

UNIVERZA V LJUBLJANI
EKONOMSKA FAKULTETA

SKUPINSKO MAGISTRSKO DELO

**ANALIZA PROJEKTA IZBOLJŠANJA POSLOVNEGA PROCESA Z
UPORABO AGILNIH METODOLOGIJ**

Ljubljana, oktober 2024

DANILO MILIĆ
JAN MITROVIĆ

IZJAVA O AVTORSTVU

Podpisani Danilo Milić študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Analiza projekta izboljšanja poslovnega procesa z uporabo agilnih metodologij, pripravljenega v sodelovanju s red. prof. dr. Talibom Damijem

IZJAVLAM

1. da sem predloženo delo pripravil samostojno;
2. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
3. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
4. da se zavedam, da je plagiatstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
5. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
6. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;
7. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
8. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
9. da na Univerzo v Ljubljani neodplačno, ne izključno, prostorsko in časovno neomejeno prenašam pravice shranitve predloženega dela v elektronski obliki, pravice reproduciranja ter pravice dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
10. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi;
11. da sem preveril verodostojnost informacij, ki izhajajo iz zapisov na podlagi uporabe orodij umetne inteligence.

V Ljubljani, dne _____

Podpis študenta: _____

IZJAVA O AVTORSTVU

Podpisani Jan Mitrović študent Ekonomske fakultete Univerze v Ljubljani, avtor predloženega dela z naslovom Analiza projekta izboljšanja poslovnega procesa z uporabo agilnih metodologij, pripravljenega v sodelovanju s red. prof. dr. Talibom Damijem

IZJAVLAM

12. da sem predloženo delo pripravil samostojno;
13. da je tiskana oblika predloženega dela istovetna njegovi elektronski obliki;
14. da je besedilo predloženega dela jezikovno korektno in tehnično pripravljeno v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani, kar pomeni, da sem poskrbel, da so dela in mnenja drugih avtorjev oziroma avtoric, ki jih uporabljam oziroma navajam v besedilu, citirana oziroma povzeta v skladu z Navodili za izdelavo zaključnih nalog Ekonomske fakultete Univerze v Ljubljani;
15. da se zavedam, da je plagiatorstvo – predstavljanje tujih del (v pisni ali grafični obliki) kot mojih lastnih – kaznivo po Kazenskem zakoniku Republike Slovenije;
16. da se zavedam posledic, ki bi jih na osnovi predloženega dela dokazano plagiatorstvo lahko predstavljalo za moj status na Ekonomski fakulteti Univerze v Ljubljani v skladu z relevantnim pravilnikom;
17. da sem pridobil vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v predloženem delu in jih v njem jasno označil;
18. da sem pri pripravi predloženega dela ravnal v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil soglasje etične komisije;
19. da soglašam, da se elektronska oblika predloženega dela uporabi za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
20. da na Univerzo v Ljubljani neodplačno, ne izključno, prostorsko in časovno neomejeno prenašam pravice shranitve predloženega dela v elektronski obliki, pravice reproduciranja ter pravice dajanja predloženega dela na voljo javnosti na svetovnem spletu preko Repozitorija Univerze v Ljubljani;
21. da hkrati z objavo predloženega dela dovoljujem objavo svojih osebnih podatkov, ki so navedeni v njem in v tej izjavi;
22. da sem preveril verodostojnost informacij, ki izhajajo iz zapisov na podlagi uporabe orodij umetne inteligence.

V Ljubljani, dne _____

Podpis študenta: _____

KAZALO

1	UVOD	1
2	PROJEKTNI MANAGEMENT.....	4
2.1	Zgodovina projektnega managementa.....	6
2.2	Trikotnik projektnega managementa (čas, strošek, obseg)	7
2.3	Življenjski cikel projekta	8
2.4	Uspešnost projekta.....	9
2.5	Kriteriji uspeha	9
2.6	Dejavniki uspešnega projektnega managementa.....	11
2.7	PMBOK	11
2.8	Področje znanja.....	15
2.9	Problemi projektnega managementa	17
3	AGILNI PROJEKTNI MANAGEMENT	20
3.1	Ekstremno programiranje	23
3.2	Metoda dinamičnega razvoja sistema	24
3.3	Kristalna metoda.....	25
3.4	Scrum	26
3.4.1	Sprint	27
3.4.2	Vloge pri metodi Scrum	28
3.4.2.1	Skrbnik Scrum-a.....	28
3.4.2.2	Produktni lastnik	29
3.4.2.3	Razvojni tim	29
3.4.3	Potek metode	30
3.4.3.1	Produktna listina.....	30
3.4.3.2	Seznam uporabniških zgodb razvijajoče verzije produkta.....	31
3.4.3.3	Uporabniške zgodbe	31
3.4.3.4	Uporabniške točke.....	32
3.4.3.5	Planiranje sprinta.....	32
3.4.3.6	Seznam sprinta	33
3.4.3.7	Dnevni stoječi sestanek	33
3.4.3.8	Pregled sprinta.....	34
3.4.3.9	Retrospektiva sprinta.....	34
3.4.4	Prednosti in slabosti Scrum metode	35
3.5	Kanban.....	38

3.5.1	Koraki metode Kanban.....	40
3.5.1.1	Vizualizacija	40
3.5.1.2	Omejitev števila opravil	41
3.5.1.3	Management toka	42
3.5.1.4	Eksplicitne procesne politike	43
3.5.1.5	Hitre povratne zanke	43
3.5.1.6	Izboljšuj sodelovalno, razvijaj eksperimentalno	44
3.5.2	Problemi z uporabo metode.....	45
4	PREDSTAVITEV PODJETJA RENTAVAN	46
4.1	Trenutno stanje (AS-IS).....	48
4.2	Organizacijska struktura podjetja.....	51
4.3	Predlog izboljšanja procesa (TO-BE).....	51
5	UPORABA PMBOK NA PODJETJU RENTAVAN	54
5.1	Začetek projekta	54
5.1.1	Razumevanje projekta	54
5.1.2	Poslovni primer	54
5.1.2.1	Definiranje ciljev projekta	54
5.1.2.2	Definiranje stroškov oziroma proračuna projekta.....	55
5.1.3	Projektna listina.....	56
5.1.3.1	Opis projekta	56
5.1.3.2	Namen projekta	56
5.1.3.3	Mejniki.....	56
5.1.3.4	Ključni deležniki	57
5.1.4	Obseg projekta.....	57
5.2	Načrtovanje projekta	58
5.2.1	Management obsega projekta.....	58
5.2.1.1	Projektne aktivnosti	58
5.2.1.2	Projektna zaloga.....	61
5.2.2	Management časa	62
5.2.3	Management kakovosti	64
5.2.4	Management komunikacij	65
5.2.5	Načrt nakupa opreme	66
5.2.6	Management tveganj	67
5.3	Izvajanje in kontrola projekta.....	68
5.3.1	Izvajanje stroškov	69

5.3.2	Kontrola stroškov	71
5.3.2.1	Osnovna analiza prislužene vrednosti - PV, AC in EV.....	71
5.3.2.2	Načrtovano in stroškovno odstopanje (SV in CV).....	72
5.3.2.3	Indeks načrtovanega in stroškovnega odstopanja (SPI in CPI).....	73
5.3.2.4	Napredna analiza prislužene vrednosti - EV (ETC in EAC).....	73
6	UPORABA SCRUM-A NA PODJETJU RENTAVAN	75
6.1	Potek metode	77
6.2	Stroški projekta.....	85
6.3	Hitrost poteka.....	86
6.4	Napoved	86
7	UPORABA KANBAN-A NA PODJETJU RENTAVAN.....	89
7.1	Kanban tabla	89
7.2	Omejitve števila opravil.....	91
7.3	Potek metode	92
7.4	Nadzor nad metodo.....	94
7.5	Trajanje projekta.....	96
7.6	Stroški projekta.....	97
8	PRIMERJAVA TRADICIONALNEGA Z AGILNIM PROJEKTNIM MANAGEMENTOM	97
9	SKLEP	103
	LITERATURA IN VIRI	107
	PRILOGE.....	115

KAZALO TABEL

Tabela 1: Poslovni izidi 2017-2022.....	55
Tabela 2: Načrtovan nakup vozil za leto 2023	56
Tabela 3: Okvirno izvajanje projekta	64
Tabela 4: Management kakovosti.....	65
Tabela 5: Načrt obvladovanja tveganj	67
Tabela 6: Strošek projekta	70
Tabela 7: Stroški nakupa opreme	71
Tabela 8: Kumulativne vrednosti PV, AC, EV	72
Tabela 9: Izračun CV, CPI, SV, SPI.....	73
Tabela 10: Proračun ob zaključku projekta (BAC)	74
Tabela 11: EAC in ETC	74
Tabela 12: Prikaz EMV ob končanem sistemskem testu	74
Tabela 13: Projektna zaloga	77

Tabela 14: Primer uporabniških zgodb	78
Tabela 15: Uporabniške točke v prvem sprintu	79
Tabela 16: Uporabniške točke v drugem sprintu	81
Tabela 17: Uporabniške točke v tretjem sprintu	83
Tabela 18: Stroški projekta	85
Tabela 19: Napoved po opravljenem tretjem sprintu	87
Tabela 20: Število zgodb v različnih aktivnostih	95

KAZALO SLIK

Slika 1: Trikotnik projektnega managementa	7
Slika 2: Primer ponudbe na spletišču	48
Slika 3: Trenutno stanje v podjetju (AS-IS).....	50
Slika 4: Izboljšanje procesa (TO-BE)	53
Slika 5: Delna Struktura razčlenitve dela (WBS).....	58
Slika 6: Projektna listina z zalogo	62
Slika 7: Gantogram.....	63
Slika 8: Graf prislužene vrednosti za projekt po 4 mesecih	75
Slika 9: Potek prvega sprinta.....	79
Slika 10: Graf izgorevanja za prvi sprint	80
Slika 11: Potek drugega sprinta.....	80
Slika 12: Graf izgorevanja za drugi sprint	82
Slika 13: Potek tretjega sprinta.....	82
Slika 14: Graf izgorevanja za tretji sprint	84
Slika 15: Graf izgorevanja do šestega sprinta	84
Slika 16: Hitrost poteka po šestih sprintih	86
Slika 17: Grafikon naraščanja po treh sprintih.....	88
Slika 18: Grafikon naraščanja po šestih sprintih.....	88
Slika 19: Seznam uporabniških zgodb	90
Slika 20: Kanban tabla	90
Slika 21: Tabla na začetku prvega dne.....	92
Slika 22: Tabla na začetku drugega dne.....	92
Slika 23: Tabla na začetku tretjega dne.....	93
Slika 24: Tabla na začetku četrtega dne	93
Slika 25: Tabla na začetku petega dne	94
Slika 26: Tabla na začetku šestega dne	94
Slika 27: Kumulativni graf poteka	96

KAZALO PRILOG

Priloga 1: WBS konfiguracije programske opreme	1
Priloga 2: WBS ostale aktivnosti projekta	2
Priloga 3: Management komunikacij	3
Priloga 4: Potek četrtega sprinta.....	7

Priloga 5: Uporabniške točke v četrtem sprintu	8
Priloga 6: Graf izgorevanja za četrti sprint.....	9
Priloga 7: Potek petega sprinta	10
Priloga 8: Uporabniške točke v petem sprintu.....	11
Priloga 9: Graf izgorevanja za peti sprint.....	12
Priloga 10: Potek šestega sprinta	13
Priloga 11: Uporabniške točke v šestem sprintu	14
Priloga 12: Graf izgorevanja za šesti sprint.....	15

SEZNAM KRATIC

angl. – angleško

AC – (angl. Actual cost); dejanski stroški

API – (angl. Application Programming Interface); programski vmesnik

BAC – (angl. budget at completion); proračun ob zaključku projekta

CFD – (angl. cumulative flow diagram); kumulativni graf

CPI – (angl. cost performance index); indeks stroškovnega odstopanja

CRM – (angl. customer relationship management); program za management strank

CV – (angl. cost variance); stroškovno odstopanje

DSDM – (angl. Dynamic System Development Method); tehnika razvoja dinamičnih sistemov

EAC – (angl. estimated cost at completion); ocenjeni strošek ob zaključku

EV – (angl. earned value); prislužena vrednost

ETC – (angl. estimate time to complete); ocena časa za dokončanje

EVM – (angl. Earned value management); metoda prislužene vrednosti

GPS – (angl. Global Positioning Systems); globalni sistem pozicioniranja

NOAA – (angl. National Oceanic and Atmospheric Administration); Ameriška nacionalna uprava za oceane in ozračje

