

Projektne metodologije u razvoju informatičkih sustava

Boban, Ivan

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, University of Zagreb, Faculty of Humanities and Social Sciences / Sveučilište u Zagrebu, Filozofski fakultet**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:131:011499>

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

Download date / Datum preuzimanja: **2024-07-16**



Sveučilište u Zagrebu
Filozofski fakultet
University of Zagreb
Faculty of Humanities
and Social Sciences

Repository / Repozitorij:

[ODRAZ - open repository of the University of Zagreb
Faculty of Humanities and Social Sciences](#)



SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
SMJER ISTRAŽIVAČKA INFORMATIKA
Ak. god. 2019.2020.

Ivan Boban

Projektne metodologije u razvoju informatičkih sustava

Diplomski rad

Mentor : izv.prof.dr.sc. Krešimir Pavlina

Zagreb, rujan, 2020.

Izjava o akademskoj čestitosti

Izjavljujem i svojim potpisom potvrđujem da je ovaj rad rezultat mog vlastitog rada koji se temelji na istraživanjima te objavljenoj i citiranoj literaturi. Izjavljujem da nijedan dio rada nije napisan na nedozvoljen način, odnosno da je prepisan iz necitiranog rada, te da nijedan dio rada ne krši bilo čija autorska prava. Također izjavljujem da nijedan dio rada nije korišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

(potpis)

Zahvale svima koji su podupirali i omogućili kreiranje ovoga rada. Posebne zahvale svim bližnjima na kontinuiranoj potpori u akademskoj karijeri. Veoma se zahvaljujem svim pojedincima koji su izdvojili svoje vrijeme za sudjelovanje u intervjuima.

Sadržaj

Sadržaj	ii
1. Uvod	1
2. Povijest i pregled	2
2.1. Povijest projektnog menadžmenta	2
2.1.1. Frederick Winslow Taylor	2
2.1.2. Henry Gantt	3
2.1.3. Henri Fayol	4
2.1.4. Razdoblje od 1950-ih	6
2.1.5. Metoda kritičnog puta (<i>eng. Critical Path method</i>)	6
2.1.6. Tehnika ocjenjivanja i pregledavanja	7
2.2. Povijest i pregled projektnih modela	8
2.2.1. Model vodopada	9
2.2.2. Spiralni model	12
2.2.3. Agilni „model“	14
2.3. Povijest i pregled projektnih metodologija	15
2.3.1. Ekstremno programiranje	16
2.3.2. <i>Lean</i> razvoj softvera	17
2.3.3. Scrum	18
2.3.4. Kanban	20
2.3.5. Ostale metodologije	22
3. PM ²	24
3.1. <i>Centre of Excellence in PM² (CoEPM²)</i>	24
3.2. <i>Open PM²</i>	24
3.3. <i>Project Support Network (PSN)</i>	25
3.4. <i>PM² Project Management Methodology</i>	25

3.4.1.	<i>The House of PM²</i>	26
3.4.2.	Životni ciklus projekta	27
4.	Project Management Institute	32
5.	Primjena projektnih metodologija.....	34
5.1.	Intervjui	35
5.1.1.	Voditeljica razvoja poslovanja	35
5.1.2.	Voditelj projekata	39
5.1.3.	<i>Senior</i> programer	41
5.1.4.	Voditelj projekata i poslovni analitičar	43
5.2.	Zaključci intervjua	45
5.3.	Kreiranje vlastitog <i>Way of Worka</i>	48
6.	Zaključak	50
7.	Literatura.....	51
8.	Dodatci.....	54
	Sažetak.....	56
	Summary.....	57

1. Uvod

Put od inicijalne ideje do finalnog produkta rijetko kada je onakav kakvoga smo inicijalno zamislili. Razni faktori konstantno utječu na naš plan aktivnosti, neovisno bili oni ljudski, vremenski, financijski ili neke druge prirode. Svaki od ovih faktora može drastično utjecati na smjer u kojem će naš projekt krenuti, te upravo radi toga bitno poznavati kako upravljati s njima. Ovdje u igru dolazi projektni menadžment, proces koji pokušava optimalno upravljati svim faktorima koji utječu na razvoj. Jedna od ključnih karakteristika projektnog menadžmenta je projektna metodologija. Jason Charvat¹ projektnu metodologiju opisuje kao „set smjernica ili principa koji mogu biti prilagođeni i primijenjeni specifičnoj situaciji.“ Bitno je istaknuti pojam specifična situacija, što upućuje na činjenicu da se na dva različita projekta, dvije različite specifične situacije, ne smije pristupiti na isti način. Pristupanje projektima na isti način može predstavljati veliki problem u raznim sektorima poslovanja, no jedan od sektora koja se najviše suočava s tim problemom je sektor informatičkih tehnologija. Kao jedan od sektora koji se najviše mijenja i razvija projektni menadžment se kontinuirano mijenja i prilagođuje sukladno s razvojem sektora informatičkih tehnologija. Upravo zbog kontinuiranog mijenjanja projektni menadžment je producirao mnogobrojne metodologije za praćenje i usmjeravanje razvoja unutar sektora informatičkih tehnologija. Cilj svake metodologije je, ugrubo rečeno, rasporediti rad kako bi se on izvršavalo u optimalnim uvjetima, no optimalno poprima različito značenje za različite članove tima. Upravo radi toga bitno je uvidjeti na koji način svi uključeni u razvoj pojedinog informatičkog sustava gledaju na projektne metodologije, te da li im one uistinu pomažu pri postizanju optimalnih uvjeta za realizaciju projekta.

¹ Charvat, J. *Project Management Methodologies*, John Wiley & Sons Inc (Verlag), 2003.

2. Povijest i pregled

Kako bismo razumjeli kako projektne metodologije zapravo utječu na rad svakog pojedinca unutar organizacije, prvo je potrebno znati kako smo došli do ove točke u projektnom menadžmentu.

2.1. Povijest projektnog menadžmenta

Počeci projektnog menadžmenta prožimaju se sve do prvih civilizacija, no tada je projektni menadžment bio veoma rudimentaran². Prema Clelandu projektni menadžment se kroz povijest koristio kako bi se stvorila promjena ili u svrhu nošenja s promjenom³. Promjena u ovom kontekstu ima pozitivnu konotaciju pošto se primjenom projektnog menadžmenta i korištenjem dostupih resursa može kreirati novi proizvod, usluga ili organizacijski proces. Ovakva definicija promjene pokriva prvi dio Clelandovog objašnjenja korištenja projektnog menadžmenta, drugi dio objašnjenja govori o nošenju s promjenom. Ovaj dio odnosi se na način kako postići željeni ili barem povoljni rezultat unatoč neočekivanim ili nepovoljnim situacijama. Prema inicijalnom objašnjenju uočljivo je kako je ključna ideja projektnog menadžmenta kroz povijest uvijek bila promjena, no prve velike promjene za sam projektni menadžment nisu došle sve do dvadesetog stoljeća.

2.1.1. Frederick Winslow Taylor

Kroz povijest projektni menadžment bio je vezan s mnogim područjima, no inženjerstvo je jedna od profesija koja je najviše pridonijela samom razvoju projektnog menadžmenta. Inženjerstvo po svojoj prirodi neprestano uzima u obzir mnogobrojne parametre u svrhu optimizacije, no to se tijekom povijesti rijetko kada fokusiralo na optimizaciju radnog okruženja. Osoba koju mnogi smatraju pretečom modernog projektnog menadžmenta je Frederick Winslow Taylor (1856. – 1915.). Taylorov cilj bio je uzeti principe i metodologije iz inženjerstva i primijeniti ih na različite aspekte poslovnog procesa. Produkt Taylorova rada je znanstveni menadžment⁴, princip upravljanja resursima čiji je cilj maksimizirati ekonomsku učinkovitost. Znanstveni menadžment fokusira se na izradi smjernica i hodograma putem kojih bi svaki od zaposlenika unutar pojedinog poslovnog procesa mogao postići svoju maksimalnu

² Cleland, D. I., Gareis, R. *Global Project Management Handbook, Planning, Organizing, and Controlling International Projects*, McGraw-Hill, 2003.

³ Cleland, D. I., Gareis, R. *Global Project Management Handbook, Planning, Organizing, and Controlling International Projects*, McGraw-Hill, 2003.

⁴ Taylor, F.W. *The principles of scientific management*, 1991.

ekonomsku učinkovitost. Taylor je svoje istraživanje primarno bazirao na optimizaciji rada u tvornicama, te u svome finalnom radu predlaže jednostavne principe optimizacije ljudskih resursa:

1. Zamijeniti stare načine obavljanja pojedinog posla s konkretnim uputama za svaki posao pojedinca
2. Znanstveno testirati, obučiti i razviti zaposlenika za pojedini posao, nasuprot postojećem modelu gdje je pojedinac samostalno birao posao koji želi obavljati
3. Suradnja između zaposlenika i menadžmenta kako bi se dogovoreni standardi održavali sukladno dogovoru
4. Preraspodjela odgovornosti i posla. Menadžment mora preuzeti sav posao za koji su oni sposobniji.⁵

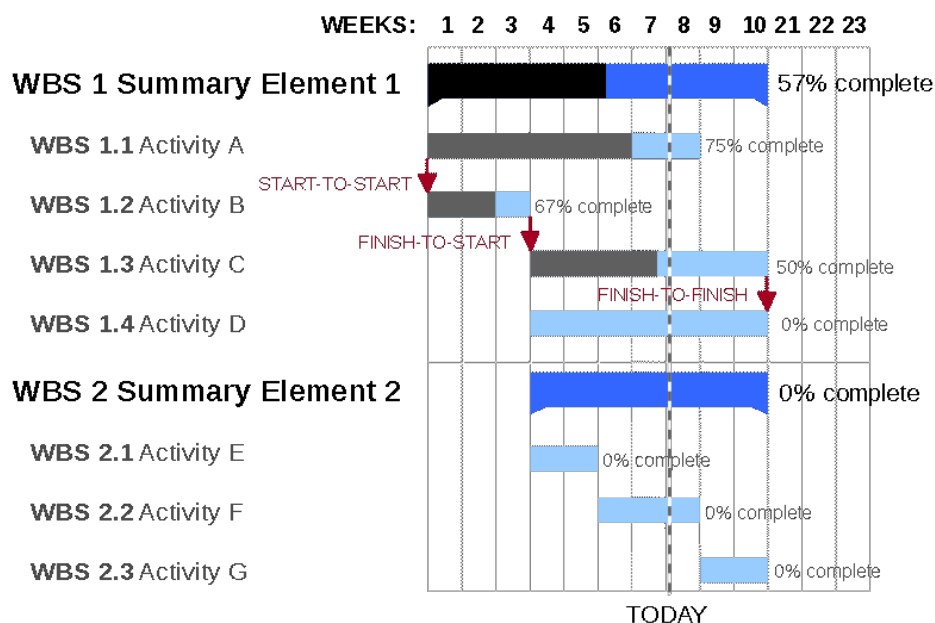
Principi koje Taylor navodi su primarno namijenjeni poslovima u tvornicama, te su stoga gotovo uvijek primjenjivi u takvim situacijama. Taylorov znanstveni menadžment postigao je svoj najveći značaj tijekom prva dva desetljeća dvadesetoga stoljeća, te je narednih godina dobivao sve više konkurencije. Tijekom vrhunca svog značaja znanstveni menadžment inspirirao je brojne druge pojedince koji su ostavili veliki značaj u području projektnog menadžmenta.

2.1.2. Henry Gantt

Jedan od pojedinaca čiji je rad bio pod utjecajem Taylorovog znanstvenog menadžmenta je Henry Gantt (1861. – 1919.). Gantt je kao i Taylor radio u inženjerstvu, te je Gantt također ostavio veliki trag u povijesti projektnog menadžmenta. Gantt je primarno poznat po svojim dijagramima. Iako su dijagrami nazvani po Ganttu, ideju iza njih prvobitno je osmislio poljski inženjer Karol Adamiecki tijekom devedesetih godina devetnaestoga stoljeća⁶. Gantt je dijagrame popularizirao u zapadnim državama, te se ovaj tip dijagrama danas naziva prema njemu. Ganttovi dijagrami vizualiziraju slijed aktivnosti i raspored pojedinog projekta, no osim toga pomažu pri definiranju međudnosa između pojedinih zadataka. Ovakav tip vizualizacije se koristi i danas, a neki od značajnijih projekata kod kojih su se koristili su izgradnja Hooverove brane i tijekom proširivanja državne autoceste u S.A.D. -u za vrijeme predsjednika Dwight D. Eisenhowera.

⁵ Taylor, F.W. *The principles of scientific management*, 1991.

⁶ *Gantt chart history*, <https://www.gantt.com/>, (20.07.2020.)



Slika 1. Primjer Ganttovog dijagram

2.1.3. Henri Fayol

Uz Gantta bitan pojedinac za ovo razdoblje je i francuski inženjer Henri Fayol (1841. - 1925.). Fayol je najpoznatiji po mnogobrojnim idejama koje iznosi u svome radu⁷, neke od kojih su ostale ne promijenjene do danas:

1. Funkcije menadžmenta
2. Principi menadžmenta

Prva bitna ideja koju Fayol iznosi je njegov pogled na funkcije menadžmenta. Fayol navodi kako menadžment ima pet funkcija:

1. Planiranje
2. Organiziranje
3. Koordiniranje
4. Upravljanje
5. Kontroliranje⁸

Ovakva raspodjela se i dalje uzima kao jedna od točaka polazišta modernog projektnog menadžmenta, iako se koordiniranje i upravljanje ponekad grupiraju u jednu funkciju, vođenje.

⁷ Fayol, H. *Administration industrielle et générale; prévoyance, organisation, commandement, coordination, controle*, Pariz, H. Dounod et e. Pinat 1916.

⁸ Fayol, H. *Administration industrielle et générale; prévoyance, organisation, commandement, coordination, controle*, Pariz, H. Dounod et e. Pinat 1916.

Druga ideja prisutna u Fayolovom radu su principi menadžmenta. Prema Fayolu postoji 14 principa menadžmenta, te mnogi eksperti njih smatraju začetcima teorije menadžmenta kakvu poznajemo danas⁹. Fayolovih 14 principa menadžmenta su:

1. Raspodjela posla – posao se raspoređuje u manje elemente i dodjeljuje pojedincima s adekvatnim vještinama za obavljanje tog posla
2. Autoritet i odgovornost – pojedinac koji ima autoritet za dodjeljivanje zadataka drugima mora sukladno time odgovarati za taj posao
3. Disciplina – međusobno poštovanje između naručitelja posla i izvršitelja posla
4. Jedinstvo naredbi – sve naredbe moraju dolaziti od jedne vrhovne osobe
5. Jedinstvo smjera – „jedna osoba, jedan plan“¹⁰
6. Podređenost individualnih interesa – interesi pojedinca ne smiju ugroziti interese organizacije
7. Renumeracija – novčani iznos plaćen trebao bi biti zadovoljavajuć zaposlenicima i organizaciji
8. Razina centralizacije – razina prema kojoj je mjerljivo na kojoj razini unutar organizacije se donose odluke
9. Skalarni lanac – „lanac superiornosti koji kreće od apsolutnog autoriteta sve do najnižih razina“¹¹
10. Red – odnosi se na materijalni red, sve ima svoje mjesto i sve je na svome mjestu
11. Pravičnost – jednakost ponašanja prema svima
12. Stabilnost zaposlenja – Fayol napominje kako je potrebno vrijeme da za prilagodbu novog pojedinca u radnu okolinu
13. Inicijativa – poticanje novih ideja i predlaganje drugačijih načina realizacija određenog projekta
14. *Esprit de Corps* – snaga u brojkama

Fayol je, kao i Taylor težio k optimalnoj učinkovitosti, te je smatrao kako će organizacije sljedeći njegove smjernice uspjeti postići performansno savršenstvo¹².

⁹ Golden Pryor, M., Taneja, S. *Henri Fayol, practitioner and theoretician – revered and relived*, Department of Marketing and Management, Texas A&M University-Commerce, Commerce, Texas, USA, 2010.

¹⁰ Golden Pryor, M., Taneja, S. *Henri Fayol, practitioner and theoretician – revered and relived*, Department of Marketing and Management, Texas A&M University-Commerce, Commerce, Texas, USA, 2010.

¹¹ Fayol, H. *Administration industrielle et générale; prévoyance, organisation, commandement, coordination, controle*, Pariz, H. Dounod et e. Pinat 1916.

¹² Golden Pryor, M., Taneja, S. *Henri Fayol, practitioner and theoretician – revered and relived*, Department of Marketing and Management, Texas A&M University-Commerce, Commerce, Texas, USA, 2010.

