupanja od osnovice potrebno je da projektni tim izradi plan za poboljšanje performansi projekta. Osnovne funkcije kontrole projekta su (PMBOK, 2004, navedeno u Beram i dr, 2016): Planiranje, mjerenje, ocjenjivanje, izvještavanje, poduzimanje korektivnih akcija i upravljanje promjenama. Praćenje i nadzor projektnih aktivnosti može se realizirati unutar projektnog menadžmenta odnosno kao interna kontrola, te izvan njega odnosno eksterna kontrola. Obje vrste kontrole su menadžerski alati za unaprjeđenje rada i rezultata projekta. „Unutarnja kontrola je usmjerena na sustav projektnog menadžmenta, konkretno na procedure za praćenje rada, izvještavanje statusa projekta, te niz potrebnih aktivnosti da bi se projekt priveo kraju. Vanjska kontrola podrazumijeva provedbu dodatnih postupaka propisanih zakonom, te dodatnih aktivnosti prema zahtjevima kupaca, uključujući i nadzor koordinacije na projektu projekta, te administrativne funkcije“ (Beram i dr., 2016:132). Vanjska kontrola je propisana zakonima, a uključuje izvješća izvođača radova o rasporedu, cijenama i tehničkim izvedbama; kontrole rada; kontrole financijskih knjiga i evidencije izvođača od strane državnih revizora. Eksterna kontrola u odnosu na internu, često se pokazuje objektivnijom, a dobivene informacije su načelno vjerodostojnije od onih dobivenih internom kontrolom.

#### 4.1.1.5. Faza zatvaranje projekta

Fazu zatvaranja projekta kao i sve ostale faze životnog ciklusa projekta, treba prethodno isplanirati i dokumentirati te provesti u skladu sa definiranim projektnim planom. „Zatvaranje projekta formalno je sredstvo kojim se signalizira završetak projektnog posla i isporuka rezultata kupcu“ (Wysocki, 2011: 21). Završna faza procjenjuje što se odvijalo tijekom projekta odnosno dokumentiraju se iskustva i spoznaje dobivene kroz rad projekta te pružaju povijesne informacije za uporabu u planiranju i izvršavanju kasnijih projekata, neovisno o tome je li projekt uspio ili nije. Projekt se ne smatra dovršenim sve dok se ne izvedu završne aktivnosti tzv. aktivnosti "pospremanja" (*engl. Clean up activities*) kojima se zatvaraju sve pravne o financijske obveze definirane projektnim ugovorom. Prema Wysocki (2011), svako dobro zatvaranje daje odgovore na sljedeća pitanja:

- Ispunjavaju li rezultati projekta očekivanja podnositelja zahtjeva?
- Ispunjavaju li rezultati projekta očekivanja voditelja projekta?
- Je li projektni tim projekt dovršio prema planu?
- Koje su informacije prikupljene koje će pomoći u kasnijim projektima?
- Koliko je dobro funkcionirala metodologija upravljanja projektima i koliko je projektni tim slijedio?
- Koje smo lekcije naučili iz ovog projekta?

Nadalje, da bi se projekt smatrao formalno zaključenim potrebno je i da naručitelj projekta prihvati isporuku kao proizvod ili uslugu koji odgovaraju njegovim zahtjevima. Faza procesa zatvaranja projekta službeno uključuje dva postupka: pravni postupak zatvaranja i postupak zatvaranja ugovorom. Prema Beram i dr. (2016), pravno zatvaranje projekta uključuje provjeru i dokumentiranje projekta, fazu formaliziranja rezultata ili završetka projekta. Tijekom pravnog postupka zatvaranja projektni tim prikuplja i ažurira projektne dokumentaciju i relevantne podatke i izvješća, a rezultati projekta se uspoređuju sa zahtjevima i očekivanjima klijenta. Pravno zatvaranje projekta uključuje i arhiviranje, a arhiviraju se dokumenti kao što su plan projekta, plan projektnih performansi, revizorska izvješća, ugovorna dokumentacija, kopije svih obavljenih komunikacijskih aktivnosti (zapisnici sa sastanka, zahtjevi za izmjenama i sl.), baza podataka s projekta, procjene kvalitete rada projektnog tima i ostalog osoblja te kompletna izvješća sastavljena tijekom projekta sa završim izvješćem projekta. Kada se govori o dokumentiranju svih iskustava i spoznaja dobivenih kroz rad, kao rezultat sastavljaju se Izvješća „naučenih lekcija“. „Izvješća naučenih lekcija su dokumenti u kojima su evidentirane



sve korisne informacije o radu na projektu. Ova izvješća ukazuju i na područja budućih poboljšanja projekta, koja mogu biti formalna ili neformalna ovisno o organizacijskim normama i/ili zahtjevima“ (Beram i dr., 2016: 145-146). Kada je riječ o definiranju uspješnosti realizacije projekta, u praksi postoje slučajevi kada je očito da je projekt uspješno dovršen ili pak neuspješno, no u određenim situacijama nije moguće sa stopostotnom sigurnošću definirati uspješnost projekta. Najčešće su to situacije kada je output projekta isporučen, ali ne u definiranom roku ili je pak izvan zadanog budžeta. Kako bi se uspješnost što bolje definirala, potrebno je u fazi planiranja unaprijed definirati kriterije uspjeha, te ukoliko se isti unaprijed dogovore, uspješnost projekta se može smatrati definiranom.

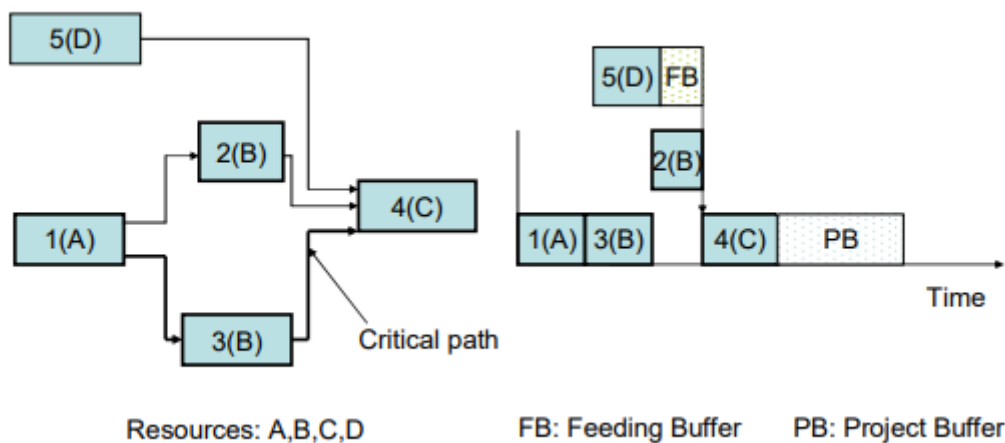
#### **4.1.2. Upravljanje projektima kritičnog lanca**

Upravljanje projektima kritičnog lanca (*Critical chain project management (CCPM)*), pristup je tradicionalnom upravljanju projektima koji je razvio i objavio dr. E.M. Goldratt u svojoj knjizi „*Critical chain*“ (Goldratt, 1997). Ova je metoda nastala kao odgovor na problem kašnjenja velikog broja projekata u odnosu na planirano, pri čemu su postojali veći troškovi od planiranih te su bili manje funkcionalni nego što je predviđeno. Pomoću ovog pristupa izračunavaju se teoretski najraniji početni i završni datumi, kao i najkasniji datumi početka i završetka za sve aktivnosti bez obzira na bilo kakva ograničenja unutar rasporeda aktivnosti. Kod izrade projektnog rasporeda (engl. *Project schedule*) projekta najčešće korištene metode su CPM (*Critical Path Method*) i PERT (*Project Evaluation and Review Technique*) koje su u svojoj osnovi imale definiranje aktivnosti, definiranje redoslijeda izvođenja aktivnosti i određivanje trajanja aktivnosti. Upravo kod određivanja trajanja pojedine aktivnosti se u obzir uzimalo tzv. sigurnosno vrijeme koje je služilo kako bi projektni raspored bio što realniji i ne bi „procurio“ van dozvoljenog vremena tijekom izvođenja projekta. Takav način dobivanja projektnog rasporeda rada ima nekoliko manjkavosti koje su dovodile do kašnjenja projekata. Javljaju se tri problema: zadatak/ aktivnost se ne izvršava sve dok nije krajnje vrijeme za njegov početak; zadaci/ aktivnosti se izvršavaju na način da se stignu izvršiti do kraja roka iako se mogu završiti i ranije te se biraju samo oni zadaci za čiji završetak ima dovoljno vremena odnosno laganiji zadaci.

Prema Raz i dr. (2001), polazna točka za CCPM je popis zadataka zajedno s njihovim procjenama trajanja i ovisnosti. Prvi se korak sastoji od razvijanja početnog rasporeda projektnih zadataka. To se radi uzimajući u obzir ovisnosti među zadacima i dostupnost resursa. Budući da barem neki od resursa imaju ograničenu dostupnost, rezultirajući raspored vjerojatno će biti duži od dobivenog rasporeda s osnovnim algoritmom „Metoda kritičnog puta“ , jer se kritične aktivnosti odgađaju dok se čeka resurse koji su potrebni. Tako CCPM identificira „Kritični put“ kao skup zadataka koji daju najduži put do završetka projekta nakon niveliranja resursa te daje očekivani datum završetka projekta. Oni resursi koji su potrebni za zadatke na kritičnom lancu definirani su kao „Kritični resursi“. Sljedeći korak u planiranju CCPM-a sastoji se od ponovnog izračunavanja raspored projekata na temelju procjena skraćenih trajanja zadataka. Prevedeno od strane autora rada prema Stratton (2009), glavni konceptualni elementi CCPM-a predstavljeni su u nastavku u kontekstu planiranja, izvršenja i kontinuiranog poboljšanja.

### **Planiranje projekta**

- CCPM uzima u obzir ovisnost o resursima kao i prvenstvu pri određivanju trajanja projekta. To se naziva kritičnim lancem. Na slici x, kritični put bi se označavao aktivnostima 1-3-4, dok se u CC (*Critical Chain*) označava 1-3-2-4 zbog zajedničkog resursa B. U takvim se slučajevima pokazuje da je kritični lanac duži od kritičnog puta i sve četiri aktivnosti trebaju se upravljati u skladu s istim.
- CCPM uvodi koncept vremenskih međuspremnik projekta i kako bi se prilagodilo učinkovito upravljanje vremenom međuspremnik koje se obično gubi na razini aktivnosti kada se njima upravlja lokalno. Međuspremnik projekta nalazi se na kraju projekta radi zaštite kritičnog lanaca, a „*Project buffer*“ za punjenje izolira sekvence aktivnosti s plutajućim djelom iz kritičnog lanaca. Dakle, takvi međuspremnici omogućuju agregiranje vremena međuspremnik kao i bolju kontrolu, omogućujući kraće i kontroliranije vrijeme izvođenja. Pri uspostavljanju međuspremnik, predložena početna točka će prepoloviti postojeće vrijeme aktivnosti i polovicu ostatka staviti u agregirani međuspremnik. Stoga je međuspremnik jednak trećini kombinacije aktivnosti i međuspremnik.
- Prilikom planiranja u višeprojektom okruženju, CCPM zagovara postepeno objavljivanje projekata oko određenog resursa koji djeluje kao bubanj. Potrebno je osigurati protok i izbjegavati previše otvorenih projekata koji rezultiraju pretjeranim višestrukim zadacima i propuštenih termina.



Slika 4. Mrežni dijagram i kritični raspored lanca koji pokazuju međuspremnike

Izvor: Stratton, R. (2009). Critical chain project management theory and practice. *Journal of Project Management and Systems Engineering*

### Izvršenje projekata i kontinuirano poboljšanje

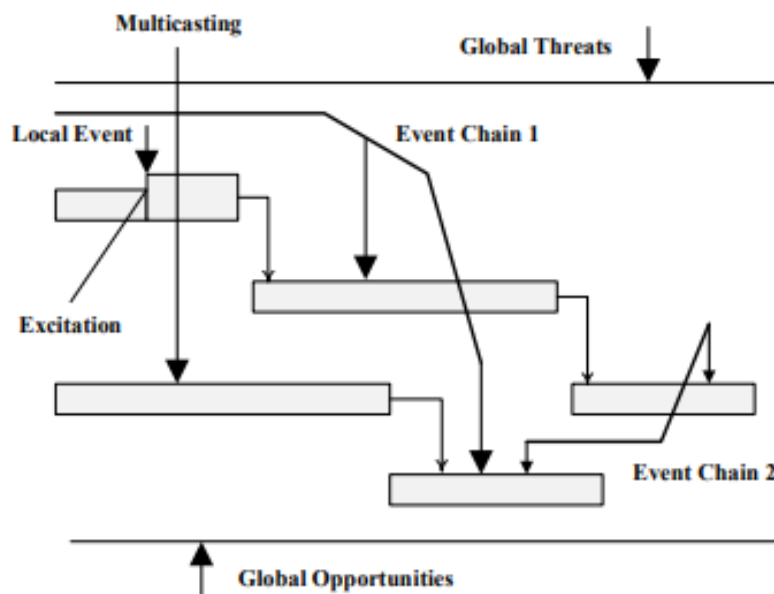
- **Izveštavanje o izvršenju zadatka** - uobičajena je praksa da se vrijeme aktivnosti izvještava o obavljenom poslu tjedno ili čak mjesečno. Uz CCPM preostalo vrijeme za završetak aktivnosti izvještava se puno češće - idealno svakodnevno.
- **Omogućavanje vidljivosti nadolazećih zadataka** - kako u sustavu planiranja ne postoje posredni datumi zadatka, podaci o preostalom vremenu pružaju naprednu najavu o nadolazećim zadacima.
- **Trenutni i nadolazeći zadaci u skladu s prioritetima kako bi se osiguralo da zadaci učinkovito napreduju** - u složenijem višeprojektom okruženju postoje mnogi zadaci u tijeku koji se natječu za vrijeme davatelja resursa. U CCPM-u je prioritet omjer završetka kritičnog lanca i potrošnje međuspremnika, obično koristeći zelene, žute i crvene kodove prioriteta u boji. Prikazuju se i nadolazeći zadaci koji prikazuju njihov relevantni prioritet, kao i predviđeno vrijeme kada se očekuje da će isti postati dostupni tom resursu.
- **Svakodnevno praćenje potrošnje međuspremnika od strane projektnog menadžera i po potrebi poduzimanje mjera za oporavak** - potrošnja međuspremnika ukazuje na to da zadatak premašuje postavljeno vrijeme i da će upravitelju zadataka možda biti potrebna pomoć. Za oporavak situacije može biti potrebno djelovanje na određenoj razini projekta.