OBD – (angl. On-board diagnostics); senzor za diagnostiko

OTE – (angl. original time estimate); planiran čas projekta

PDCA – (angl. Plan, Do, Check, Act); Demingov cikel planiranja, izvedbe, preverjanja in ukrepanja

PERT – (angl. Program Evaluation and Review Technique); tehnika vrednotenja in pregleda projekta

PMI – (angl. Project Management Institute); Inštitut projektne managementa

PMBOK – (angl. Project Management Book of knowledge); Vodnik po znanju projektnega vodenja

PV – (angl. planned value); planirana vrednost

SLA – (angl. Service Level Agreement); Sporazum o ravni storitev

SMART – (angl. specified, measurable, attainable, relevant, and time-bound); metoda za postavljanje ciljev

SPI – (angl. Schedule Performance Index); indeks načrtovanega odstopanja

SV – (angl. Schedule variance); načrtovano odstopanje

WBS – (angl. work breakdown structure); struktura razčlenitve dela

WIP – (angl. Work in progress); omejitve števila opravil

XP – (angl. Extreme Programming); Ekstremno programiranje

ZDA – (angl. United States of America); Združene države Amerike

1 UVOD

Marsikatero malo ali srednje majhno podjetje se v poslovanju sreča s projekti ali prenovo svojih procesov. Tradicionalni pristop projektnega managementa v mnogo primerih prinaša časovne prekoračitve, višje stroške izvedbe ter posledično nezadovoljstvo naročnika oziroma končnega uporabnika. Cilj projekta je uspešna izvedba z najvišjo kvaliteto, z najmanj porabljenimi sredstvi ter v najkrajšem času. V realnosti se projekti neprestano srečujejo s spremembami, naloga projektnega tima je, da te spremembe predvidi ali minimizira njihov vpliv na končni rezultat. Zaradi rigidnosti tradicionalnega pristopa projektnega managementa se v praksi vedno bolj uporabljajo agilne metode projektnega managementa.

Projektni management je s strani Inštituta Projektnega Managementa (angl. Project Management Institute, v nadaljevanju PMI) opredeljen, kot pojem uporabe znanja, veščin, orodij in tehnik za projektne dejavnosti, da se izpolnijo projektne zahteve. Oprijemljivejši opis je, da je management projektov vse, kar je potrebno, da se projekt zgodi pravočasno in v okviru proračuna, da se doseže potreben obseg in kakovost (PMI, 2020).

Ker se projekti znotraj svoje izvedbe srečujejo s spremembami (od proračuna projekta, časovnih omejitev, kakovostnih zahtev), lahko rečemo, da je projektni management prilagajanje sprememb, s katerimi se srečuje projekt. Osredotoča se na razvoj posebej opredeljenih rezultatov projektov, ki jih je treba zagotoviti v določenem času, do določene kakovosti in z dano stopnjo virov, tako da bodo doseženi načrtovani rezultati. Učinkoviti management projektov je bistvo za uspeh projekta. Pri uporabi katerekoli splošne metode management projektov je pomembno upoštevati korporacijsko in poslovno kulturo, ki tvori okolje določenega projekta (Egovernment, 2011).

Večja podjetja imajo temeljne in podporne procese v svoji organizaciji. Davenport (1993) piše, »da je proces strukturiran, merljiv sklop aktivnosti, katerih cilj je ustvariti določen proizvod za kupca ali trg. Je posebna ureditev delovnih aktivnosti skozi čas in prostor, z začetkom in koncem in z jasno identificiranimi vhodi in izhodi. Struktura procesov je ključen element za doseg prednosti pri prenovi procesov«. Management poslovnih procesov (angl. Business process management) je pristop, ki obravnava problematiko poslovnih procesov v organizacijah. Glavna naloga je izboljšanje organizacijske zmogljivosti z managementom in optimiziranjem procesov podjetja, pri čemer zajema njihov celotni življenjski cikel (Davenport, 1993).

Agilni pristop oziroma agilnost organizacijam omogoča ohranjanje konkurenčnosti v primerjavi z drugimi podjetji. Agilnost pomeni hitro možnost prilagajanja in reagiranja na spremembe, s katerimi se projekti in procesi srečujejo med svojo izvedbo. Agilni pristop v primerjavi s tradicionalnim pristopom predvideva in pričakuje deviacije v projektu. To lahko zajema zahteve za izvedbo, časovne zamike in cilje projekta. Pogosto se zgodi, da se

naročnikove zahteve spremenijo med izvedbo, kar povzroči preglavice pri izvedbi projekta. Čas med zbiranjem zahtev in izvedbo se podaljša, kar privede do nezadovoljstva in neizpolnitve pričakovanj naročnika. Pri agilnem pristopu je naročnik bolj povezan in vključen v projekt. V primerjavi s tradicionalnim pristopom, kjer naročnik vidi rezultate po več mesecih, v agilnem pristopu prejema povratne informacije na tedenski ravni (Lester, 2014). Agilni razvoj programske opreme temelji na inkrementalnem iterativnem pristopu. Namesto poglobljenega načrtovanja na začetku projekta so agilne metode odprte za spreminjanje zahtev in spodbujajo stalne povratne informacije končnih uporabnikov. Med funkcionalni timi v določenem času delajo na iteracijah izdelka in to delo je organizirano v seznamu uporabniških zgodb (angl. backlog), ki jih je treba izvesti v okviru projekta. Agilni projekt se izvaja v iteracijah, cilj vsake iteracije je izdelava določenega dela. V agilnem načinu vodstvo spodbuja timsko delo, odgovornost in osebno komunikacijo. Poslovne interesne skupine in razvijalci morajo sodelovati, da izdelek uskladijo s potrebami strank in cilji podjetja. Agilnost se nanaša na vsak postopek, ki je usklajen s koncepti agilnega manifesta. Februarja 2001 se je 17 razvijalcev programske opreme sestalo v mestu Utah, Združenih državah Amerike in razpravljalo o lahkah razvojnih metodah. Objavili so manifest za agilno programsko razvijanje, ki opisuje, kako so našli boljše načine za razvoj programske opreme. Manifest vključuje štiri vrednote in dvanajst načel. Agilni manifest je dramatično nasprotje tradicionalnemu vodniku (angl. Project Management Book of knowledge, v nadaljevanju PMBOK) in njegovim standardom (Eby, 2017)

Scrum je najbolj priljubljena in široko uporabljena agilna metoda. Scrum okvir je bil zasnovan za obravnavo dveh ključnih postavk pri razvoju programske opreme, hitrost in spreminjajoče se zahteve strank. V tem pristopu se projekt razvoja programske opreme izvaja v iteracijah, vsaka iteracija je znana kot sprint, ki traja od enega do štirih tednov. Ustvariyo se majhni timi (5-10 članov), ki skupaj delujejo v smeri zelenih rezultatov. Metoda Scrum omogoča aktivno sodelovanje strank na vsaki stopnji, tako da se vse potrebne spremembe obravnavajo takoj tako kot tudi ukrepi, ki jih je treba narediti. To zagotavlja, da se projekt izvede v roku in učinkovito izpolni zahteve naročnika.

Kanban je vizualni sistem za management dela, ki se giba skozi celoten postopek. Kanban vizualizira tako postopek (potek dela) kot tudi dejansko delo, ki poteka skozi ta postopek. Kanban je metoda managementa delovnega toka (angl. workflow), zasnovana tako, da pomaga vizualizirati delo, povečati učinkovitost in približa k agilni miselnosti ter pripomore k pospešenemu izvajanju nalog in kvaliteti le teh. Iz japonščine je beseda Kanban dobesedno prevedena kot pano ali napis oziroma je sestavljena iz dveh besed in pomeni znakovna kartica (angl. signal card). Čeprav izvira iz proizvodnje je postal temelj razvoja programske opreme. Zaradi svoje učinkovitosti sej je začel uporabljati tudi v poslovnih enotah na različnih področjih.

PMBOK temelji na naučenih lekcijah (angl. lessons learned), ki zajema prakso managementa projektov, vključno s formalnimi, pisnimi okviri praks ter predhodno

neformalnimi in nenapisanimi znanji. Projektni management je sestavljen iz petih procesnih faz in desetih področij znanja. Vsako področje znanja zajema vseh pet faz. Vsak proces v načrtu projekta je medsebojno povezan z vsaj enim področjem in fazo. Ker PMBOK vključuje dinamične, integrativne procese, ki so v stalnem razvoju, ko se praksa spreminja in raste, se PMBOK redno posodablja. Podobno kot v drugih disciplinah vsebino metode definirajo, preizkušajo in reformirajo posamezniki, ki uporabljajo in širijo znanje v okviru projektnega managementa (Alby, 2008). Večina vodnika pokriva teme, kot so proračun, načrtovanje, finančno napovedovanje, osebje, organizacijsko vedenje in znanost o managementu. V skladu s podobnimi standardi managementa projektov je PMBOK oblikovan iz množice povezanih procesov, ki konvergirajo k dokončanju projekta. Po vodniku tri faze projekta vključujejo vložke, orodja in tehnike ter rezultate. Uporabljajo se za ustvarjanje splošnega življenjskega cikla projekta, v katerem so opisni tipični koraki in tehnike managementov projektov (Lewis, 2018).

Manjša podjetja v Sloveniji imajo v večini lastnike, ki so tudi direktorji. V mnogo primerih so lastniki skeptični do agilnih metod, saj nimajo vse pod nadzorom, potrebno je zaupanje v delavce in razvijalce ter v vse, ki so vključeni v prenovi procesa. Ker je pri prenovi procesov mnogo sprememb in se projekti ne izvedejo, tako kot je bilo planirano (čas, finance), agilne metode ponujajo bolj transparentni management ter boljši pregled nad časovno izvedbo in finančno ter kakovostno kontrolo.

Namen magistrskega dela je s pomočjo domače in tuje literature pregledati in razumeti učinke agilnih metod na izboljšanje procesa znotraj manjšega podjetja. Preučiti prednosti oziroma slabosti agilnih metod v primerjavi s klasično metodo projektnega managementa. V nalogi bodo podrobneje predstavljene agilne metode Scrum in Kanban ter tradicionalno metodo projektnega managementa PMBOK. V Sloveniji se agilne metode šele uvajajo v večje organizacije, v nalogi pa bomo želeli preučiti, kakšen vpliv, oziroma želene rezultate doprinesejo znotraj manjšega podjetja.

Agilne in klasične metode želiva preučiti na praktičnem primeru prenove procesa v podjetju Rentavan. Podjetje Rentavan se ukvarja z oddajo tovornih in potniških vozil, prevozi oseb in tovora na lokacije po vsej Evropi. Vozni park vsebuje 32 vozil. Zaradi povečanja števil strank, števila in okvar obstoječih vozil, povečanega povpraševanja po nabavi novih vozil, trenutnega poslovnega modela ni zmožen predelati in zadovoljiti potrebnih zahtev za učinkovito poslovanje. Komunikacija s strankami poteka preko telefona ali elektronskih sporočil, rezervacije se ročno vpišejo na liste papirja, ročno vnašajo v Excel datoteke, sledljivost servisov vozil se vodi ročno, večina procesa naročaja poteka v papirnati obliki. Vsi elementi procesa niso med seboj povezani. Digitalizacija praktično ne obstaja (pozitiven primer so shranjena elektronska sporočila, dostopna preko Windows operacijskega sistema, na NAS strežniku znotraj podjetja). Hranjenje podatkov je neučinkovito, saj je večina občutljivih podatkov v mapah.

Panoga je zelo konkurenčna. Na trgu se pojavljajo podjetja, ki ponujajo nižje cene, za kar se je podjetje Rentavan prisiljeno prilagoditi z bolj učinkovitim poslovnim modelom, tako da lahko zniža stroške in ponudi konkurenčnejšo ceno. Z novim poslovnim procesom najema vozila je podjetje tudi bolj fleksibilno na spremembe v poslovnem okolju. Tveganje predstavlja tudi primanjkljaj delovne sile, vendar bi se z novim informacijskim sistemom, ta nevarnost minimizirala. Vodenje evidence strank (zgodovina najemov), zgodovina vozil (servisi, gume, poškodbe) bi se vodilo iz baze podatkov, kar drastično zmanjša možnost napak, zamudnin, napačnih podatkov itd. Kot posledica bi bilo zadovoljstvo zaposlenih, vodstva in strank večje, kar bi vplivalo na poslovanje podjetja.