2.1.4. Razdoblje od 1950-ih

Neispravno bi bilo reći kako se projektni menadžment nije razvijao od početnih desetljeća dvadesetoga stoljeća, no u tome periodu većina projekata još uvijek nije imala projektni menadžment kakav mi poznajemo danas. Do sredine dvadesetoga stoljeća u svrhu projektnog menadžmenta pretežito su se koristiti Ganttovi dijagrami, te je ostatak planiranja bio rađen kako se koja faza završavala. Tijekom 1950-ih razvile su se dvije metode planiranja projekata:

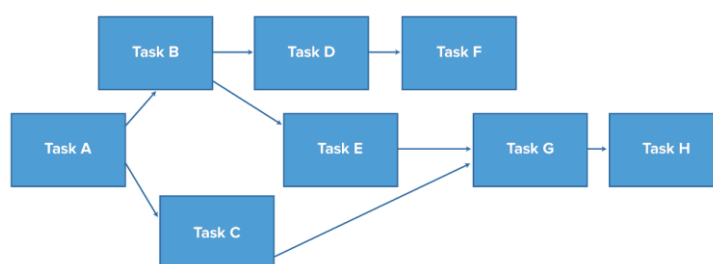
1. Metoda kritičnog puta (*eng. Critical Path method*)
2. Tehnika ocjenjivanja i pregledavanja programa (*eng. Program evaluation and review technique*)

2.1.5. Metoda kritičnog puta (*eng. Critical Path method*)

Metoda kritičnog puta razvijena je od strane Morgan R. Walkera i James E. Kelley Jr. tijekom 1950-ih. Iako se 1950-e uzimaju kao vremenski period razvoja, preteče ove metode pronalazimo u 1940-ih tijekom projekta Manhattan¹³, u kojemu je sudjelovao Walkerov poslodavac DuPont. Osnovni koncept metode kritičnog puta je izrada modela putem kojega možemo pronaći kritični put. Model se sastoji od:

1. Popisa aktivnosti potrebne za završetak projekta
2. Vrijeme potrebno za izvođenje pojedinog zadatka
3. Povezanost zadataka
4. Krajne točke projekta (*eng. Milestones*) ¹⁴

Koristeći navedene vrijednosti moguće je kreirati dijagram (Slika 2.) putem kojega možemo uočiti koje aktivnosti su ključne (kritične) za naš projekt.



Slika 2. Jednostavni CPM dijagram

¹³ Thayer, H., *Management of the Hanford Engineer Works in World War II, How the Corps, DuPont and the Metallurgical Laboratory fast tracked the original plutonium works*, ASCE Press, 1996.

¹⁴ Baker, S.L., *Critical Path Method (CPM)*, 2004.,

<https://web.archive.org/web/20100612142236/http://hspm.sph.sc.edu/COURSES/J716/CPM/CPM.html>
(21.7.2020.)

Cilj ovakvih dijagrama je optimalno rasporediti resurse kako bi projekt mogao biti gotov u što kraćem vremenu. Prilikom izrade *CPM* dijagrama vrijeme potrebno za izvođenje pojedinog zadatka uzima se kao jedna fiksna vrijednost, primjerice očekivani datum završetka aktivnosti ili broj dana od početka zadatka. *CPM* model aktivnosti prikazuje kao čvorišta koja se međusobno povezuju strelicama kako bi se prikazale međuodnosi između pojedinih aktivnosti.

2.1.6. Tehnika ocjenjivanja i pregledavanja

Tehnika ocjenjivanja i pregledavanja (*eng. Program evaluation and review technique*) razvila se istovremeno kao i metoda kritičnog puta. Ovu metodu razvila je Ratna mornarica Sjedinjenih Američkih Država kao potpora razvoju balističkih raketa¹⁵. *PERT* je veoma sličan *CPM-u* pošto je cilj oba modela optimalno iskoristiti dostupne resurse. Oba modela u konačnici kreiraju veoma slične dijagrame, no postoji par ključnih razlika. Dok *CPM* dijagram aktivnosti pozicionira na čvorišta, a putem strelica prikazuje povezanosti između aktivnosti, *PERT* dijagram aktivnosti pozicionira na strelice te ih međusobno povezuje putem čvorišta. Druga bitna razlika je podatak o vremenu potrebnom za završetak pojedine aktivnosti. *CPM* model prilikom izrade dijagrama uzima jednu fiksnu vremensku vrijednost, dok *PERT* model uzima tri vrijednosti:

1. Optimističnu procjenu (o)
2. Realnu procjenu (m)
3. Pesimističnu procjenu (p)

Putem ove tri vremenske vrijednosti *PERT* model može približno izračunati vrijeme potrebno za pojedinu aktivnost. Informacija o vremenu se može dobiti putem više funkcija no najčešća funkcija je:

$$te = \frac{o + 4m + p}{6}$$

U suvremenom projektnom menadžmentu *CPM* i *PERT* se i dalje koriste za jednostavno vizualiziranje potrebnih aktivnosti i vremena za realizaciju pojedinog projekta.

¹⁵ Malcom, D.G., Roseboom, J.H., Clark. C.E., Fazar, W., *Application of a Technique for Research and Development Program Evaluation*, Operations Research, 1959.

2.2. Povijest i pregled projektnih modela

Prema svemu navedenom lako je uvidjeti zašto je se projektni menadžment sredinom druge polovice devetnaestoga stoljeća nazivao „slučajnom profesijom“¹⁶. Prethodno navedeni modeli za vođenje i planiranje projekata naravno nisu prvi primjeri vođenja i planiranja projekta. Garel¹⁷ početak projektnog menadžmenta plasira u Srednji vijek, kada je improviziranje prestalo biti adekvatan pristup u građevini zbog brojnih faktora nedostatak, primjerice ljudskih ili materijalnih resursa. Ovakve problematike u suvremenom poslovanju rješava projektni menadžment, konkretnije različiti modeli i metodologije kako maksimizirati dostupne resurse. Taylorov rad u području znanstvenog menadžmenta konkretno se fokusirao na aspektu maksimiziranju uporabe svih dostupnih resursa i u konačni postizanju maksimalne ekonomske učinkovitosti. Ključna razlika između Taylorovih i Fayolovih pristupa je činjenica što su pristupi bili primjenjivi na raznolikim područjima. Primjenjivost na različitim područjima je ključan princip kod današnjih projektnih modela i metodologija, no bitno je razlikovati što je praksa, a što model u kontekstu projektnog menadžmenta. Praksa je nešto što je vezano uz pojedinu profesiju ili aktivnost, te je relativno teško praksu iz određene profesije koristiti u drugoj. Kao primjer možemo uzeti dvije kompanije koje se bave razvojem softvera, firma A se bavi razvojem aplikacija za banke, dok se firma B bavi razvojem aplikacija za uređivanje fotografija. Obje kompanije u srži obavljaju isti posao, razvoj aplikacija, no način na koji će oni doći do finalnog rezultata može biti veoma različit. Firma B će primjerice uložiti više vremena u izgled korisničkog sučelja svoje aplikacije, nego firma A, no to ne znači da bi firma A morala uložiti više vremena u takve aktivnosti. Iz tog razloga kada bismo mi kao projektni menadžeri htjeli maksimizirati učinkovitost firme A rad na izgledu aplikacije ne bismo trebali stavljati u fokus, dok bi za svrhu maksimizacije učinkovitosti firme B to potencijalno imali u fokusu. Ovakav individualni pristup karakterizira menadžersku praksu, kao nešto što je primjenjivo u veoma malom broju slučajeva, te čak i na razini individualnog slučaja. Poznavajući limitacije menadžerskih praksi model menadžmenta se može karakterizirati po četiri kriterija:

1. Model menadžmenta podržava reprezentativnost kompanije. Na kompaniju se gleda kroz više aspekata, te zanemaruje primarnu okupaciju kompanije.

¹⁶ Davis, J. G. *The accidental profession. Project Management Journal*, 15 (Special Summer Edition), 6. 1984.

¹⁷ Garel, G. *A history of project management models: From pre-models to the standard models*, Cnam Paris, Lirsa, France, 2012.

2. Model zanemaruje specifičnosti pojedinog sektora i sektora iz kojeg model proizlazi. Modeli Taylora i Fayola, iako su primarno razvijeni u inženjerstvu primjenjivi su i u drugim sektorima.
3. Postojanje institucija koje formuliraju i generaliziraju modele menadžmenta. Standardiziranje modela, edukacija o alatima, terminologiji, funkcijama itd.
4. Model menadžmenta karakteriziran je po kompaniji i/ili primjernim projektima. Putem konkretnih primjera uočljiva je prednost korištenja pojedinog modela.¹⁸

Posljednja točka upućuje na važnost dokazivanja učinkovitosti modela na konkretnom primjeru. Modeli Taylora i Fayola su svoju učinkovitost primijenili u području inženjerstva, Taylor u području strojarstva, te Fayol u području rudarstva, no drugom polovicom dvadesetoga stoljeća računalno inženjerstvo je adaptiralo postojeće modele, te je u narednim godinama postala jedna od glavnih grana unutar koje se razvijaju, ne samo projektni modeli, već i principi cijeloga projektnog menadžmenta.

2.2.1. Model vodopada

Model vodopada se najčešće smatra prvim modelom za razvoj softvera. Problematika povijesnog pregleda ovog modela je u činjenici što nije moguće konkretno odrediti vrijeme njenog nastanka i njenog autora ili autore. Unatoč tome za polazišni točku se najčešće uzima rad Dr. Winston W. Roycea *Managing the development of large software systems*. Royce u svom radu predlaže način vođenja projekata koji on smatra kvalitetnim¹⁹, te je do tih zaključaka došao radećim na raznim projektima. Ovime Royceov model zadovoljava točku 4 Garelovih kriterija, dok ostale točke nije moguće primijeniti (Garelov rad izašao je preko 40 godina nakon Royceovog).

Model vodopada svoji naziv dobiva zbog načina kako se on najčešće interpretira. Aktivnosti se prilikom razvoja informacijskog sustava nastavljaju ili „prelijevaju“ jedna na drugu, te su na taj načina povezane. Unutar prvobitnog prijedloga modela definirane su sljedeće aktivnosti:

1. Sistemski zahtjevi
2. Softverski zahtjevi
3. Analiza
4. Dizajn programa

¹⁸ Garel, G. A history of project management models: From pre-models to the standard models, Cnam Paris, Lirsa, France, 2012.

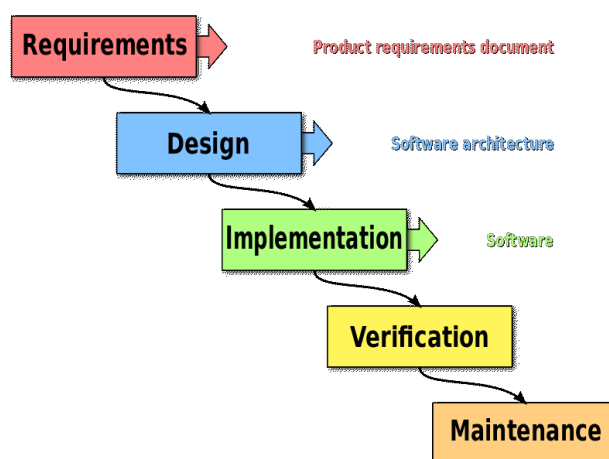
¹⁹ Ryoce, W. W., *Managin the development of large software systems*, Proceedings of IEEE WESCON, 1970.

5. Pisanje koda
6. Testiranje
7. Operacije

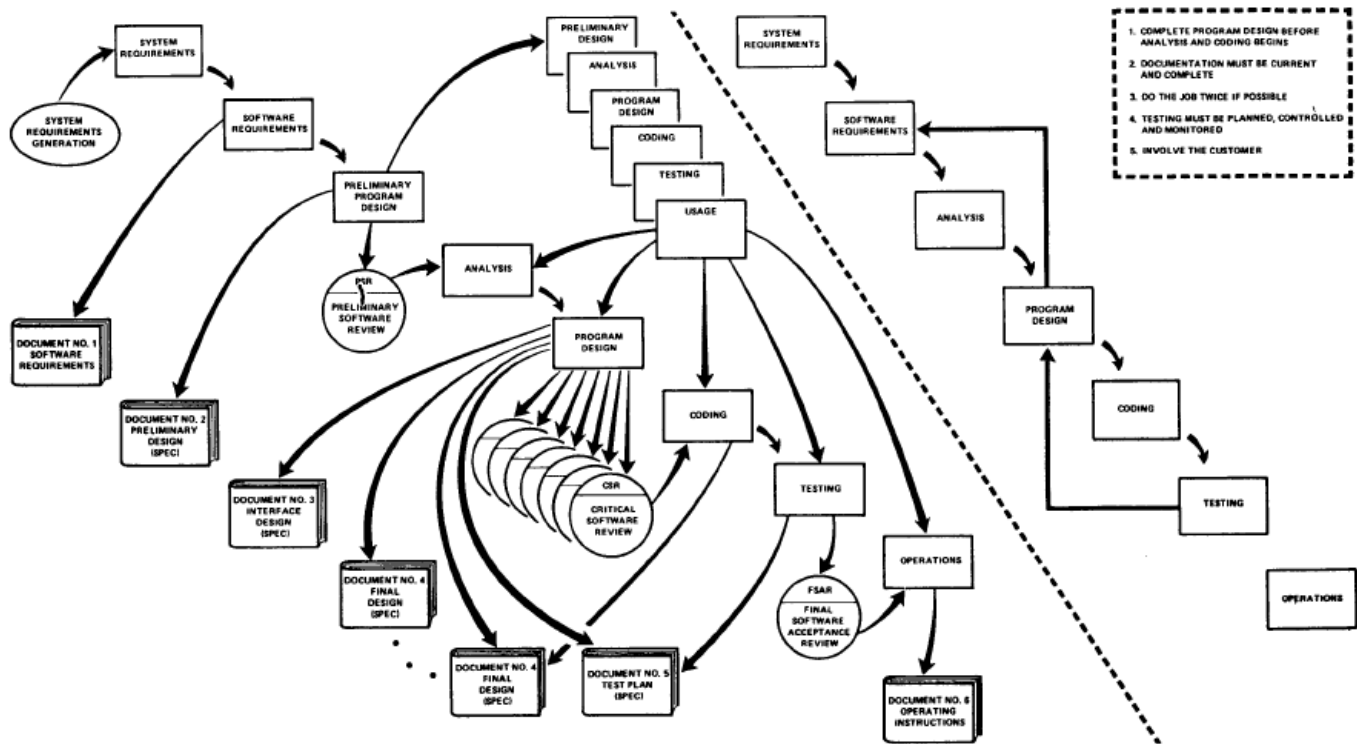
Koncepti koraka se od svog nastanka nisu znatno mijenjali, no danas se za model vodopada govori da ima 5 koraka:

1. Zahtjevi
2. Dizajn
3. Implementacija
4. Verifikacija
5. Održavanje

Neovisno govorimo li o 5 ili 7 aktivnosti, one unutar sebe sadržavaju mnogo podaktivnosti, Royce u obzir uzima i naručitelje informatičkih sustava prilikom predlaganja svog modela. Kako naručitelje najčešće ne zanima način kako se došlo do željenog proizvoda ovakav pristup pomaže u definiranju rokova i opravdavanju cijene razvoja. Jedna veoma bitna točka koja nije direktno spomenuta u aktivnostima je dokumentiranje svakog aspekta unutar razvoja. Velika pažnja je stavljena na dokumentiranje unutar Royceovog rada, pošto s kvalitetnom dokumentacijom moguće je parametrizirati gotovo svaku od razvojnih aktivnosti, te time izbjeci česte probleme koje susrećemo u razvoju. Uzimajući to u obzir, iako se za vizualiziranje ove metode najčešće koristiti vizual sa slike 3. Royceov finalni model je zapravo prikazan u slici 4.



Slika 3. Standardna vizualizacija modela vodopada



Slika 4. Royceov finalni vizual svoje predložene metode

Pregledom punog modela uočljivo je kako se skoro iz svakoga čvorišta povlači grana dokumentacije. Dokumentacija je jedna od jedinica u razvoju informatičkih sustava koja je oduvijek bila polarizirajuća, programeri ju rijetko kad žele pisati, dok bi vrhunski projektni menadžeri trebali inzistirati na njoj. Kako bi razumjeli važnost koju Royce stavlja na dokumentaciju moramo se prisjetiti jednog od ključnog aspekta projektnog menadžmenta, maksimalna učinkovitost dostupnih resursa. Ulaganjem u kvalitetnu dokumentaciju od početka projekta ostvarujemo uvjete za postizanje maksimalne učinkovitosti. Jedan od najboljih argumenata za ovakav pristup je situacija koja se dešava prilikom testiranja informatičkog sustava, greške u kodu. Bez kvalitetne dokumentacije za rješavanje grešaka u kodu potrebno je mnogo više vremena za otklanjanje grešaka, te ju najčešće mora ukloniti pojedinac koji ju je uzrokovao²⁰. U slučaju da pojedinac odgovoran za grešku trenutačno radi na nekome drugome projektu uklanjanje njegove greške ugrožava realizaciju dva projekta, te time smanjuje maksimalnu učinkovitost na dvaju projekata. S druge strane, ako je projekt ispravno dokumentiran projektni menadžer može angažirati druge pojedince za uklanjanje greške, pod pretpostavkom da posjeduju jednake vještine u pisanju koda kao i osoba koja je kreirala grešku.

²⁰ Royce, W. W., *Managing the development of large software systems*, Proceedings of IEEE WESCON, 1970.

Ovakav pristup primjenjiv je na svim aktivnostima unutar modela vodopada, te je važnost dokumentacije prisutna i u mnogo drugih projektnih modela.

Kada govorimo o povijest modela vodopada bitno je spomenuti razvoj SAGE-a (*eng. Semi-Automatic Ground environment*)²¹, prilikom kojega je također bio korišten sličan projektni model. Model primijenjen prilikom ovoga projekta smatra se među prvima koji je odbacio do tada standardni način razvoja, pisanje koda i ispravljanje grešaka u kodu, te je slično kao i model vodopada koristio model baziran na raspodijeli po aktivnostima.

Interesantna činjenica je kako Royce unutar svojeg prijedloga ne spominje izraz vodopada, pretežito zato što on sam smatra da tako jednostavan pristup nije adekvatan prilikom izrade informatičkih sustava, te pogledom na finalnu vizualizaciju model se ne može s lakoćom karakterizirati. Porijeklo naziva ovog modela također ne možemo jednoznačno odrediti, no par godina nakon objavljivanja Royceovog rada, Bell i Thayer²² koriste izraz vodopad govoreći o modelu koji Royce predlaže.