- **Viši menadžeri prate stanje svih projekata i poduzimaju mjere po potrebi** - na ovoj se razini status prioriteta svih projekata pregledava radi praćenja i rješavanja oporavka programa na višoj razini.
- **Razlozi kašnjenja se prate i fokusiraju na poboljšanje** – relevantni razlozi odgode izdvajaju se radi poboljšanja aktivnosti.

#### 4.1.3. Metodologija lanca događaja

Metodologija lanca događaja (*Event chain methodology* (ECM)) nadopunjuje CPM i CCPM. Prema Virine i Trumper (2016), to je tehnika modeliranja nesigurnosti i mrežnog planiranja koja se usredotočuje na identificiranje i upravljanje događajima i lancima događaja, koji mogu utjecati na projekte. Ova metodologija uvelike poboljšava točnost planiranja te pomaže umanjiti negativan utjecaj raznih događaja na izvršavanje svih projektnih aktivnosti. Kada je riječ o riziku, upravljanje rizikom obično podrazumijeva upotrebu tijeka rada i raznih alata te tehnike vizualizacije. Jedan od ključnih koraka kada dođe do razvoja pouzdanog rasporeda projekta jest njegovo identificiranje. Ono što se preporučuje kod identifikacije su različite tehnike kao što su kontrolni popisi, brainstorming, intervjui, analiza pretpostavki, SWOT analiza, temeljni uzrok i razne vrste dijagrama. Što se tiče analize rizika, PMI predlaže korištenje nekoliko tehnika, kao što su očekivana novčana vrijednost, analiza osjetljivosti i Monte Carlo analiza. „Očekivana novčana vrijednost zahtijeva upotrebu stabla odlučivanja kako bi se identificirale projektne alternative i podrška odlučivanja. Analiza osjetljivosti određuje kojih rizičnih događaja ima najviše te potencijalni utjecaj na raspored projekata. Monte Carlo simulacija koristi vjerojatnosti koje ne ulaze za generirajuću raspodjelu potencijalnih rezultata, što omogućuje uključivanje rizika i nesigurnost u rasporedu projekata“ (Schuyler, 2001, navedeno u Avlijaš i Radunović, 2019). Monte Carlo i analiza osjetljivosti su temelj za primjenu ECM-a. Kada je riječ o primjeni ECM-a kao metodi, prema Virine i Trumper (2016), identificiranje i upravljanje tim događajima ili lancima događaja (kada jedan događaj uzrokuje drugi događaj) fokus je metodologije lanca događaja. Metodologija se fokusira na događaje više kao kontinuirani postupak za promjenu projektnog okruženja jer s kontinuiranim problemima unutar projekta moguće je otkriti i popraviti prije nego što imaju značajne probleme koji mogu utjecaj na projekt. ECM se temelji na šest principa koji uključuju identifikaciju pojedinačnih i višestrukih

događaja, definiciju trenutka događaja i uzbuđenja, vizualizacija lanaca događaja, Monte Carlo analiza, analiza osjetljivosti i mjere tehnike (Intaver Institute, 2011, navedeno u Avlijaš i Radunović., 2019). Slika 5 prikazuje primjer lančanog dijagrama događaja te većinu principa i tehnika.



Slika 5. Primjer lančanog dijagrama događaja

Izvor: Avlijaš, G., & Radunović, M. (2019). Application of Event Chain Methodology in Schedule Risk Analysis. *European Project Management Journal*, 9(2), 26-34

Avlijaš i Radnunović (2019) razlikuju šest principa na kojima se temelji ECM, a to su:

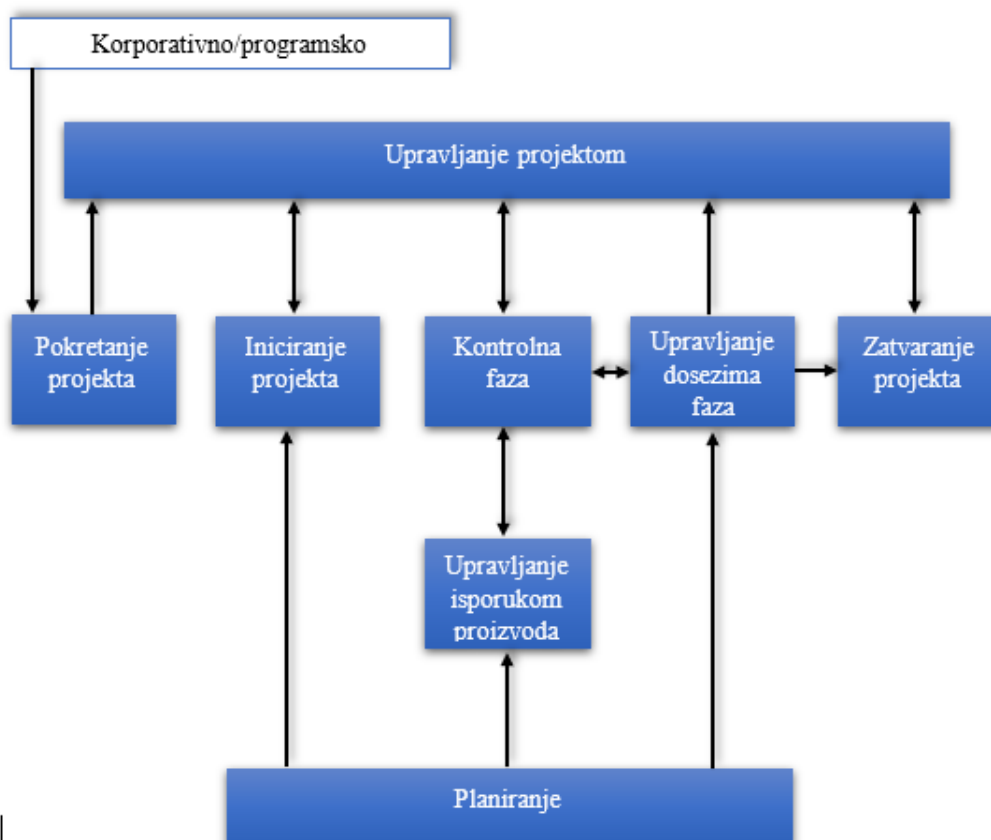
1. **Trenutak događaja i uzbuđenja** – najviše projektnih aktivnosti u stvarnom životu nisu jednolike i kontinuirane i one su pod utjecajem događaja koji se transformiraju iz jednog oblika u drugi. Ti događaji uključuju promjene u resursima, materijalima ili dijelovima koje mogu izmijeniti jedno ili više trajanja aktivnosti. Zadatak od događaja do aktivnosti uključuje utjecaj na vjerojatnost pojave, uzbuđeno stanje i trenutak događaja, koji također mogu biti vjerojatni. Događaji mogu biti pozitivni ili negativni, a mogu transformirati aktivnost od osnovnog (prizemnog) do uzbuđenog stanja, kao i obrnuto (napori ublažavanja).

2. **Lanci događaja** - događaju se kada jedan rizični događaj (pošiljatelj) uzorkuje drugi događaj (primatelj), koji obično dovode do stvaranja efekta valovitosti kroz projekt i imaju značajni utjecaj. Događaj može uzrokovati više događaja u različitim projektnim aktivnostima ili mogu utjecati na različite resurse, stvarajući efekt višestrukog slanja. Svaki primatelj može djelovati kao i pošiljatelj ako pokrene drugi događaj.
3. **Dijagrami lanca događaja** - omogućuju vizualizaciju složenih odnosa između događaja i rasporeda, a koriste se za samo značajne događaje (Virine, & McVean, 2004; Virine i Rapley, 2003, navedeno u Avljaš i Radunović, 2019). Dijagrami lanca događaja nadograđuju se na originalne Ganttove dijagrame s navedenim izmjenama (slika 5). Neka od osnovnih pravila koja se mogu lako razumijeti uključuju: strelice prema dolje koje predstavljaju negativne utjecaje, strelice prema gore pozitivne utjecaje, pojedinačne strelice predstavljaju događaje, višestruko povezane linije predstavljaju *multicasting*, strelice izvan grafikona predstavljaju globalne događaje koji utječu na sve aktivnosti, veličina strelica predstavlja relativnu vjerojatnost događaja, vodoravni položaj traka predstavlja srednji trenutak, itd.
4. **Monte Carlo analiza** - Monte Carlo simulacija provodi se kako bi se kvantificirao kumulativni utjecaj identificiranih događaja i lanaca događaja. Osim raspodjele vjerojatnosti i utjecaja rizičnih događaja, inputi također uključuju raspodjele vjerojatnosti i utjecaje koji se odnose na fluktuacije u trajanju aktivnosti, koje ne bi trebale imati isti osnovni uzrok kao i događaji kako bi se izbjeglo dvostruko računanje rizika. Uz tradicionalne rezultate Monte Carlo analize, ECM uključuje izračun stope uspjeha, vjerojatne novčane tijekove, uvjetno grananje, itd.
5. **Kritični lanci događaja** - rizični događaji i lanci događaja s najznačajnijim utjecajima nazivaju se kritičnim. Analiza osjetljivosti se koristi za identifikaciju kritičnih događaja i ublažava njihov negativan utjecaj analizom korelacija između lanaca događaja, trajanja projekta i troška. Kritični događaji mogu biti vizualizirani pomoću tablice osjetljivosti, koja omogućuje izračunavanje korelacije koeficijenta, troškova događaja i ukupni trošak projekta s rizicima i neizvjesnosti.
6. **Mjerenje performansi** – korištenje stvarnih podataka o izvedbi osigurava korištenje ažuriranih podataka i ponovni izračun vjerojatnosti pojava i trenutaka događaja. To dalje omogućuje stvaranje ažuriranog rasporeda projekata i trajanja, koji su obično popraćeni grafikonima koji ilustriraju varijacije mogućnosti za

dovršetak projekta u određenom roku. Šansa da se ispoštuje rok stalno se ažurira i to se obično s vremenom smanjuje zbog različitih projektnih rizika, ali se također može poboljšati zbog mjera za ublažavanje rizika koje izvode menadžeri.

#### 4.1.4. PRINCE2

PRINCE2 predstavlja akronim za *PR*ojects *IN* *C*ontrolled *E*nvironments, *version 2*, odnosno to su projekti u kontroliranom okruženju verzija 2. Razvijen je od strane vlade Ujedinjenog Kraljevstva i predstavlja standard za upravljanje javnim projektima u UK. PRNCE2 nastao je iz ranije metode pod nazivom PROMPTII i iz PRINCE metode upravljanja projektima koja je inicijalno razvijena 1989. godine. od strane *Central Computer and Telecommunications Agency* (CCTA) kao standard vlade UK za upravljanje IT projektima. PRINCE2 je bazirana na 7 principa odnosno tema i procesa. Sedam tema predstavlja: poslovni slučaj, organizacija, kvaliteta, plan, rizik, promjena i napredak. Principi i teme se realiziraju kroz procese.



Slika 6. Dijagram PRINCE2 procesa

Izvor: Prevedeno i prilagođeno od strane autora ovog rada prema Priščan, A. (2021). Upravljanje projektima u RH: Stanje nacije. Dostupno na: <http://www.infotrend.hr/clanak/2008/10/upravljanje-projektima-u-rh-stanje-nacije,23,517.html> [pristupljeno 17. rujna 2021]

Kako se principi i teme realiziraju kroz procese, prema Vaničková (2017) PRINCE2 metodologija definira sljedeće procese tijekom životnog ciklusa projekta:

- **Pokretanje projekta** (*Starting up a project - SU*) - ovaj postupak prikuplja informacije potrebne za projekt, imenuje izvršnog direktora i tim za upravljanje projektom, donosi odluku o stavu za primijenjenu na projektu, definira kvalitetu koju očekuje kupac, planira rad potreban za pripremu sporazuma između kupca i dobavljača, stvara zapisnik voditelja projekta (bilježenje prvih rizika, itd.).
- **Iniciranje projekta** (*Initiating a project - IP*) - cilj postupka je provjeriti ima li projekt poslovne zasluge, uključujući definiranje parametara projekta. Dokument pod nazivom Projekt Inicijalna dokumentacija sastavlja se i koristi za procjenu uspjeha projekta. Nadalje, sastavljaju se strategije za kvalitetu, rizik, konfiguraciju i komunikaciju iskrvarila.
- **Upravljanje projektom** (*Directing a project - DP*) - postupak je u vlasništvu višeg menadžmenta čiji je cilj prenijeti odgovornost za uspjeh projekta na Projektni odbor. Znači postavljanje standarda aktivnosti koje delegiraju voditelj projekta i njegov tim. Viši menadžment provodi pripremu i odobravanje projektnog plana, odobravanje provedbe projekta, provjera održivosti projekta, praćenje napretka i uspješnosti završetka projekta.
- **Kontrolna faza** (*Controlling a stage - CS*) - ovaj postupak opisuje svakodnevne aktivnosti na koje projekt menadžer provodit i prati napredak projekta i provjerava pojedine aktivnosti. Voditelj projekta provodi aktivnosti kao što su odobravanje posla, prikupljanje informacija, praćenje rizika, pregled situacije, izvještavanje i moguće korektivne mjere.
- **Upravljanje isporukom proizvoda** (*Managing product delivery - MP*) - postupak osigurava komunikaciju između proizvođača, voditelja projekta i tima stručnjaka, posebno ako projektni tim potječe iz izvođača radova. Proces uključuje, na primjer, planiranje, provjeru, pomirenje i dodjeljivanje dodijeljenog timskog rada prema potrebnoj kvaliteti, izvještavanje voditelja i prihvaćanje proizvoda.
- **Upravljanje dosezima faza** (*Managing stage boundaries - SB*) - cilj procesa je završiti fazu i planirati drugu, uključujući ažuriranje plana projekta; ažuriranje poslovnog slučaja projekta; ažuriranje registra rizika; izvještavanja o kraju faze; i prijelaz u sljedeću fazu.