Glavni cilji magistrskega dela so:

- izvedba projekta izboljšanje poslovnega procesa v podjetju Rentavan z uporabo agilnih metod Scrum in Kanban,
- ugotoviti, zakaj se agilne metode šele uvajajo v Sloveniji in zakaj so managerji skeptični do njihove uporabe,
- primerjava rezultatov uporabe agilnih metod in tradicionalne metode PMBOK.

V magistrskem delu bova uporabljala deskriptivno metodo, saj bova preučevala odkrita dejstva o delovanju podjetja Rentavan in sekundarne analize, kot so članki in literatura o tej tematiki. Literatura bo pridobljena pri tujih in domačih avtorjih, sklicevala se bova na članke v visoko rangiranih revijah in kvalitetne spletne strani. Pri empiričnem delu naloge bo opravljena analiza del uspešnih in neuspešnih aplikacij agilnih in klasičnih metod v podjetjih. Z uporabo metod Scrum, Kanban, PMBOK bova prikazala prednosti, slabosti določenih metod in predlagala izboljšave le teh.

Pri izdelovanju magistrskega dela bova uporabljala znanje in kompetence, ki sva jih pridobila pri podiplomskem študiju poslovna informatika v okvirjih projektov pri predmetih smeri.

2 PROJEKTNI MANAGEMENT

Projektni management je eden najpomembnejših sestavnih delov uspešnega poslovanja. Vpliva na prihodke, obveznosti in na koncu vpliva na zadovoljstvo in zadrževanje strank. Manjša podjetja imajo po navadi naenkrat manjše število projektov, medtem ko večja podjetja, družbe žonglirajo z mnogimi projekti hkrati. Projekti so po svoji naravi začasni in predstavljajo sredstvo za doseganje cilja, ki bo sčasoma dosežen. Projekti spodbujajo naraščajočo potrebo po delovni sili. PMI je ocenjeval, da bo v obdobju 2010-2020 po vsem svetu dodanih več kot 15 milijonov novih delovnih mest za management projektov.

Management projektov ni celotno delovanje podjetja. To je samo en segment, določen projekt s podrobnim načrtom, kako posameznik in podjetje dosežeta določen cilj. Je

podrobno opisan načrt v korakih, od katerega je vsak enako pomemben. Če želite pravilno preiti na naslednji korak, morate doseči prejšnjega. Management projektov si lahko predstavljamo kot lestev, na katero se moramo povzpeti. Ne moremo takoj skočiti na vrh. Za največjo učinkovitost se je treba lotiti korak za korakom.

Tim mora uporabiti orodja, ki so jim na voljo ter njihovo strokovno znanje in znanje za izvedbo vsakega koraka in prehod na naslednji. Dovolj enostavno je reči, da želite priti do polja A, zato boste v tej smeri naredili 25 korakov. Vendar je treba v projektnem načrtu upoštevati faktor časa in najverjetneje bo treba upoštevati tudi določen proračun. Lahko se plazite tistih 25 korakov ali pa boste morda tekli. Odvisno od tega, kako hitro je treba priti tja za uspešen zaključek projekta. Denar lahko prihranite s potovanjem peš ali pa najemom voznika. Vse je odvisno od proračuna, ki je namenjen projektu. Ni nobenega pristopa, sistema ali načrta, ki bi ustrezal vsem. Vsak projekt, ki se ga boste lotili, bo najverjetneje časovno, ciljno in proračunsko opredeljen (Reh, 2019).

Projektni management je strukturiran način managementa sprememb. Osredotoča se na razvoj posebej opredeljenih rezultatov projektov, ki jih je treba zagotoviti v določenem času, do določene kakovosti in z dano stopnjo virov, tako da bodo doseženi načrtovani rezultati projekta. Učinkovit management projektov je bistvo za uspeh projekta. Pri uporabi katerekoli splošne metode managementa projektov je pomembno upoštevati korporacijsko in poslovno kulturo, ki tvori okolje določenega projekta. Povečane zahteve glede odgovornosti v javnem sektorju so privedle do večje osredotočenosti na uspešnost in uspešno izvajanje poslovanja. Roberts in Furlonger (2000) pišeta: »Podjetje Gartner ocenjuje, da uporaba zmerno stroge metode managementa projektov v primerjavi z ohlapno metodo izboljša produktivnost za 20 do 30 odstotkov« (Roberts in Furlonger, 2000). Uporaba formalnega okvira managementa, metode za projekte lahko pomaga pri doseganju formalnega soglasja s cilji projekta, razjasnitev obsega, določitev potrebnih virov, določanje odgovornosti za rezultate in uspešnost ter spodbujanje osredotočenosti na končne rezultate projekta, ki jih je treba doseči (Egovernment, 2011).

Uradna opredelitev projektnega managementa s strani PMI iz leta 1969 je opredeljen, kot pojem uporabe znanja, veščin, orodij in tehnik za projektne dejavnosti, da se izpolnijo projektne zahteve. Povzeto po Williamsu (2008) pomeni, da je management projektov vse, kar je potrebno, da se projekt zgodi pravočasno in v okviru proračuna, da se doseže potreben obseg in kakovost.

Na projekt se ne gleda kot na običajno delo, ampak kot na enkratno prizadevanje spreminjanja stvari. Na primer izdelava nove spletne strani predstavlja projekt, običajno delo pa predstavlja tekoče vzdrževanje in manjše posodobitve. Čas in proračun sta že znana izraza, morda naj bi projekt trajal šest tednov in imel proračun 5.000 evrov. Obseg se nanaša na seznam dogovorjenih rezultatov ali funkcij (tukaj je določena lestvica zahtevane rešitve). Na primer ustvarjanje novega spletnega mesta za podjetje je resnično mogoče končati v

šestih tednih, pisanje računovodske programske opreme pa ne.

Te tri omejitve (čas, proračun, obseg) tvorijo tako imenovani trikotnik projektnega managementa (Williams, 2008).

2.1 Zgodovina projektnega managementa

Čeprav projektni management deluje kot sodobna disciplina, njegovi začetki segajo v drugo polovico 19. stoletja. Takrat se je predvsem v Združenih državah Amerike (v nadaljevanju ZDA) začejalo industrijsko prebujanje. Izvajali so se veliki državni projekti, ki pa so bili zelo kompleksni in običajno na odročnih lokacijah (npr. železnice, jezovi itd.). Pri projektu gradnje železnice, ki bi prečkala celotno državo ZDA, so se vodje in projektanti soočili z zastrašujočo nalogo organiziranja delovne sile več tisočih delavcev ter predelave in sestavljanja neverjetnih količin materiala, potrebnega za dokončanje železnice. Tako so taki projekti spodbudili idejo o projektnem managementu (Microsoft, 2014).

Ob koncu 19. stoletja je Frederick Taylor začel preučevati delo in njegove procese. Želel je pokazati, da je mogoče analizirati in izboljšati delo, z osredotočanjem na njegove osnovne komponente. Njegova študija je temeljila na industrijskih obratih (npr. jeklarna), kjer je spremljal najmanjše in na videz nepomembne procese (npr. premikanje delov po jeklarni, nalaganje peska itd.). Do takrat je veljalo, da se produktivnost poveča tako, da se od delavcev zahteva več dela in daljši delovni čas. Taylor je namesto težjega ter daljšega dela predstavil in uvedel nov koncept bolj učinkovitega dela. Še večji preboj v projektnem managementu je dosegel Henry Gantt, ki je podrobno preučeval vrstni red delovnih operacij. Na njegove študije je močno vplivala prva svetovna vojna, kjer je preučeval management gradnje mornariških ladij. Njegov izum je gantogram, ki opisuje zaporedje in trajanje vseh nalog v procesu. Ta izum se je izkazal za tako pomembnega in za tako močno analitično orodje v managementu, da je po stoletju obstoja ostal skoraj nespremenjen (Microsoft, 2014).

Začetki modernega projektnega managementa segajo v obdobje druge svetovne vojne, kjer so obsežni in zahtevni vladni in vojaški projekti, ob zmanjšani delovni sili, zahtevali nove organizacijske strukture. Zasnovani so bili zapleteni diagrami, imenovani diagrami PERT (angl. Program Evaluation and Review Technique) oziroma tehnike managementa in pregledovanja projekta in metoda kritične poti (angl. critical path method), kar je managerjem omogočilo več nadzora nad zapletenimi in obsežnimi projekti (Haughey, 2010).

Z dvigom kapitalizma po drugi svetovni vojni in prihod hitro spreminjajočega in konkurenčnega sveta so managerji iz različnih panog začeli nove tehnike managementa uporabljati pri iskanju novih strategij managementa in različnih orodij za obvladovanje rasti. V poznih šestdesetih letih so podjetja začela uporabljati splošne systemske teorije za poslovne interakcije. Richard Johnson, Fremont Kast in James Rosenzweig so prvi opisali podjetje kot človeški organizem, ki ima skeletni sistem, mišični sistem, krvni sistem, itd.

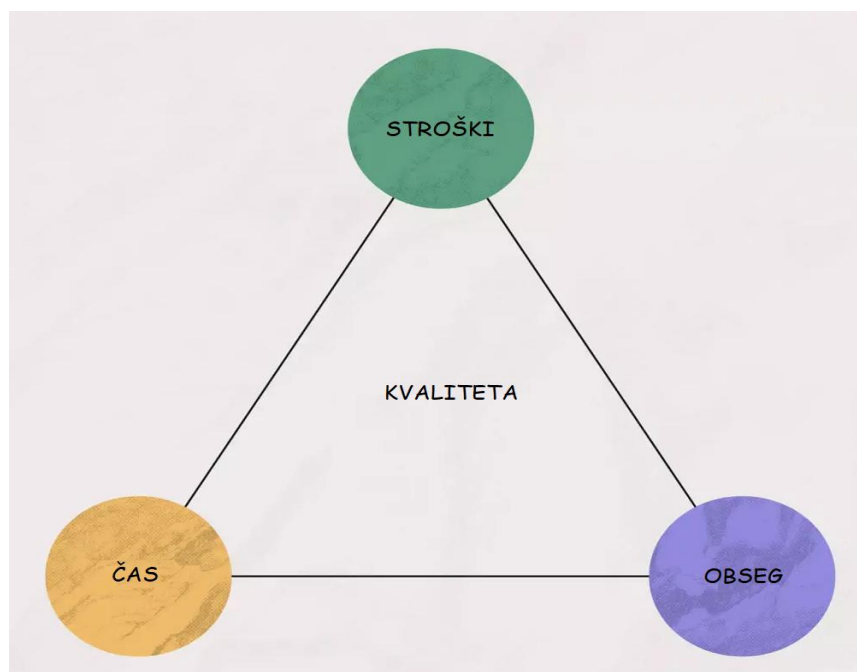
Pogled na podjetje, kot človeški organizem pomeni, da so vsi sistemi, ki v njem delujejo, pomembni za uspeh podjetja, morajo delovati usklajeno v smeri specifičnih ciljev ali projektov. V zadnjem desetletju se je management projektov še naprej razvijal. Predvsem sta se pojavila dva pomembna trenda (Microsoft, 2014):

- načrtovanje od spodaj navzgor. Ta princip poudarja enostavnejše zasnove projektov, krajše projektne cikle oziroma iteracije ter učinkovito sodelovanje, vključenost in komuniciranje med vsemi deležniki projekta. To je izredno pomembno pri sprejemanju pomembnih odločitev. Princip je znan tudi kot agilni management in vključuje številne metode, kot so Scrum, Kristalna metoda, Ekstremno programiranje in mnoge druge,
- načrtovanje in pregledovanje od zgoraj navzdol. Za ta princip je značilno sprejemanje odločitev na nivoju celotnega podjetja o projektih, ki bi jih morala imeti organizacija, s podporo različnih sodobnih tehnologij (npr. podatkovno rudarjenje).

2.2 Trikotnik projektnega managementa (Čas, strošek, obseg)

Projektni managerji uporabljajo trikotnik projektnega managementa, da svojemu timu pomagajo ustvariti kakovosten izdelek. To dosegajo z obvladovanjem treh omejitev: časa, stroškov in obsega, ki so prikazani na sliki 1. Tako zmanjšajo projektna tveganja, omogočijo lažji management sprememb in poenostavijo komunikacijo z vsemi deležniki projekta.

Slika 1: Trikotnik projektnega managementa



Vir: prirejeno po ASANA (2024).