2.2.2. Spiralni model

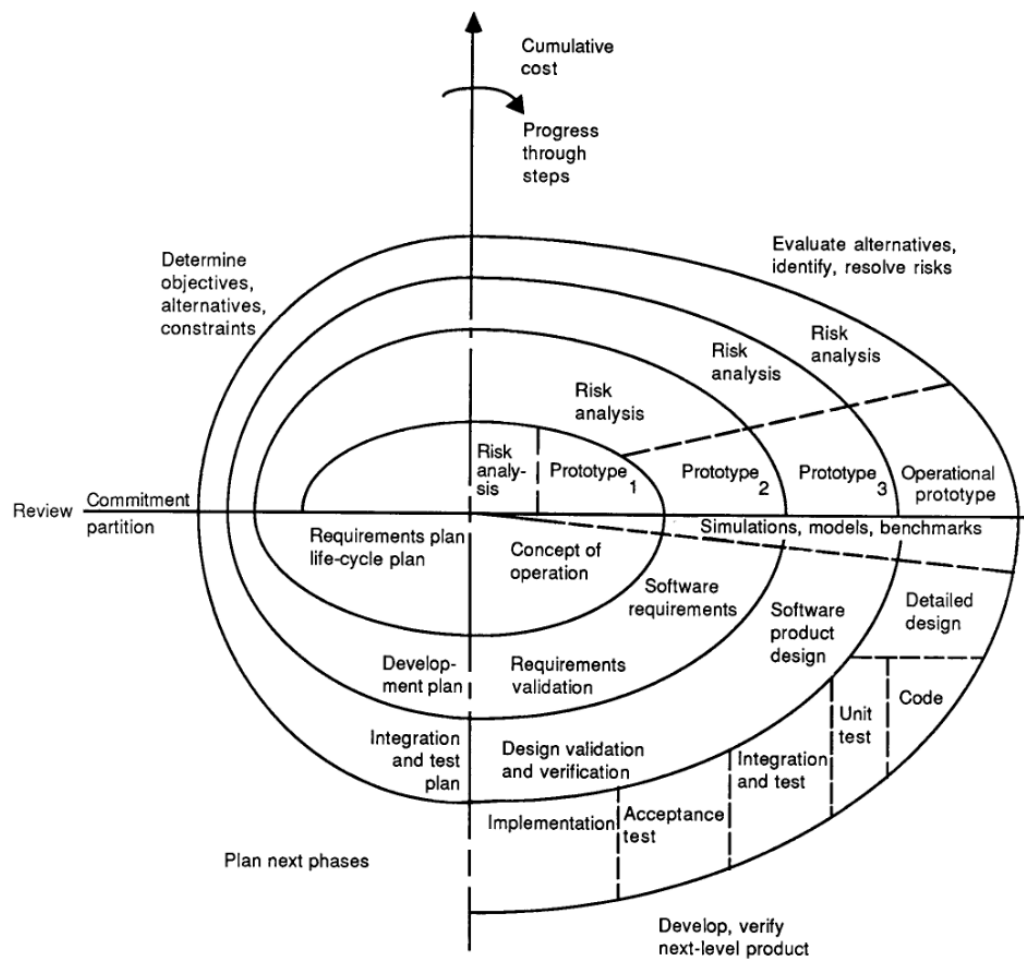
Model vodopada je ubrzo nakon svojeg nastanka postao primarni način planiranja razvoja mnogih informatičkih sustava, te je putem mnogih alternacija probao doraditi na svojim nedostacima. Unatoč svemu tome jedna ključna mana modela vodopada je uvijek ostala prisutna i vidljiva, što ako se projekt iznenada prekine? Cilj svakog projekta je da producira nekakav produkt ili usluga, no unatoč tomu mogućnost prekida projekta je uvijek prisutna pošto postoje faktori koje većina tadašnjih modela nije mogla uključiti u svoj plan, npr. gubitak ključnog člana razvojnog tima ili izlazak konkurentnog proizvoda na tržište koji je kvalitetniji nego naš proizvod u razvoju. Koristeći model vodopada uložili bismo veliku količini resursa u pripreme aktivnosti koje bi u slučaju prekida razvoja predstavljale veliki gubitak za kompaniju. Ovakvi problemi specifični su za modele koji se baziraju na dokumentaciji²³, te je Bohem u svome radu iz 1986. predložio način koji bi uklonio probleme tadašnjih projektnih metoda, spiralni model. Spiralni model je prilikom svoga prezentiranja zadovoljavao Garelov četvrti kriterij pošto je on bio primijenjen na razvoju TRW-ovog softvera za produktivnost.

²¹ Boehm, B.W., *A spiral model of software development and enhancement*, ACM SIGSOFT Software Engineering Notes, 1986.

²² Bell, T.E., Thayer, T.A., *Software requirements: Are they really a problem?*, TRW Defense and Space Systems Group Redondo Beach, California, 1976.

²³ Boehm, B.W., *A spiral model of software development and enhancement*, ACM SIGSOFT Software Engineering Notes, 1986.

Spiralni model baziran je na rizicima, te se prije prelaska iz jedne aktivnosti projekta u drugu uvijek moraju analizirati rizici za sljedeću fazu (Slika 5.).



Slika 5. Vizualizacija spiralnog modela

Koristeći spiralni model moguće je kontinuirano planirati faze razvoja prije nego li se one započnu, nasuprot planiranju svih faza istovremeno. Takvim pristupom omogućene su pravovremene intervencije tijekom projekta, te tako minimizirati uzaludno uložene resurse. Rizik se smatra bilo kakvom akcijom koja može ugroziti završetak projekta, kako je već navedeno gubitak člana tima ili novi konkurentni proizvod, no rizici mogu biti i promijene u dizajni ili funkcionalnostima razvijenog informacijskog sustava. U svrhu umanjnja tog rizika unutar spirale (Slika 5.) se kreiraju mnogi prototipovi. Izrada prototipova se može kategorizirati kao zasebna projektna metodologija čiji je fokus na izradi primjernih verzija nekog proizvoda kako bi se dokazala mogućnost izrade proizvoda ili kao pokazatelj dostupnih funkcionalnosti pojedinog informatičkog sustava. Poznavajući svrhu prototipa lako je uočljivo zašto su toliko bitni za spiralni model. Putem njih možemo prilikom kretanja spirale dobiti uvid u stanje

sustava koji razvijamo, dobiti povratne informacije od pojedinaca zaduženih za testiranje, te s tim informacijama planirati korake za sljedeću fazu aktivnosti.

2.2.3. Agilni „model“

Početak 21. stoljeća i razvoj Interneta informatički sektor se uvelike promijenio. Dostupnost Interneta sve većoj populaciji omogućila je razvoj sve više „e-“ usluga i potaknula prijelaz s programa direktno instaliranih na računalima, na usluge dostupne preko preglednika. U takvome svijetu razvoj je neprekidan, te se specifikacije mogu promijeniti u bilo kojem trenutku. Upravo radi prilagodbe vođenja projekata u informatičkome svijetu su se još krajem 20. stoljeća razvile različite metodologije koje svoj fokus stavljaju na sam proces razvoja informatičkih sustava. Metodologije nastale u ovome periodu biti će promatrane u sljedećem poglavlju, no ključno je navesti kako je nekolicina veoma utjecajnih ljudi za njihov razvoj 2001. prezentirali svoju ideju za optimalno vođenje projekata u modernome informatičkom dobu²⁴. Svoj prijedlog su nazvali agilnim, te on nije nešto što bi se moglo nazvati modelom, naime dok se prethodno navedeni modeli baziraju na sistematičnosti agilni „model“ se bazira na vrijednostima:

1. Pojedinci i interakcije, nasprema procesa i alata
2. Funkcionalan softver, nasprema iscrpne dokumentacije
3. Suradnja s klijentima, nasprema pregovaranje ugovora
4. Reagirane na promijene, naprema slijedenju plana²⁵

Navedene vrijednosti su ono što čine srž agilnog „modela“, te se zbog fokusa na njima agilni „model“ ne naziva modelom, već se češće koriste izrazi agilni razvoj ili agilni pristup. Agilni razvoj stoga nema striktno definirani raspored ili slijed aktivnosti pošto se, kako točka 4 navodi, nužno ne poštuje ustanovljeni plan već se razvoj kontinuirano prilagođava očekivanjima i potrebama klijenta. Kako je agilni pristup produkt mnogih zagovaratelja različitih projektnih metodologija, pogodan je integraciji s različitim metodologijama.

²⁴ Beck, Kent. Beedle, Mike. van Benneku, Arive. Cockburn, Alistair. Cunningham, Ward. Fowler, Martin. Grenning, James. Highsmith, Jim. Hunt, Andrew. Jeffries, Ron. Kern, Jon. Marick, Brian. Martin, Rober C. Mellor, Steve. Schwaber, Ken. Sutherland, Jeff. Thomas. Dave *Manifesto for Agile Software Development*, 2001. <https://agilemanifesto.org/>, (26.07.2020.)

²⁵ Beck, Kent. Beedle, Mike. van Benneku, Arive. Cockburn, Alistair. Cunningham, Ward. Fowler, Martin. Grenning, James. Highsmith, Jim. Hunt, Andrew. Jeffries, Ron. Kern, Jon. Marick, Brian. Martin, Rober C. Mellor, Steve. Schwaber, Ken. Sutherland, Jeff. Thomas. Dave *Manifesto for Agile Software Development*, 2001. <https://agilemanifesto.org/>, (26.07.2020.)

2.3. Povijest i pregled projektnih metodologija

Jedan od češćih problema prilikom pregleda projektnog menadžmenta je razumijevanje pojmova. U svrhu razumijevanja ciljanog korištenja pojedinih pojmova važno je poznavati osnove:

- Okvir se uzima kaj najbliži prijevod engleskog pojma *framework*
- Za definiciju pojma *framework* se uzima definicija iz Meriam-Webster rječnika
- *Framework* – osnovna konceptualna struktura (primjerice ideja)²⁶

Unutar raznih literatura uočljivo je kako se izrazi okvir i metodologija koriste naizmjenično, te je tako teško doći do jednoznačne razlike. Prema hrvatskoj enciklopediji pojam metodologija definiramo kao „nauk o metodi“²⁷, dok se metoda definira kao „planiran ili unaprijed smišljen postupak za postizanje određenoga teorijskog ili praktičnog cilja“²⁸. Pojam koji također moramo definirati je model, te njegova najprimjenjivija definicija bila bi: „Oblik (lik, tip) u koji se netko ugleda, koji uzima za primjer ili ga oponaša (reproducira); obrazac, uzorak, predložak, uzor, primjer“²⁹. Prema navedenome uočljiva je veoma velika sličnost između značenja pojma model i pojma *framework*. Unatoč sličnosti u značenjima model se u literaturi veoma rijetko koristi kao sinonim za metodologiju ili *framework*, dok se ta dva pojma koriste naizmjenično unatoč mnogo manjoj sličnosti u značenju. Prema tome možemo zaključiti kako pojmovi metodologija i *framework* imaju drugačije značenje kada o njima govorimo u kontekstu projektnog menadžmenta, točnije imaju mnogo sličnije značenje. Neovisno o tome koristi li se izraz projektna metodologija ili projektni *framework* ideja koja se prenosi ciljanoj publici je jasna, stoga u svrhu jednostavnosti u nastavku pojam projektna metodologija koristiti će se za obuhvaćanje obuhvaćanje pojmova projektna metodologija i projektni *framework*.

Poznavajući definicije ključnih pojmove možemo ih primijeniti u kontekstu projektnog menadžmenta:

- Simulirajući hijerarhiju projektni model može se smatrati podređenim projektnoj metodologiji
- Projektne modele možemo smatrati jednom od metoda projektnih metodologija

²⁶ Meriam-Webster riječnik, *framework*, <https://www.merriam-webster.com/dictionary/framework> (17.08.2020.)

²⁷ Leksikografski zavod Miroslav Krleža, *metodologija*, *Hrvatska enciklopedija, mrežno izdanje*. 2020. <https://www.enciklopedija.hr/natuknica.aspx?id=40441> (27.07.2020.)

²⁸ Leksikografski zavod Miroslav Krleža, *metoda*, *Hrvatska enciklopedija, mrežno izdanje*. 2020. <https://www.enciklopedija.hr/Natuknica.aspx?ID=40437> (27.07.2020.)

²⁹ Leksikografski zavod Miroslav Krleža, *model*, *Hrvatska enciklopedija, mrežno izdanje*. 2020. <https://www.enciklopedija.hr/Natuknica.aspx?ID=41453> (27.07.2020.)

- Projektni modeli mogu se smatrati projektnom metodom za raspoređivanje aktivnosti unutar projekta

Saznanjem o razlikama između modela, metodologija i okvira možemo usmjeriti svoj fokus na područje interesa, razvoj informatičkih sustava. Prethodno je navedeno da je razvoj informatičkog sektora potaknuo rapidni razvoj projektnog menadžmenta stoga će se naredne projektne metodologije sagledavati iz perspektive primjene u informatičkom sektoru i razvoju informatičkih sustava. Bitno je napomenuti kako se neke od sljedećih metodologija fokusiraju na sam proces razvoja informatičkih sustava, te tako aktivnosti poput dokumentacije nisu eksplicitno navedene u njima.

2.3.1. Ekstremno programiranje

Prethodno navedeni agilni razvoj produkt je okupljanja nekolicine veoma utjecajnih pojedinaca u području projektnog menadžmenta informatičkog sektora, te je jedan od sudionika bio i Kent Beck. Beck je tijekom 1990-ih bio angažiran od strane Chryslera kao programer u svrhu optimizacija performansi sustava koji je tada bio razvoju, te je njegov rad na tom projektu bio inspiracija za kreiranje nove projektne metodologije nazvane ekstremno programiranje (*eng. Extreme programming*)³⁰. Naziv metodologije proizlazi od činjenice što Beck fokus postavlja na već postojeće metode, ali se koriste u velikome intenzitetu („do ekstrema“). Jedno od područja na koje je Kent stavio veliki fokus je testiranje koda putem *unit* testova. Prvi aspekt *unit* testova na koji se ekstremno programiranje fokusira je na redosljed testiranja, te se predlaže da se prvo kreiraju testovi, pa zatim produkcijski kod³¹. Iako se doima kako bi kreiranje *unit* testova oduzelo dodatno vrijeme nasuprot razrješavanju grešaka u kodu kako ih pronalazimo, ova metodologija navodi kako pravovremeno uočavanje grešaka ušteduje resurse kasnije u razvoju kada greške nisu toliko očite. Testiranje dakako ne stane nakon što mi uspješno kreiramo neku novu funkcionalnost koda, već tek kada se novo kreirana funkcionalnost uspješno integrira s ostatkom sustava, stoga je bitno ponovno testirati cijeli sustav nakon što se ugradi nova funkcionalnost. Prilikom pisanja koda predlaže se i očuvanje jednostavnosti koda, te se prilikom pisanja novog koda ne smiju kreirati funkcije koje obavljaju više od jedne funkcionalnosti. Održavanje jednostavnosti koda uvelike je olakšano koristeći programiranje u paru. Zagovaratelji ekstremnog programiranja navode kako programiranje u

³⁰ Lee Copeland, *Extreme programming*, 2001., <https://www.computerworld.com/article/2585634/extreme-programming.html> (27.07.2020.)

³¹ Wells, D., *Code unit tests first*, 2000., <http://www.extremeprogramming.org/rules/testfirst.html> (28.07.2020.)

paru proizvodi jednaku količinu funkcionalnosti pojedinog sustava, no s mnogo većom kvalitetom koda³².

Kako je Beck jedan od stvaratelja ideje agilnog razvoja ekstremno programiranje također se zalaže za otvorenost u komunikaciji s krajnjim korisnikom, te mogućnošću za iznenadnim promjenama u specifikacijama zatraženog projekta. Ovaj aspekt se unutar ekstremnog programiranja često realizira odgovorima na testove prihvatanja. Testovi prihvatanja oblik su osiguranja kvalitete unutar metodologije, te služe kao finalna verifikacija funkcionalnosti prije nego li se ona u potpunosti integrira s ostatkom sustava³³. Sukladno s *unit* testovima broj testova prihvatanja trebao bi biti velik. Svrha velikog broja raznih testova činjenica što ekstremno programiranje, kao i agilni razvoj, smatraju kako je funkcionalni sustav bitniji dokaz rada tima, nego iscrpna dokumentacija.

2.3.2. *Lean* razvoj softvera

Kao što je uočljivo kroz povijesni pregled projektnog menadžmenta mnogobrojne industrije su kreirale principe vođenja projekata koji su primjenjivi na razvoj informatičkih sustava. Jedna od industrija koja do sad nije spomenuta je automobilska industrija, te je jedan od najvećih proizvođača automobila, Toyota, odgovoran za kreiranje temelja na kojima se bazira *lean* razvoj softvera. *Lean* pristup je adaptacija Toyotinog produkcijskog sustava, koji je produkt novog načina razmišljanja o manufakturi automobila vođena od strane Taiichi Ohnoa. Ohno je od početka imao jedan cilj, uklanjanje otpada³⁴. Otpadom se prema Ohnou smatra bilo koji aspekt koji ne pridonosi vrijednost kupcu, te je stoga fokus stavljen samo na nužne aktivnosti. Koncept uklanjanja otpada je ključan za *lean* razvoj softvera, te je polazišna točka za bilo kakav razvoj. Kada *lean* pristup primjenjujemo u razvoju informatičkih sustava otpadom smatramo sve aktivnosti koje ne pridonose završetku projekta, stoga bilo kakva čekanja, prebacivanja zadataka, dodatne funkcionalnosti smatramo otpadom. Osim uklanjanja otpada *lean* pristup stavlja fokus na ojačavanje razvojnog tima. Snažan razvojni tim ne znači samo zapošljavanje najkvalitetnijih pojedinaca, već izgradnju kvalitetnih odnosa između cijelog razvojnog tima. Važnost kvalitetnih odnosa između svih uključenih u proces razvoja informatičkih sustava je ključan aspekt ne samo *lean* pristupa, već i cijelog agilnog razvoja, pošto se smatra kako će kvalitetan i sretan tim kreirati kvalitetniji proizvod u manje vremena.