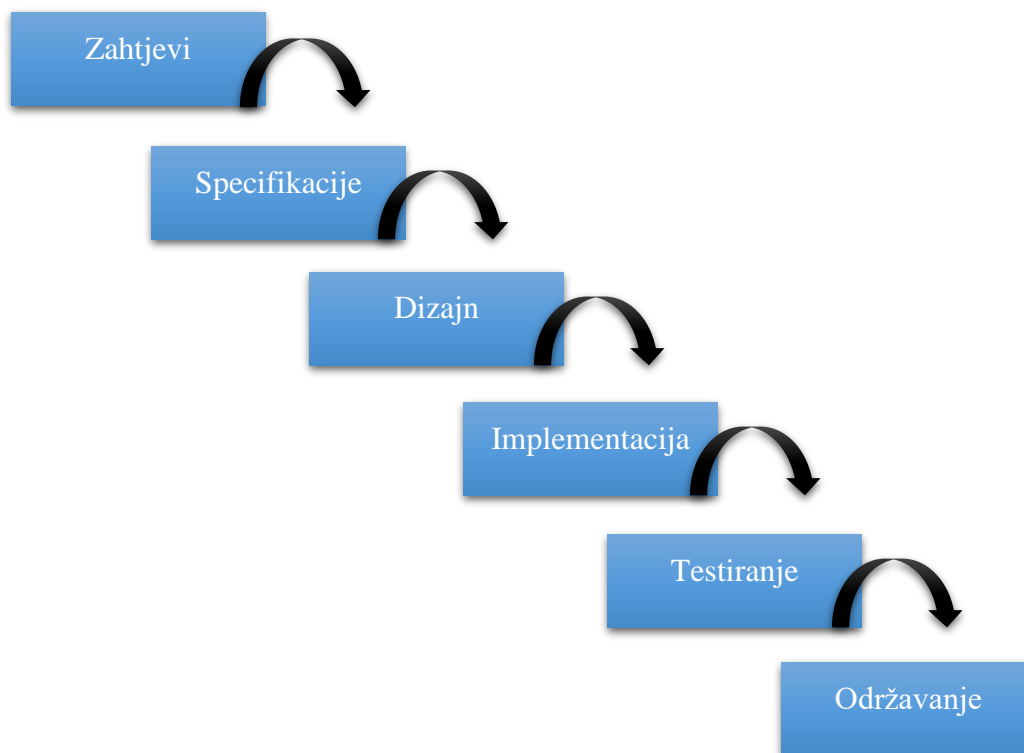
- **Zatvaranje projekta** (*Closing a project* – CP) - obuhvaća postupak za udovoljavanje zahtjevu voditelja projekta kako bi pravovremeno projekt bio završen nakon što su ciljevi projekta ispunjeni. Rezultat završne faze projekta je bilježenje opsega ciljeva projekta, kako bi se potvrdilo zadovoljstvo kupca proizvodom, kako bi se osigurale mjere prikladne za održavanje i podrške projekta, steklo iskustvo od provedbe projekta i izvijestilo o doprinosu projekta.

Ova metodologija se još naziva i strukturnim upravljanjem projektima zato što može upravljati projektom na logičan i organiziran način slijedeći definirane procese. PRINCE 2 opisuje što bi projekt trebao raditi odnosno opisane su aktivnosti u pojedinim fazama te kada se odvija koja faza, a to čini nizom procesa koji pokrivaju aktivnosti potrebne za projekt, od pokretanja do zatvaranja.

#### **4.1.5. Vodopadna metodologija**

Vodopadna metodologija odnosno model vodopada (*Waterfall model*) je u softversku industriju preuzet iz dvije industrije, a to su proizvodnja i građevinarstvo. Obje industrije imaju strukturirano fizičko okruženje i bilo kakve naknadne promjene nakon isporuke proizvoda su gotovo nemoguće. Tijekom početka industrije proizvodnje softvera, nisu postojale formalne metodologije razvoja softvera, stoga je preuzet vodopadni model. Vodopadna metodologija je prva metodologija koja definira proces odvijanja životnog ciklusa nekog sustava. „Vodopadna metodologija je bio prvi moderan pristup korišten u izgradnji softvera, a definirao je Winston W. Royce davne 1970. godine“ (*The Waterfall Development Methodology*, 2006, navedeno u Beram i dr., 2016). Model vodopada je sekvencijski proces i ima svojstvo linearnosti, što znači da nakon jedne faze u potpunom životnom ciklusu slijedi druga, bez da ima mogućnosti povratka na bilo koju drugu fazu. Kvalitetni i dobro dizajnirani objekti imaju nacрте i planove. Kada se jednom planira dizajn ili struktura projekta, postupak provedbe je uvelike jednostavniji i učinkovitiji, pa tako vrijedi i za informatičke projekte odnosno programiranje i izradu softverskih rješenja. Programi imaju svoje algoritme i dijagrame tokova koji se planiraju, točnije gdje su navedeni koraci i sve funkcije koje se trebaju izvesti. Prema McCormick (2012), faze razvoja su: specifikacija zahtjeva, koncepcija, programiranje, testiranje i uklanjanje pogrešaka, implementacija i održavanje. U ovakvom strukturiranom načinu razvojni tim ne ide u sljedeću fazu provedbe dok prethodna nije izvršena u potpunosti, primjenom ovog modela na

razvoj softvera stvara se veliki trošak i iskorištava se velika količina vremena u svakoj pojedinoj fazi razvoja, jer se također javljaju razni problemi i nedoumice koji se moraju riješiti. Najveći napor i velika količina vremena uglavnom se utroši u početku kod dizajniranja i unaprijed ispravljanja pogrešaka. Kada faza dizajna završi slijedi programiranje bez naknadnih promjena. Timovi za analizu, dizajn i programiranje često rade na manjim dijelovima u razvojnom procesu, a naglasak se stavlja na dokumentaciju svake pojedine faze. Kako je već navedeno, model vodopada je sekvencijski, slikom 7 prikazan je redosljed kako se faze odvijaju jedna iza druge bez mogućnosti povratka.



Slika 7. Model vodopada

Izvor: Prevedeno i prilagođeno od strane autora ovog rada prema Cohn, M. L., Sim, S. E., & Lee, C. P. (2009).

What counts as software process? Negotiating the boundary of software work through artifacts and conversation. *Computer Supported Cooperative Work (CSCW)*. Dostupno na:

<http://www.drsusansim.org/papers/cscwj-process-cohn.pdf> [pristupljeno 11. srpnja 2021.]

Aktivnosti koje se provode u svakoj pojedinoj fazi, objašnjene su u nastavku. Prema McCormick (2012):

1. Faza specifikacije zahtjeva uključuje razumijevanje onog što treba dizajnirati i koje su funkcije, svrha, itd. Ako se ne zna što se želi dizajnirati ne može se ni nastaviti sa projektom. Čak i mali kod, poput dodavanja dva cjelobrojna broja treba napisati tako da ima neki „izlaz“. U ovoj fazi navedeni su zahtjevi koje će softver trebati zadovoljiti. Zatim se zahtjevi prezentiraju timu programera. Ako je ova faza uspješno dovršena, osigurava rad preostalih faza kao što programer ne mora biti opterećen uvođenjem promjena u kasnijim fazama zbog promjena u zahtjevima.
2. Faza analize prema zahtjevima, softver i hardver moraju biti pravilni kako bi se izvršio projekt. Odlučuje se koji će se računalni jezik koristiti za dizajniranje softvera, do sustava baze podataka koji se može koristiti za nesmetano funkcioniranje softvera, takve se važne značajke odlučuju u ovoj fazi.
3. Faza dizajna u kojoj se bilježi algoritam ili dijagram toka programa ili softverskog koda. To je vrlo važna faza koja se oslanja na prethodne dvije faze provedbe. Ispravan dizajn u ovoj fazi osigurava izvršenje u sljedećoj fazi. Ako se tijekom faze provedbe uoče da postoje još neki zahtjevi za projektiranje koda, faza analize se ponovo provjerava i faza provedbe provodi se prema novom skupu resursa.
4. Faza programiranja se odvija na temelju dizajniranog algoritma ili dijagrama toka i odvija se programiranje softvera. Ovo je faza u kojoj se ideja i dijagram aplikacije fizički kreiraju ili materijaliziraju. Ispravno izvršavanje prethodnih faza osigurava jednostavniju i lakšu provedbu.
5. Faza testiranja, s dovršenim programiranjem aplikacije, testira napisani kod. Testiranje provjerava ima li nedostataka u dizajniranom softveru i je li softver bio dizajniran prema navedenim specifikacijama. Ispravno izvršavanje ove faze osigurava da zainteresirani klijent bude zadovoljan izrađenim softverom. Ako ima nedostataka, proces razvoja softvera se mora vratiti u fazu dizajniranja. U fazi dizajna provode se promjene, a zatim slijedi provedba faza programiranja i testiranja.
6. Faza prihvatanja je zadnja faza razvoja softvera u vodopadnom modelu. Ispravno izvršenje svih prethodnih faza osigurava provedbu prema predviđenim zahtjevima i štiti je najvažnije osigurava zadovoljstvo klijenta. Međutim, u ovoj fazi će možda trebati i definirati podršku klijentu vezano za softver koji je stvoren. Ako klijent zahtijeva daljnja poboljšanja postojećeg softvera, postupak razvoja se mora početi iznova od prve faze odnosno zahtjeva.

Iako ova metoda ima nedostatka što se tiče ne mogućnosti vraćanja na određene faze, visokih troškova, utrošenog vremena, ne mogućnost prilagodbe modela i dr., međutim ovaj model je i dalje jedan od najčešće korištenih metodologija. Korišteni su i novi modeli i metodologije koje će biti u nastavku navedene i pojašnjene, ali razlog široke upotrebe ovog modela je to što je proveden na raznim projektima što se tiče razvoja softvera.

## 4.2. Agilne metodologije

S obzirom na ne mogućnost balansiranja između fleksibilnosti i stabilnosti prilikom provođenja projekata koristeći se tradicionalnim pristupima te kao rezultat rastućih zahtjeva za neprekidnim inovacijama, te zahtjeva za smanjenjem troškova osmišljen je relativno novi, drugačiji pristup procesu razvoja, a to je agilni pristup odnosno agilna metodologija. Agilne metodologije su također i predstavnik modernog ili iterativnog pristupa. Primjena agilne metodologije, prema Beram i dr. (2016.), temelji se na poslovnim principima kao što su: neprekidna inovacija, prilagodba proizvoda, smanjenje vremena isporuke, prilagodba ljudi i procesa te pouzdani rezultati. Sve spomenuto veže se uz pojam „agilnost“, a pojam „agilnost predstavlja sposobnost organizacije da stvara i istodobno fleksibilno odgovara na nastale promjene, kako bi projekt uspješno završio u turbulentnom poslovnom okruženju“ (Highsmith, 2004, navedeno u Beram i dr. 2016.). Temeljna ideja ovog pristupa je potaknuti programere da razviju dovoljno napredne modele koji mogu podržavati akutne probleme dizajna i što manje dokumentacije, a najvažnije načelo jest zadovoljstvo kupaca davanjem brzih i kontinuiranih isporuka korisnog softvera. 2001. godine, u SAD-u, stvoren je dokument „Manifest agilne metodologije“ koji je prihvaćen kao definicija agilnog razvoja sustava i pridruženih agilnih principa. Prema, tome uspostavljen je novi sustav vrijednosti temeljen na sljedećim pretpostavkama (Beck, 2001, navedeno u Beram i dr., 2016):

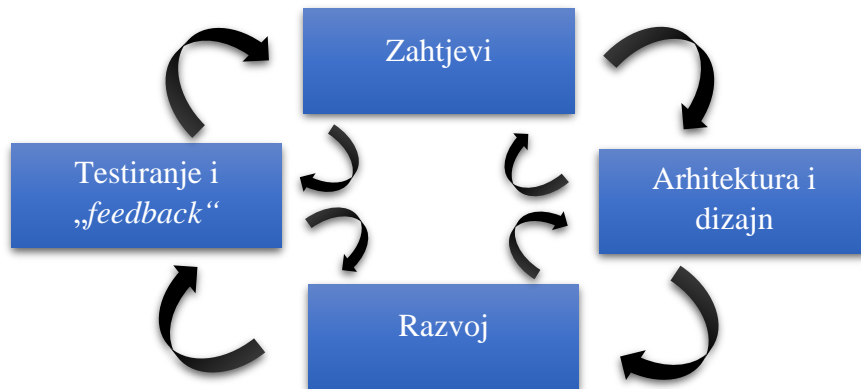
- pojedinci i interakcija važniji su od procesa i alata
- proizvod koji funkcionira važniji je od detaljne dokumentacije
- suradnja s korisnikom važnija je od ugovornog pregovaranja
- odgovor na promjenu važniji je od praćenja plana.

Prethodno navedeni principi ni u kojem slučaju ne umanjuju vrijednost procesa, alata, dokumentacija, ugovora i planova, već ukazuju na prioritete u radu i djelovanju. Iako je Manifest agilne metodologije napisan prvenstveno za agilni razvoj softvera, on se može

primijeniti i na agilno upravljanje informatičkim projektima. Važno je naglasiti i osnovne principe primjene i razvoja agilne metodologije (Beck, 2001, navedeno u Beram i dr., 2016):

- naručitelj i izvođač informatičkih projekata su u interakciji i zajedno proaktivno funkcioniraju
- potrebno je u projektom timu stvoriti motivirajuću radnu okolinu i podupirati promjene koje su, zbog turbulentnosti radnog okruženja, sve učestalije
- način komunikacije među sudionicima procesa je „licem u lice“
- agilne metode ne podupiru nikakav dodatni napor, primjerice prekovremeni rad zbog uvođenja promjena
- ističe se važnost konstantnog rada, razvijanja novih vještina i sposobnosti sudionika u tom procesu
- umjesto strogog nadzora naglašava se samoorganizacija i samostalnost svakog sudionika te se podupire pružanje savjetodavnih usluga sa strane stručnog kadra, s ciljem lakšeg prihvaćanja promjena, osnovna mjera napretka je proizvod (završen ili s određenim stupnjem završenosti) koji funkcionira i kojim je klijent zadovoljan
- povratne informacije o radu tima utječu na poboljšanje efikasnosti daljnje provedbe projekta
- zahtjevi za promjenama su dobrodošli u svakom trenutku
- na projektu rade razvojni inženjeri i ljudi iz poslovnog područja koji su inicirali sam projekt
- projektni tim periodično ispituje dobre i loše postupke te ih nastoji popraviti ukoliko je to potrebno

Prema svim navedenim principima može se reći kako proizlazi da je primjena i razvoj agilnih metoda jako važna za provedbu informatičkih projekata, neovisno o tome radi li se o projektima implementacije informacijskih sustava ili razvoju softvera. Prema McCormick (2012), razvoj softvera koristi planirani i strukturirani proces za razvoj softverskih proizvoda. Pojam "Razvoj softvera" postupak je koji uključuje pisanje i održavanje kod-a koji se koristi za razvoj softvera. Međutim, često proces razvoja softvera uključuje ideju softvera, dizajniranje softvera do konačnog razvoja softvera koji se zatim predaje klijentu. Slikom 8 prikazani su koraci agilne metodologije, a to su prikupljanje korisničkih podataka, dizajn i dokumentacija, razvoj, testiranje i primjena.