Trikotnik predstavlja model projektnega managementa. Vse tri stranice trikotnika so med seboj povezane in jih je zato treba nenehno uravnavati. Ko se spremeni ena od omejitev, je

treba prilagoditi eno ali obe omejitvi, da se ohrani kakovost, ki je znotraj trikotnika. Projektni managerji tako lažje določajo in upravljajo prednostne naloge projekta in imajo jasno komunikacijo s strankami. Z dobrim managementom teh dejavnikov je večja verjetnost, da bo projekt ustvaril kakovosten izdelek. Trikotnik jasno prikazuje korelacijo med obsegom, stroški in časom. Na primer, če ima projekt strogo določen rok in pride do spremembe obsega projekta, projektni managerji lahko prilagodijo tretjo omejitev. Poznavanje treh omejitev tako omogoča lažje določanje in management prednostnih nalog projekta. Posebej pri komunikaciji s strankami projektni trikotnik ponuja jasno sliko, ki pomaga razložiti odvisnosti med stroški, časom in obsegom ter kako ob spremembah omejitve vplivajo druga na drugo. Tako projektni manager pomaga razumeti vpliv sprememb v jasnih in laičnih izrazih. S tem se tudi gradi zaupanje med projektnim timom in stranko.

Poznavanje, management in nadzorovanje treh omejitev znatno zmanjša tveganje, s katerim se lahko sooča projekt. Poznavanje ključnih prednostnih nalog omogoča upoštevanje tveganj pri načrtovanju projekta. Pomembno je, da se pri managementu sprememb zagotovi, da sprememba ene omejitve ne poruši druge in tako ogroža kakovost končnega produkta. Tudi če se pojavijo tveganja, dobra komunikacija s stranko, preprečuje, da bi se zadovoljstvo in kakovost poslabšala (Farrar, 2021).

Obseg, čas in stroški predstavljajo tri stranice trikotnika. Ko so v ravnotežju, zagotavljajo kakovost. Naloga projektnega managerja je, da se ohranja ravnovesje, tako da se trikotnik ne zlomi. Obseg predstavlja velikost projekta in zajema kakovost, kompleksnost, količino ali obseg, številko funkcij in zmogljivosti končnih izdelkov. Projektni manager mora znati upravljati in nadzorovati dodatne zahteve, saj lahko povečanje le teh negativno vpliva na obseg projekta in tako poruši ravnotežje s stroški in časom projekta. Sprememba lahko drastično poviša stroške projekta. Stroški vključujejo vse razpoložljive vire, ki jih projekt potrebuje za uspešno dokončanje. To zajema znanje, surovine, opremo, prostore, inventar, orodja itd. Strošek in čas sta tesno povezana, saj lahko izostanek enega člana projektnega tima znatno vpliva na dolžino projekta. Na čas se v nekaterih primerih gleda tudi kot na vir projekta, ki lahko vrže trikotnik iz ravnovesja. Časovna komponenta je tretja stranica projektnega trikotnika, ki predstavlja celotno časovnico projekta, število delovnih ur, ki jih projektni tim opravi med projektom, časovne iteracije oziroma intervale, potrebne za doseg mejnikov, ter čas za načrtovanje in oblikovanje strategije projekta. Management časa običajno zahteva prilagoditve na drugih dveh komponentah ali stranicah trikotnika. Da se na primer ohrani časovnica projekta, se zaposli nov član (Rudder in Main, 2023).

2.3 Življenjski cikel projekta

Življenjski cikel projekta predstavlja linearno napredovanje projekta. To pomeni od definiranja in določitve projekta, izdelave načrta, izvedbe do zaključka projekta. Definiranje, načrtovanje, izvajanje in nadzor predstavljajo odločitvene točke projekta, ki jih določi, spremlja in upravlja projektni manager (Verzuh, 2021):

- definiranje oziroma določanje: faza se začne, ko sta določena projektni manager in projektna listina in se zaključi, ko so potrjena vsa potrebna pravila projekta. Odobritev dokumenta pomeni, da se vse zainteresirane strani strinjajo s cilji, proračunom in obsegom projekta,
- načrtovanje: naslednja faza se nadaljuje, ko so pravila projekta odobrena. Projektni manager začne graditi projektni načrt. V tej fazi se načrt in tudi pravila običajno pogosto spreminjajo, saj se določajo podrobnosti o izvedbi projekta. Zato je ob zaključku faze načrtovanja treba odobriti ne le načrt, ampak tudi vse spremembe pravil iz prve faze. Ker se obe fazi lahko hitre, se pogosto zgodi, da se izvedejo večkrat, preden se potrdi končna različica ali pa se fazi izvajata istočasno, vendar to ni priporočljivo, saj mora biti načrt podrobno izdelan. Če so osnovne predpostavke in dogovori, ki so izdelani med fazo definicije, je običajno faza načrtovanja bolj produktivna,
- izvedba: v fazi izvajanja se opravi dejansko delo na projektu, kot je bilo potrjeno v načrtu. Večino časa in truda zajame prav ta faza. Ko je cilj projekta dosežen, je faza izvedbe končana,
- zaključitev: faza zaključka je najhitrejša od vseh štirih faz, vendar je ena najpomembnejših. Tukaj se opravljajo tri pomembne funkcije: prehod v naslednjo fazo razvoja novega izdelka ali operacije, formalen zaključek projekta s strani stranke in pregled uspešnosti oziroma neuspešnosti projekta z namenom izboljšave prihodnjih projektov.

Treba je poudariti pomen prvih dveh faz projekta. Čeprav definiranje in načrtovanje predstavljata 10 odstotkov celotnega projekta, sta bistveni pri pripravi projektnega tima na efektivno in kakovostno delovanje v fazi izvedbe (Verzuh, 2021).

2.4 Uspešnost projekta

Namen projektov je narediti nekakšno spremembo v organizaciji. Te spremembe se merijo z uspešnostjo časa, proračuna in obsega. Čas je izrednega pomena pri zagotavljanju uspešnosti projekta. Izdelek je treba izdelati po določenem urniku in dostaviti pravočasno. Vsak projekt vsebuje proračun in stroške, ki nastanejo pri njegovem izvajanju. Delujejo kot naložbe, ki lahko prispevajo k organizaciji ali pa jo stanejo več, kot so v projekt vložili. Obseg projekta je rezultat, ki zadovolji zahteve stranke. Vendar pa določitev obsega in njegove uspešnosti težje določiti ali bo projekt, končan pravočasno in znotraj, proračuna. Cilji obsega so določeni z zahtevami. In prav ti cilji morajo doprinesti rezultat, ki bo izpolnil namen in zagotovil vrednost uspešnega projekta. Po mnenju Verzuha (2021) primeri ciljev obsega so poslovne zahteve (npr. kaj naj bi produkt delal) in zmogljivostne zahteve (npr. ali je izdelek funkcionalen).

2.5 Kriteriji uspeha

Parametri, zaradi katerih se bo ocenilo, ali je projekt bil zaključen uspešno ali ne in zadovoljil

potrebe stranke, se imenujejo kriteriji uspeha. Čeprav je lahko projekt končan znotraj zastavljenega časovnega, proračunskega in kakovostnega okvirja, ta pogled ne pokriva zadovoljstva stranke. Če pričakovanja projekta niso izpolnjena oziroma projektni managerji teh pričakovanj ne razumejo in ne obvladujejo, se projekt lahko smatra kot neuspešen. Pri ocenjevanju uspešnosti projektnega managementa je treba upoštevati veliko stvari. Nekaterih ni mogoče ovrednotiti, dokler projekt ali naloga ni končana oziroma opravljena. Treba se je zavedati, da uspešen projektni management in uspešen projekt pomenita dve različni stvari (Verzuh, 2021).

Nekatere kriterije uspeha je težko določiti, saj lahko trajajo tedni, meseci ali celo leta, da se popolnoma razkrije njihov učinek na uspešnost. Vendar pa se lahko določeni kriteriji za uspeh projekta določijo vnaprej in se s pomočjo njihovih karakteristik določi uspešnost projekta (Simplilearn, 2023):

- obseg: če obseg in cilji projekta niso dobro načrtovani in definirani, lahko projektni tim izgubi fokus tekom projekta. Zato, da ne pride do zmede, običajno skrbijo projektni manager in deležniki projekta, vendar se lahko v primeru večjih projektov cilji in zahteve razbijejo na manjše, bolj kontrolirane cilje za posamezni tim. Cilj mora biti določen, merljiv, dosegljiv, relevanten in časovno omejen ali krajše SMART (angl. Specified, Measurable, Attainable, Relevant, and Time-bound),
- urnik: običajno se večina projektov sooča z zamudami, vendar je obvezno določiti rok projekta pred njegovim začetkom. Zato je dobra ideja, da se alocira nekaj dodatnega časa, tako da se lahko projekt izvede v dogovorjenem roku,
- proračun: projektni manager in deležniki projekta morajo oceniti stroške in povrnitev naložbe. Tako se določi in razporedi proračun. Projekt mora biti končan v okviru proračunskega načrta. Projekt se šteje za uspešnega, če je končan znotraj tega okvirja,
- cilji stranke: cilji stranke morajo vedno biti na prvem mestu. Projekt je lahko neuspešen, če ni v skladu z zahtevami stranke. Potrebno je spremljati in upoštevati strankine povratne informacije tekom celotnega projekta.
- kvaliteta: glavni cilj katerega koli projekta je zagotavljanje najboljše možne kakovosti. Tim izkazuje profesionalnost s tem, ko doseže in opravi dodatno delo za stranko,
- cilji tima: cilji tima niso samo uspešno dokončanje projekta, ampak tudi kaj se je in bo naučil, tako da lahko pridobljeno znanje vpliva na nadaljnje projekte. Ker je tim globoko vpleten v projekt in pozna vsako njegovo malenkost, je s strani projektnega managerja pridobivanje povratnih informacij tima zelo pomembno,
- končni izdelki: končni izdelki morajo zagotavljati kvaliteto in opravljati delo, za katero so bili proizvedeni/razviti. Torej se uspešnost projekta ne zagotovi samo po izdaji izdelka, ampak se spremlja tudi njegova uporaba po zaključku projekta,
- zmogljivost vira: projektni manager mora efektivno in racionalno alocirati vse vire, ki so potrebni za izvedbo projekta,
- management tveganj: prepoznavanje in razumevanje tveganj pripomore k uspešnosti

projekta. Vsak projekt ima otipljive in neotipljive nevarnosti, ki jih projektni manager mora predvideti.

- dokumentacija: celotna projektna dokumentacija prikazuje jasno sliko vsake faze projekta. Dobro pripravljena dokumentacija je lahko v pomoč tudi pri nadaljnjih projektih.

2.6 Dejavniki uspešnega projektnega managementa

Dejavniki, ki vplivajo na uspešnost projekta, zajemajo tudi sposobnost managementa. Zato so običajno projektni managerji tudi izjemne vodje. Značilnosti, ki spremljajo dobre managerje, so običajno sposobnost motiviranja, imeti jasno vizijo in združiti ljudi k skupnemu cilju. Vendar sposobnost projektnega managerja ni glaven faktor pri zagotavljanju uspešnosti projekta. Verzuh (2021) je določil pet karakteristik oziroma faktorjev:

- sporazum med projektnim timom, strankami in vodstvom glede ciljev projekta. Cilji morajo biti jasno zastavljeni, saj je to običajno razlog za neuspeh projekta,
- načrt celotnega poteka projekta ter definirane zadolžitve in odgovornosti vseh deležnikov. S tem se lahko spremlja uspešnost že med samo izvedbo projekta. Ker je vsak projekt edinstven, je jasno definiran načrt obvezen pri razumevanju in izvedbi, predvsem pa, kaj se lahko izvede in kaj ne. Načrt mora zajemati vsa področja projektnega managementa. To so človeški viri, finance, oprema in material, ki so potrebni za dokončanje projekta. Če je načrt dobro zasnovan, se lahko predvidi, ali bo projekt presegel zastavljeni proračun,
- stalna in efektivna komunikacija med vsemi deležniki projekta. Uspešen projekt je rezultat uspešnega dogovarjanja med ljudmi o ciljih projekta in njihovo uresničitev. Od koncepta do izvedbe je uspeh, odvisen od sposobnosti dogovarjanja, usklajevanja, pravičnega ukrepanja, prepoznavanja in reševanja problemov ter odzivanje na spremembe,
- nadzorovan obseg projekta. Uspešen projektni manager zagotavlja, da vsi deležniki projekta natančno razumejo, kaj je bilo in kaj še bo izvedeno v danem časovnem in finančnem okviru. Tako je management pričakovanj deležnikov izredno pomembna naloga, ki jo je treba izvajati čez celoten proces, saj se pogosto projekti soočajo s konstantnimi spremembami. Zato se naročniki ne smejo strinjati samo z začetnim izvirnikom obsega projekta, vendar morajo razumeti tudi morebitne spremembe obsega,
- podpora managementa. Dobri managerji morajo znati delegirati delo, saj sami običajno ne morejo imeti pregleda nad celotnim projektom.