³² Wells, D., *Pair programming*, 2000., <http://www.extremeprogramming.org/rules/testfirst.html> (28.07.2020.)

³³ Wells, D., *Acceptance test*, 2000., <http://www.extremeprogramming.org/rules/functionaltests.html> (28.07.2020.)

³⁴ Poppendieck, M., Poppendieck, T. *Lean Software Development: An Agile Toolkit*, Addison Wesley, 2003.

Osim do sad navedenih principa, uklanjanje otpada i ojačavanje tima, *lean* pristup bazira se na još pet principa:

1. Pojačavanju učenja
2. Rok isporuke postaviti što dalje
3. Isporučiti što prije
4. Ugrađivanju cjelokupnosti
5. Optimizirati cjelinu³⁵

Većina navedenih principa dijeli veliku sličnost s ekstremnim programiranjem, pogotovo principi 4 i 5. Principi 2 i 3 preuzete su iz Toyotinog produkcijskog sustava koji se od početka bazira na što kraćem vremenu čekanja za kupca ili krajnjeg korisnika. Princip 1 zapravo najviše surađuje s inicijalnim konceptom uklanjanja otpada. Kontinuiranim učenjem razvojni tim neprekidno pronalazi nove načine rješavanja problema, te putem toga mogu pronaći dodatne aspekte unutar razvoja koji mogu biti eliminirani, jer se u tom slučaju smatraju otpadom.

2.3.3. Scrum

Jedna od metodologija koja najviše fokusa stavlja na međuljudske odnose je *scrum*. Metodologija je prvi puta prezentirana 1995-te na OOPSLA (*eng.* Object-Oriented Programming, Systems, Languages & Applications) od strane Ken Schwabera i Jeff Sutherlanda. *Scrum* je metodologija koja fundamentalno preispituje način kako se projekti planiraju prije nego li je napisana ita jedna linija koda. Model vodopada i spiralni model određuju kontekst i konkretne rokove na samom početku projekta, dok *scrum* planiraju kontekst i okvirne rokove na početku, te ih tijekom trajanja projekta adaptiraju i prilagođavaju sukladno napretku projekta³⁶. Za razliku od prethodno navedenih metodologija *scrum* potiče suradnju između svih pojedinaca uključenih u razvoj informatičkog sustava, te ih tako kategorizira u tri uloge:

1. Vlasnik proizvoda
2. Razvojni tim
3. *Scrum master*

Prva od uloga je vlasnik proizvoda i njegova/njena uloga je određivanje i komuniciranje željenih funkcionalnosti koje sustav u izgradnji mora imati. Prema *scrum* metodologiji vlasnik

³⁵ Poppendieck, M., Poppendieck, T. *Lean Software Development: An Agile Toolkit*, Addison Wesley, 2003.

³⁶ Schwaber, Ken. *SCRUM Development Process*, <http://www.jeffsutherland.org/oopsla/schwapub.pdf> (29.07.2020.)

proizvoda mora biti jedna osoba, no ona može biti predstavnik/predstavnica mišljenja i očekivanja većeg broja ljudi³⁷. Kako je vlasnik proizvoda odgovorna osoba za komuniciranje očekivanja, on je ujedno i odgovoran za konačni proizvod, pošto razvojni tim sukladno njegovim uputama i očekivanja razvija novi sustav. Ideja kako jedna osoba mora snositi odgovornost za proizvod koji su oni zatražili, neovisno o njegovoj kvaliteti, definitivno će u pojedinim vlasnicima proizvoda sigurno probuditi strah i nelagodu, no dubljim pogledom ovakav pristup postaje razumljiv. Razvojni tim sve svoje aktivnosti bazira na popisu potrebnih funkcionalnosti dobivenog od strane vlasnika proizvoda, te će u suradnji s njim davati prijedloge za dodatne aktivnosti. Dodatne aktivnosti mogu, ali ne moraju biti dodane na popis, ali je ključno znati kako je to bila svjesna odluka vlasnika proizvoda, stoga je ključno da cijela organizacija poštuje odluke vlasnika proizvoda.

Razvojni tim, kako je već navedeno, obavlja proces razvoja informatičkog sustava. *Scrum* metodologija podosta se razlikuje od ostalih, te jedan od ključnih aspekata je činjenica što se od razvojnog tima očekuje da samostalno rasporede aktivnosti potrebne za ostvarivanje ciljeva u pojedinom vremenskom intervalu³⁸. Veoma interesantan aspekt ovog dijela *scrum* metodologije je činjenica što se unutar razvojnog tima pojedinci ne kategoriziraju prema titulama ili prema aspektu posla koji obavljaju, već se na cijeli razvojni tim gleda kao jednu cjelinu. Time se pokušava ukloniti mentalitet „mi vi“, pošto je međusobno poštovanje i razumijevanje svih uključenih u razvoj ključno za uspjeh *scrum* metodologije.

Posljednja uloga unutar *scrum* metodologije je *scrum master*. Njegova primarna namjena je poticanje i podržavanje *scrum* metodologije u organizaciji, ali i pomaganje istoj prilikom adaptacije i učenja *scruma*. Prilikom prvobitne adaptacije *scrum master* će prividno biti član razvojnog tima, ali i jedan od savjetnika vlasnika proizvoda. Cilj ovakvog pristupa je prvotno mentoriranje svih uključenih u razvoj, kako bi razumjeli *scrum* metodologiju i uvidjeli njene prednosti, te naknadno savjetovanje istih pojedinaca. *Scrum master* također drži odgovornost provođenja jednog od najbitnijih aspekata *scrum* metodologije, dnevni *scrum*. Dnevni *scrum* je događaj na kojemu razvojni tim međusobno komuniciraju stanje razvoja u tom trenutku. Na prvu dnevni *scrum* zvuči kao običan sastanak razvojnog tima, no postoji par ključnih aspekata koji ga čine unikatnim:

- Održava se svaki dan na istom mjestu u isto vrijeme

³⁷ *Scrum guide*, <https://www.scrumguides.org/scrum-guide.html>, 2020. (31.07.2020.)

³⁸ *Scrum guide*, <https://www.scrumguides.org/scrum-guide.html>, 2020. (31.07.2020.)

- Vrijeme trajanja je 15-ak minuta
- Svaki od članova razvojnog tima ima ograničeno vrijeme
- Tema je određena od strane razvojnog tima, no najčešće se komunicira odgovaranjem na pitanja:
 - Što sam jučer napravio ?
 - Što ću danas napraviti?
 - Problemi koje imam u radu?
- *Scrum master* osigurava da se dnevni *scrum* održi
- *Scrum master* može biti prisutan, ali samo kao moderator³⁹

Posljednji bitni aspekt za *scrum* metodologije je ideja sprinta. Sprint je period unutar kojeg se implementiraju potrebne funkcionalnosti koje se nalaze na popisu kreiranog od strane vlasnika proizvoda. Prije početka sprinta razvojni tim odabire koje funkcionalnosti će biti implementirane unutar tog sprinta i same načine implementacije. Razvojni tim samostalno odlučuje o načinu implementacije (odabir tehnologije), te sami moraju organizirati kako će rasporediti potrebne razvojne aktivnosti. Nakon odabira razvojni tim svoj plan za sprint prezentiraju vlasniku proizvoda i *scrum masteru*. Iako je *scrum* metodologija koja prihvaća promjene, nakon što su ciljevi za pojedini sprint ustanovljeni, ovaj period razvoja ne dozvoljava promjene koje mogu ugroziti dogovoreni plan, no manje promjene koje ne ugrožavaju dogovoreni rok se mogu uključiti uz prethodni dogovor razvojnog tima i vlasnika proizvoda. Stanje sprinta je gotovo uvijek tema dnevnog *scruma*⁴⁰, te se putem njega zamjenjuju statusni sastanci koji pretežito zahtijevaju više vremena od dnevnog *scruma*.

2.3.4. Kanban

Jedna od metodologija koja je nastala sukladno s *lean* razvojem softvera je i *kanban*. Naziv metodologije potječe iz japanskog jezika i prevodi se kao pano. Metodologija se koristila sukladno s *lean* pristupom za vrijeme poboljšanja manufakture u Toyotinim tvornicama. Način primjene bio je sljedeći:

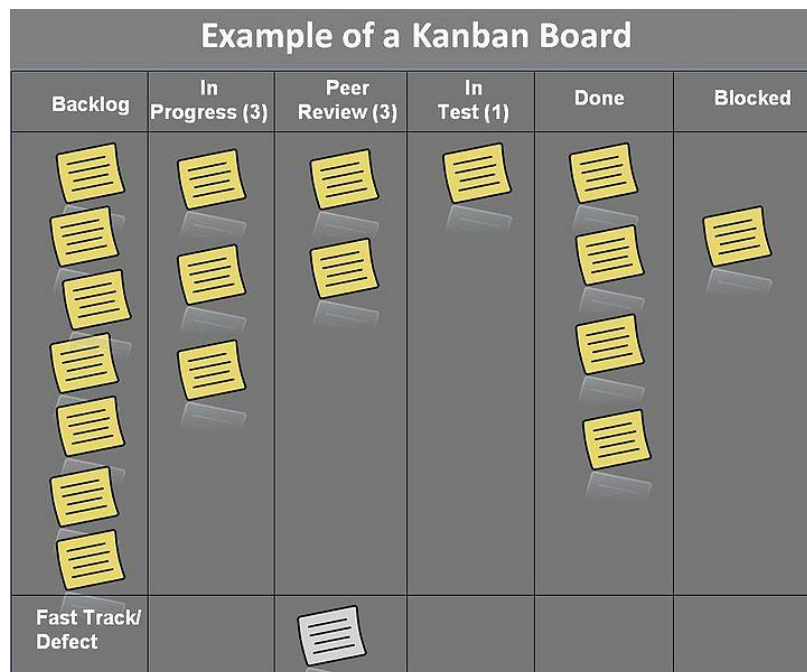
- Svaki završeni proizvod dobio bi zasebnu karticu
- Nakon što se proizvod proda dodijeljena kartica vraćala bi se na početak produkcijske linije

³⁹ *Scrum guide*, <https://www.scrumguides.org/scrum-guide.html>, 2020. (31.07.2020.)

⁴⁰ *Scrum guide*, <https://www.scrumguides.org/scrum-guide.html>, 2020. (31.07.2020.)

- Radnici su mogli raditi samo na dodijeljenim karticama i tek nakon što se postigao određeni broj kartica
- Svaki materijal također je imao svoju karticu, su se gornji procesi ponavljali sve dok resursi nisu napustili zadatke tvornice⁴¹

Cilj kartica bio je u skladu s ciljanim *lean* pristupom unutar kojega se sve nepotrebne ili ne profitne aktivnosti smatraju otpadom, no unatoč svojoj povezanosti s *lean* pristupom *kanban* se nastavio razvijati kao zasebna metodologija. Kada govorimo primjeni *kanban* metodologije u razvoju informatičkih sustava, kartice se ne dodjeljuju proizvodima. Kartice unutar ovog konteksta reprezentiraju potrebne aktivnosti unutar razvoja, a aktivnosti „dodajemo“ na *kanban* ploču. *Kanban* ploča prvi je korak pri uspješnoj implementaciji *kanban* metodologije u razvoju. Ploča s lakoćom reprezentira trenutačne aktivnosti u razvoju, njihove međudnose i međusobnu povezanost između pojedinih aktivnosti(Slika 6.), te tako pomaže u prvom koraku vizualiziranja potrebnih aktivnosti⁴². Aktivnosti se na ploči kategoriziraju prema njihovom trenutačnom stanju unutar razvojnog procesa. Kategorije nisu unaprijed zadane, no najčešće se koriste kategorije poput: u analizi, u tijeku, razriješeno, u pregledu, završeno i zatvoreno.



Slika 6. Primjer *kanban* ploče

⁴¹ *A history of Kanban*, <https://kanbantool.com/kanban-guide/kanban-history> (04.08.2020.)

⁴² *Visualizing the workflow*, <https://kanbantool.com/kanban-guide/kanban-fundamentals/visualize-the-workflow> (04.08.2020.)

Navedene kategorije predstavljaju očekivani ciklus pojedine potrebne aktivnosti, no kako razvoj informatičkih sustava nije uvijek bez prepreka postoje kategorije poput blokiran od strane druge aktivnosti. Današnji alati omogućuju mnogo različitih međuodnosa između aktivnosti i različite kategorizacije, stoga navedene kategorije ne obuhvaćaju sve moguće međuodnose.

Ploča je naravno samo prvi korak u uspješnom primjenjivanju *kanban* metodologije u razvoju informatičkih sustava. Drugi korak, limitiranje broja aktivnosti unutar kategorije u tijeku⁴³, podsjetnik je na činjenicu kako je *kanban* blisko vezan uz *lean* pristup. Broj aktivnosti unutar ove kategorije morao bi biti što manji. Metodologija ne nalaže koji je optimalan broj aktivnosti za ovu kategoriju, no do zaključka je veoma lako doći. Uzmemo li u obzir kako je cilj projektnog menadžmenta maksimizirati dostupne resurse uviđamo kako prilikom ostvarenja tog cilja resurse moramo ciljano rasporediti. Ponekad je veći broj manjih aktivnosti potreban kako bi se implementirala pojedina funkcionalnost, te je u tom slučaju veliki broj aktivnosti unutar kategorije u tijeku prihvatljiv. U takvim slučajevima potrebno je dakako kvalitetno promisliti o mogućoj preraspodijeli resursa kako bi se nužne aktivnosti što prije obavile ili o promijeni trenutnog pristupa kako razvoj ne bi bio opterećen.

Kako bismo izbjegli prethodno navedenu situaciju preopterećenosti pojedinih resursa *kanban* metodologija predlaže treći korak mjerenja i upravljanja protokom⁴⁴. Kontinuiranim mjerenjem možemo uočiti mane u našem pristupu i pravovremeno ih ispraviti. Ispravljanje grešaka i uočavanje područja koja se mogu poboljšati posljednji je korak u uspješnom primjenjivanju *kanban* metodologije⁴⁵, no postoji još jedan način kako možemo osigurati kvalitetu razvoja. Utvrđivanjem pravila i njihovim primjenjivanjem⁴⁶ osiguravamo da svi uključeni u razvoj poznaju „pravila igre“. Pravila mogu varirati od načina pisanja koda do kategoriziranja kada se aktivnost smatra gotovom.

2.3.5. Ostale metodologije

Sve do sada spomenute metodologije dakako nisu jedine koje se koriste, no *scrum* i *kanban* su dakako jedne od najčešći metodologija koje se koriste unutar informatičkog sektora. *Scrum* i

⁴³ *Limit work in progress*, <https://kanbantool.com/kanban-guide/kanban-fundamentals/limit-work-in-progress>, (04.08.2020.)

⁴⁴ *Managing and Measuring Flow*, <https://kanbantool.com/kanban-guide/kanban-fundamentals/manage-and-measure-flow> (04.08.2020.)

⁴⁵ *Recognize Improvement Opportunities*, <https://kanbantool.com/kanban-guide/kanban-fundamentals/improve-continuously> (04.08.2020.)

⁴⁶ *Making Process Policies Explicit*, <https://kanbantool.com/kanban-guide/kanban-fundamentals/make-process-policies-explicit>, (04.08.2020.)

kanban se ponekad koriste u kombinaciji, te se takav pristup ponekad naziva *scrumban*. Od ostalih metodologija vrijedno je napomenuti:

- *DevOps*
- *Rapid-application development (RAD)*
- *Adaptive software development (ASD)*

3. PM²

Cilj svih metodologija navedenih u prethodnom poglavlju je identičan, osigurati što brži i kvalitetniji razvoj. Svaka od navedenih metodologija ima svoje prednosti i mane, *scrum* metodologija se čak i od samih kreatora tumači kao „jednostavna za razumjeti, teška za savladati“. Sudeći prema tome lako je zaključiti kako je za uspješnu integraciju projektnih metodologija u projekte nužna osoba koja posjeduje znanja u tom području, projektni menadžer. Projektni menadžer je veoma koristan u razvoju informacijskih sustava, no kada govorimo o manjim organizacijama ili timovima zaposlenje takve osobe nije uvijek moguće. Znači li to da znanje o uspješnom implementiranju projektnih metodologija nije dostupno manjim organizacijama i timovima? Upravo suprotno. Kako bi se znanje o vođenju projekata što više približilo svima uključenima u projekte Europska Komisija je 2008. godine započela s PM² inicijativom⁴⁷. Inicijativa je prvobitno započela 2007. godine raspodjelom uloga i odgovornosti za projekte u sektoru informatičkih tehnologija, no s vremenom je preuzela vodeću ulogu u razvoju projektnog menadžmenta na području Europske Unije. Utjecaj inicijative uočljiv je po rezultatima njezinog rada i razvoja:

- *Centre of Excellence in PM² (CoEPM²)*
- *Open PM²*
- *Project Support Network (PSN)*
- *PM² Project Management Methodology*

3.1. *Centre of Excellence in PM² (CoEPM²)*

Cilj CoEPM² ili Centar izvrsnosti u PM² je pružiti Europskoj Komisiji i ostalim institucijama Europske Unije kvalitetna rješenja i pomoć u području projektnog menadžmenta⁴⁸. CoEPM² je blisko vezan uz *Open PM²* pošto se putem njega promovira šira uporaba *Open PM²*.