Slika 8. Proces agilnog razvoja softvera

Izvor: Prevedeno i prilagođeno od strane autora ovog rada prema Umbreen, M., Abbas, J., & Shaheed, S. M. (2015). A Comparative Approach for SCRUM and FDD in Agile. *International Journal of Computer Science and Innovation*

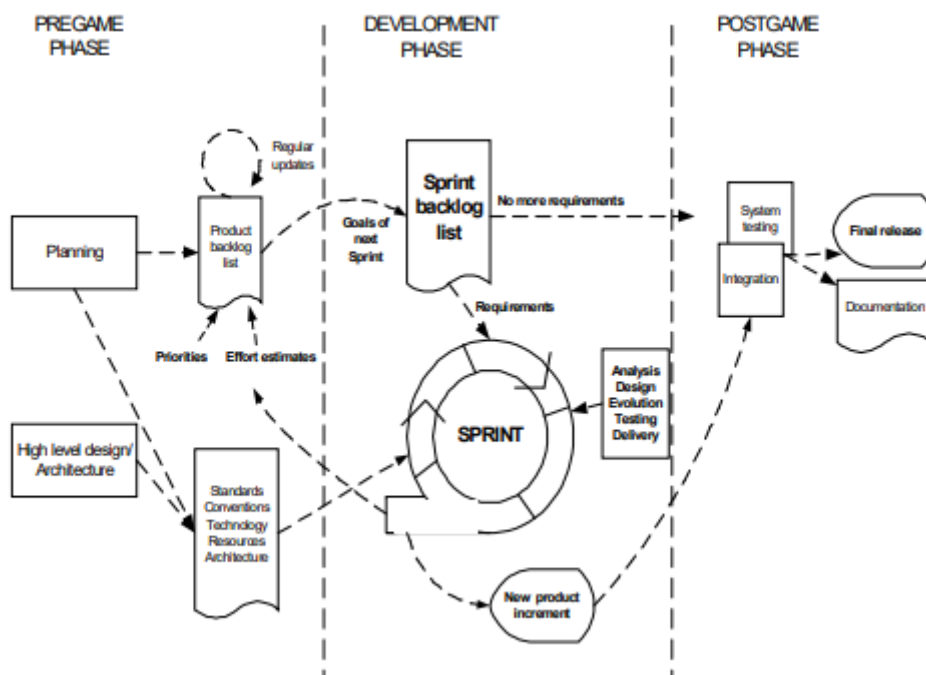
Agilni pristup broji nekoliko metodologija i modela, a to su: SCRUM metodologija, Model brzog razvoja aplikacija, Kanban metodologija, Metoda ekstremnog programiranja, Crystal, Metoda dinamičkog razvoja sustava, Lean metoda i Metoda razvoja temeljena na osobinama. Sve navedene metode u nastavku će biti prikazane i pojašnjene.

#### 4.2.1. SCRUM

Kada je riječ o SCRUM metodi, ona je jedinstvena po empirijskom procesu kontrole od svih agilnih metodologija. „To je empirijski pristup koji primjenjuje ideje teorije industrijskog procesa upravljanja razvojem sustava što rezultira pristupom koji ponovno uvodi ideje fleksibilnosti, prilagodljivosti i produktivnosti“ (Schwaber i Beedle, 2002, navedeno u Umbreen i dr. , 2015). Ova metoda koristi stvarnu procjenu projekta, a ne koristi prognoze napretka projekta ili optimistične procjene. SCRUM pristup je pristup koji je razvijen za upravljanje razvojem softvera i fokus je na to kako članovi tima trebaju funkcionirati kako bi sustav bio fleksibilan u stalnim promjenama koje se odgađaju u okolini. Prema Abrahamsson i dr. (2017), glavna ideja je da razvoj sustava uključuje nekoliko varijabli iz okoline i tehničkih varijabli (npr. zahtjevi, vremenski okvir, resursi i tehnologija) koja će se vjerojatno primijeniti tijekom procesa. To razvojni proces čini vrlo nepredvidljivim i složenim, a zahvaljujući

fleksibilnosti ove metode, proces razvoja sustava može pratiti i odgovoriti na promjene uzrokovane okolinom ili pak nekim drugim čimbenicima. Također, ova metoda pomaže poboljšati postojeće inženjerske prakse u organizaciji jer često uključuje aktivnosti koje su usmjerene na identificiranje bilo kakvih nedostataka ili prepreka u procesnom razvoju kao i ostale prakse koje se koriste. SCRUM metoda predstavlja razvoj softvera „korak po korak“, a „osnovna vremenska cjelina je *engl. Sprint*, što predstavlja zaokruženu jedinicu razvojnog procesa koja najčešće traje 30 kalendarskih dana“ (Beram i dr., 2016). SCRUM prolazi kroz sve faze razvoja: planiranje, programiranje, testiranje i isporuka. Na kraju svakog *sprinta* razvojni tim isporučuje do tada završeni dio proizvoda. Razdvajanje razvojnog procesa na sprintove uvelike smanjuje rizik od isporuke lošeg softvera te se lakše može prilagoditi promjenama i zahtjevima korisnika. Svaki pojedini *sprint* se razdvaja na manje dijelove koji traju po 24 sata odnosno započinju i završavaju Dnevnim SCRUM-om. Dnevni SCRUM je obveza cijelog razvojnog tima da održavaju sastanke, a za takve sastanke je karakteristično da nema stolica, sudionici stoje i pišu svoje ideje na ploči, ograničeno vrijeme sastanka je 15 minuta, a svrha sastanaka je odgovoriti na pitanja: Što si jučer napravio? Što ćeš danas napraviti? Koje su ti prepreke na putu? Takvim načinom procesa se razvojnom timu lakše prilagoditi promjenama u dinamičnoj okolini koja kontinuirano utječe na razvoj softvera ili informacijskog sustava. Cho (2008) u svom radu navodi kako u SCRUM-u postoje 3 različite uloge procesa: Vlasnik proizvoda, Tim i Scrum Master. **Vlasnik proizvoda** odgovoran je za dobivanje početnih i tekućih sredstava za projekt do stvaranje ukupnih zahtjeva projekta, Povrat ulaganja (ROI) i plan objavljivanja. **Tim** je odgovoran za provedbu funkcionalnosti opisanom u zahtjevima. Timovi trebaju biti samoupravno, samoorganizirano i višestruko funkcionalni za maksimaliziranje učinka cijelog tima. Za uspjeh su zaslužni svi članovi tima. **Scrum Master** (SM) je odgovoran da osigura vrijednosti, prakse i pravila te jesu li donesene i provedene. SM predstavlja uprava i tim međusobno. SM također pokušava ukloniti sve prepreke nametnute programerima.

Kada se govori o samom procesu SCRUM metode, on uključuje 3 faze koje su prikazane slikom 9, a one su: faza „prije igre (*engl. Pregame phase*), faza razvoja – *Sprint* (*engl. Development phase*) i faza „poslije igre“ (*engl. Postgame phase*)



Slika 9. SCRUM proces

Izvor: Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*.

Abrahamsson i dr. (2017), objašnjavaju sljedeće faze:

- **Faza „prije igre“** (*engl. Pregame phase*) uključuje dvije podfaze: planiranje i arhitektura / dizajn visoke razine.

Planiranje uključuje definiciju sustava koji se razvija. Stvara se popis zaostalih proizvoda koji sadrži sve trenutno poznate. Zahtjevi mogu potjecati od odjela za kupce, prodaje i marketinga, korisničke podrške ili programera softvera. Zahtjevi imaju prioritet i procjenjuju se napori potrebni za njihovu provedbu. Popis zaostalih proizvoda stalno se ažurira novim i detaljnijim stavkama, kao i točnijim procjenama i novim prioritetnim narudžbama. Planiranje također uključuje definiranje projektnog tima, alata i drugih resursa, procjenu rizika i pitanja kontrole, potrebe za obukom i odobrenje uprave za verifikaciju. U svakoj iteraciji, SCRUM tim (ovi) pregledavaju zaostale proizvode kako bi stekli svoju predanost sljedećoj iteraciji.

U fazi arhitekture, planiranje visoke razine sustava, uključujući arhitekturu, planira se na temelju trenutnih stavki u „Zaostalim proizvodima“. U slučaju poboljšanja postojećeg sustava, promjene potrebne za implementaciju zaostalih stavki identificiraju se zajedno s problemima koje mogu uzorkovati. Održava se sastanak za pregled dizajna

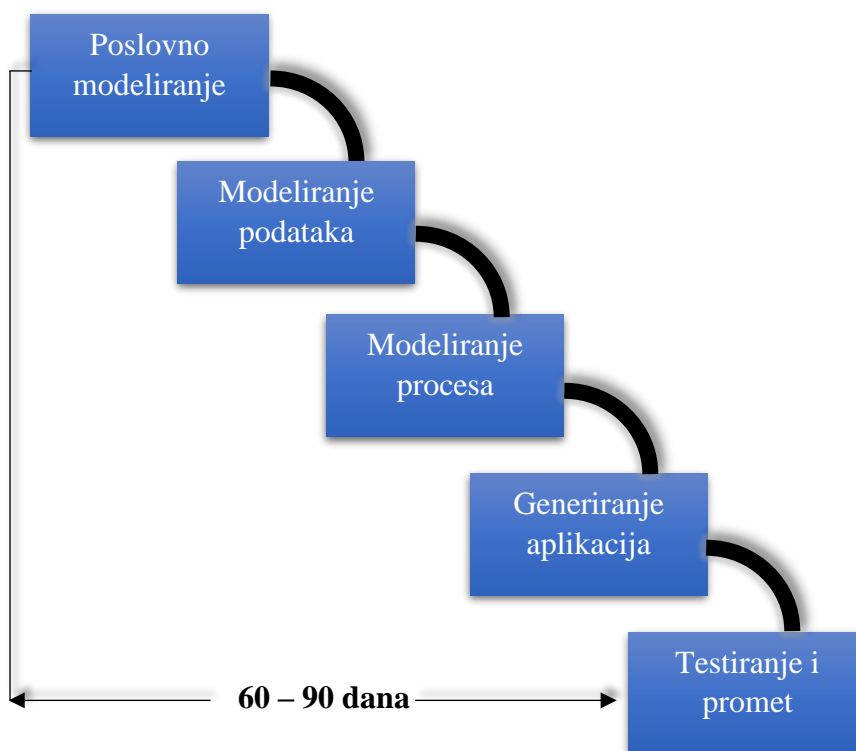


radi razmatranja prijedloga za provedbu i odluke se donose na temelju tog pregleda. Osim toga, pripremaju se preliminarni planovi za sadržaj izdanja.

- **Razvojna faza (engl. *Development phase*)**, koja se naziva i fazom igre, je agilni dio SCRUM pristupa. Ova se faza tretira kao "crna kutija" ("*Black box*") u kojoj se očekuje nepredvidivo. Različite okolišne i tehničke varijable (poput vremenskog okvira, kvalitete, zahtjeva, resursa, implementacijske tehnologije i alata, pa čak i razvojne metode) identificirane u SCRUM-u, koje se mogu promijeniti tijekom procesa, promatraju se i kontroliraju kroz različite SCRUM prakse tijekom *Sprintova*. Umjesto da te stvari uzme u obzir tek na početku projekta razvoja softvera, SCRUM ima za cilj njihovu stalnu kontrolu kako bi se mogao fleksibilno prilagoditi promjenama.  
U fazi razvoja sustav se razvija u *Sprintovima*. *Sprintovi* su iterativni ciklusi u kojima se funkcionalnost razvija ili poboljšava za stvaranje novih koraka. Svaki *Sprint* uključuje tradicionalne faze razvoja softvera: zahtjevi, analiza, dizajn, evolucija i isporuka. Arhitektura i dizajn sustava razvijaju se tijekom razvoja *Sprinta*. Planirano je da jedan *Sprint* traje od tjedan dana do mjesec dana. Na primjer, može postojati tri do osam *Sprinteva* u jednom procesnom razvoju prije nego što sustav bude spreman za distribuciju. Također može postojati više od jednog tima za povećanje prirasta.
- **Faza „poslije igre“ (engl. *Postgame phase*)** jest zatvaranje projekta. U novu fazu se ulazi kada je postignut dogovor da su varijable iz okoline poput zahtjeva, ispunjeni. U ovom slučaju više se ne mogu pronaći stavke i problemi niti se mogu izmisliti bilo kakvi novi zahtjevi. Sustav je sada spreman za isporuku i pripremu, te uključuje zadatke kao što su integracija, ispitivanje sustava i dokumentacija.