2.7 PMBOK

Leta 1969 je pet prostovoljcev ustanovilo PMI (Weber, 2018). Njihov cilj je bil določiti standarde za management projektov, izvajati raziskave za izboljšanje načina managementa

projektov in vedno večjemu številu projektnih managerjev zagotoviti možnost izmenjave znanj in izobraževanja v disciplinah projektnega managementa. Od takrat je ameriški nacionalni inštitut za standarde (angl. American National Standard Institute) priznal PMI kot akreditiranega razvijalca standardov. Eden poseben standard je vodnik po knjigi znanja projektnega managementa bolj znane kot PMBOK. Standard se je začel uporabljati leta 1987 kot poskus standardizacije informacij in praks projektnega managementa, ki jih skupnost projektnih managerjev splošno sprejema kot dobro prakso (ITBusinessEdge, 2010).

PMBOK opisuje celo vrsto teorij in učenja, ki zajema prakso managementa projektov, vključno s formalnimi pisnimi okviri praks ter predhodno neformalnimi in nenapisanimi znanji. Projektni management je sestavljen iz petih procesnih skupin oziroma faz in desetih področij znanja. Vsako od teh področij znanja vsebuje še pet procesnih skupin. Vsako področje znanja potrebuje vseh pet procesnih skupin, da bi lahko delovalo. Vsak proces v načrtu management projektov postane medsebojno povezan z vsaj enim področjem in eno procesno skupino. Ker PMBOK vključuje dinamične, integrativne procese, ki so v stalnem razvoju, ko se praksa spreminja in raste, se PMBOK redno posodablja. Podobno kot v drugih disciplinah vsebino knjige znanja projektnega managementa definirajo, preizkušajo in reformirajo posamezniki, ki uporabljajo in širijo znanje v okviru projektnega managementa (Alby, 2008). Večina vodnika zajema teme, ki so značilne za management projektov, vendar omenja širše teme, ki se prekrivajo, kot so proračun, načrtovanje, finančno napovedovanje, osebe, organizacijsko vedenje in znanost o managementu.

V skladu s podobnimi standardi managementa je PMBOK oblikovan okoli procesov, ki sovpadajo in vplivajo drug na drugega za dokončanje projekta. Po vodniku tri faze projekta vključujejo vložke, orodja in tehnike ter rezultate. Uporabljajo se za ustvarjanje splošnega življenjskega cikla projekta, v katerem so opisani tipični koraki in tehnike managementu projektov (Lewis, 2018). Obstaja pet določenih faz management projektov. Te faze predstavljajo časovni načrt za doseganje zastavljenih ciljev vsakega projekta. Faze vključujejo (Eby, 2017):

Zasnova in začetek:

Tu bi raziskovali, kako naj se razvije projektna listina. V projektni listini je navedeno, kaj naj bi projekt dosegel, kdaj bi bil dosežen, z uporabo katerih virov in zakaj se bomo lotili projekta. Poleg tega bo naveden status financiranja in kaj naročnikova organizacija pričakuje od zaključenega projekta. V bistvu gre za krajše poročilo o projektu od naročnika, zato morajo biti managerji projektov sposobni analizirati listino in razjasniti vse elemente, za katere je potrebna jasnejša opredelitev ali o katerih se je treba pogajati. V prvi fazi se opredelijo in določijo cilji projekta. V tej fazi se izvedljivost določi z razpravo in testiranjem izvedljivosti. Ko projekt zainteresirane strani sprejmejo, se oblikuje listina o projektu ali dokument o začetku projekta, v katerem so navedeni cilji in zahteve projekta.

Opredelitev in načrtovanje:

Na tej stopnji se oblikuje načrt, ki bo skupinam pomagal usmerjati določene cilje v celotnem projektu. Opredeljen je obseg projekta in razvit je celovit projektni načrt. V načrtu so opredeljene spremenljivke, kot so stroški, kakovost, razpoložljivi viri, opredeljeni so tudi kritični dejavniki uspeha in tveganja, pripravljeni pa so tudi načrti za izredne razmere. Proces si prizadeva zajeti vsako priložnost, da se zagotovi uspeh projekta. Združuje vseh deset področij znanj managementa projektov. Projektni načrt tudi zajema merilnike uspešnosti, ki se določijo na podlagi obsega, časovne razporeditve in stroškov projekta.

Zagon ali izvedba:

Faza izvajanja projekta se začne, takrat ko se končni rezultati (angl. deliverables) razvijejo in dokončajo. V tej fazi je veliko gibljivih delov, kot so razvoj tima, dodeljevanje virov, sledenje sistemu, načrtovanje projektov in spreminjanje načrtov. V tej fazi se zaključi bistvo projekta, dosledna poročila o stanju in posodobitve pa poskrbijo, da se vsi cilji preverjajo. Poleg zagotavljanja, da tim ali izvajalci izvajajo fizična dela, ki jih projekt sestavlja, procesi vključujejo tudi zagotavljanje kakovosti, management tima, komunikacijo, nabave in management poslov zainteresiranih strani. Čeprav priročnik PMBOK nikoli ne more določiti tehničnih podrobnosti vseh vrst projektov, vam daje okvir za uporabo najboljših praks na teh širokih področjih managementa projektov.

Uspešnost in nadzor:

Ta faza je usmerjena v merjenje napredka in uspešnosti projektov, da se zagotovi, da so vse naloge usklajene s prvotnim načrtom projekta. Ključni kazalniki uspešnosti pomagajo ugotoviti, ali projekt poteka na podlagi komponent, kot so cilji projekta, kakovostni rezultati, napor in sledenje stroškov, uspešnost projekta, potrjevanje rezultatov glede na obseg projekta v povezavi z zainteresiranimi stranmi ter nadzor komunikacij in naročil. V okviru tega procesa bo vsak element analiziran glede odstopanj ali tveganj in po potrebi se izvede korektivni ukrep. Vse spremembe, ki se jih bo management projekta odločil implementirati, bodo vključene v načrt managementa projekta.

Zaključek projekta:

Ta faza signalizira zaključek projekta, vendar se to ne zgodi samo na koncu projekta. Veliki projekti so sestavljeni iz faz in pri vsaki bo potekal postopek zapiranja oziroma zaključevanja. Naloge managerjev projektov so, da prekinejo delo izvajalcev, prepoznajo dragocene člane tima in ocenijo projekt, da prepoznajo slabosti in prednosti. Ko je projekt končan, morajo projektni managerji sestaviti seznam neizpolnjenih nalog in opraviti končno analizo proračuna, ki bo vključena v končno poročilo o projektu. Vendar pa obstaja končni postopek zaprtja, v katerem se dokumentacija formalno dokonča in zaključi, sprostijo se

zunanji izvajalci in dobavitelji, dokončan projekt postane funkcionalen. Vsi dokumenti morajo biti zbrani in shranjeni za nadaljnjo uporabo.

Ljudje porabijo večino časa v fazah izvajanja in nadzora, dejansko opravljajo naloge, izdelujejo izdelek in poskrbijo, da je vse na pravi poti. Seveda je to delo zelo dragoceno, brez njega sploh ne bi bilo smisla zagnati projekt, vendar se v teh fazah običajno ne narekuje uspeh ali neuspeh projekta. To se dogaja v drugih treh fazah (začetek, načrtovanje, zaključek) zaradi česar se te faze tretirajo kot najpomembnejše. Vendar zakaj so te faze tako pomembne? Treba je razmisliti o posledicah, če bi bile te faze slabo končane ali celo popolnoma prezrte. Če se začetka projekta ne lotimo pravilno, se pogosto podjetja znajdejo v situaciji, ko imajo člani projektne skupine zelo različne predstave o namenu projekta in se na koncu ne strinjajo o tem, v katerem trenutku je projekt res končan. Če imamo za uspeh dober dizajn in nekaj statičnih spletnih strani, vendar je strankina zahteva prvi rezultat v brskalniku Google, lahko ponudimo odličen izdelek, ki ga ocenjujejo kot neuspešnega. Začetna faza je priložnost, da se vsem, ki so v skupini že od začetka, zagotovi, da so na isti strani in da se odpravljajo napačne predstave in konflikti, da se z njimi med projektom ne srečujemo. Dobra zasnova bo poskrbela, da se vnaprej lahko določijo vse zainteresirane strani na projektu, kar preprečuje verjetnost, da se bodo pojavili ob neprimernem trenutku med projektom (Weber, 2018).

Neuspešno izvedena faza načrtovanja ima lahko na projekt katastrofalni vpliv. Če se slabo ali pa sploh ne načrtuje, kako bomo vedeli, kaj bomo počeli. Naivno in nevarno je tudi načrtovanje celotnega projekta ob samem začetku in pričakovati, da se bo načrtu lahko sledilo. Težko je načrtovati, kaj naj se stori čez nekaj dni, kaj šele čez nekaj mesecev. Najboljši pristop k načrtovanju je po navadi tisti, ki omogoča natančno načrtovanje neposredne prihodnosti projekta in načrtovanje nalog, ki so na višji prioritetni ravni. To je znano kot pristop kotalnega valovanja (angl. rolling wave). Naloge v naslednjih treh do štirih tednih so razdeljene na dele, tako da je mogoče vsak dan spremljati napredek projekta. Vse, kar je daljše od enega meseca, ostane nenačrtovano kot naloge visoke prioritete, tako da se lahko spremlja, kaj nas čaka, ne da bi se preveč osredotočali na podrobnosti (Williams, 2008).

Če se ne posveča pozornost zaključku projekta, je lahko to prav tako problematično kot slaba zasnova ali načrtovanje. V večini primerov se šteje, da je projekt končan z izdelavo izdelka, vendar ni tako. Zainteresirane strani se lahko znova in znova vračajo, zahtevajo le še eno spremembo ali vztrajajo, da je treba odpraviti preostale napake in težave, ki jih najdejo. Del zaključka projekta z odliko je zagotavljanje, da ima izdelek, ki se je zgradil, prihodnost. Če se nagibamo k izbiri pogodbe o podpori za vse projekte, ki jih izvajamo, je potrebno fazo zaključevanja pravilno izvesti z ad hoc dogovori. Če nameravate prihodnje vzdrževanje in podporo izdelka izročiti komu drugemu, morda osebi, ki je znotraj organizacije stranke, potem je potrebno za to poskrbeti. Samo, da se izdelek preda hitreje in se tako lotimo naslednjega projekta, bo to verjetno zagotovilo nezadovoljstvo tako na eni kot drugi strani

(Williams, 2008).

2.8 Področje znanja

PMBOK poleg ustvarjanja postopkov, iz katerega je projekt sestavljen, zajema najboljše prakse na desetih področjih znanja o managementu projektov. Ta področja znanja so ključne komponente, povezane z uspešnim managementom projektov. Seznam vseh desetih komponent vključuje (Kothari, 2024; PMI, 2013):

Management integracije projektov vključuje procese in dejavnosti za prepoznavanje, definiranje, združevanje, poenotenje in usklajevanje različnih procesov in dejavnosti managementa projektov znotraj procesnih skupin za management projektov. V kontekstu projektnega managementa integracija vključuje značilnosti poenotenja, konsolidacije, komunikacije in integrativnih ukrepov, ki so ključnega pomena za nadzorovano izvajanje projekta vse do njegovega zaključka ter uspešen management zahtev zainteresiranih strani in izpolnjevanje zahtev. Management integracije projektov vključuje izbiro dodeljevanja virov, sklepanje kompromisov med konkurenčnimi cilji in alternativami ter managementom soodvisnosti med področji znanja o opravljanju projektov. Procesi projektnega managementa so po navadi predstavljeni kot diskretni procesi z definiranimi vmesniki, medtem ko se v praksi prekrivajo.

Management obsega projekta vključuje procese, ki zagotavljajo, da projekt vključuje vsa nujno potrebna dela, ki bodo vodila projekt do njegove uspešne zaključitve. Management obsega projekta je primarna skrb z definiranjem in kontrolo, kaj je in kaj ni vključeno v projektu.