3.2. *Open PM²*

Upravo je *Open PM²* odgovoran za rapidno širenje ideja ove inicijative i sve većem prihvaćanju ovakvog pristupa prema projektnom menadžmentu. Cilj *Open PM²* je omogućiti slobodan pristup svim dokumentima povezanih uz PM² inicijativu svima koji su zainteresirani za

⁴⁷ History of PM², https://www.pm2alliance.eu/history-of-pm_2/ (27.08.2020.)

⁴⁸ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

područje projektnog menadžmenta, te time povećati efikasnost projektnih menadžera i projekata koji oni vode. Ovakvim pristupom teže poboljšati efikasnosti i uspješnost projekata diljem Europe. *Open PM²* sve navedeno uspijeva koristeći mnoge različite pristupe no neki od ključnih su:

- Uvođenje zajedničkog jezika i procedura, razumljivi svima u svrhu poboljšane komunikacije
- Omogućuje transparentnost za među-organizacijske projekte
- Omogućuje poboljšano monitoriranje i kontroliranje projekata financiranih od strane Europske unije
- Publikacije osigurane od strane *Centre of Excellence in PM²*⁴⁹

3.3. Project Support Network (PSN)

PSN reprezentira zajednicu zagovaratelja, pojedinaca i organizacija, *PM²* inicijative i metodologije. *PSN* se zapravo sastoji od manjih lokalnih ureda za projektnu pomoć (eng. *Local Project Support Offices, LPSOs*) koji su pod sponzorstvom CoEPM²-a. Kao takvi *PSN*-ovi omogućuju razmjenu iskustava i znanja zajednice pojedinog područja koja time obogaćuju ne samo lokalnu zajednicu, već i direktno pomažu u daljnjem razvoju cijele inicijative.

3.4. PM² Project Management Methodology

Metodologija koju je *PM²* inicijativa kreirala možemo smatrati najbitnijim produktom cijele inicijative. Kao što je već spomenuto ovu metodologiju trebali bi moći koristiti organizacije i timovi neovisno o prethodnom znanju projektnog menadžmenta i broju uključenih u razvoj. U svrhu toga metodologija prvobitno definira pojam koji je u centru cijele tematike, projekt. Prema ovoj metodologiji projekt ima četiri ključne karakteristike:

- Privremenost
- Jedinstvenost
- Rezultat
- Ograničenja⁵⁰

⁴⁹ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

⁵⁰ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

Prvu karakteristiku možemo smatrati vremenskim aspektom projekta. Unutar tog aspekta podrazumijeva se kako svaki projekt mora posjedovati definirani početak i kraj. Jedinstvenost i rezultat možemo sagledati kao zajednički aspekt. Svaki projekt mora rezultirati neakvim rezultatom, neovisno da li je konkretni produkt ili usluga. Uparujući rezultat s jedinstvenošću dolazimo do zaključka kako rezultat svakog projekta mora biti nov. *PM²* metodologija prilikom definiranja ovih karakteristika uvažava projekte čiji je rezultat direktna konkurencija nekom drugom proizvodu ili usluzi, no navodi kako će rezultat takvog projekta uvijek imati razlike.⁵¹ Posljednja karakteristika odnosi se na aspekt za koji možemo tumačiti da je razlog postojanja cijele profesije, ograničenja. Pod ograničenjima smatramo bilo kakve limitacije koje utječu na planiranje, izvedbu, financiranje⁵² i niz drugih limitacija koje su postavljene na sam projekt. Definiranje svih navedenih aspekata nužno je, ne samo radi poznavanja osnovnih koncepata projektnog menadžmenta, već kako bismo poznavali razlike između projekata i operacija. Operacije možemo smatrati polarnom suprotnošću projekata, pošto se njima smatra svakodnevni rutinski rad pojedine organizacije. Spominjući operacije u pitanje se mogu dovesti informatičke kompanije čiji je primarni fokus nuđenje poslovnih rješenja (izrada posebnih sustava za posebne slučajeve), te se kao takve doimaju kao kompanije koje nemaju operacije. U usporedbi s drugim kompanijama koje imaju gotove proizvode kod kojih se operacijama mogu smatrati aktivnosti uporabe proizvoda, kompanije koje se bave nuđenjem rješenja operacijama se mogu smatrati aktivnosti održavanja kvalitete i dostupnosti ponuđenog rješenja. Dorade na postojećim sustavima se također mogu smatrati operacijama pošto one ne kreiraju novi proizvod, već su samo nadogradnje već kreiranog proizvoda. Poznavanje razlike između projekata i operacija važna je komponenta kako bi organizacija imala uspješno poslovanje.

3.4.1. *The House of PM²*

Kao i sve prethodne metodologije *PM²* metodologija bazira se na nekolicini ključnih ideja. U slučaju *PM²* metodologije to su:

- Modelu projektne vladavine
- Životnom ciklusu projekta
- Procesima
- Projektni artefakti

⁵¹ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

⁵² European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

Navedene ideje se unutar metodologije nazivaju *The House of PM²*. Model projektne vladavine nastavak je nad originalnim motivatorom cijele inicijative, definiranjem uloga unutar projekata u informatičkom sektoru. Putem njega metodologija definira niz uloga koje se pojavljuju unutar projekta, njihove zadatke i međusobne interakcije tijekom projekta. Životni ciklus projekta definira faze koje projekt vođen prema *PM²* metodologiji prolazi. Faze prema raspodijeli izgledaju gotovo identično modelu vodopada, te se to može doimati kao velika mana metodologije. Ove mane su razriješene s dvije ključne karakteristike metodologije, mogućnost prilagođavanje metodologije potrebama organizacije i dokumentiranom integracijom s agilnim pristupom. Nerealno je očekivati da organizacije nisu prilagođavale već spomenute metodologije prema svojim potrebama, no *PM²* je jedna od rijetkih koja otvoreno uvažava promijene, te čak nudi smjernice kako adekvatno izmijeniti metodologiju. Proces izmjene metodologije i rješenja koji odgovara pojedinoj organizaciji često se naziva kreiranjem vlastitog načina rada (*eng. Way of Work, WoW*). Integracija faza životnog ciklusa s agilnom metodologijom blisko je vezana uz preostale ključne ideje, te će životni ciklus projekta biti sagledan u sljedećem poglavlju. Procesi su aktivnosti koje su nužne za tijek projekta, te nekolicina njih služi za kreiranje artefakata. Artefaktima se najčešće smatraju dokumentacija koja nastane tijekom životnog ciklusa projekta, no unutar *PM²* metodologije postoje specifični dokumenti koji su usko vezani uz životni ciklus *PM²* projekta.

3.4.2. Životni ciklus projekta

Kada govorimo o metodologijama u razvoju informatičkih sustava važno je ponovno spomenuti njihovu bazičnu svrhu, maksimizacija dostupnih resursa i omogućiti što kvalitetniji i brži razvoj. Različite metodologije to pokušavaju osigurati na drugačije načine, *kanban* putem vizualizacije i raspodijele zadataka ili primjerice *scrum* putem *sprinta* unutar kojega se radi na konkretnim zadacima. Metodologije nisu toliko fokusirane na aktivnosti potrebne za dolaženje do dijela razvoja, dok prema *PM²* metodologiji projekt počinje mnogo prije nego li se napiše prva linija koda. Jedan od ključnih aspekata agilnog pristupa je reagiranje na promijene, nasprema slijeđenju plana, što je u realnom svijetu veoma česta pojava, no postavlja se pitanje zašto se promijene u planu dešavaju? Promijene u planu se najčešće dešavaju zbog nedovoljnog definiranja rezultata projekta ili pak uvažavanja svih promjena kojih se naručitelj projekta dosjeti, što u konačnici ugrožava projekt i može dovesti do zaustavljanja projekta. Gledajući iz ovakve perspektive dovodi se u pitanje zašto agilni pristup zagovara reagiranje na promijene. Kako bismo to razumjeli moramo prvobitno kontekstualizirati što znači promjena u planu i koja je važnost plana. Kao primjer možemo uzeti organizaciju koja je dobila zahtjev izrade

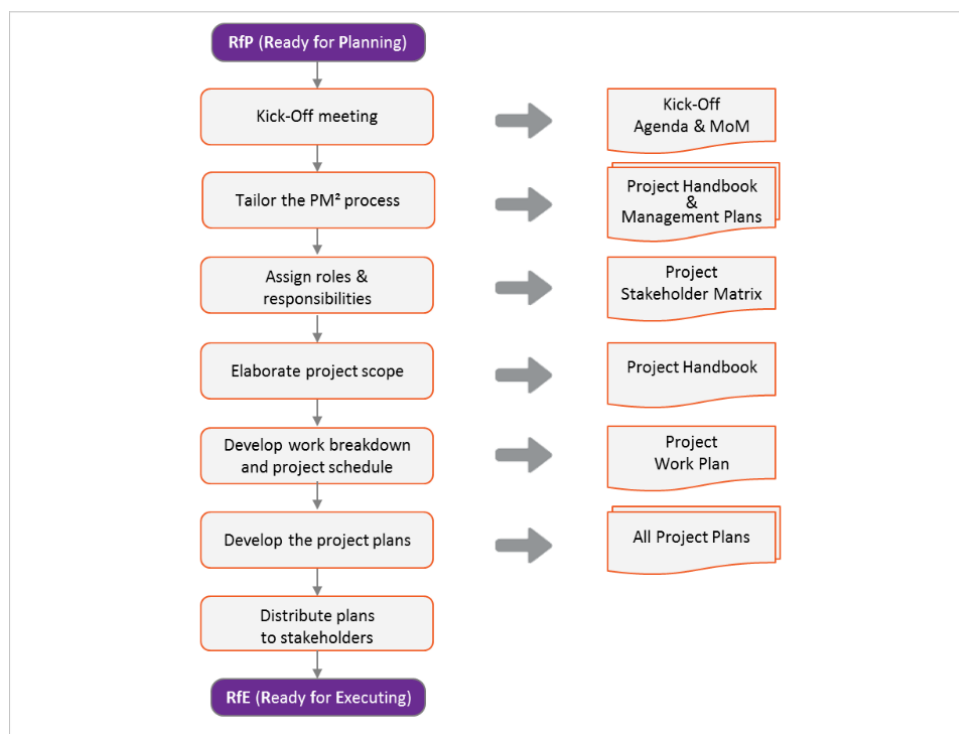
sustava putem kojega se mogu unositi i obrađivati podatci. Zahtjev se na prvu doima veoma jednostavan no kada bismo projekt započeli samo putem osnovnog zahtjeva postojala bi velika vjerojatnost neuspjeha. Kako bismo izbjegli takvu situaciju bitno je razumjeti što svaka od komponenata zahtjeva zapravo znači. Dobiveni zahtjev sastoji se od dvije ključne komponente unošenje podataka i obrada podataka, te je nužno dobiti ideju što svaka od tih komponenti znači za naručitelja projekta. Unositi podatke u sustav aktivnost je koja se može odvijati na mnogo različitih načina, unošenje putem forme ili pak očitavanje podataka sa slika, u konačnici naručitelj može očekivati kako bi on morao moći glasom unositi podatke. S druge strane obrada podataka također može imati više značenja, radi li se o financijskim izvješćima pa podatke treba obrađivati prema računovodstvenim procesima, ili se možda radi o slikovim podacima gdje se očekuje mogućnost retuširanja slika, dok se u konačnici možda jednostavno misli na sustav koji služi kao znanstveni kalkulator te bi kao takav morao moći raditi matematičke operacije. Kada znamo obujam (*eng. scope*) našega projekta mnogo lakše možemo reagirati na promjene unutar našega projekta, te ih ponekad i sami inicirati u svrhu kvalitetnijeg konačnog rezultata. Dobivanje zahtjeva i definiranje obujma projekta osnovni su procesi prve faze životnog ciklusa projekta prema *PM²* metodologiji. Ova faza naziva se faza iniciranja, te se unutar nje definiraju aspekti koji su u konačnici najbitniji za samog naručitelja projekta poput nužnog budžeta, nužnog vremena i ključno obujma projekta. U svrhu definiranja postoje tri artefakta (dokumenta). Prvi od tih dokumenata je zahtjev za inicijaciju projekta putem kojeg naručitelj dolazi pojedinoj organizaciji u svrhu započinjanja projekta. Nakon dobivanja zahtjeva za iniciranje projekta kreira se sljedeći nužan dokument poslovni slučaj (*eng. Business Case*)⁵³. Poslovni slučaj kreiraju članovi organizacije koja je zaprimila zahtjev, najčešće poslovni menadžer (*eng. Business manager*), te se on prezentira upravi organizacije. Unutar ovog dokumenta definira se isplativost projekta za samu organizaciju o kojoj u konačnici odlučuje sama uprava organizacije. Poslovni slučaj je dokument koji se neprestano može mijenjati ovisno o napretku projekta, te je njegova reevaluacija poželjna nakon svake prekretnice (*eng. milestone*) unutar projekta⁵⁴. Ako je poslovni slučaj odobren od strane uprave kreira se posljednji artefakt unutar faze iniciranja, projektna povelja (*eng. project charter*). Unutar povelje finaliziraju se obujam, nužno vrijeme, cijena projekta i rizici za projekt. Nužno je napomenuti kako je ovaj dokument kreiran od strane projektnog menadžera, poslovnog

⁵³ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

⁵⁴ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

menadžera i pružatelja rješenja (*eng. solution provider*). Pružatelj rješenja je osoba koja predstavlja ljude koji će u konačnici realizirati projekt, programere, te je njegova uključenost veoma bitna kako bi projekt imao izvediva očekivanja. Koristeći ovakav pristup ovako rano u životnom ciklusu projekta vidljiva je činjenica kako *PM²* metodologija slijedi ideje agilnog pristupa, ali i pomaže u usvajanju njenih ideja ne samo od strane organizacije već i od strane naručitelja projekta.

Nakon kreiranja poslovne povelje projekt prelazi u fazu planiranja. Tijekom ove faze dogovaraju se aspekti ključni za samu realizaciju projekta. Važnost ove faze je utvrđivanje načina vođenja projekta i raspodjele potrebnih aktivnosti na sve uključene u razvojni proces. Upravo zbog tih faktora ovo je faza koja producira povećani broj artefakata, konkretna raspodjela prikazana je na Slici 7.



Slika 7. Aktivnosti i produkti faze planiranja

Dokument koji je važan za razumijevanje projektne metodologije u kontekstu razvoja informatičkih sustava je projektni priručnik (*eng. Project Handbook*). Priručnik je rezultat inicijalnog sastanka pojedinaca koji su uključeni u realizaciju projekta. Cilj sastanka je razumijevanje očekivanja, uloga i plana, kao i utvrđivanje potrebnih akcija za finalnu realizaciju. Artefakti iz prethodne faze ključni su za kreiranje projektnog priručnika, no unutar njega se nalazi još jedan bitan aspekt koji ovu metodologiju čini veoma interesantnom.

Dogovoreni načini vođenja projekta su također navedeni unutar njega⁵⁵, što znači da se tijekom inicijalnog sastanka postiže konsenzus oko pristupa projektu. Ovakav pristup omogućuje prilagođavanje *PM²* metodologije konkretnom projektu, te otvara mogućnost uvođenja ideja drugih metodologija prema potrebi. Projektni tim može postići dogovor kako će tijekom faze izvršavanja (*eng. Executing Phase*) koristiti dnevne *scrum* sastanke ili bilo koji aspekt druge metodologije koji smatraju prikladnim za aktivni projekt. Drugi artefakt od velike važnosti je plan rada projekta (*eng. Project Work Plan*), artefakt proširuje stavke iz projektne povelje, te raspoređuje aktivnosti na dubljoj razini. Sastoji se od tri ključne komponente:

- Raščlanjivanje posla (*eng. work breakdown*)
- Procjene napora i cijene (*eng. effort and cost estimates*)
- Raspored projekta (*eng. project schedule*)

Komponente su međusobno ovisne o komponenti koja dolazi prije nje, stoga ih je prikladno objasniti kroz primjer. Nastavno na originalni zahtjev, sustav za unos i obradu podataka, pretpostavimo da je konkretan zahtjev sustav za praćenje plaća zaposlenika. Podatci se unose ručno putem predefinirane forme, te se potom mogu uspoređivati i kalkulirati. U svrhu primjera uzeti ćemo funkcionalnost unošenja podataka putem forme. Kako bismo izradili kvalitetan plan rada projekta moramo razumjeti što je sve potrebno za ovu funkcionalnost (izgled forme, polja, način pohrane u bazu podataka, validacija podataka itd.). Nakon što smo raščlanili funkcionalnost dobivamo ideju o potrebnim resursima za realiziranje te funkcionalnosti (broj ljudi, vrijeme i cijenu uloženog napora). Prethodna dva koraka nam omogućuju određivanje vremena unutar projekta kada će se raditi na toj konkretnoj funkcionalnosti. Ovakav proces pogodan je implementaciji sa *scrum* i *kanban* metodologijom. *Kanban* za raščlanjivanje aktivnosti na podaktivnosti, dok se *sprintovi* mogu uključiti u raspored projekta kako bi se unaprijed znalo vrijeme implementiranja. Plan rada smatra se živim dokumentom, te bi se također morao ažurirati sukladno s tijekom životnog ciklusa projekta. Faza planiranja uz navedeno producira još nekolicinu artefakata i obuhvaća procese poput tranzicijskog plana i plana poslovne implementacije.