#### 4.2.2. Model brzog razvoja aplikacija

Model brzog razvoja aplikacija (*engl. Rapid application model - RAD*) prvi je puta predložio IBM davnih 1980-ih godina. Krična značajka primjene ovog modela kod razvoja softvera je upotreba moćnih razvojnih alata i tehnika. Softverski projekt može se implementirati pomoću RAD modela ako se može rastaviti na manje module kojima se svaki modul može neovisno dodijeliti zasebnim timovima, a da se ti moduli mogu kombinirati u konačni proizvod. „Model RAD (brzi razvoj aplikacija) temelji se na prototipiranju i iterativnom razvoju bez posebnog planiranja. Sam postupak pisanja softvera uključuje planiranje potrebno za razvoj proizvoda“ (Tutorialspoint, 2021). Prototip je radni model koji je funkcionalno ekvivalentan komponenti gotovog zamišljenog proizvoda. Brzi razvoj aplikacija se fokusira na prikupljanje zahtjeva kupaca putem određenih radionica ili fokus grupa, rano testiranje izrađenog prototipa od strane kupaca pomoću iterativnog koncepta, ponovnu upotrebu postojećih prototipova, dok je manji fokus na planiranju. Ovaj model, smanjujući vrijeme planiranja i naglašavajući ponavljanje prototipa, omogućuje voditeljima projekata te svim dionicima da točno mjere napredak i komuniciraju u stvarnom vremenu o novim problemima, preprekama ili promjenama što rezultira većom učinkovitošću, bržim razvojem i učinkovitom komunikacijom. Prema Agarwal (2000), najvažnije karakteristike RAD alata uključuju sposobnost za planiranje, modeliranje podataka i procesa, generiranje koda te ispitivanje i uklanjanje pogrešaka. Što se tiče timova unutar ovog modela, oni se obično sastoje od pojedinaca različitih vještina, programera, stručnjake za modeliranje podataka i razvojne stručnjake. Što se tiče funkcionalnosti modula, oni se razvijaju paralelno kao prototipovi i integrirani su kako bi se postigao cjeloviti proizvod i brže isporučio. Kada je riječ o samom procesu RAD modela, on je prikazan slikom 10, a dijelovi procesa su: poslovno modeliranje, modeliranje podataka, modeliranje procesa, generiranje procesa, generiranje aplikacija, ispitivanje i promet.



Slika 10. Dijagram modela RAD

Izvor: Prevedeno i prilagođeno od strane autora ovog rada prema Singh, V. (2020). What is Rapid Application Development Model (RAD)?. Dostupno na: <https://hackr.io/blog/rapid-application-development-model> [pristupljeno 13. rujna 2021.]

RAD model distribuira i faze analize, projektiranja, razvoja i ispitivanja, a glavnih šest faza su (Tutorialspoint, 2021):

- **Poslovno modeliranje** - poslovni model za proizvod u razvoju osmišljen je u smislu protoka informacija i distribucije informacija između različitih poslovnih kanala. Provodi se cjelovita poslovna analiza kako bi se pronašle vitalne informacije za poslovanje, kako ih se može dobiti, kako i kada se informacije obrađuju i koji su faktori koji pokreću uspješan protok informacija.
- **Modeliranje podataka** - podaci prikupljeni u fazi poslovnog modeliranja pregledavaju se i analiziraju kako bi se oblikovali skupovi podatkovnih objekata vitalnih za poslovanje. Identificirani su i definirani atributi svih skupova podataka. Odnos između

ovih podatkovnih objekata uspostavljen je i detaljno definiran u odnosu na poslovni model.

- **Modeliranje procesa** - skupovi podataka definirani u fazi modeliranja podataka pretvaraju se kako bi se uspostavio tok poslovnih informacija potreban za postizanje određenih poslovnih ciljeva prema poslovnom modelu. U ovoj fazi definiran je procesni model za bilo kakve promjene ili poboljšanja skupova objekata podataka. Dani su opisi procesa za dodavanje, brisanje, dohvaćanje ili izmjenu podatkovnog objekta.
- **Generiranje aplikacija** - stvarni sustav izgrađen je i kodiranje se vrši pomoću alata za automatizaciju za pretvaranje modela procesa i podataka u stvarne prototipove.
- **Ispitivanje i promet** - ukupno se vrijeme testiranja smanjuje u modelu RAD jer se prototipovi neovisno testiraju tijekom svake iteracije. Međutim, protok podataka i sučelja između svih komponenata moraju se temeljito testirati s potpunim pokrivanjem ispitivanja. Budući da je većina programskih komponenata već testirana, to smanjuje rizik od bilo kakvih većih problema.

Ova metodologija se može koristiti kada sustav treba izraditi u kratkom vremenskom periodu od 2 do 3 mjeseca, kada su svi zahtjevi poznati, kada će korisnik biti uključen tijekom cijelog životnog ciklusa, kada je tehnički rizik manjim, kada postoji neka potreba za stvaranjem sustava koji se može modulariti također za 2 do 3 mjeseca te kada je proračun dovoljno visok da se mogu priuštiti dizajneri za modulariranje zajedno sa svim troškovima automatiziranih alata za generiranje koda.

#### 4.2.3. Kanban

Kanban pristup razvio je krajem četrdesetih godina prošlog stoljeća, inženjer Taiichi Ohno, za Toyotu (Japanska automobilska industrija). Stvoren je kao jednostavan sistem planiranja, čiji je cilj bio optimalni nadzor i upravljanje radom i zalihama u svakoj fazi *lean* proizvodnje. „Kanban je Japanska riječ koja znači natpis i koristi se u proizvodnja kao sustav rasporeda“ (Ahmad i dr., 2013). To je kontrola protoka odnosno mehanizam za povlačenje pogona Just –

in – time, u kojoj uzlazne aktivnosti obrade pokreću signalni zahtjevi za nizvodnim procesom. Ključni razlog za razvoj Kanban metodologije bila je neadekvatna produktivnost i efikasnost Toyote u usporedbi sa američkim automobilskim divovima. Pomoću Kanban metodologije, Toyotin sistem kontrole proizvodnje je postao fleksibilan i efikasan te je povećao produktivnost, istovremeno smanjujući trošak sirovina, poluproizvoda i gotovih proizvoda. S vremenom je Kanban postao i efikasan način upravljanja raznim proizvodnim sistemima jer je jedan od najjednostavnijih okvira koji se koristi, a omogućava projektnim menadžerima da efikasno upravljaju i prate svoje projekte. Kanban metodologija je do nedavno bila korištena u mnogim proizvodnim sustavima, montažnim sustavima i sustavi lanaca opskrbe na vrlo učinkovit način. To je sustav rasporeda rada koji maksimizira produktivnost tima uz smanjenje vremena praznog hoda. Vrijeme praznog hoda može se dogoditi unutar bilo kojeg procesa, tijekom rada ili postupka. Kako se Kanban koristio u raznim industrijama i sustavima pa se tako počeo koristiti i kod izrade softverskih rješenja. „Kanban je metoda za razvoj softverskih proizvoda i procesa s naglaskom na *Just-in-time* isporuku dok se pri tome ne opterećuje softver developere. Naglasak je na tome da developeri izvlače posao iz redova i proces je do isporuke korisniku vidljiv svima koji sudjeluju u razvoju“ (Info Novitas, 2021). Zapravo formalni počeci primjene kanban pristupa su povezani sa J. Andresonom kada je pomogao Microsoftu u postizanju bolje vizualizacije protoka rada i ograničavanja tijekom rada. S obzirom na uspješnost dokazano je pet temeljnih načela, a Kirovska i Koceski (2015) kako počinje proces uspješnoj usvajanja tih načela:

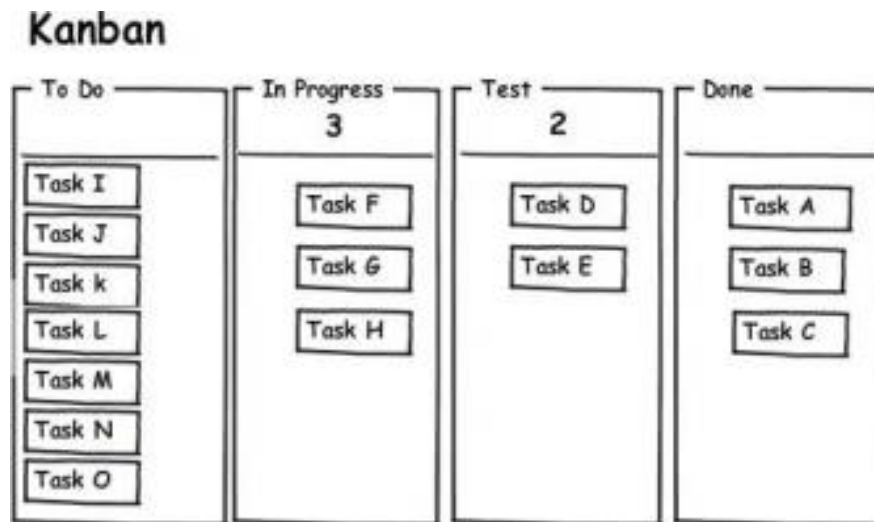
- Početi s onim što imate sada - to je vaš trenutni postupak
- Dogovorite kako slijediti evolucijski pristup promjenama i poboljšanjima
- Poštujte trenutne uloge i odgovornosti tima / organizacije

Na temelju tih principa sljedeće što treba implementirati su pet glavnih Kanban načela koje vode do uspjeha, a to su (Kirovska i Koceski, 2015):

1. Vizualizacija posla i tijek rada koji slijedi
2. Ograničavanje rada u tijeku (WIP) pomoću virtualnog Kanban sustava
3. Upravljanje protokom
4. Neka politike upravljanja budu eksplicitne
5. Koristiti modele i znanstvenu metodu te poboljšavati suradnju

U agilnom načinu razvoja softvera postalo je česta praksa da se napredak na projektu vizualizira i da bude dostupan svima na zidu u obliku ploče na kojoj se nalaze zadaci, stoga je prednost

Kanban-a da može poslužiti za vizualizaciju u bilo kojoj metodi razvoja, poput RUP (*Rational Unified Process*) ili Waterfall. Primjer vizualizacije Kanban ploče prikazan je slikom 11.



Slika 11. Primjer Kanban ploče

Izvor: Kirovska, N., & Koceski, S. (2015). Usage of Kanban methodology at software development teams. *Journal of applied economics and business*, 3(3), 25-34.

Kanban ploča je alat koji vizualizira čitav tijek projekta i omogućava da bilo koji novi član ili vanjski entitet može razumjeti što se trenutno događa te koji su zadaci izvršeni, a koji se tek trebaju realizirati. Kanaban sistemi su jednostavni po dizajnu i fleksibilni po prirodi zbog jednostavnog definiranja koraka procesa u kolone i dokumentiranjem radnih zadataka. Funkcionalnost kanban ploče ogleda se u jednostavnosti i sposobnosti da se projekt prikaže na praktičan način. Svaka kolona je faza u projektu i tako daje članovima tima i projektnom menadžeru jasan pregled odmah na prvi pogled. Timovi mogu vidjeti na kojim su zadacima dodijeljeni, a menadžer mogu pratiti napredak projekta. Project management Srbija (2021) navodi osnovne elemente Kanban ploče i detaljnija objašnjenja, a oni su sljedeći:

- **Kolone** – koje predstavljaju definirane korake u procesu
- **Kartice** – koje predstavljaju radne zadatke koji se kreću kroz proces

Kanban ploča ukazuje na:

- trenutne zadatke koji se izvršavaju
- zadatke koje treba obaviti u budućnosti
- zadatke koji su završeni

Podijeljene kolone su međusobno povezane i zadaci se postepeno povlače iz krajnje lijeve kolone (budući zadaci) u krajnju desnu kolonu (izvršeni zadaci). Kanban sistem mjeri radni ciklus koji se završava principom *Work in Progress* (WIP). WIP ima određena ograničenja i unaprijed definirani specifični status. Ograničavanje WIP-a radi održavanja dosljednih standarda jedan je od osnovnih principa koji upravljaju Kanban metodologijom. Za sam tim je izuzetno važno da tekuće zadatke izvršava po propisanom redosljedju. Ova metodologija zahtjeva stroga WIP ograničenja obujma nedovršenog posla u bilo kojem trenutku. Timovi dodjeljuju ograničenje broju kartica u bilo kojoj koloni aktivnog rada. Kada je ograničenje ispunjeno, niti jedan novi rad ne može ući u novu kolonu dok se prethodni zadatak ne dovrši i premjesti u sljedeću kolonu. Kanban kolone kako je ranije navedeno predstavljaju način za razbijanje različitih faza procesa, a kartice su organizirane pod naslovima kolona i prevlače se u sljedeću kolonu desno kako bi se naznačilo gdje su u proizvodnom ciklusu. Kanban kartice se postavljaju na ploču i svaka predstavlja individualni zadatak. Svaka je kartica ispunjena informacijama vezanim uz određeni zadatak (kratak naziv i opis). Zadaci će također biti dodijeljeni članovima tima koji će biti odgovorni za izvršenje do krajnjeg roka. Na prednjoj strani kartice potrebno je unijeti detalje koji olakšavaju razumijevanje svakom tko promatra plan projekta, a ti detalji su: naziv, dodjela kartice, tip kartice i datum dospijea., dok su informacije Kanban kartice: opis, prilozi ili datoteke te povijest komentara.