Management časa projekta vključuje procese, potrebne za management pravočasnega zaključka projekta. V tem primeru se management časa nanaša na mejnike in roke projektov. Manager projekta mora poleg zagotavljanja, da se projekti dostavijo v zahtevano specifikacijo in v okviru proračuna, zagotoviti tudi, da je delo končano ob določenih rokih ali pred njimi.

Management stroškov projekta vključuje procese, ki so vključeni v načrtovanje, ocenjevanje, oblikovanje proračuna, financiranje, management in nadzor stroškov, tako da se lahko projekt zaključi v okviru odobrenega proračuna. Proračun za projekt predstavlja finančni načrt. Sledenje temu načrtu in dokončanje projekta v okviru tega proračuna vključuje skrbno usklajevanje in management. Tu igra vlogo management časa. Izrek »čas je denar« ni nikoli primernejši kot takrat, ko ga uporabljate pri zapletenih projektih.

Management kakovosti projekta vključuje procese in dejavnosti izvajalske organizacije, ki določajo kakovostne politike, cilje in odgovornosti, tako da projekt zadovolji potrebe, za katere je bil izveden. Management kakovosti projektov uporablja politike in postopke za

izvajanje sistema managementa kakovosti organizacije v okviru projekta in po potrebi podpira dejavnosti nenehnega izboljševanja procesov, ki se izvajajo v imenu izvajajoče organizacije. Management kakovosti projektov deluje tako, da zagotavlja, da so zahteve projekta, vključno z zahtevami za izdelek, izpolnjene in potrjene. Nenehno izboljševanje je del filozofije kakovosti.

Management človeških virov projekta vključuje procese, ki organizirajo, upravljajo in vodijo projektno skupino. Projektno skupino sestavljajo ljudje z dodeljenimi vlogami in odgovornostmi za dokončanje projekta. Člani projektnih skupin imajo lahko različne spretnosti, lahko so dodeljeni polni ali krajši delovni čas in jih lahko dodate ali odstranite iz skupine, ko projekt napreduje. Čeprav so določene posebne vloge in odgovornosti članov projektna skupine, je sodelovanje vseh članov skupine pri načrtovanju projektov in odločanju koristno. Sodelovanje članov skupine pri načrtovanju doda njihovo strokovnost v postopek in okrepi njihovo zavezanost projektu.

Management projektnih komunikacij vključuje procese, ki so potrebni za pravočasno in ustrezno načrtovanje, zbiranje, ustvarjanje, distribucijo, shranjevanje, iskanje, management, nadzor, spremljanje in končno razporejanje informacij o projektu. Manager projektov večino časa porabijo za komunikacijo s člani skupine in drugimi deležniki v projektu, ne glede na to, ali so notranji (na vseh organizacijskih ravneh) ali zunanji v organizaciji. Učinkovita komunikacija ustvarja most med različnimi deležniki, ki imajo lahko različne kulturna in organizacijska ozadja, različne ravni strokovnega znanja in različne perspektive in interese, ki vplivajo na izvedbo ali rezultat projekta.

Management tveganj projekta vključuje procese izvajanja načrtovanja, identifikacije, analiza, načrtovanja odzivanja in nadzora tveganj na projektu. Cilji managementa s tveganji projekta so povečati verjetnost in vpliv pozitivnih dogodkov ter zmanjšati verjetnosti in vpliv negativnih dogodkov v projektu. Čeprav bi vsak manager projektov želel pričakovati najboljše od svojega projekta, morajo managerji projektov določiti najslabše možne scenarije, jih skrbno pretehtati in pripraviti načrte ukrepov ob nepredvidljivih dogodkih. Prav tako si mora prizadevati za določitev strategij, ki omejujejo tveganje.

Management projektnih naročil vključuje procese potrebne, za nakup ali pridobitev izdelkov, storitev ali rezultatov, potrebnih zunaj projektna skupine. Organizacija je lahko kupec ali prodajalec izdelkov, storitev ali rezultatov projekta. Management javnih naročil vključuje management pogodb in postopke nadzora sprememb, potrebnih za razvoj in management pogodb ali naročil, ki jih izdajo pooblaščen člani projektna skupine. Management projektnih naročil vključuje tudi nadzor katere koli pogodbe, ki jo je izdala zunanja organizacija (kupec), ki od izvajalca (prodajalca) pridobiva končne izdelke iz projekta in management pogodbenih obveznosti, ki jih s pogodbo položi projektna skupina.

Management deležnikov projekta vključuje procese, ki so potrebni za identifikacijo ljudi,

skupin ali organizacij, ki bil lahko vplivali na projekt ali bi le ta vplival na njih, za analizo pričakovanj zainteresiranih strani in njihov vpliv na projekt ter na razvoj ustreznih strategij za vključevanje deležnikov v odločanje glede projekta in izvršbo. Vsak, na katerega projekt vpliva je deležnik. Managerji projektov morajo prepoznati zainteresirane strani in si prizadevati za vzdrževanje dobrih odnosov z njimi. Management deležnikov se osredotoča tudi na nenehno komuniciranje z zainteresiranimi stranmi, da bi razumeli njihove potrebe in pričakovanja, reševanje problemov, ki se pojavljajo, management nasprotujočih si interesov in spodbujanje ustreznega vključevanja zainteresiranih strani v projektne odločitve in dejavnosti. Zadovoljstvo deležnikov bi moralo biti upravljano, kot da je ključni cilj projekta

2.9 Problemi projektnega managementa

Vsako leto se podjetja po vsem svetu soočajo z astronomskimi stopnjami neuspeha projektov, pri čemer pogosto zapravijo milijone denarnih sredstev na neuspešen projekt. Ista podjetja zaradi vzrokov neuspeha pokličejo drage svetovalce, da ocenijo in obnovijo neuspešne projekte in pogosto opustijo tisto, kar se je sprva zdelo kot dobro načrtovani, dobro organizirani projekt, namenjen uspehu. Veliki projekti propadejo z osupljivo hitrostjo. Ne glede na to, ali gre za tehnološke instalacije, post združitvene integracije ali nove strategije rasti, ta prizadevanja porabijo ogromno sredstev v nekaj mesecih ali celo letih.

Po mnenju Matta in Ashkenasa (2003) kažejo študija za študijo, da ta prizadevanja pogosto prinašajo razočaranje, po nekaterih ocenah več kot polovico časa in davek, ki ga plačajo, ni le finančni. Ti neuspehi demoralizirajo zaposlene, ki so vztrajno delali, da bi dokončali svoj del dela. Težava nastane, ko tradicionalni pristop managementa projektov preusmeri fokus projektnih skupin stran od končnega rezultata, k razvoju priporočil, novih tehnologij in delnih rešitev. Seveda je njihov namen združiti vse to v načrt, ki bi dosegel končni cilj, a ko projekt vključuje veliko ljudi, ki delajo v daljšem časovnem obdobju, je managerjem, ki ga načrtujejo, zelo težko predvideti vse dejavnosti in delovne tokove, ki bodo potrebni. Razen, če je končni izdelek zelo dobro razumljen, je to v visoko tehničnih inženirskih projektih skoraj neizogibno, da bodo nekatere stvari izpuščene iz načrta. In tudi, če so bile predvidene vse prave dejavnosti, se lahko izkaže, da jih je težko ali celo nemogoče povezati skupaj, ko so končane.

Ne obstaja ena sama metoda ali organizacijska kultura, ki bi jo lahko uporabili za uspešen management projektov. Različne organizacije različno obravnavajo funkcionalne projekte. Nekateri imajo razdrobljene in decentralizirane skupine z mnogimi projekti, medtem ko imajo drugi velike agregacije strokovnjakov za management projektov v centralizirani podporni organizaciji (Matta in Ashkenasa, 2003).

Neuspeh projekta se lahko zgodi v katerikoli organizaciji in vsakem projektu. Razlogov za to je mnogo. Včasih je uspeh ali neuspeh izven nadzora managerja projekta in/ali članov tima. Včasih je neuspeh nemogoče nadzorovati. Neuspešni projekti in ljudje, ki so vpleteni

v neuspeh, imajo nekaj skupnih stvari. V obeh primerih dobijo recept za hitre popravke, ki se običajno izkažejo za neučinkovite in lahko povzročijo katastrofalne stranske učinke. V medicinskem žargonu bi rekli, da gre za gripo, ki se ne odziva na antibiotike. Kot antibiotik ali zdravilo za težave se na projektu uporabi in implementira sodobna tehnologija.

Na voljo je mnogo predlogov, kot je nadgradnja programske opreme za sledenje projekta, uporaba kritične verige namesto kritične poti ali uporaba Monte Carlo simulacije za izračun projektnih tveganj. V mnogih primerih ti grobi posegi ne uspejo, saj se uporabljajo neprimerno. Cilj projektnega managementa je izdelati uspešen izdelek ali storitev. Ta cilj pogosto ovirajo napake opustitve, pa tudi naročilo vodstva, managerjev projektov, članov skupin in drugih, povezanih s projekti. Namen te naloge je prikazati oziroma identificirati pogoste vzroke za neuspele projekte, da se zagotovijo informacije za ublažitev njihovega pojava in v mnogih primerih popravilo povzročene škode in povrnitve projektov (Ledig, 2021).

Uspešni projekti se v veliki meri uresničijo zaradi angažiranega nabora deležnikov. Celotna veriga poveljevanja oziroma vodstveni delavci, sponzorji ali izvršno vodstvo, morajo biti aktivni udeleženci v projektu. Jasno je, da bo vsaka pobuda zelo trpela, če sponzor ne bo zavezan ali če ključni igralci, ne morejo razviti kohezivne projektne strategije in nadzorovati usmeritev prizadevanj.

Problem uskladitve je kritičen v primerih, ko cilji projekta niso v skladu z osnovno vizijo organizacije. Preprost primer je prizadevanje za odprtje ključnih trgov v Aziji, ko organizacija namenja sredstva v evropsko tržišče. Upoštevati je treba tudi to, da projekti pogosto propadejo, ker oddelki določijo osebo, s katero najlažje shajajo in ne tiste osebe, ki prispevajo največ znanja in izkušenj. Na primer, če se projekt vrti okoli uvajanja novega nabavnega sistema in se dodeli nov zastopnik za nabavo, bo projektni tim po vsej verjetnosti projekt zaključil neuspešno (Ledig, 2021).

Najboljša priporočena praksa je opredelitev jasnih ciljev, ki se pregledujejo skozi celotno obdobje projekta. Vsak popravek smeri ali celo odpoved projekta lahko pri tem postane rutina pri tem procesu. Usklajevanje spretnosti in ustreznega strokovnega znanja s projekti je ključni dejavnik uspeha. Dosledna komunikacija v standardiziranem formatu je v pomoč tudi glavnim deležnikom. Komunikacija in izmenjava informacij sta bistvena elementa za uspeh projekta. Poleg tega delo timsko delo razvija ustvarjalnost, gradi močne vezi in razvija zaupanje med zaposlenimi. Na koncu tim deluje v dobrem vzdušju in postane bolj produktiven (Ledig, 2021).

Morda je eno od področij neuspeha projekta, o katerem se manj poroča, management tveganj (angl. Risk management). Ker je vsak projekt edinstven, obstaja vedno določena stopnja nepredvidljivosti. V mnogih primerih projektna tveganja niso proaktivno identificirana, analizirana in ublažena. Tudi v primerih, ko je tveganje aktivni del izvedbe postopka, je

strogost, namenjena temu področju, zanemarljiva. Prepogosto se težave rešujejo reaktivno, kar povzroči spremembo terminskega plana in proračuna. Posledica tega je zamuda pri terminskem planu, prekoračitev proračuna in prekomerno število ur in izgorelost zaposlenih. Najboljša priporočljiva praksa je uporaba integriranega in proaktivnega managementa tveganj, pristop za vsa projektna prizadevanja. To vključuje razvoj in objavo načrta managementa tveganj ter izobraževanje celotnega projektnega tima o prednostih izvajanja managementa tveganj. Na bolj podrobni ravni je treba integrirati ugotovljena tveganja v obseg, urnik in stroške. Poleg tega, vodenje dnevnika tveganj in izdelava podatkov, ki so na voljo, vsem izredno pomaga (Kashyap, 2022).