Tijekom faze izvršavanja se napokon počinju pisati linije koda, te se tijekom ove faze kreira zatraženi sustav. Ova faza je konceptualno veoma jednostavna i veoma varira od projekta do projekta. Varijacije pretežito ovise o složenosti projekta, no unatoč za svaki dobro vođeni

⁵⁵ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

projekt mogu se primijeniti ideje iz *PM²* metodologije. Kao i prethodne faze tijekom faze izvršavanja kreiraju se artefakti te su za ovu fazu najbitniji projektni izvještaji. Tri najbitnija projektna izvještaja su izvještaj statusa projekta (*eng. Project Status Report*), izvještaj napretka projekta (*eng. Project Progress Report*) i izvještaj pregleda kvalitete (*eng. Quality Review Report*). Svaki od navedenih dokumenata služi kao alat za monitoriranje napretka projekta, te se putem njih mogu uvidjeti napreduje li projekt prema očekivanom rasporedu i prepoznati potencijalne rizike za projekt.

Završna faza je faza zatvaranja koja započinje kada vlasnik projekta prihvati rezultat faze izvršavanja, te se u ovu fazu ne prelazi sve dok se to ne dogodi. Faza služi za finaliziranje dokumentacije oko projekta i njegova zatvaranja, no sastoji se od jednog veoma bitnog aspekta, naučenih lekcija (*eng. Lessons Learned*)⁵⁶. Ovaj aspekt je veoma bitan za napredak i razvoj *PM²* metodologije, ne samo na razini organizacije, već i u svrhu poboljšanja same metodologije. Služi kao osvrt projektnog tima na projekt tijekom kojega mogu navesti pozitivne i negativne aspekte provedenog projekta i tako unaprijediti način rada na budućim projektima.

Dio životnog ciklusa koji se ne smatra fazom već aktivnošću koja se prožima je monitoriranje i kontrola. Ovaj aspekt ključan je za projektne menadžere kako bi osigurali uspješan napredak projekta. Tijekom svake od navedenih faza postoji niz faktora koji mogu kritično ugroziti projekt, stoga se od projektnog menadžera očekuje znanje kako bi pravovremeno uspio reagirati na potencijalne rizike i opasnosti. Aktivnosti su mnogobrojne, te je većina njih u sržni projektnog menadžmenta:

- Monitoriranje projektnih performansi
- Kontroliranje zadanog rasporeda
- Kontroliranje cijene
- Upravljanje s ključnim sudionicima projekta
- Upravljanje rizikom⁵⁷

⁵⁶ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

⁵⁷ European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

4. Project Management Institute

PM² inicijativa promiče primjenjivanje projektnog menadžmenta na razini Europske Unije, no organizacija koja je mnogo ranije započela sa standardiziranjem pozicija i koncepata u području projektnog menadžmenta je *Project Management Institute*. Početci instituta sežu unatrag 60 godina kada je projektni menadžment počeo biti primjenjivan unutar raznih industrija⁵⁸. Službeno osnivanje institucije dogodilo se 1969. godine u Atlanti od strane:

- James R. Snyder
- Eric Jenett
- Gordon Davis
- E.A. "Ned" Engman
- Susan Gallagher⁵⁹

Svrha instituta je u početku bila formalizirati područje projektnog menadžmenta i poziciju projektnog menadžera. U cilju realizacije tog cilja kreirani se različiti resursi koji formaliziraju, opisuju i standardiziraju mnoge aspekte projektnog menadžmenta. Od ovih aspekata nužno je spomenuti certificiranje koje institut provodi. Najpoznatiji certifikat je *Project Management Professional (PMP)*. Dobivanjem ovog certifikata osoba je u očima instituta sposobna voditi projekte unutar bilo koje domene, no unatoč tome pojedinci smatraju kako je pozadina u području prednost prilikom vođenja projekata. Osim *PMP* certifikata institut omogućuje certificiranje za :

- Certified Associate in Project Management (CAPM)
- Program Management Professional (PgMP)
- Portfolio Management Professional (PfMP)
- PMI Agile Certified Practitioner (PMI-ACP)
- PMI Risk Management Professional (PMI-RMP)
- PMI Scheduling Professional (PMI-SP)
- PMI Professional in Business Analysis (PMI-PBA)

Certifikati su valjani samo na određeni period (najčešće 3 godine), te je za obnavljanje certifikata nužno aktivno sudjelovanje u *PMI* zajednici. Institut veliki fokus stavlja na kontinuirano učenje, stoga se mogućnost produživanja certifikata mjeri u *professional*

⁵⁸ Patrick L. Healy, *Project Management: Getting the Job Done on Time and in Budget*. 1997.

⁵⁹ John Leo Weber, *A History of PMI & Its Role in Project Management*, <https://www.projectmanager.com/blog/history-of-pmi> (28.08.2020.)

*development units (PDU)*⁶⁰. PDU-i se mogu prikupiti na dva načina, educiranjem i vraćanjem zajednici. Educiranjem se smatra sudjelovanje na seminarima organiziranih od strane instituta, pohađanjem predavanja ili *webinara*. Pod vraćanjem zajednici smatraju se aktivnosti poput mentoriranja ili volontiranja. Svrha relativnog kratkog trajanja certifikata i potreba za kontinuiranim učenjem je održavanje kontinuirane visoke kvalitete pojedinaca koji posjeduju pojedine certifikate.

⁶⁰ *What are PDUs?*, <https://www.pmi.org/certifications/maintain/pdu> (31.08.2020.)

5. Primjena projektnih metodologija

Poznavajući osnovne principe i navedene prednosti pojedinih metodologija nužna je osnova kako bi ih mogli kritički proučavati. Povijesnim pregledom uočljivo je kako je maksimizacija ekonomske učinkovitost bila jedan od primarnih motivatora za istraživanja unutar područja projektnog menadžmenta. S razvojem projektnog menadžmenta i njegovom sve većom primjenom unutar informatičkog sektora ideja maksimizacije ekonomske učinkovitosti odbacila je „prefiks“ ekonomski, te se danas fokusira na generalnu učinkovitost ne samo pojedinca, već raznih timova i organizacije kao jedinstvene cjeline. Maksimalna učinkovitost pri razvoju informatičkih sustava drastično se razlikuje od one za kojom je Taylor tragao prilikom provođenja svojih istraživanja, pošto ne samo da se područje rada drastično razlikuje, nego je i sam put od ideje do konačnog produkta mnogo kaotičniji. Metodologije poput scruma i kanbana pokušavaju olakšati uvođenje reda u kaotičnost razvoja, no u konačnici one su samo „teorija“. Sljedeći „teoriju“ svaka organizacija trebala bi svaki svoj projekt odraditi predviđenom roku, sve tražene funkcionalnosti bi trebale biti ostvarene i u konačnici svaki projekt bi trebao teći glatko, no unatoč tome projekti ponekad ipak kasne ili se određene funkcionalnosti nikada ne finaliziraju. Razni faktori mogu utjecati na neuspjeh projekta, no nužno je sagledati način na koji svi uključeni u razvoj informatičkih sustava gledaju na i u konačnici koriste projektne metodologije. U svrhu takvog pregleda intervjuiran je niz pojedinaca koji su uključeni u razvoj informatičkih sustava. Svaki od ispitanika prvobitno je upitan o njihovim pozicijama, te na osnovi njihovih odgovora intervju je proveden s:

- Voditeljicom razvoja poslovanja
- Voditeljem projekata
- Senior programerom
- Voditeljem projekata i poslovnim analitičarom

Osim pitanja o poziciji koju obavljaju svi ispitanici su bili upitani i sljedeća pitanja:

1. Koliko godina radite unutar IT sektora?
2. Kako projektne metodologije utječu na vaš rad?
3. Osnovne prednosti koje vam projektne metodologije donose u radu?

Ovisno o odgovorima na pitanja postavljena su i dodatna pitanja koja su ovisila od ispitanika do ispitanika i smjeru intervjuja, te će ta pitanja i njihovi odgovori biti spomenuti u narednim poglavljima.

5.1. Intervjui

Fokus intervjua stavljen je iskustvo i mišljenje intervjuiranih pojedinaca o primjeni projektnih metodologija unutar IT sektora, konkretnije njihova primjena prilikom razvoja informatičkih sustava. Sukladno tome nikakvi osobni podatci nisu bili traženi, osim pozicije koju obavljaju, niti će se navoditi

5.1.1. Voditeljica razvoja poslovanja

Voditeljica razvoja poslovanja koja je bila intervjuirana prije obavljanja trenutne pozicije u organizaciji obavljala je pozicije *solution architecta* i projektnog menadžera. Analizom aktualnih tržišnih projektnih praksi uočljivo je kako pojedine metodologije stavljaju veći fokus na praćenje i vođenje projekata s ciljem zadovoljavanja zahtjeva uprave ili tržišta te nisu prilagođene metodi rada razvojnim stručnjacima kojima je izazov pratiti agilne metodologije te je stoga često izazov zadovoljiti ključne zahtjeve projekta (vrijeme, opseg, trošak i kvalitetu). Kao osoba s velikom količinom iskustva u različitim aspektima razvoja informatičkih sustava jedno od prvih postavljenih pitanja voditeljici je:

Smatrate li kako vam vaša prethodna radna iskustva pomažu u razumijevanju kako programeri i projektni menadžeri gledaju na projekte i primjenu projektnih metodologija?

Voditeljica razvoja poslovanja: Osobno smatram kako voditelji IT projekta moraju imati tehnička predznanja ukoliko vode tehničke projekte. To je moj subjektivni dojam, tržište smatra da voditelj ne mora biti stručnjak u području unutar kojega vodi projekte, već mora biti stručnjak u vođenju ljudi i projekata.

Koliko godina radite unutar IT sektora?

Voditelj poslovanja razvoja: 12 godina.

Koliko od spomenuti 12 godina koristite neku od projektnih metodologija?

Voditelj poslovanja razvoja: 7 godina.

Unutar 7 godina koje ste projektne metodologije koristili?

Voditeljica razvoja poslovanja: Zapravo sam oduvijek bila povezana s projektnim menadžmentom i pratila razvoj *Project Management Institute*, te sukladno s njihovim razvojem adaptirali metodologije poput *scrum* i agilnog modela. Moje iskustvo u vođenju projekata primarno je vezano za višegodišnje projektne implementacije aplikacija renomiranih vendora (Oracle, IBM) te rad s većim projektnim timovima u velikim domaćim i

internacionalnim kompanijama i javnom sektoru. Zbog složenosti i uključenosti velikog broja internih i vanjskih dionika te potrebe za ranijim budžetiranjem, projekti su bili rađeni korištenjem vodopad metode. Vođenje kroz agilne metode počela sam primjenjivati unatrag 3 godine na manje složenim projektima.

Kako je model vodopada dosta zastarjeli pristup, kako se on pokazao u praksi na vašim projektima?

Voditeljica razvoja poslovanja: Osobno dosta ne primjenjiv, zbog velikog razvoja tehnologija. Primjerice projekt od 2 godine započinjali smo s jednom tehnologijom, a završavali je s potpuno drugom. Osobno smatram kako bi postigli bržu i bolju isporuku da smo projekt razvijali kroz niz *sprintova*. Jedan od najvećih problema s takvim pristupom bila je činjenica što su prema planu budžet i vremenska ograničenja bila predodređena, no unatoč tome korisnička očekivanja su se kontinuirano mijenjala.

Kako projektne metodologije utječu na vaš rad?

Voditeljica razvoja poslovanja: Postoji više načina kako projektne metodologije pozitivno utječu na moj rad. Prva je svakako raspodjela posla između ljudi, unutar timova s kojima sam radila uvijek postoje *doeri* koji preuzmu veliku količinu posla na sebe, te mi metodologije poput *kanbana* pomažu ravnomjerno rasporediti potrebne zadatke. Druga je mogućnost praćenja učenja razvojnog tima, primjerice treba li za pojedine aktivnosti manje vremena.

Postoji li aspekt primjene projektnih metodologija za koji bi rekli da je negativno utjecao na razvoj timove s kojima ste surađivali?

Voditeljica razvoja poslovanja: Praćenje utrošenog vremena projektnog tima često je izazov projektnog *managera* i važno je naći pravi balans između zahtjeva *managementa* i uprave za detaljnim praćenjem troškova resursa i zahtjeva prema tehničkom tipu. Osobno mi je bio izazov kada sam radeći u kompaniji imala zahtjev uprave da na projektu dnevno isporučujem po pojedinom članu timu detaljan opis aktivnostima na razini sata podijeljen na naplative i nenaplative sate. U takvoj situaciji društveni odnosi između članova razvojnog tima postali su veoma loši zbog velike količine stresa unutar tima, ali i manjka kolegijalnosti. Članovi tima nisu bili spremni pomoći jedni drugima pošto nisu znali kako da evidentiraju vrijeme uloženo u pomaganje drugom članu tima.

Smatrate li kako članovi razvojnih timova koji su veliku količinu vremena radili bez projektnih metodologija teže adaptiraju metodologije nasprema pojedinaca koji ih primjenjuju od početka svog rada?

Voditeljica razvoja poslovanja: Otpor promjeni će uvijek postojati u pojedincima, no ne bih pripisala dob kao jedan od faktora koji utječe na spremnost k adaptaciji. Postoje pojedinci koji su tek ulazili u IT industriju te su pružali veliki otpor primjenjivanju projektnih metodologija u svom radu. Moj osobni cilj prilikom korištenja raznih metodologija i alata za njihovu implementaciju je pronaći pristup metodologije koji će odgovarati pojedinom članu tima. Važno mi je uskladiti očekivanja uprave i mogućnosti tehničkih timova kroz razne prilagodne tražiti *WoW (Way of Work)* na razini pojedinog projekta. Za neke će to biti Kanban, a za neke i obično planiranje i praćenje kroz excel.

Na koji način pronalazite pristupe/metodologije koji su prikladni vašim timovima?

Voditeljica razvoja poslovanja: Aktivno sudjelujem u okupljanjima PMI Hrvatska gdje s kolegama iz struke prolazimo kroz aktualne tržišne projekte i dijelimo iskustva. Također blisko surađujem s tržišnim PM-ovima s kojima prolazim kroz postojeće izazove i tražim savjete za vođenje i praćenje pojedinih projekata. Pogrešku smatram učiteljem, a metodologije koristim kao smjernice te ih prilagođavam projektima i timu. Pošto sam trenutno u firmi koja je u fazi tranzicije iz obiteljske firme u srednje veliku firmu koristim “light” metode kako bi se prilagodila mogućnostima tima i zadovoljila zahtjeve uprave.

Smatrate li kako je striktno praćenje „teorije“ metodologije ispravan pristup za korištenje projektnih metodologija?

Voditeljica razvoja poslovanja: Odgovor na ovo pitanje ovisi o nizu faktora. Kod manjih timova inzistiranje na svim aspektima pojedine metodologije nije efikasno. Jedan programer ne bi trebao sam sebi dodijeliti zadatak, obaviti zadatak, napraviti provjeru kvalitete obavljenog zadatka samo zato što metodologija nalaže da su svi koraci nužni. Također ovisi o dinamici razvojnog tima, jedan od ključnih razloga zašto *sprintovi* nisu bili uspješni unutar razvojnog tima je činjenica što razvojni tim ima veoma specifičnu dinamiku rada. Nadovezujući se na dinamiku rada, smatram da u situacijama gdje se razvojni tim jedino bavi projektnim aktivnostima projektne metodologije najviše uspijevaju. U svim ostalim slučajevima s razvojnim timovima potrebno je kreirati svoj način rada (*Way of Work*) koji odgovara dinamici posla.

Koliko vremena bi jedan programer trebao dnevno uložiti u aktivnosti vezane za projektne metodologije?

Voditeljica razvoja poslovanja: Do 15-ak minuta dnevno, no jedino ako on također uviđa prednost u takvome radu. Osobno ne inzistiram na formalnosti nekoga alata, te bih prihvatila

običan.txt dokument s popisom odrađenih tjednih aktivnosti. Aktivnosti jednog programera vezane uz prakticiranje projektnih metodologija ne smiju biti posao, već trebaju služiti kao pomoć u praćenju svog rada.

5.1.2. Voditelj projekata

Slično kao i voditelj razvoja poslovanja intervjuirani voditelj projekata prethodno je obavljao poziciju programera i web developera, te je stoga jedno od pitanja na koje je postavljen fokus bilo:

Smatrate li vašu tehničku prošlost kao prednost u obavljanju vaše trenutne pozicije?

Voditelj projekata: Znanje o području rada svakako smatram prednošću u obavljanju svog rada pošto mogu razumjeti probleme s kojima se razvojni tim suočava, no također to ponekad rezultira mojom prevelikom uključenosti u njihov rad. U globalu ne smatram kako bi voditelj projekata morao imati preveliko znanje o području unutar kojeg vodi projekte no to je svakako prednost.

Koliko dugo radite unutar IT sektora?

Voditelj projekata: Oko 14 godina.

Od tih 14 godina koliko dugo koristite bilo koji oblik projektnih metodologija?

Voditelj projekata: Agilni pristup koristim unazad 10-ak godina, pretežito zbog nedostataka starijih pristupa poput modela vodopada.

Koje mane ste uočavali prilikom uporabe modela vodopada?

Voditelj projekata: Pristup u svojoj srži nije prigodan za korištenje unutar ove domene. Metoda vodopada nije najprikladnija za softverske projekte prvenstveno u domeni definicije i potvrde opsega. Agile iterativnim pristupom dopušta izmjene (ili bolje shvaćanje) prilikom definicije opsega i time bolju komunikaciju između klijenta i izvođača, smanjuje troškove eventualnih izmjena i doprinosi zadovoljstvu, kvaliteti i boljem upravljanju očekivanjima.