#### **4.2.4. Ekstremno programiranje**

Budući da softverski inženjering postoji u dinamičnom okruženju, tradicionalni pristupni upravljanju projektima više nisu održivi u tolikoj mjeri, što znači da IT stručnjaci moraju pronalaziti nove načine rješavanja razvojnih zadataka. Ekstremno programiranje je jedna od najpoznatijih agilnih metoda u svijetu, a njeni počeci su utemeljeni 1990. godine. Temeljena je tada na dobro poznatoj metodologiji vodopada (*engl. Waterfall*) iako su se razlikovale po nekim karakteristikama. Vodopadna metodologija je poznata po predvidljivosti, a ekstremno programiranje to nije podržavala već se fokusirala na adaptibilnost. Ekstremno programiranje (*Extreme Programming - XP*) razvilo se iz problema uzrokovanih dugotrajnim razvojnim ciklusima tradicionalnih razvojnih modela (Beck, 1999 navedeno u Abrahamsson i dr., 2017). XP je jedan od brojnih agilnih okvira koje primjenjuju IT tvrtke, a njegova ključna značajka je na tehničkim aspektima razvoja softvera po čemu se i razlikuje od ostalih pristupa. „XP je skup inženjerskih praksi. Programeri moraju tijekom svojih postupaka ići dalje od svojih

možnosti. Otuda potječe "krajnost" u nazivu okvira" (Altexsoft, 2021). Kako bi se bolje razumjelo na koji način funkcionira Ekstremno programiranje prilikom razvoja softverskog rješenja, aktivnosti su sljedeće (Altexsoft, 2021):

1. **Planiranje**, prva faza, je kada se kupac sastane s razvojnim timom i prezentira zahtjeve u obliku korisničkih priča kako bi opisao željeni rezultat. Tada tim procjenjuje priče i kreira plan izdanja podijeljen na iteracije potrebne za pokrivanje potrebne funkcionalnosti dio po dio. Ako se jedna ili više priča ne može procijeniti, mogu se uvesti takozvani *skokovi*, što znači da su potrebna daljnja istraživanja.
2. **Dizajn** je zapravo dio procesa planiranja, ali se može izdvojiti kako bi se naglasila njegova važnost. Povezan je s jednom od glavnih XP vrijednosti koja će se razmotriti u nastavku - jednostavnošću. Dobar dizajn donosi logiku i strukturu u sustav i omogućuje izbjegavanje nepotrebnih složenosti i suvišnosti.
3. **Kodiranje** je faza tijekom koje se stvarni kôd stvara primjenom specifičnih XP praksi kao što su standardi kodiranja, programiranje parova, kontinuirana integracija i kolektivno vlasništvo nad kodom.
4. **Testiranje** je srž ekstremnog programiranja. Redovita je aktivnost koja uključuje i jedinična ispitivanja (automatizirano testiranje kako bi se utvrdilo radi li razvijena značajka ispravno) i testove prihvatanja (ispitivanje kupaca kako bi se potvrdilo da je cjelokupni sustav stvoren u skladu s početnim zahtjevima).
5. **Slušanje** se sastoji od stalne komunikacije i povratnih informacija. Kupci i voditelji projekata uključeni su u opisivanje poslovne logike i vrijednosti koja se očekuje.

Prethodno pojašnjene aktivnosti kroz životni ciklus razvoja softverskog rješenja primjenom ovog modela prikazane su slikom 12.





Slika 12. Životni ciklus projekta primjenom Ekstremnog programiranja

Izvor: Info Novitas (2021). Ekstremno programiranje (XP). Dostupno na: <https://www.info-novitas.hr/ot-nama/metodologije-rada/ekstremno-programiranje-xp/> [pristupljeno 13. srpnja 2021.]

Iako je pet navedenih faza one se mogu, prema Hutagalung (2006), a kako je navedeno u Beram i dr.(2016), sažeti u četiri glavne aktivnosti: kodiranje, testiranje, slušanje i dizajniranje. Takav razvojni proces podrazumijeva suradnju između nekoliko sudionika koji imaju svoje uloge, zadatke i odgovornosti. Ekstremno programiranje u fokus stavlja ljude ističući vrijednost i važnost socijalnih vještina kao što su komunikacija, suradnja, reagiranje i povratne informacije. Uloge koje su obično povezane sa ekstremnim programiranjem su (Beck, 2000, navedeno u Abrahamsson i dr., 2017):

- **Programer** - programeri pišu testove i održavaju programski kod jednostavnim i određenim. Prvo pitanje koje XP čini uspješnim je komunikacija i koordinacija s ostalim programerima i članovima tima.
- **Kupac** - kupac piše priče i funkcionalne testove i odlučuje kada će se neki zahtjev zadovoljiti. Kupac postavlja prioritet zahtjeva za implementacijom.
- **Tester** - testeri pomažu kupcu da napiše funkcionalne testove. Provede funkcionalne testove redovito emitirajte rezultate ispitivanja i održavajte alate za testiranje.
- **Trakeri ili upravitelji** - traker daje povratne informacije u XP-u. Ulazi u trag procjenama tima (npr. procjene napora) i daje povratnu informaciju o tome koliko su točne te buduće procjene za poboljšanje. Također prati napredak svake iteracije i procjenjuje je li cilj postignut u danim resursima i vremenskom ograničenju ili ako su u procesu potrebne bilo kakve promjene

- **Trener** - trener je osoba odgovorna za proces u cjelini. Razumijevanje XP-a je važno u ovoj ulozi, omogućujući treneru da vodi ostale članovi tima tijekom postupka.
- **Konzultant** - konzultant je vanjski član koji posjeduje specifična tehnička znanja koja su potrebna. Savjetnik vodi tim pri rješavanju njihovih specifičnih problema.
- **Menadžer** - menadžer donosi odluke. Da bi to mogao, on komunicira s projektnim timom kako bi utvrdio trenutnu situaciju i razlikovao bilo koju poteškoću ili nedostatak u procesu.

Krajem 90-ih godina, Beck je sažeo skup određenih vrijednosti i principa koji mogu na najjednostavniji način opisati ekstremno programiranje te do dovode do učinkovitije suradnje unutar projektnog tima, a u konačnici i kvalitetnijeg proizvoda. Pa su vrijednosti ekstremnog programiranja sljedeće (Altexsoft, 2021):

1. **Komunikacija.** Svi u timu rade zajedno u svakoj fazi projekta.
2. **Jednostavnost.** Programeri se trude pisati jednostavan kôd koji donosi veću vrijednost proizvodu jer štedi vrijeme i trud.
3. **Povratne informacije.** Članovi tima često isporučuju softver, dobivaju povratne informacije o njemu i poboljšavaju proizvod u skladu s novim zahtjevima.
4. **Poštovanje.** Svaka osoba dodijeljena projektu doprinosi zajedničkom cilju.
5. **Hrabrost.** Programeri objektivno procjenjuju vlastite rezultate bez opravdanja i uvijek su spremni odgovoriti na promjene.

Te vrijednosti predstavljaju specifičan način razmišljanja motiviranih timskih igrača koji daju sve od sebe na putu do postizanja zajedničkog cilja. Načela XP proizlaze iz ovih vrijednosti i odražavaju ih na konkretniji način. Načela ekstremnog programiranja su (Artexsoft, 2021):

1. Brza povratna informacija. Članovi tima razumiju dane povratne informacije i odmah reagiraju na njih.
2. Pretpostavljena jednostavnost. Programeri se trebaju usredotočiti na posao koji je trenutno važan i slijediti načela YAGNI (nećete trebati) i SUHI (ne ponavljajte se).
3. Inkrementalne promjene. Male promjene napravljene na proizvodu korak po korak djeluju bolje od velikih odjednom.
4. Prihvaćajući promjene. Ako klijent misli da proizvod treba promijeniti, programeri bi trebali podržati ovu odluku i planirati kako primijeniti nove zahtjeve.
5. Kvalitetan rad. Tim koji dobro radi, stvara vrijedan proizvod i osjeća se ponosnim na njega.

#### 4.2.5. Crystal

*Crystal* metodologija je obitelj metodologija koja uključuje niz raznih metodologija za odabir najprikladnije metodologije za svaki pojedini projekt. Osim prilagodbe metodologije, ovaj pristup uključuje i principe za prilagođavanje metodologija u skladu sa raznim okolnostima na različitim projektima. „*Crystal* je agilni okvir koji se fokusira na pojedince i njihove interakcije, za razliku od procesa i alata. Drugim riječima, ovaj je okvir izravni izdanak jedne od temeljnih vrijednosti artikuliranih u Agilnom Manifestu“ (ProductPlan, 2021). Izvorno prema Cockburn, navedeno u Chang (2010), *Crystal* je obitelj metodologija (Čisto, Žuta, Narančasta i Crvena), a temelji se na veličini projekta (gdje je broj ljudi koji su uključeni oko 1-6, 1-20, 1-40, 1-100) i kritičnosti (tamo gdje bi nedostaci mogli prouzročiti gubitak direktnog novca, osnovnog novca ili života). Kada se radi o veličini tima, kada raste, implementiranje *Crystal* metodologije mijenja se kako bi se strukturi dodalo više formalnosti upravljanja projektom. Kritičnost projekta također povećava složenost projekta te je potrebno osigurati ispunjenje očekivanih zahtjeva. Ova metodologija također i prepoznaje kako svaki projekt može zahtijevati prilagođeni skup politika, praksa i procesa koji udovoljavaju jedinstvenim karakteristikama projekta. Kada govorimo o članovima obitelji *Crystal*, svaki pojedini član je obojan različitom bojom koja označava „težinu“ metodologije, drugim riječima što je boja tamnija to je teža metodologija. Ova metodologija predlaže odabir odgovarajuće boje za projekt temeljen na njegovoj veličini i kritičnosti (prikazano na slici 13).

	L6	L20	L40	L80
E6	E6	E20	E40	E80
D6	D6	D20	D40	D80
C6	C6	C20	C40	C80
Clear	Clear	Yellow	Orange	Red

Slika 13. Dimenzije *Crystal* metodologije

Izvor: Cockburn (2002) navedeno u Abrahamsson i dr. (2017)

Što je sustav koji se razvija kritičniji to je potrebna veća i strogost. „Simboli na slici 13 označavaju potencijalni gubitak uzrokovan kvarom sustava (tj. razina kritičnosti): Udobnost (C), Diskrecijski novac (D), bitan novac (E) i život (L)“ (Cockburn, 2002a, navedeni u Abrahamsson i dr., 2017). Drugim riječima, razina kritičnosti C označava da pad sustava zbog nedostataka uzrokuje gubitak udobnosti za korisnika dok pogreške u sustavu kritičnom za život mogu doslovno prouzročiti gubitak života sustava.

Ovaj model izrađen je na dva temeljna pristupa, a to su: timovi mogu sami pronaći načine kako poboljšati i optimizirati svoje tijekove rada i svaki je projekt jedinstven i uvijek se mijenja, zato je tim tog projekta najprikladniji za određivanje načina na koji će se baviti poslom. Postoje određena pravila, značajke i vrijednosti koje su zajedničke svim metodama u obitelji *Crystal*. Projekti se uvijek koriste inkrementalnim razvojnim ciklusom s maksimalnom duljinom prirasta od 4 mjeseca, ali po mogućnosti između 1 i 3 mjeseca. Naglasak se stavlja na komunikaciju i suradnju unutar projektnog tima. Trenutno su izgrađene tri glavne *Crystal* metodologije: *Crystal Clear*, *Crystal Orange* i *Crystal Orange Web* (Cockburn 2002a, navedeno u Abrahamsson i dr. 2017). Kada je riječ o karakteristikama *Crystal* metodologije, one uključuju:

- ***ljudsku snagu*** gdje je fokus na postizanju uspjeha u projektu kroz poboljšanje rada uključenih ljudi (mogu biti i druge metodologije usmjeren na proces, na arhitekturu ili na alat, ali *Crystal* jest usmjeren na ljude).
- ***ultralight*** gdje za bilo koju veličinu projekta i prioriteta, metodologija *Crystal* obitelji za projekt će raditi na smanjenju papirologije, režijskih troškova i birokracije u najmanju ruku što je praktično za parametre tog projekta.
- ***istegnite se kako biste se uklopili*** u ono gdje započinjete s nečim što je manje nego što mislite da trebate i uzdižite se tek toliko da dobijete potrebnu veličinu (Cockburn, 2008, navedeno u Chang, 2010)

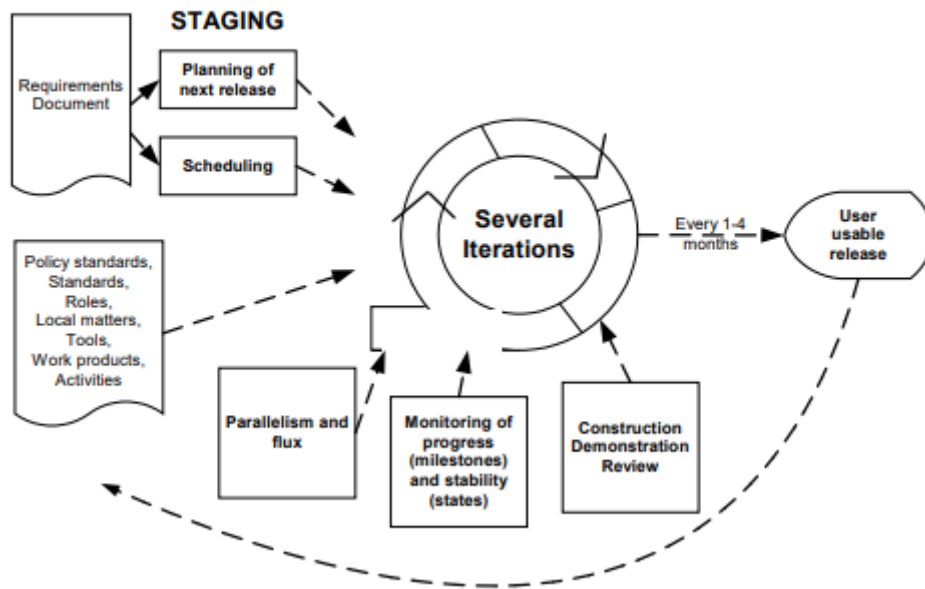
Kada je riječ o procesu *Crystal* metodologije, svaki član *Crystal* obitelji daje smjernice politike, standarada, proizvoda za rad, lokalna pitanja, alate i uloge koje je potrebno slijediti u procesu razvoja. *Crystal Clear* i *Crystal Orange* su dva člana obitelji *Crystal* koji su konstruirani i korišteni (Cockburn 1998; Cockburn 2002a, navedeno u Chang, 2010). *Crystal Orange* predstavlja aktivnosti promatrajući njihov kontekst, sličnosti i razlike te ilustrirajući cijeli proces i njegove aktivnosti,. Također, *Crystal Orange* dizajniran je za srednje velike projekte s

ukupno od 10 do 40 članova projektnog tima te sa trajanjem od 1 do 2 godine. *Crystal Clear* dizajniran je za vrlo male projekte koji obuhvaćaju samo šest programera te bi se tim za razvoj projekta trebao nalaziti u istom prostoru zbog ograničenosti u međusobnoj komunikacijskoj strukturi. Standarde i politike koji se moraju primijeniti tijekom razvojnog postupka, prema Cockburn (2002a), a kako je navedeno u Abrahamsson i dr. (2017), sugeriraju sljedeće standarde i politike:

- Inkrementalna dostava u regularnoj osnovi
- Praćenje napretka prema *milestonesima* na temelju isporuke softvera i glavnih odluka, a ne pisanjem dokumenata
- Izravno sudjelovanje korisnika
- Automatizirano regresijsko ispitivanje funkcionalnosti
- Dva korisnička pregleda po realizaciji
- Radionice za podešavanje proizvoda i metodologija na početku i tijekom svakog rasta

Jedina razlika između standarda politike ove dvije metodologije je da *Crystal Clear* predlaže postupnu isporuku u roku od 2 do 3 mjeseca, dok se u *Crystal Orange* koraci mogu produljiti na maksimalno 4 mjeseca. Standardi i politike ovih metodologija su obvezni, ali se mogu i mijenjati ekvivalentnim praska drugih metodologija kao što su *Extreme Programming* i SCRUM. Kada je riječ o ulogama i odgovornosti unutar tima potrebno je prvenstveno naglasiti kako su timovi unutar *Crystal Clear* i *Crystal Orange* različiti. Osnovna razlika između navedene dvije metodologije je činjenica da u *Crystal Clear* projektu postoji samo jedan tim, dok u *Crystal Orange* postoji više timova koje treba pratiti kroz projekt. U obe metodologije, jedan zadatak može sadržavati više uloga. U *Crystal Clear* glavne uloge imaju: sponzor, stariji dizajner – programer, dizajner – programer i korisnik. Ove uloge se još mogu dijeliti na poduloge kao npr. dizajner-programer sastoji se od dizajnera poslovne klase, programera, dokumentatora softvera i tester (Cockburn 1998 navedeno u Abrahamsson i dr. 2017). *Crystal Orange* predstavlja nekoliko novih uloga, kao što su dizajner korisničkog sučelja, dizajner baza podataka, stručnjak za upotrebu, tehnički voditelj, poslovni analitičar / dizajner, arhitekt, mentor dizajna, točka ponovne upotrebe, pisac i tester (Cockburn 1998; Cockburn 2002a, navedeno u Abrahamsson i dr. 2017).