Meritev uspešnosti je področje, ki je deležno mnogo besed, a vendar je slabo razumljena. Je praksa analize napredka projekta in razumevanje, kaj to pomeni v kontekstu celotnega načrta in ciljev projekta. Omogoča projektnim strokovnjakom, da primerjajo napredek naloge ali projekta s trendi in merili uspešnosti, kot sta pretekla uspešnost in pričakovani napredek. Uporaba analize variance uspešnosti jim omogoča, da natančno določijo, katera področja projekta so v zaostanku in katera ne ter na katera se je treba osredotočiti. Pomanjkanje meritev uspešnosti projekta vodijo do tega, da imajo vse strani malo vpogleda v to, kje so projekti glede na to, kje bi morali biti v različnih časovnih obdobjih. Posledično projekti v težavah niso pravočasno poudarjeni za popravne ukrepe in ustrezni korektivni ukrepi niso opredeljeni, kar vodi v slabo kakovost izdelkov in storitev. Priporočen pristop je uporaba standardiziranih meril uspešnosti projekta in vzpostavitev izhodišča projekta za urnik, trud, izdelek ipd. Vloga managementa prislužene vrednosti (angl. earned value management) je tukaj pomembna tudi pri majhnih projektih. Ta primerja napredek stroškov in načrtov z izhodišči projekta, proračuni in pričakovanimi prihodnjimi rezultati (AresPrism, 2023).

Eden bolj pogostih razlogov, zakaj načrt managementa projekta ne uspe, je povezan z nejasnim obsegom dela. Načrt managementa projekta mora imeti opredeljen, jasn in dosegljiv obseg, ki ga je mogoče slediti, in dokončati z orodji in viri, dodeljenimi projektu. Načrti managementa projekta ne uspejo, ker člani tima nimajo jasnih pričakovanj glede obsega in proračuna. Treba je biti prepričan, da sta začetek projekta in projektna listina socializirana in deljena z vsemi ključnimi deležniki. Ne zajema se področja obsega samo zato, da bi zadovoljili specifične deležnike projekta, ampak je potrebno biti skladen z načrtovanim obsegom.

Projekt ima lahko nejasno napisane definicije obsega. Težave se lahko pojavijo pri zbiranju samih zahtev uporabnikov. Obstaja lahko pritisk za čimprejšnjo izvedbo, preden je projekt ustrezno opredeljen. To je eden od klasičnih primerov potencialno neuspešnega projekta. Obseg projekta mora biti jedrnat, jasn in nedvoumen. Projekt mora biti jasno in splošno razumljiv vsem deležnikom. Priporočen pristop je, da se pregleda obseg projekta z uporabniki in pridobi popolna podpora o tem, kar bo izvedeno in dostavljeno. Bistveno je jasno razumevanje obsega, za pridobitev zavezanosti in uspešne izvedbe. To pomeni, pridobitev dogovora o tem, kaj je znotraj in izven obsega. Mogoče je primerno oblikovanje

in uporaba uradnega postopka za nadzor sprememb, vključno z odborom za nadzor sprememb. Oblikovanje popolne rešitve z zelo širokim obsegom pogosto vodi do zapletenih, večletnih projektov s kompleksnimi soodvisnostmi. Kjer je mogoče, je treba obseg omejiti na dosegljiva natančno opredeljena prizadevanja. S strožjim obsegom projekta lahko organizacija bolje opravi spremljanje napredka in nadzor rezultatov. Za zapletene drage projekte, ki vsebujejo veliko neznank in spremenljivih tveganj, je priporočeno uvesti fazo preiskave obsega pred odobritvijo in izvedbo projekta. Preiskava obsega je lahko na primer v obliki pilota ali raziskovalnega dokumenta kot dokaz koncepta, primerjalne analize podobnih projektov in simulacij. Ne glede na pristop, bo izbrana tehnika uporabe t. i. pred projekta za opredelitev obsega prinesla prepotrebno jasnost primarnemu projektu in izboljšala možnosti za njegov uspeh (Rodriguez, 2018).

Za managerje projektov in zainteresirane strani je ključnega pomena, da se zavedajo napredka projekta in izzivov na vsaki stopnji. Na žalost so deležniki pogosto obveščeni o kritičnih vprašanjih v fazi, ko vpliva na stroške, časovne okvire in je pogostokrat obsežna in nepovratna. Neustrezna komunikacija o statusu projekta in težavah je posledica potrebe in pričakovanja deležnikov, ki niso obvladovana ustrezno. Očitno je, da reševanje težav zahteva čas, stran od načrtovanih projektnih aktivnosti. Ta težava bo vplivala na kateri koli del projekta. Najprej in predvsem je treba ustvariti načrt managementa komunikacij. To bi moralo biti sestavljeno iz dveh delov, projektno komunikacijo in komunikacijo z deležniki. Te dejavnosti je potrebno začeti ob začetku projekta, s posebnim prizadevanjem za izvedbo analize zainteresiranih strani za ugotavljanje pričakovanj in komunikacijskih potreb. Potrebno je biti pripravljen na posredovanje statusa projekta in posodobitve preko več kanalov, za dostavo informacij, ki bi zadovoljile različne potrebe zainteresiranih strani. Sodobna orodja za management projektov imajo vključeno tehnologijo socialnega sodelovanja, ki pomagajo izboljšati prizadevanja za komuniciranje pri projektih (Carlos, 2011). Izogibanje zgoraj navedenim pastem ne bo zagotovilo uspešnega projekta. Vendar bo zagotavljalo trdne temelje in temelje, iz katerih lahko začnemo proces izvajanja v skladu s cilji projekta in močno vpliva na uspešen izidi.

3 AGILNI PROJEKTNI MANAGEMENT

Agilne metode so bile prvotno razvite za programsko opremo, da bi racionalizirali in izboljšali razvojni proces, da bi hitro ugotovili in prilagodili težave in pomanjkljivosti. Razvijalcem in timom omogočajo hitro dostavo boljšega izdelka s pomočjo kratkih, iterativnih, interaktivnih sej tako imenovanih sprintov. V dobi digitalne preobrazbe, ko se mnoga podjetja selijo na digitalno delovno mesto, je agilnost popolnoma primerna za organizacije, ki si želijo preoblikovati način managementa projektov in celotnega poslovanja. Agilnost lahko pomaga zagotoviti proces in metodološko usklajenost na ravni celotnega podjetja. Z vidika poslovne koristi, digitalno delovno mesto in agilnost zagotavljata: povečano fleksibilnost in produktivnost, večjo preglednost in kakovost rezultatov, zmanjšano nevarnost zgrešenih ciljev, večjo angažiranost in zadovoljstvo

zainteresiranih strani. Na področju managementa projektov agilnost ponuja projektnim skupinam, sponzorjem, managerjem projektov in strankam številne projektu specifične prednosti, med drugim: hitrejši razvoj rešitve, zmanjšanje odpadkov z minimizacijo resursov, večja prilagodljivost in prilagojenost spremembam, boljši uspeh zaradi večjega prizadevanja, hitrejša odkrivanje težav in napak, optimizirani razvoj proces, optimalni nadzor projekta, povečana osredotočenost na posebne potrebe strank, povečana pogostost sodelovanja in povratnih informacij (Alexander, 2023).

Medtem ko se tradicionalni projektni management osredotoča na vnaprejšnje načrtovanje, kjer so pomembni dejavniki stroški, obseg, čas, agilni management daje velik pomen na timsko delo, sodelovanje s strankami in fleksibilnosti. Gre za iterativni pristop, ki se bolj osredotoča na vključitev povratnih informacij strank in nenehnih izdaj z vsako iteracijo projekta razvoja programske opreme (Narasimman, 2024). Osnovni koncept agilnega razvoja programske opreme je, da se poglobi v spremljajoče spremembe in skupna prizadevanja za doseganje rezultatov vnaprej določenega procesa. Prilagodljivo načrtovanje je morda največja lastnost agilne metode, zaradi česa je priljubljena med timi projektov. Agilni projektni management je slog managementa projektov, ki se osredotoča na zgodnje zagotavljanje poslovne vrednosti, nenehno izboljševanje izdelka in procesov projekta, fleksibilnost obsega, vnos tima in zagotavljanje dobro preizkušenih izdelkov, ki odražajo potrebe kupcev. Layton (2012) ugotavlja, da so agilne vrednote, načela in prakse prisotne že dolgo časa. Agilni pristopi temeljijo na empirični metodi nadzora, postopku sprejemanja odločitev na podlagi realnosti, opažene v projektu. V okviru metod razvoja programske opreme je lahko empirični pristop učinkovit tako pri razvoju novih izdelkov kot pri projektih za izboljšanje in nadgradnjo. S pogostim pregledovanjem dosedanjih del se potrebne prilagoditve zgodijo dokaj hitro. Empirični nadzor zahteva prosto preglednost. Vsi, ki sodelujejo v agilnem projektu vedo, kaj se dogaja in kako projekt napreduje. Ljudje, ki so nekaj vložili v izdelek in postopek, redno ocenjujejo sam potek. Zahteva tudi takojšno prilagoditev, kar pomeni, da hitrejša možnost prilagajanja omogoča boljšo sposobnost zmanjševanja težav (Layton, 2012).

Agilni projektni management je skupek metod, ki pomagajo skupini, da razmišlja učinkoviteje, deluje bolj uspešno in sprejema boljše odločitve. Te metode obravnavajo vsa področja tradicionalnega programskega inženiringa, vključno z managementom projektov, oblikovanjem in arhitekturo programske opreme ter izboljšanjem procesov. Vsaka od teh metod je sestavljena iz praks, ki so racionalizirane in optimizirane, da bi jih bilo mogoče čim lažje osvojiti. Agilnost je tudi miselnost, saj lahko pravilna miselnost močno vpliva na to, kako učinkovito tim uporablja prakse. Ta miselnost pomaga ljudem v skupini, da si izmenjujejo informacije med seboj, tako da lahko skupaj sprejemajo pomembne projektne odločitve, namesto da imajo managerja, ki vse te odločitve sprejema sam. Agilna miselnost je v odprtem načrtovanju in izboljšanju procesov celotnega tima. Agilni tim uporablja prakse na tak način, da si vsi delijo iste informacije in vsaka oseba v timu ima besedo o tem, kako naj se prakse uporabljajo. Resničnost agilnosti za številne time, ki niso dosegle toliko

uspeha, je precej drugačna od njene obljube, ključ do te razlike pa je pogosto miselnost, ki jo tim prinaša pri vsakem projektu (Stellman in Green, 2014).

Po mnenju Stellman in Green (2014) je večina podjetij, ki izdelujejo programsko opremo, eksperimentirala z agilno metodo. Čeprav so bila nekatera uspešna, so nekateri timi dosegli nezadovoljive rezultate. Sicer so dosegli nekaj izboljšav pri managementu svojih projektov, dovolj, da se jim je trud uporabe agilnosti splačal, ampak niso videli bistvenih sprememb, ki naj bi jih agilnost obljubljala. Bistvo je sprememba miselnosti, kar pomeni, da agilnost timu pomaga postati bolj učinkovito.

Agilni razvoj programske opreme temelji na inkrementalnem iterativnem pristopu. Namesto poglobljenega načrtovanja na začetku projekta so agilne metode odprte za spreminjanje zahtev sčasoma in spodbujajo stalne povratne informacije končnih uporabnikov. Več funkcionalni timi v določenem času delajo na iteracijah izdelka in to delo je organizirano v seznam uporabniških zgodb, ki je prednostno določen na podlagi poslovne ali kupčeve vrednosti. Cilj vsake iteracije je izdelava delujočega dela izdelka. V agilni metodi vodstvo spodbuja timsko delo, odgovornost in osebno komunikacijo. Poslovne interesne skupine in razvijalci morajo sodelovati, da izdelek uskladijo s potrebami strank in cilji podjetja. Agilnost se nanaša na vsak postopek, ki je usklajen s koncepti agilnega manifesta. V manifestu razvijalci opisujejo najboljše načine za razvoj programske opreme, kako to prakticirajo in s tem pomagajo drugim. Agilni manifest je dramatično nasprotje tradicionalnemu vodniku PMBOK in njegovim standardom (Eby, 2017). Agilni manifest je sestavljen iz štirih temeljnih vrednot in dvanajst ključnih načel, ki določajo agilen pristop razvoja programske opreme. Agilne metode implementirajo štiri vrednote na različne načine, vendar se vse zanašajo nanje pri razvoju visokokakovostne delujoče programske opreme.

Štiri vrednote agilne metode so (Naydenov, 2015):

- posamezniki in interakcije nad procesi in orodji,
- delujoča programska oprema nad obsežno dokumentacijo,
- sodelovanje s strankami nad pogodbenim pogajanjem
- odziv na spremembe nad sledenjem načrta.