Koje prednosti vam primjena projektnih metodologija donosi u radu?

Voditelj projekata: Agilne metodologije omogućuju nam mogućnost utjecanja na promjene. Promjene utječu na budžet, opseg i kvalitetu, što kasnije se uvode, to se više nešto mora žrtvovati, tj. veći su troškovi izmjene. Ponovno ću napomenuti već spomenutu točku, iterativni pristup tijekom definicije opsega umanjuje ovu vrstu rizika i omogućava izmjene u trenutku kada su one “najjeftinije”.

Postoje li nedostaci unutar projektnih metodologija koje utječu na vaš rad ili na rad razvojnog tima?

Voditelj projekata: Ne bih to nužno nazvala nedostatkom, no uviđanje prednosti koje primjena projektnih metodologija donosi u rad je jedan aspekt koji zahtjeva dosta vremena i volje svih uključenih. Primjerice scrum i dnevni stand upovi, u početku sama primjena metodologije neće biti 'glatka', ukoliko tim ima manjka u znanju o metodologiji i iskustva u primjeni. No upravo kroz kontinuiranu primjenu metodologije tijekom više projekata tim dobiva iskustvo. Timove ne treba prisiliti na metodologije, one najbolje uspijevaju kada su svi uključeni u njihovo korištenje spremni i voljni njihovom korištenju.

Koliko bi vremena pojedini član trebao odvojiti na aktivnosti vezane uz projektne metodologije?

Voditelj projekata: Na dnevne scrum stand-upove okvirno 15-ak minuta te još 15-ak minuta na eventualnu pripremu. Ovakav vremenski raspored naravno vrijedi za sve članove osim voditelja projekta. Ukoliko se ukaže da pojedini članovi prečesto odvajaju više od 30 minuta dnevno na takve aktivnost potrebno je reevaluirati trenutačni pristup.

Koliko članova bi razvojni tim trebao imati kako bi primjena projektnih metodologija bila učinkovita?

Voditelj projekata: Bolje je pitanje koju metodologiju primijeniti ovisno o članovima tima Scrum npr. idealno navodi mislim 3-5 članova tima (developeri, tester, analitičari) + Scrum Master + Product Owner

Koje su uloge koje najviše nedostaju u razvojnim timovima?

Voditelj projekata: Ovo nema egzaktan odgovor, ovisi o potrebama projekta, metodologiji, resursima tvrtke... npr. po scrumu PO i SM nikad ne bi trebala biti ista osoba. Ili nekad u praksi razvoj nema PM-a, nego to pada na lead developera, što je previše za tu ulogu i sl.

Zbog mnogih ograničenja jedan pojedinac u razvojnom timu ponekad mora obavljati više uloga, koje „kombinacije“ uloga su poželjne, a koje nisu? (npr. smije li programer sam sebi raditi QA?)

Voditelj projekata: Ne bih rekla da su poželjne kombinacije, osim ako se tako mora zbog uštede na resursima developer ne bi smio biti QA/QT.

5.1.3. Senior programer

Za kreiranje potpune slike kako projektne metodologije utječu na članove razvojnog tima intervjuiran je i programer s višegodišnjim iskustvom u razvoju informatičkih sustava.

Koliko dugo radite unutar IT sektora?

Senior programer: 12 godina.

Od tih 12 godina koliko dugo koristite bilo koji oblik projektnih metodologija?

Senior programer: Unazad 2 godine smo počeli primjenjivati projektne metodologije. Prije toga planovi su se prenosili i komunicirali putem *e-maila*, telefonski ili kroz *Bugzillu*. Putem *Bugzille* smo bilježili *taskove* vezane uz uklanjanje *bugova*, ali i bilježili zadatke vezane uz projektne aktivnosti.

Koje prednosti uočavate primjermom projektnih metodologija u svakodnevnom radu?

Senior programer: Jedna od ključnih prednosti je kvalitetnija organiziranost rada. Do prije uvođenja projektnih metodologija zadatci i dorade su samo pristizali, te su sve aktivnosti bile „veoma hitne“, no ako je sve hitno što je ustvari hitno? S trenutnim pristupom aktivnosti raspoređujemo na tjednoj razini, te tako posao pokušavamo jednakomjerno rasporediti na sve članove razvojnog tima. Ovakvim pristupom osiguravamo da ne zaboravimo na niti jednu aktivnost, ali i barem u mom slučaju smanjujemo količinu stresa u razvoju.

Koji pristup sada koristite u razrješavanju „hitnih aktivnosti“?

Senior programer: Prvobitno pokušavamo prosuditi istinsku „hitnost“ aktivnosti, te potom ovisno o hitnosti odrađujemo ili ne odrađujemo hitnosti. Ne predviđene aktivnosti svakako utječu na predviđeni tjedni plan, no u takvim slučajevima rokovi za neke aktivnosti budu pomaknuti.

Koliko vremena u prosjeku odvajate za aktivnosti vezane uz primjenu projektnih metodologija?

Senior programer: Za svoje (JIRA) *taskove* na dnevnoj razini moram odvojiti okvirno 15-ak minuta, no kako kao *senior* moram zadatke zadavati i mlađim kolegama, te tu ponekad moram odvojiti i do 2 sata na tjednoj razini. Primarni razlog za takvu količinu vremena je činjenica što mlađim kolegama pokušavam detaljno objasniti što se očekuje od njih. Pošto je naš sustav veoma opširan ponekad se i sam moram prisjetiti kako koji aspekt njega radi, kako bih mogao ispravno zadati zadatak nekom od mlađih kolega.

Kako prosuđujete koje aktivnosti prepustiti mlađim kolegama, a koje da vi odradite?

Senior programer: Primarni faktor prilikom te odluke je moje razumijevanje o njihovom poznavanju našeg sustava. Ukoliko smatram da mlađi kolega aktivnost može odraditi u razumnom vremenu aktivnost prosljedim na njega, u suprotnom aktivnost preuzmem na sebe. Prilikom donošenja takvih odluka svakako uzimam u obzir činjenicu kako mlađi kolege izgube priliku da savladaju i taj aspekt naših sustava, no ponekad su aktivnosti ili previše hitne ili je vrijeme za upoznavanje mlađih kolega s potrebnim aspektom mnogo veće od vremena potrebnog da ja odradim aktivnost.

Kako mlađi kolege doživljavaju projektne metodologije?

Senior programer: U glavnom ih koriste kvalitetno. Ponekad se dogodi situacija gdje ih moram podsjetiti da ažuriraju stanje pojedinog *taska* ili unesu komentar. Generalno kao organizacija smo na početku pune primjene projektnih metodologija u razvoju, no i ovaj pomak koji smo trenutačno napravili je uvelike poboljšao kvalitetu našeg razvoja.

5.1.4. Voditelj projekata i poslovni analitičar

Kako različite organizacije koriste različite projektne metodologije intervjuiran je još jedan voditelj projekata u svrhu dobivanja drugačijeg pogleda na projektne metodologije u razvoju informatičkih sustava.

Koliko dugo radite unutar IT sektora?

Voditelj projekata: 8 godina.

Od 8 godina koliko dugo primjenjujete neki oblik projektnih metodologija?

Voditelj projekata: Svih 8 godina.

Kao voditelj projekata smatrate li kako je poznavanje područja unutar kojega vodite projekte nužna predispozicija?

Voditelj projekata: U srži vođenje projekata se svodi na iste aktivnosti neovisno radi li se o izgradnji bolnice ili razvoju informatičkog sustava, no kvaliteta rada voditelja projekta će biti drugačija ovisno o njegovom poznavanju područja. Zbog svojeg obrazovanja u području IT-a puno bolje mogu predvidjeti i razumjeti potrebne korake za pojedine aktivnosti ili funkcionalnosti, primjerice nove API metode. U srži vođenje projekata se svodi na komunikaciju te će kvaliteta komunikacije biti osjetno bolja ukoliko voditelj projekata posjeduje predznanje tematike.

Smatrate li da je moguće „teoriju“ ijedne metodologije/frameworka primjenjivati u potpunosti?

Voditelj projekata: Smatram da je. Ne bih to nazvala "zašto to teorija tako nalaže" jer nekako se došlo i do te teorije, a ona je proizašla iz praktičnih problema koji su riješeni primjenom neke teorije na velikom broju primjera/projekata. Ako pritom pričamo o scrumu ili waterfallu, poznata su mi oba pristupa u svojoj "hardcode" primjeni koja su završila dobro nakon 100% primjene. Mišljena sam da najčešće svi mi balansiramo i uzimao ono najbolje od poznatih pristupa, no ako funkcionira - ne vidim potrebu za promjenom. Ako ne funkcionira - onda očito trebamo nešto promijeniti i pokušati implementirati 100% pa vidjeti koji se "lanci" mogu otpustiti ili prilagoditi našem okruženju i projektima.

Koje su direktne prednosti koje primjena projektnih metodologija donosi u vaš rad?

Voditelj projekata: Planiranje i organiziranje aktivnosti i pojedinaca što u konačnici dovodi do kreiranja najbitnijeg aspekta za klijenta, rok završetka projekta. Poznavanje odnosa između

aktivnosti, tko je od članova projektnog tima na kojem aspektu, kolika je izvršenost pojedinog zadatka sve su metrike koje projektne metodologije, uz primjenu alata kreiran za njih, prate i u konačnici pomažu meni u svakodnevnom radu.

Na koji način odlučujete kojim pristupom voditi projekt/kako ste došli do trenutnog pristupa?

Voditelj projekata : Svaki projekt je specifičan i način vođenja dogovaramo prilikom pripreme ponude projektni tim (uski tim ljudi koji sudjeluje u planiranju) i voditelj tima, ali također i naručitelj/klijent. Paralelno radimo na *waterfall*, hibridnim i agilnim projektima, a za svaki je projekt ovisno o tipu, svrsi, cilj, obimu, odabran najprikladniji način vođenja. Trenutačno najviše imamo hibridno vođenih projekata, a došli smo do njih temeljem iskustva rada na prethodnim projektima

Smatrate li da vlasnik projekta/krajnji korisnik treba posjedovati dublje znanje o procesima razvoja i primjeni projektnih metodologija u razvoju?

Voditelj projekata: Prvobitno bih napomenula kako mi je veoma teško povjerovati da su se projekti ikada vodili prema najčešćoj definiciji modela vodopada (interakcija s vlasnikom projekta samo na početku i kraju projekta). Prilikom primjene modernijih pristupa neke aktivnost se odvijaju kroz jedan oblik modela vodopada. Vlasniku projekta/krajnjem korisniku nužno je objasniti na koji način neka promjena utječe na rokove, ali on nužno ne mora znati sve aspekte putem kojih mi dolazimo do tog roka. Važno je napomenuti kako, barem prema mojem iskustvu, prilikom realizacije nekog projekta rijetko kada komuniciramo isključivo s vlasnikom projekt/krajnjim korisnikom, već najčešće komuniciramo s njihovom informatičkom službom koji su upućeniji u tematiku. U suradnji s njima jednostavnije je komunicirati tehnička ograničenja.

Koje aspekte projektnih metodologija mlađi članovi razvojnog tima najčešće ne razumiju?

Voditelj projekata: Većina mladih članova zanemaruju važnost komunikacije tijekom projekta. Komunikacija je u srži cijelog projektnog menadžmenta, ali nije nešto što se lako definira. Aspekte komunikacije između tima, prema korisniku i slično ne možemo zapisati na papir i očekivati da ih svi poštuju. Sve ovisi od osobe do osobe, no svi veoma brzo nauče kako ne komunicirati nakon prve veće greške prilikom komunikacije.

Koliko bi vremena pojedini član trebao odvojiti na aktivnosti vezane uz projektne metodologije?

Voditelj projekata: Apsolutno minimalno, do 15-ak minuta dnevno. Pojedinci koji bi eventualno trebali uložiti više vremena su *lead developeri* u svrhu opširnijeg pregleda stanja projekta, ali ne uviđam nikakvu potrebu da itko drugi odvajaju veću količinu vremena. Naravno ovisi o alatu i aktivnosti koje pojedina organizacija koristi, no pošto alat treba olakšavati posao trebao bi biti posložen da njegova uporaba oduzima što manje vremena.

Koliko znanje bi ostali članovi razvojnog tima trebali imati o projektnim metodologijama i sveukupnom projektnom menadžmentu?

Voditelj projekata: Trebali bi posjedovati općenito znanje što je projekt, kako se vodi i kako njegovo sudjelovanje u projektu i obavljanje aktivnosti (npr. ažuriranje statusa pojedinog *taska*) utječe na cjelokupni projekt. Poznavanjem ovih točaka pojedini član razumije što sve utječe na projekt i tako može izbjeći situacije koje mogu ugroziti projekt. Primjerice vlasnik projekta s programerom dogovori funkcionalnost bez uzimanja u obzir ostatak projekta i ostale članove, takva jednostavna aktivnost ne ugrožava samo *scope* projekta već potencijalno njegov termin isporuke, financiranje i mnoge druge dijelove projekta. U svrhu izbjegavanja takvih situacija, ali i drugih prednosti smatram kako je dobro dublje obrazovati članove razvojnog tima o vođenju projekata i projektnim metodologijama.

5.2. Zaključci intervjua

Cilj intervjua bio je dobiti ideju o načinu kako različiti članovi različitih razvojnih timova percipiraju projektne metodologije koje koriste u svome svakodnevnom radu. Kroz intervjue postavljen je niz potpitanja putem kojih se pokušalo uvidjeti koji su trenutačni trendovi u projektnom menadžmentu. Na osnovi navedenih i pojedinih ne navedenih pitanja dolazimo do sljedećih zaključaka:

Stariji pristupi poput modela vodopada sve više se zamjenjuju agilnim pristupima. Ispitanici su se složili kako takav pristup jednostavno nije realističan u suvremenom razvoju. Interesantno je napomenuti kako nitko od intervjuiranih pojedinaca nije koristio „standardni“ model vodopada. Prilikom korištenja modela vodopada komunikacija s vlasnikom projekta/krajnjim korisnikom bila je česta, no unatoč dogovorenim specifikacijama očekivanja za konačni proizvod su se ipak mijenjala. Upravo u ovakvim situacijama su svi se svi intervjuirani pojedinci složili da model vodopada propada i potrebno je koristiti neki od agilnih pristupa.

Prilikom korištenja agilnih pristupa komunikacija s naručiteljem projekta/vlasnikom projekta/krajnjim korisnikom je ključna, no jedan aspekt o kojemu su intervjuirani pojedinci bili ispitani je razina znanja koju pojedinci s kojima komuniciraju trebaju posjedovati o razvoju i eventualno o vođenju projekata. Odgovori na ovo pitanje su varirali, no u konačnici su svi intervjuirani napomenuli kako su gotovo uvijek tijekom razvoja komunicirali s pojedincima koji su posjedovali određeno razumijevanje o razvoju i projektnim metodologijama, te su zbog toga lakše mogli dogovarati očekivanja oko konačnog proizvoda.

Iako se svi intervjuirani slažu kako je velika prednost ako voditelj projekta posjeduje tehničke vještine, nitko ne napominje kako je to nužna predispozicija. Tijekom intervjua određeni pojedinci naveli su kako je spomenuta točka razlog mnogobrojnih diskusija unutar zajednice projektnog menadžmenta. U suštini voditelj projekta trebao bi posjedovati znanja samo o vođenju ljudi i projekata, dok je sve ostalo samo prednost.

Scrum + alat za praćenje aktivnosti su najpopularniji pristupi vođenja projekata. Ovakav rezultat nije neočekivan, te je najbolje objašnjenje ovakvog trenda tvrdnja jednog intervjuiranog pojedinca: „*Scrum* je u srži pristup koji razvojni tim educira kako da se samostalno organiziraju“. Jednom kada razvojni tim u potpunosti savlada sve aspekte *scruma* voditelj projekta, koji je u većini slučajeva prema iskustvu intervjuiranih pojedinaca i *scrum master*, jednostavno može pratiti napredak projekta i svoje vrijeme posvetiti drugim aktivnostima vezanim u vođenje projekta. Iako bi se prema laičkome shvaćanju uloge voditelja projekta „mikromenadžiranje“ razvojnog tima smatralo ključnom aktivnošću voditelja projekta, intervjuirani voditelji smatraju kako takav pristup nije ispravan. Razvojni timovi sami trebaju znati prosuditi potrebno vrijeme i potrebne resurse za realizaciju pojedine aktivnosti, dok im voditelj treba osigurati to vrijeme i resurse. U svrhu praćenja osiguranog vremena i resursa koriste se razne druge metodologije, primjerice *kanban*.

Kroz intervjuje ponovno je potvrđena važnost PMI-a kao vodeće organizacije unutar područja projektnog menadžmenta, te se metodologije i pristupi integriraju u razvojne timove sukladno njihovim preporukama. Prilikom integracije veliki naglasak mora biti stavljen na pravilno educiranje razvojnog tima o prednostima pojedine metodologije ili pristupa. Ukoliko razvojni tim ne uviđa prednosti novoga načina rada, neće doći do poboljšanja u efikasnosti razvojnog tima. Vještina koju bi svaki voditelj trebao posjedovati je procjena o efikasnosti pojedinog pristupa, ponekad preporuka PMI-a jednostavno nije primjenjiva unutar razvojnog tima radi mnogobrojnih faktora. Kontinuirano izmjenjivane pristupa i metodologija također nije

produktivno, stoga voditelji preporučuju periode od 3-4 mjeseca za isprobavanje novog pristupa, idealno na manjim projektima jednakog trajanja.