## ONE INCREMENT



Slika 14. Primjer Crystal Orange procesa

Izvor: Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis.

Razvoj informatičkih projekata korištenjem ove metode uključuju razne aktivnosti kao što su inscenacija, praćenje, pregledavanja te paralelizam i tok. Također, javljaju se i druge aktivnosti koje mogu biti identificirane i uključene u proces, a one su (Abrahamsson i dr. 2017):

- **Inscenacija** - uključuje planiranje sljedećeg povećanja sustava. Trebalo bi biti planirano da se maksimalno proizvede radno izdanje u svaka 3 ili 4 mjeseca. Poželjan je raspored od 1 do 3 mjeseca. Tim odabire zahtjeve koji će se implementirati u povećanje i zakazati ono što smatraju da su u stanju isporučiti.
- **Revizija i pregled** - svaki prirast uključuje nekoliko ponavljanja. Svaka iteracija uključuje sljedeće aktivnosti: izgradnja, demonstracija i pregled ciljeva povećanja.
- **Praćenje** - napredak se prati u pogledu rezultata tima tijekom razvojnog procesa s obzirom na njihov napredak i stabilnost. Napredak se mjeri prekretnicama (početak, pregled 1, pregled 2, testiranje, isporuka) i fazama stabilnosti (vrlo fluktuirajuće, fluktuirajuće i dovoljno stabilno za pregled). Nadzor je potreban i u *Crystal Clear* i *Crystal Orange*.

- **Paralelizam i tok** - jednom kada praćenje stabilnosti daje rezultat "dovoljno stabilan za pregled" rezultata, sljedeći zadatak može započeti. U *Crystal Orange* to znači da višestruki timovi mogu uspješno nastaviti s maksimalnim paralelizmom. Kako bi to osigurali, nadzorni i arhitektonski timovi pregledavaju svoje planove rada, stabilnost i sinkronizaciju.
- **Strategija cjelovite raznolikosti** - *Crystal Orange* uključuje metodu koja se naziva holistička strategija raznolikosti za razdvajanje velikih funkcionalnih timova u višefunkcionalne skupine. Središnja je ideja uključiti više stručnjaka u jedan tim. Strategija holističke raznolikosti također omogućuje formiranje malih timova s potrebnim posebnim znanjem i radom, a također razmatra pitanja poput lociranja timova, komunikacije i dokumentacije i koordinacije više timova.
- **Tehnika podešavanja metodologije** - jedna je od osnovnih tehnika *Crystal Clear* i *Orange*. Koristi projektne razgovore i timske radionice za izradu specifične *Crystal* metodologije za svaki pojedini projekt. Jedna od središnjih ideja postupnog razvoja jest omogućiti popravljavanje ili poboljšanje procesa razvoja. U svakom koraku projekt može naučiti i koristiti stečeno znanje za razvoj procesa za sljedeći priraštaj.
- **Pregledi korisnika** - za *Crystal Clear* se preporučuju dva pregleda korisnika po jednom izdanju. U *Crystal Orange*, recenzije korisnika trebale bi se organizirati tri puta za svaki prirast.
- **Radionice refleksije** - *Crystal Clear* i *Orange* uključuju pravilo da bi tim trebao održavati radionice refleksije prije i nakon koraka (s preporukom i za radionice refleksije sa srednjim korakom) *Crystal Clear* i *Crystal Orange* ne definiraju nikakve posebne prakse ili tehnike koje projekt treba koristiti u njihovim zadacima razvoja softvera. Usvajanje praksi iz drugih metodologija, poput XP-a i SCRUM-a, dopušteno je u *Crystal* da zamijeni neke vlastite prakse, kao što su, na primjer, radionice razmišljanja

## 5. Zaključak

U vremenu složenih projekata i projekata velikih razmjera, kvaliteta vođenja i upravljanja projektima dobiva na iznimnoj važnosti. Kroz povijest stručnjaci sa područja projektnog menadžmenta dolazili su do novih rješenja kako efikasnije i kvalitetnije upravljati projektima, iskoristiti sve tehnike i alate u svrhu ostvarivanja svih postavljenih ciljeva u određenom roku na zadovoljstvo svih dionika. Kvalitetno vođenje projekata bilo koje vrste postaje sve značajnije u poduzećima u gotovo svim granama industrije neovisno o veličinama odnosno neovisno o tome radi li se o malom, srednjem ili velikom poduzeću. Nagle i intenzivne tehnološke promjene odrazile su se na poslovne organizacije, a posljedica toga je razvoj kompleksnih poslovnih procesa što donosi i veći broj informatičkih rješenja. Investicije u efikasna i kvalitetna informatička rješenja sve više rastu pa se tako povećava potreba i za kvalitetnim upravljanjem informatičkim projektima. Razvijene su razne metodologije kojima se definiraju temeljna pravila, faze i procesi kako na najefikasniji način pristupiti takvoj vrsti projekta, koje u konačnici donose novi i kvalitetan proizvod ili uslugu. Sve metode koje imaju temelj u vodopadnom modelu pripadaju tradicionalnom pristupu. To su uglavnom metode u kojima se projekt raspoređuje prema slijednim fazama, a fokus je na procesima i ostvarenju ciljeva u zadanom vremenskom roku i budžetu. Ono što je najveća prednost tradicionalnih metodologija je da su projektni zadaci i faze unaprijed poznati što umanjuje rizik jer se zna što će se i kada odvijati, no problemi na koje su naišli projekt menadžeri su strogo definirane, nefleksibilne metode i veliki utrošak vremena na planiranje i pripremu kompletne projektne dokumentacije. Kod IT projekata, kada je riječ o korištenju tradicionalnih metoda, svi zahtjevi nisu poznati na samome početku pa je potrebna visoka razina fleksibilnosti kako bi se dao odgovor na navedene promjene, a u tom slučaju su tradicionalne metode ograničene i ne moguće je jasno definirati projekt i projektni plan. Dakle, agilne metodologije su idealne za IT projekte jer one imaju spomenutu fleksibilnost i česte promjene proizvoda, a upućenost naručitelja je veća što u konačnici dovodi do većeg zadovoljstva i same uspješnosti projekta. Agilne metode su one metode gdje je projektni tim više uključen vlastitim idejama, više se i slobodnije komunicira te dolazi do boljeg odgovora na definirane zahtjeve proizvoda. U slučaju softverskih rješenja, najčešće se koriste agilne metodologije jer takvim vrstama projekata najviše i odgovaraju, no njihova pojava ne znači da bi se tradicionalne metode trebale u potpunosti prestati koristiti, već svaka poslovna organizacija posebno odabire onu metodu koja im najviše odgovara s obzirom na poslovanje. Postoje ona poduzeća koja kombiniraju najbolje prakse iz raznih metoda koje



proizlaze iz tradicionalnih i suvremenih metodologija. Ne postoji jedinstven pristup koji bi odgovarao baš svim vrstama projekata. Zbog toga kombinacija tradicionalnog i agilnog pristupa može dovesti do efektivnijeg i efikasnijeg upravljanja. Usporedno sa tradicionalnim pristupom u kojem je projektni menadžer odgovoran za realizaciju projekta, od njega se očekuje da primijeni i nove vještine specifične za agilno okruženje i pruži podršku timu da se samoorganizira. Kod projekata gdje nastaju male promjene tijekom njegove realizacije može se reći da se primjenjuje tradicionalni pristup, jer kako je ranije navedeno kod takvih projekata nema nepredviđenosti jer naručitelj projekta jasno definira zahtjeve, a projektni tim zna kako doći do rezultata. Kod projekata gdje su ciljevi jasno definirani, a način dolaska nije jasan, kaže se da se primjenjuje agilni pristup. Odabir ili pak kombinacija metoda ovisi prije svega o samim početnim zahtjevima i proizvodu/usluzi koji projektni tim razvija. Postoje situacije kada će se projekti unutar IT sektora bolje provoditi prema tradicionalnim metodama, ali su agilne one koje su najadekvatnije. Izbor pojedinih metoda također ovisi i o tome koliko je projekt kompleksan i specifičan, koliki broj članova tima je potreban za razvoj projekta, jesu li svi zahtjevi projekta poznati, koji su dostupni resursi te poslovni i pravni čimbenici. Također, mora se voditi i računa o tome da odabrane metodologije (u ovom slučaju agilne) utječu i na ljude i na procese kao što su planiranje, sva potrebna dokumentacija i razvojni procesi, te na sam projekt (vrsta projekta, poslovnih i pravni čimbenici i sl.). Kada je riječ o budućem razvoju suvremenih metoda, česta je situacija da svatko iz trenutno dostupnih metodologija izvuče vlastitu "izvedenicu" koja je više ili manje uspješna i na takav način radi. Potrebno je voditi računa više o metodološkoj kvaliteti nego o količini metode. Svaka metodologija se samostalno proučava i predstavlja kao jedino rješenje problema, no ranije spomenutom pravilnom kombinacijom metoda se mogu očekivati i novije metodologije, a u konačnici i neki konkretni alati za izradu softvera pomoću kojih bi se projekti uspješno izvršavali.

## LITERATURA

1. Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2017). Agile software development methods: Review and analysis. *arXiv preprint arXiv:1709.08439*. Dostupno na: <https://arxiv.org/ftp/arxiv/papers/1709/1709.08439.pdf> [pristupljeno 13. srpnja 2021.]
2. Agarwal, R., Prasad, J., Tanniru, M., & Lynch, J. (2000). Risks of rapid application development. *Communications of the ACM*, 43(11es), 1-es. Dostupno na: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.617.6417&rep=rep1&type=pdf> [pristupljeno 13. srpnja 2021.]
3. Ahmad, M. O., Markkula, J., & Oivo, M. (2013, September). Kanban in software development: A systematic literature review. In *2013 39th Euromicro conference on software engineering and advanced applications* (pp. 9-16). IEEE. Dostupno na: [https://www.researchgate.net/profile/Muhammad-Ovais-Ahmad/publication/260739586\\_Kanban\\_in\\_Software\\_Development\\_A\\_Systematic\\_Literature\\_Review/links/548fe4830cf2d1800d862c6b/Kanban-in-Software-Development-A-Systematic-Literature-Review.pdf](https://www.researchgate.net/profile/Muhammad-Ovais-Ahmad/publication/260739586_Kanban_in_Software_Development_A_Systematic_Literature_Review/links/548fe4830cf2d1800d862c6b/Kanban-in-Software-Development-A-Systematic-Literature-Review.pdf) [pristupljeno 13. srpnja 2021.]
4. Altexsoft; software r&d engineering (2021). Extreme Programming: Values, Principles, and Practice. Dostupno na: <https://www.altexsoft.com/blog/business/extreme-programming-values-principles-and-practices/> [pristupljeno 13. srpnja 2021.]
5. Artto, K. A., & Wikström, K. (2005). What is project business?. *International Journal of Project Management*, 23(5), 343-353. Dostupno na : <https://aaltodoc.aalto.fi/bitstream/handle/123456789/45757/isbn9789529285358.pdf?sequence=1> [pristupljeno 8. srpnja 2021.]
6. Atkinson, R. (1999). Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria. *International journal of project management*, 17(6), 337-342. Dostupno na: [https://d1wqtxts1xzle7.cloudfront.net/50939896/s0263-7863\\_2898\\_2900069-620161217-30660-1size9j-with-cover-page-v2.pdf?Expires=1632764195&Signature=Es~OPCougWo4YUV6ns9aaeFJVuaZVTHhJA7Jd2zGHvuj0fC99ha5uOq3Nws5XF1nZRyCRrZkBcLlp5XA5D9z69wvvP1LtDEpTbyHZ0itCt31nFbbpBFN3d4YxeVXBntl-8iLay2KYL4LCNfJEKVN954Cmi3NDTIW0ACsgRwtPoPOWrBN6IGzdA8ihSwoPyNrP~yc74MN-JvyVcaoUMLNhJUqhCFFsCC30ve0G6HPZOLUlq0QmyaBbSDgB-](https://d1wqtxts1xzle7.cloudfront.net/50939896/s0263-7863_2898_2900069-620161217-30660-1size9j-with-cover-page-v2.pdf?Expires=1632764195&Signature=Es~OPCougWo4YUV6ns9aaeFJVuaZVTHhJA7Jd2zGHvuj0fC99ha5uOq3Nws5XF1nZRyCRrZkBcLlp5XA5D9z69wvvP1LtDEpTbyHZ0itCt31nFbbpBFN3d4YxeVXBntl-8iLay2KYL4LCNfJEKVN954Cmi3NDTIW0ACsgRwtPoPOWrBN6IGzdA8ihSwoPyNrP~yc74MN-JvyVcaoUMLNhJUqhCFFsCC30ve0G6HPZOLUlq0QmyaBbSDgB-)

[74EiQWpQMRL8IMWmYeuENSvkjtoclQZmXoYwmRQfnOtC9q118KVhp~pRV453p3GmH-K-yA4zddKZPt2UL0cUNQBIPh1Q &Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://www.researchgate.net/publication/338498700) [pristupljeno 9. srpnja 2021.]

7. Avlijaš, G., & Radunović, M. (2019). Application of Event Chain Methodology in Schedule Risk Analysis. *European Project Management Journal*, 9(2), 26-34. Dostupno na: [https://www.researchgate.net/profile/Goran-Avlijas/publication/338498700\\_Application\\_of\\_Event\\_Chain\\_Methodology\\_in\\_Schedule\\_Risk\\_Analysis/links/5e1a10a692851c8364c62697/Application-of-Event-Chain-Methodology-in-Schedule-Risk-Analysis.pdf](https://www.researchgate.net/profile/Goran-Avlijas/publication/338498700_Application_of_Event_Chain_Methodology_in_Schedule_Risk_Analysis/links/5e1a10a692851c8364c62697/Application-of-Event-Chain-Methodology-in-Schedule-Risk-Analysis.pdf) [pristupljeno 11. srpnja 2021.]
8. Beram, A. i dr. (2016). *Menadžment informatičkih projekata*. Split: Ekonomski fakultet u Splitu.
9. Chang, M. (2010). Agile and Crystal Clear with Library IT Innovations. In *VALA2010 Conference*. Dostupno na: [http://www.vala.org.au/vala2010/papers2010/VALA2010\\_14\\_Chang\\_Final.pdf](http://www.vala.org.au/vala2010/papers2010/VALA2010_14_Chang_Final.pdf) [pristupljeno 14. srpnja 2021.]
10. Cho, J. (2008). Issues and Challenges of agile software development with SCRUM. *Issues in Information Systems*, 9(2), 188-195. Dostupno na: <https://pdfs.semanticscholar.org/402c/77bb003ee7888f6d83d6bbd173737bd09521.pdf> [pristupljeno 13. srpnja 2021.]
11. Cleland, D.I. (1990). *Project management: Strategic Design and Implementation*. New York: McGraw-Hill, Inc.
12. Cohn, M. L., Sim, S. E., & Lee, C. P. (2009). What counts as software process? Negotiating the boundary of software work through artifacts and conversation. *Computer Supported Cooperative Work (CSCW)*. Dostupno na: <http://www.drsusansim.org/papers/cscwj-process-cohn.pdf> [pristupljeno 11. srpnja 2021.]
13. Horváth, V. (2019). Project management competence—definitions, models, standards and practical implications. *Vezetéstudomány-Budapest Management Review*, 50(11), 2-17. Dostupno na: [http://unipub.lib.uni-corvinus.hu/4289/1/VT\\_2019n11p2.pdf](http://unipub.lib.uni-corvinus.hu/4289/1/VT_2019n11p2.pdf) [pristupljeno 9. srpnja 2021.]
14. Info Novitas (2021). Ekstremno programiranje (XP). Dostupno na: <https://www.info-novitas.hr/o-nama/metodologije-rada/ekstremno-programiranje-xp/> [pristupljeno 13. srpnja 2021.]

15. Info Novitas (2021). Project Management (PMBOK). Dostupno na: <https://www.info-novitas.hr/o-nama/metodologije-rada/project-management-pbok/> [pristupljeno 22. lipnja 2021.]
16. Info Novitas (2021). Kanban agilna metoda. Dostupno na: <https://www.info-novitas.hr/o-nama/metodologije-rada/kanban-metoda/> [pristupljeno 13. srpnja 2021.]
17. Jenjić, M. (2018). Primjena metoda i vještina upravljanja projektima na projektima u javnim organizacijama (Doctoral dissertation, University of Split. Faculty of economics Split). Dostupno na: <https://repozitorij.efst.unist.hr/en/islandora/object/efst%3A2047/datastream/PDF/view> [pristupljeno 27. rujna 2021]
18. Kirovska, N., & Koceski, S. (2015). Usage of Kanban methodology at software development teams. *Journal of applied economics and business*, 3(3), 25-34. Dostupno na: <https://eprints.ugd.edu.mk/14949/1/030302.pdf> [pristupljeno 13. srpnja 2021.]
19. Linke, K. (2019). Traditional and Agile Management Approaches. In *12th ILERA European Congress, Düsseldorf, Deutschland*. Dostupno na: [https://www.researchgate.net/profile/Knut-Linke/publication/335724209\\_Traditional\\_and\\_Agile\\_Management\\_Approaches/links/5d77c3de299bf1cb8097b0a3/Traditional-and-Agile-Management-Approaches.pdf](https://www.researchgate.net/profile/Knut-Linke/publication/335724209_Traditional_and_Agile_Management_Approaches/links/5d77c3de299bf1cb8097b0a3/Traditional-and-Agile-Management-Approaches.pdf) [pristupljeno 22. lipnja 2021.]
20. McCormick, M. (2012). Waterfall vs. Agile methodology. *MPCS, N/A*. Dostupno na: [http://www.mccormickpcs.com/images/Waterfall\\_vs\\_Agile\\_Methodology.pdf](http://www.mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf) [pristupljeno 11. srpnja 2021. ]
21. Prišćan, A. (2021). Upravljanje projektima u RH: Stanje nacije. Dostupno na: <http://www.infotrend.hr/clanak/2008/10/upravljanje-projektima-u-rh-stanje-nacije,23,517.html> [pristupljeno 17. rujna 2021]
22. ProductPlan (2021). Crystal Agile Framework. Dostupno na: <https://www.productplan.com/glossary/crystal-agile-framework/> [pristupljeno 14. srpnja 2021.]
23. Project Management Institute Zagreb Croatia (2021). O nama. Dostupno na: <https://www.pmi-croatia.hr/hr/pmi-hrvatska/o-nama/> [pristupljeno 27. rujna 2021.]
24. Project management Srbija (2021). Kanban metodologija. Dostupno na: <https://project-management-srbija.com/project-management/kanban-metodologija> [pristupljeno 13. srpnja 2021.]

25. Raz, T., Barnes, R., & Dvir, D. (2003). A critical look at critical chain project management. *Project management journal*, 34(4), 24-32. Dostupno na : [https://d1wqtxts1xzle7.cloudfront.net/43924217/A\\_critical\\_look\\_at\\_critical\\_chain\\_projec20160320-19446-1yj33y9.pdf?1458492395=&response-content-disposition=inline%3B+filename%3DA\\_critical\\_look\\_at\\_critical\\_chain\\_projec.pdf&Expires=1625862588&Signature=Ga9zA7b83QF9DLlaa1araOucBTqkMUwfm32YsyF3yu4k8iR~EumuHB2Xmk9zuB9PfmPEat~b8W0bYZgL65U4PcZzGx2yxVBX4I4I5QggJ6LhFbX9iZiXxwGETQBuna~VbXWnAuq8M5wHM8uiXqiGevfQ6geenuMc8dSlc wdE~Jr5IB0OfQue7LOnkX-SsnEOQ5kSrMEjMcSATYXJn7r6mtqFyBu7hZYv0PFIJxATQaYSal~K~gtJWnG4~Pkbw4QgDSvAP9x9Vx4MFWD1xvbIpepxJx~~nuQbrkaPyjNmkPqqb2GH4tqSzq5iWBOaTqwm4Jvrv3lhRZKST~~v8FA3sQ\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/43924217/A_critical_look_at_critical_chain_projec20160320-19446-1yj33y9.pdf?1458492395=&response-content-disposition=inline%3B+filename%3DA_critical_look_at_critical_chain_projec.pdf&Expires=1625862588&Signature=Ga9zA7b83QF9DLlaa1araOucBTqkMUwfm32YsyF3yu4k8iR~EumuHB2Xmk9zuB9PfmPEat~b8W0bYZgL65U4PcZzGx2yxVBX4I4I5QggJ6LhFbX9iZiXxwGETQBuna~VbXWnAuq8M5wHM8uiXqiGevfQ6geenuMc8dSlc wdE~Jr5IB0OfQue7LOnkX-SsnEOQ5kSrMEjMcSATYXJn7r6mtqFyBu7hZYv0PFIJxATQaYSal~K~gtJWnG4~Pkbw4QgDSvAP9x9Vx4MFWD1xvbIpepxJx~~nuQbrkaPyjNmkPqqb2GH4tqSzq5iWBOaTqwm4Jvrv3lhRZKST~~v8FA3sQ_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA) [pristupljeno 9. srpnja 2021.]
26. Singh, V. (2020). What is Rapid Application Development Model (RAD)?. Dostupno na: <https://hackr.io/blog/rapid-application-development-model> [pristupljeno 13. rujna 2021.]
27. Stratton, R. (2009). Critical chain project management theory and practice. *Journal of Project Management and Systems Engineering*, 21(4), 149-173. Dostupno na: [https://d1wqtxts1xzle7.cloudfront.net/36683155/Critical\\_chain\\_managemet-with-cover-page-v2.pdf?Expires=1625923858&Signature=Bdy22k8CSk26ofHvdhJ8NEY4WUxf5OIOSOkZVEneroVy8v5Ek7RgBQ5lmSwyAHEIZDrmVuw6Fr9uVvR3pLwSzNB~xDIsRvW7MWJgwQnKFrhNp~OAI9oo5OY6wnOIbcZ-fsK-sU8DcTDc6nQB5xJVzAJ56Glh6hGOrdTyGnIc10x0u~QWXTvb4ncTRaCDoHHoyGDPOEIUAqfgxPftbnD1FEObProAWdmv-3UzXENqtF5NgEVFNqXMzdfEpwky0LF7c5Wx~mVT-W4xo~cj1-ZVaUG7Sonw8ZrvyIEzEOwSf19X~vJ6ofhf0g5XvbbKlvy5swEN31G5IE6VdVXQv\\_sQg\\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://d1wqtxts1xzle7.cloudfront.net/36683155/Critical_chain_managemet-with-cover-page-v2.pdf?Expires=1625923858&Signature=Bdy22k8CSk26ofHvdhJ8NEY4WUxf5OIOSOkZVEneroVy8v5Ek7RgBQ5lmSwyAHEIZDrmVuw6Fr9uVvR3pLwSzNB~xDIsRvW7MWJgwQnKFrhNp~OAI9oo5OY6wnOIbcZ-fsK-sU8DcTDc6nQB5xJVzAJ56Glh6hGOrdTyGnIc10x0u~QWXTvb4ncTRaCDoHHoyGDPOEIUAqfgxPftbnD1FEObProAWdmv-3UzXENqtF5NgEVFNqXMzdfEpwky0LF7c5Wx~mVT-W4xo~cj1-ZVaUG7Sonw8ZrvyIEzEOwSf19X~vJ6ofhf0g5XvbbKlvy5swEN31G5IE6VdVXQv_sQg_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA) [pristupljeno 10. srpnja 2021.]
28. Thackeray, V. (2002). Što je to projekt?. *Informatica museologica*, 33(1-2), 57-60.
29. Tutorialspoint (2021). SDLC – RAD model. Dostupno na: [https://www.tutorialspoint.com/sdlc/sdlc\\_rad\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_rad_model.htm) [pristupljeno 13. srpnja 2021.]

30. Umbreen, M., Abbas, J., & Shaheed, S. M. (2015). A Comparative Approach for SCRUM and FDD in Agile. *International Journal of Computer Science and Innovation*, 2, 79-87. Dostupno na: [https://www.researchgate.net/profile/Jalil-Abbas/publication/336835327\\_A\\_Comparative\\_Approach\\_for\\_SCRUM\\_and\\_FDD\\_in\\_Agile/links/5db51e3492851c577eca0bd6/A-Comparative-Approach-for-SCRUM-and-FDD-in-Agile.pdf](https://www.researchgate.net/profile/Jalil-Abbas/publication/336835327_A_Comparative_Approach_for_SCRUM_and_FDD_in_Agile/links/5db51e3492851c577eca0bd6/A-Comparative-Approach-for-SCRUM-and-FDD-in-Agile.pdf) [pristupljeno 13. srpnja 2021. ]
31. Vaníčková, R. (2017). Application of PRINCE2 project management methodology. *Studia Commercialia Bratislavensia*, 10(38), 227.
32. Virine, L., & Trumper, M. (2016). Event chain methodology in details. *Retrieved May, 10, 2016.* Dostupno na: [http://www.projectdecisions.org/paper/Paper\\_EventChainMeethodology.pdf](http://www.projectdecisions.org/paper/Paper_EventChainMeethodology.pdf) [pristupljeno 11. srpnja 2021.]
33. Vresk, A., Pihir, I., & Furjan, M. T. *Agilne vs tradicionalne metode za upravljanje IT projektima–studija slučaja.*
34. Wysocki, R. K. (2011). *Effective project management: traditional, agile, extreme.* John Wiley & Sons. Dostupno na: <http://103.38.12.142:8081/jspui/bitstream/123456789/174/1/Effective%20Project%20Management%20Robert%20K.%20Wysocki%2C.pdf> [pristupljeno 5. srpnja 2021.]

## POPIS SLIKA

Slika 1. Tri perspektive projekta .....	4
Slika 2. Pet faza tradicionalnog pristupa u vođenju projekata .....	9
Slika 3. Faza planiranja projektnog menadžmenata .....	12
Slika 4. Mrežni dijagram i kritični raspored lanca koji pokazuju međuspremnike.....	19
Slika 5. Primjer lančanog dijagrama događaja.....	21
Slika 6. Dijagram PRINCE2 procesa .....	23
Slika 7. Model vodopada.....	26
Slika 8. Proces agilnog razvoja softvera .....	30
Slika 9. SCRUM proces .....	32
Slika 10. Dijagram modela RAD .....	35
Slika 11. Primjer Kanban ploče.....	38
Slika 12. Životni ciklus projekta primjenom Ekstremnog programiranja.....	41
Slika 13. Dimenzije Crystal metodologije .....	43
Slika 14. Primjer Crystal Orange procesa .....	46