Utjecaj dobi i iskustva unutar IT sektora članova razvojnog tima preispitan je u kontekstu adaptacije na projektne metodologije. Nitko od intervjuiranih naveo dob ili iskustvo kao faktor koji utječe na brzinu adaptacije ili voljnost korištenja projektnih metodologija u razvoju. Jedan od često spomenutih faktora je činjenica kako mlađi pojedinci koji tek ulaze u IT sektor često već posjeduju znanja o projektnim metodologijama i pristupima. S druge strane pojedinci s mnogo godina iskustva unutar sektora s lakoćom uviđaju prednosti, jer im najčešće novi pristupi olakšavaju organizaciju rada i umanjuju stres. U ponekim slučajevima su određeni mlađi kolege pružali veći otpor adaptiranju od starijih kolega, stoga možemo zaključiti kako dob i iskustvo ne bi smjeli utjecati na uključivanje projektnih metodologija i principa u razvojne timove.

Faktor koji dosta utječe je jednostavnost i intenzitet uporabe različitih alata i aspekata pojedinog pristupa ili metodologije. Konsenzus je kako je 15 minuta dnevno idealna količina vremena koju članovi razvojnog tima trebaju odvojiti za svoje aktivnosti povezane uz projektne metodologije. Aktivnostima se smatraju *scrum* dnevni *stand-upovi* i ažuriranje *taskova* u određenom alatu. Za jednostavnost dnevnih *scrum stand-upova* odgovoran je *scrum master*, dok se za jednostavnost ažuriranja *taskova* trebaju dogovoriti pravila. Kako različite organizacije koriste različite alate za praćenje *taskova* ne postoji set univerzalnih pravila, no neka od češće spomenutih pravila su:

- Definirati obavezne informacije prilikom kreiranja *taska*
- Pravila dodjeljivanja i preuzimanja *taskova*
- Pravila otvaranja/zatvaranja *taskova* (tko, što, kako)

Jedan aspekt koji se ne bi mogao nazvati pravilom, ali ga je jednako važno istaknuti je percepcija aktivnosti. U intervjuima je navedeno kako nitko u razvojnom timu na aktivnosti ne bi trebao gledati kao dodatni posao, već kao prednost i pomagalo u radu. Kada govorimo o percepciji važno je ponovno istaknuti pravilnu edukaciju razvojnog tima o osnovnim konceptima projektnog menadžmenta, projektnih metodologija i pristupa. Kako je spomenuto u jednom od intervjuja važno je ne samo radi njih, već za cijelu organizaciju.

Iako su pretežito bili intervjuirani voditelji intervju s programerom prikazuje nam više od samo njegovog iskustva. Putem njegovih odgovora uviđamo koliku promjenu projektne metodologije mogu imati u relativno kratkom roku. Kako sam navodi metodologije nisu

integrirane u potpunosti, no integracija samo određenih aspekata rezultirala je značajnim pomakom u kvaliteti i brzini razvoja.

Generalni zaključak oko svih intervjuja je kako svaki intervjuirani pojedinac ima svoju listu prednosti koje im primjena projektnih metodologija i pristupa donosi u radu. Programerima pretežito olakšavaju organizaciju rada, kvalitetnije procjenjuju važnost pojedinih aktivnosti te im se količina stresa prilikom razvoja smanjuje. Voditeljima s druge strane omogućuju lakši pregled stanja pojedinog projekta, pravovremeno uočavanje rizika za projekt, omogućuje balansiranje očekivanja za vlasnika projekta i niz drugih prednosti. No jedan aspekt cijele priče koji se često izostavlja je put potreban za postizanje istinske sinkronizacije između svih uključenih u razvoj informatičkih sustava.

5.3. Kreiranje vlastitog *Way of Work*

Pitanje na koje su svi dali odgovor ne bilo je:

Smatrate li da razvojni tim može koristiti sve aspekte određene metodologije?

Na prvu ovakav odgovor može biti iznenađujuć, no uvijek moramo uzeti u obzir kako je svaka metodologija, svaki *framework*, svaki pristup u konačnici samo teorija. Povijesnim pogledom mnogi pristupi trebali bi funkcionirati, u teoriji naravno. No jedan se aspekt kontinuirano protivi svim teorijama koje čovjeka imaju kao ključan aspekt, nepredvidivost života. Jedna od produkcijskih baza podataka je prestala reagirati, no u tijeku je *sprint*. Znači li to da se nitko iz razvojnog tima ne smije razriješiti problem baze dok *sprint* ne završi? U nekim situacijama da, u nekim ne, no prema steknutim informacijama lako je zaključiti kako će velika većina će prvo htjeti razriješiti problem s bazom te zatim nastaviti sa *sprintom*. Znači li to da ne bismo smjeli primjenjivati *scrum* i *sprintove* ako nam se mogu dogoditi neočekivani problemi? Naravno da ne, neočekivani problemi su, ironično, očekivani u razvoju informatičkih sustava. Aspekt koji je ključno savladati i dogovoriti je način kako se nositi i kako postupati u slučaju takvih situacija. Samo zato što određena metodologija ne predviđa kako reagirati u pojedinoj situaciji ne znači nužno da ona nije prikladna za primjenu unutar organizacije. Uzimanjem, odbacivanjem i posuđivanjem raznih aspekata iz raznih metodologija organizacija može kreirati način rada koji najbolje odgovara njoj, ili njihov *Way of Work*. Kreiranje svog *WoW*-a nikako nije jednostavan i kratak zadatak, te je za kreiranje uspješnog *WoW*-a ponekad potrebno nekolicina godina, no u konačnici rezultira s pristupom koji je daleko bolji za organizaciju nego bilo koja od navedenih metodologija. Važno je napomenuti kako ovime ne isključujemo ideju da određene organizacije mogu u cijelosti koristiti sve aspekte pojedine metodologije, u takvim

slučajevima njihov *WoW* može biti ta metodologija. Za većinu organizacija dovoljno je početi s primjenom jedne metodologije (*sprintovi* iz *scruma* ili korištenje *kanban* ploče) i s vremenom uviđati kako primjena te metodologije pomaže ili odmaže članovima razvojnog tima. Uspješnost primjene pojedinog pristupa ili metodologije mogu se mjeriti na različite načine:

- Smanjenje potrebnog vremena za pojedine ponavljajuće aktivnosti
- Brzina realizacije aktivnosti
- Razlika u predviđenom i realno potrebnom vremenu za aktivnost
- Samostalnost razvojnog tima u organizaciji projekta

Navedene točke su izvedene na osnovi provedenih intervjua, te metrike po kojemu se uspješnost variraju od organizacije do organizacije. Neovisno o metrikama koje organizacija prati nužno je odabrati par metrika prema kojima možemo ustanoviti uspješnost primjene projektnih metodologija u organizaciji. Ukoliko se ustanovi da metrike predugo stagniraju ili čak unazađuju, nužno je reevaluirati metodologije koje koristimo.

Naravno ponovno je važno napomenuti kako prečesto mijenjanje pristupa također nije poželjno, stoga je prije odluke o promjeni pristupa ili odbacivanju samo jednog aspekta pojedine metodologije ili pristupa nužno savjetovati se s cijelim razvojnim timom. Organizacije koje slijede *PM²* metodologiju će najčešće inicirati mogućnost promjene pristupa tijekom *lessons learned* sastanka, no za organizacije koje ne slijede *PM²* metodologiju ovakve odluke najbolje je donositi na kraju projekta, osim ako trenutni pristup previše ugrožava aktualni projekt. Prilikom reevaluacije nužno je identificirati razlog zašto korišteni pristup nije efikasan za organizaciju. Pronalaženjem razloga možemo ustanoviti postoje li aspekti metodologije ili pristupa koji nam odgovaraju i njih uključiti u svoj *WoW*.

Ključni aspekt prilikom kreiranja svog *WoW*-a je kontinuirano unaprjeđenje. Saznanje o činjenici da određena metodologija ili određeni pristup nije prikladan za organizaciju također pomaže pri unaprjeđenju *WoW*-a. Kroz niz pokušaja, promišljanja i uspjeha s različitim pristupima sa sigurnošću možemo znati što funkcionira za našu organizaciju, a što ne.

6. Zaključak

Projektne metodologije drugačije utječu na svakog člana razvojnog tima, no ne nužno na način na koji bismo inicijalno pomislili. Kao što je uočljivo kroz intervjuje svatko svoj fokus postavlja na drugačiju prednost koju im primjena projektnih metodologija donosi u radu, te rijetko tko spominje mane tijekom primjene. „Mane“ ili „negativne“ aspekte projektnih metodologija intervjuirani pojedinci pretežito prepisuju manjku iskustva u njihovoj primjeni. Prema intervjuima možemo zaključiti kako projektne metodologije uistinu olakšavaju rad svima uključenima u razvoj informatičkih sustava, no komponenta koja je često izostavljena je vrijeme potrebno za uspješno primjenjivanje projektnih metodologija u razvoju. Kao što je navedeno doima se kako je budućnost projektnog menadžmenta usmjerena prema kreiranju vlastitog *Way of Work*, što je zapravo u jednu ruku prirodna evolucija. Garel je naveo kako je model menadžmenta karakteriziran je po kompaniji i/ili primjernim projektima,⁶¹ no kako bismo izbjegli situaciju gdje svaka organizacija ima pristup vođenja projekata nazvan po sebi koristimo izraz *WoW*. Izradom vlastitog *WoW*-a osiguravamo da svaki član razvojnog tima ima konkretne prednosti od korištenja projektnih metodologija ili pristupa. Slijedimo li pristup izrade *WoW*-a projektnu metodologiju prilagođavamo projektu i projektnom tim, što u konačnici ostvaruje prvobitni cilj cijelog projektnog menadžmenta, povećanje (ekonomske) učinkovitosti.

⁶¹ Garel, G. A history of project management models: From pre-models to the standard models, Cnam Paris, Lirsa, France, 2012.

7. Literatura

Baker, S.L., Critical Path Method (CPM), 2004.,

Beck, Kent. Beedle, Mike. van Benneku, Arive. Cockburn, Alistair. Cunningham, Ward. Fowler, Martin. Grenning, James. Highsmith, Jim. Hunt, Andrew. Jeffries, Ron. Kern, Jon. Marick, Brian. Martin, Rober C. Mellor, Steve. Schwaber, Ken. Sutherland, Jeff. Thomas. Dave *Manifesto for Agile Software Development*, 2001. <https://agilemanifesto.org/>, (26.07.2020.)

Bell, T.E., Thayer, T.A., *Software requirements: Are they really a problem?* , TRW Defense and Space Systems Group Redondo Beach, California, 1976.

Boehm, B.W., *A spiral model of software development and ehnhancement*, ACM SIGSOFT Software Engineering Notes, 1986.

Charvat, J. *Project Management Methodologies*, John Wiely & Sons Inc (Verlag), 2003.

Cleland, D. I., Gareis, R. *Global Project Manament Handbook, Planning, Organizing, and Controlling International Projects*, McGraw-Hill, 2003.

Davis, J. G. *The accidental profession. Project Management Journal*, 15 (*Special Summer Edition*), 6. 1984.

European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

Fayol, H. *Administration industrielle et générale; prévoyance, organisation, commandement, coordination, controle, Pariz, H. Dounod et e. Pinat* 1916.

Gantt chart history, <https://www.gantt.com/> , (20.07.2020.)

Garel, G. *A history of project management models: From pre-models to thestandard models*, Cnam Paris, Lirsa, France, 2012.

Golden Pryor, M., Taneja, S. *Henri Fayol, practitioner and theoretician – revered and relived*, Department of Marketing and Management, Texas A&M University-Commerce, Commerce, Texas, USA, 2010.

<http://www.extremeprogramming.org/rules/functionaltests.html> (28.07.2020.)

<https://web.archive.org/web/20100612142236/http://hsppm.sph.sc.edu/COURSES/J716/CPM/CPM.html> (21.7.2020.)

<https://www.computerworld.com/article/2585634/extreme-programming.html> (27.07.2020.)

John Leo Weber, *A History of PMI & Its Role in Project Management*,
<https://www.projectmanager.com/blog/history-of-pmi> (28.08.2020.)

Lee Copeland, *Extreme programming*, 2001.,

Leksikografski zavod Miroslav Krleža, *metoda*, *Hrvatska enciklopedija*, mrežno izdanje. 2020.
<https://www.enciklopedija.hr/Natuknica.aspx?ID=40437> (27.07.2020.)

Leksikografski zavod Miroslav Krleža, *metodologija*. *Hrvatska enciklopedija*, mrežno izdanje.
2020. <https://www.enciklopedija.hr/natuknica.aspx?id=40441> (27.07.2020.)

Leksikografski zavod Miroslav Krleža, *model*, *Hrvatska enciklopedija*, mrežno izdanje. 2020.
<https://www.enciklopedija.hr/Natuknica.aspx?ID=41453> (27.07.2020.)

Malcom, D.G., Roseboom, J.H., Clark. C.E., Fazar, W., *Application of a Technique for
Research and Development Program Evaluation*, Operations Research, 1959.

Patrick L. Healy, *Project Management: Getting the Job Done on Time and in Budget*. 1997.

Poppendieck, M., Poppendieck, T. *Lean Software Development: An Agile Toolkit*, Addison
Wesley, 2003.

Ryoce, W. W., *Managin the development of large software systems*, Proceedings of
IEEE WESCON, 1970.

Schwaber, Ken. SCRUM Development Process,
<http://www.jeffsutherland.org/oops/schwapub.pdf> (29.07.2020.)

Taylor, F.W. *The principles of scientific management*, 1991.

Thayer, H., *Management of the Hanford Engineer Works in World War II, How the Corps,
DuPont and the Metallurgical Laboratory fast tracked the original plutonium works*, ASCE
Press, 1996.

Wells, D., *Acceptance test*, 2000.,

Wells, D., *Code unit tests first*, 2000., <http://www.extremeprogramming.org/rules/testfirst.html>
(28.07.2020.)

Wells, D., *Pair programming*, 2000., <http://www.extremeprogramming.org/rules/testfirst.html>
(28.07.2020.)

A history of Kanban, <https://kanbantool.com/kanban-guide/kanban-history> (04.08.2020.)

History of PM², https://www.pm2alliance.eu/history-of-pm_2/ (27.08.2020.)

<https://kanbantool.com/kanban-guide/kanban-fundamentals/visualize-the-workflow>
(04.08.2020.)

Limit work in progress, <https://kanbantool.com/kanban-guide/kanban-fundamentals/limit-work-in-progress> , (04.08.2020.)

Making Process Policies Explicit, <https://kanbantool.com/kanban-guide/kanban-fundamentals/make-process-policies-explicit> , (04.08.2020.)

Managing and Measuring Flow, <https://kanbantool.com/kanban-guide/kanban-fundamentals/manage-and-measure-flow> (04.08.2020.)

Recognize Improvement Opportunities, <https://kanbantool.com/kanban-guide/kanban-fundamentals/improve-continuously> (04.08.2020.)

Scrum guide, <https://www.scrumguides.org/scrum-guide.html> , 2020. (31.07.2020.)

Visualizing the workflow,

What are PDUs?, <https://www.pmi.org/certifications/maintain/pdu> (31.08.2020.)

8. Dodatci

Slika 1. Primjer Ganttovog dijagrama, preuzeto s:

<https://commons.wikimedia.org/wiki/File:GanttChartAnatomy.svg>

Slika 2. Jednostavi *CPM* dijagram, preuzeto s:

<https://www.smartsheet.com/critical-path-method>

Slika 3. Standardna vizualizacija modela vodopada, preuzeto s:

https://commons.wikimedia.org/wiki/File:Waterfall_model.svg

Slika 4. Royceov finalni vizual svoje predložene metode, preuzeto s:

https://commons.wikimedia.org/wiki/File:1970_Royce_Managing_the_Development_of_Large_Software_Systems_Fig10.PNG

Slika 5. Vizualizacija spiralnog modela, preuzeto s:

<http://www-scf.usc.edu/~csci201/lectures/Lecture11/boehm1988.pdf>

Slika 6. Primjer *kanban* ploče, preuzezo s:

https://commons.wikimedia.org/wiki/File:Kanban_board_example.jpg

Slika 8. Aktivnosti i produkti faze planiranja, preuzeto iz:

European Commission Centre of Excellence in Project Management (CoEPM²), *PM² Project Management Methodology Guide 3.0*, Brisel, Luksemburg, 2018.

Projektne metodologije u razvoju informatičkih sustava

Sažetak

Prilikom razvoja informatičkog sustava preporučuje se korištenje projektnih metodologija kako bi se razvoj kretao što preglednije i jednostavnije. Svaka od metodologija predlaže određene faze i korake kako bi projekt nesmetano napredovao, te kako bi vrijeme sudionika na projektu bilo što kvalitetnije iskorišteno. Iako razvoj informatičkog sustava uključuje različite tipove sudionika, od svih se očekuje da se pridržavaju iste projektne metodologije, no da li je takav pristup poželjan? Pomažu li projektne metodologije uistinu svima u jednakoj mjeri ili je realnost ipak drugačija?

Ključne riječi: diplomski rad, projektne metodologije, informatičkih sustavi

Usage of project management methodologies in the development of information systems

Summary

When developing an information system, it is recommended to use project management methodologies in order to make the development as clear and simple as possible. Each of the methodologies suggests certain phases and steps in order for the project to progress smoothly, and for the time of project participants to be used as well as possible. Although the development of an information system involves different types of participants, everyone is expected to adhere to the same project methodology, but is such an approach desirable? Do project management methodologies really help everyone equally or is the reality different after all?

Key words: master's thesis, project methodologies, information systems