12 ključnih načel, ki danes vodijo agilni managementa projektov so (Alexander, 2023):

- zadovoljstvo strank je vedno najpomembnejše in se doseže s hitro in neprekinjeno dostavo,
- spreminjajoče se okolje je vključeno v katero koli fazo postopka, da se kupcu zagotovijo konkurenčne prednosti,
- izdelki ali storitve se dobavljajo z višjo frekvenco,
- zainteresirane strani in razvijalci vsakodnevno tesno sodelujejo,

- vse zainteresirane strani in člani skupine ostajajo motivirani za optimalne rezultate projekta, timi pa imajo vsa potrebna orodja in podporo ter jim zaupajo, da bodo dosegli cilje projekta,
- sestanki iz oči v oči veljajo za najučinkovitejše in najučinkovitejši format za uspeh projekta,
- končni delujoč izdelek je merilo uspeha,
- trajnostni razvoj je dosežen z agilnimi procesi, kjer lahko projektni tim ohranja stalen in trajen tempo,
- agilnost se stopnjuje s stalnim poudarkom na tehnični odličnosti in pravilnem oblikovanju,
- enostavnost je bistven element,
- samoorganizacijske skupine imajo največjo zmožnost razvijanja najboljše arhitekture, zasnove ter izpolnjevanja zahtev,
- timi uporabljajo redne intervale za izboljšanje učinkovitosti skozi prilagajanje svojega vedenja.

Vsako metodo, okvir ali vedenje, ki sledi tem vrednotam, lahko imamo za agilno. Ne glede na katero agilno metodo tim uporablja, koristi ni mogoče v celoti uresničiti brez zavzetosti vseh vpletenih. V nadaljevanju bo opisanih nekaj najbolj priljubljenih agilnih metod.

3.1 Ekstremno programiranje

Ekstremno programiranje (angl. Extreme Programming, v nadaljevanju XP) je bolj radikalna agilna metoda, ki se bolj osredotoča na proces inženiringa programske opreme in obravnava faze analize, razvoja in preizkušanja z novimi pristopi, ki bistveno vplivajo na kakovost končnega izdelka. XP lahko uporabimo, če imamo morda sistem, katerega funkcionalnost naj bi se spreminjala vsakih nekaj tednov ali mesecev. Včasih kupci nimajo konkretne predstave o tem, kaj bi moral sistem natančno storiti. V mnogih panogah programske opreme se zahteve dinamično spreminjajo in ta sprememba je edina stalnica. V takem primeru bo XP metoda delovala, medtem ko druge metode ne bodo. Glavni cilj XP-ja je, da se stranki dostavi uporabna programska oprema, kadar in ko jo potrebuje. Tu moramo postaviti pričakovanja, tako da lahko kupci v omejenem obdobju dobijo novo zgradbo sistema z najbolj prednostnimi funkcijami. Potem lahko naredimo nove načrte za naslednjo izdajo. Prav tako kot pri Scrum-u, XP vključuje veliko interakcij s strankami in skozi projekt naslavlja povratne informacije, predloge strank, kar vodi k največjemu zadovoljstvu strank (Ghahrai, 2017).

XP je tipičen okvir agilnega razvoja, ki ga je razvil Kent Beck in ga je mogoče prilagoditi razvojnim podjetjem različnih dimenzij. Gre za metodo, ki poudarja vrednote, kot so komunikacija, preprostost, povratne informacije, pogum, spoštovanje ter daje prednost zadovoljstvu strank nad vsem ostalim. Ta metoda razvijalcem ponuja zaupanje, saj jih motivira, da sprejemajo spremembe v zahtevah strank, tudi če pridejo v poznejšo fazo

razvojnega cikla. Delo je v XP-ju izjemno pomembno, saj kadar pride do težav, ga rešuje celotni tim managerjev, razvijalcev ali kupcev. Vsi so bistveni koščki iste sestavljanke, kar ustvarja plodno okolje za visoko produktivnost in učinkovitost v timu. V XP-ju se programska oprema preizkuša že od prvega dne tako kot tudi zbiranje povratnih informacij za izboljšanje nadaljnjega razvoja (Espírito, 2022).

Vsak XP projekt ima številne korake, ki vodijo od začetka do zaključka. Vse se začne z načrtovanjem. V tem koraku se tim pogovarja s stranko, da bi razumel želeni izdelek ter zbrala uporabniške zgodbe, ki so osnova za cilje projekta. Te zgodbe so ocenjene in prednostno razvrščene ter nato pretvorjene v naloge. Ta korak je ključnega pomena za postavitev temeljev za uspešen projekt. Po načrtovanju sledi dizajn. Razvijalci vzamejo uporabniške zgodbe in oblikujejo arhitekturo kode, da izpolni pričakovanja strank. To vključuje odločanje o programskem jeziku, okolju, okvirjih, ki se bodo uporabljali pri razvoju programske opreme. Ko je dizajn končan, sledi kodiranje. Tukaj razvojni tim dela na dejanskem kodiranju programske opreme. Koda je napisana iterativno, kar pomeni, da je koda, napisana v majhnih delih in redno preizkušena. To zagotavlja, da je koda visoke kakovosti in da je mogoče hitro opraviti morebitne napake. Potem sledi testiranje, ki je bistveni del XP-ja. Razvijajo se avtomatizirani testi, ki zagotovijo, da koda deluje po pričakovanjih. To pomaga prepoznati morebitne napake, preden je programska koda izdana. Zadnji korak metode predstavljajo povratne informacije. Stranko se spodbuja, da poda povratne informacije, predloge o programski opremi, ki jo bodo samo še izboljšale. Tako se zagotovi, da programska oprema izpolnjuje pričakovanja strank in je najvišje kakovosti (Paula, 2023). Običajno timi razvijalcev delujejo v kratkih iterativnih ciklih. Vsak cikel je namenjen točno določenemu delu končnega izdelka. Cikel traja od enega do dveh tednov. Po vsakem ciklu je rezultat na voljo stranki. Načrt cikla se lahko spremeni v skladu z željami stranke. Ko so vsi cikli končani in so testirani vsi deli izdelka, je ta na voljo stranki. Če so bile potrebe stranke izpolnjene, se XP projekt zaključí.

Glavna prednost ekstremnega programiranja je, da ta metoda podjetjem za razvoj programske opreme omogoča prihranke stroškov in časa, potrebnih za realizacijo projektov. Prihranki časa so na voljo zaradi dejstva, da se XP osredotoča na pravočasno dostavo končnih izdelkov. Timi prihranijo denar, ker ne uporabljajo veliko dokumentacije. Težave običajno rešijo z razpravami znotraj tima. Ena izmed slabosti je, da je metoda XP preveč osredotočena na kodo, ne pa na oblikovanje. To je lahko težava, ker je dober dizajn izredno pomemben za programske aplikacije. Poleg tega je v XP projektih dokumentacija o napakah pomanjkljiva in se lahko z istimi težavami, napakami soočimo pri naslednjih projektih (Kukhnavets, 2018).

3.2 Metoda dinamičnega razvoja sistema

Tehnika razvoja dinamičnih sistemov (angl. Dynamic System Development Method, v nadaljevanju DSDM) je pristop agilnega razvoja programske kode, ki zagotavlja okvir za

gradnjo in vzdrževanje sistemov. Filozofija DSDM je izposojena iz spremenjene različice načela sociologov, 80 % aplikacije je narejeno, oddano v 20 % časa, mi pa želimo, da je aplikacija 100 %. DSDM je iterativna metoda kode, v kateri vsaka iteracija sledi 80 % pravilu, da je za vsako stopnjo potrebno dovolj dela, da se olajša premik do naslednje stopnje (Ghahrai, 2017). Preostali detajli so pogosto dokončani pozneje, ko se zabeležijo poslovne potrebe ali se zahteve spremenijo.

DSDM je okvir, ki ga sestavlja osem načel (osredotočenost na potrebe podjetja, pravočasna dostava, sodelovanje, kakovost se ne ogroža, postopna gradnja iz trdnih temeljev, iterativno razvijanje, neprestana komunikacija, demonstracija kontrole), življenjski cikel, izdelki, vloge, odgovornosti in primeri najboljših praks. Ti temelji podpirajo filozofijo čim hitrejšega zagotavljanja strateško usklajenih poslovnih koristi, da bi si organizacija zagotovila čim boljšo donosnost naložbe.

DSDM je agilna metoda, ki obravnava potrebe tako preprostega razvoja izdelkov, pri katerih bo verjetno, da bodo timi alocirani in bolj zapletenih projektnih situacij, ki lahko vključujejo več timov, ki so razdeljeni in so celo v različnih državah in časovnih pasovih (Richards, 2023). Uporabniki so močno prisotni pri razvoju sistema, zato obstaja velika verjetnost, da bodo lažje razumeli projekt razvoja programske opreme. V tem modelu se osnovna funkcionalnost dostavi hitro, več funkcionalnosti pa se zagotavlja v pogostih intervalih. Ta metoda omogoča razvijalcem enostaven dostop do končnih uporabnikov. Največja težava je, da je to metodo drago izvajati, saj od uporabnikov in razvijalcev zahteva, da so usposobljeni uporabljati to metodo učinkovito. Zato morda ni primerna za majhne organizacije ali enkratne projekte. Gre za razmeroma nov model, zato ni pogost in ga ni tako lahko razumeti (Rosencrance, 2019).

3.3 Kristalna metoda

Kristalna metoda (angl. Crystal) je ena izmed najbolj lahkih in prilagodljivih pristopov k razvoju programske opreme. Metoda dejansko obsega družino agilnih procesnih modelov vključno s Čistim kristalom (angl. Crystal Clear), Rumenim kristalom (angl. Crystal Yellow), Oranžnim kristalom (angl. Crystal Orange) in drugih. Vsak model ima edinstvene značilnosti, ki jih vodi več dejavnikov, kot so velikost tima, kritičnost sistema in prednostne naloge projekta. Ta družina kristalov obravnava spoznanje, da lahko vsak projekt zahteva nekoliko prilagojen nabor politik, praks in postopkov, da bi dosegel edinstvene lastnosti izdelka.

Metoda, ki jo je predstavil Alistar Cockburn, se osredotoča predvsem na ljudi in interakcije med njimi, medtem ko delajo na agilnem projektu razvoja programske opreme. Poudarek je tudi na kritičnosti poslovanja in prednostni nalogi sistema v razvoju. Za razliko od tradicionalnih razvojnih metod ta metoda ne poskuša popraviti orodij in tehnik razvoja, ampak ljudi in procese drži v središču procesa. Vendar pa niso najpomembnejši samo ljudje

ali procesi, temveč interakcije med njimi. Ključni principi metode vključujejo timsko delo, komunikacijo, preprostost in razmišljanje za pogosto prilagajanje ter izboljšanje procesa.

Tako kot druge agilne metode tudi ta spodbuja zgodnjo, pogosto ponudbo delujoče programske opreme, večjo vključenost uporabnikov, prilagodljivost in odstranjevanje birokracije ali motenj (Blueprint, 2022). Metoda zagotavlja pogoste dobave, ki razvijalcem omogočajo, da v vsaki fazi prepoznajo morebitne težave. Ta metoda omogoča tesnejšo komunikacijo in spodbuja člane tima k sodelovanju in izmenjavi znanja. Ne glede na to, kako dober ali slab je izdelek, vedno obstajajo področja za izboljšave. Člane tima spodbuja, da si vzamejo nekaj časa od razvoja programske opreme in spregovorijo o tem, kako izpopolniti procese. Dejstvo, da obstajajo različice v družini metode, pomeni, da se lahko načela razlikujejo glede na velikost skupine in velikosti projekta, kar ima za posledico projekte, ki niso tako enostavni. Zaradi nenehne potrebe po komunikaciji in razmišljanju morda ta metoda ne bo delovala pri timih, razpršenih po različnih območjih. Največja težava pri uporabi te metode je, da načrtovanje in sam razvoj aplikacije nista odvisna od zahtev uporabnikov (Espírito, 2022).

Ta metoda sledi sedmim ciklom oziroma korakom: projektni cikel, dostavni cikel, iteracijski cikel, delovni teden, integracijsko obdobje, delovni dan in epizoda. V projektnem ciklu je vključeno več aktivnosti, razvoj začetnega načrta, ustvarjanje in določanje članov razvojnih timov ter opravljanje analize izvedljivosti. Pri dostavnem ciklu tim posodobi, nadgradi načrt izdaje, zagotavlja izpolnjevanje zahtev z eno ali več ponovitvami in dobavlja integrirani izdelek. Vodilni razvijalec pregleduje načrt projekta in po potrebi prilagaja razvoj izdelka. Iteracijski cikel sledi korakom ocenjevanja, razvoja in testiranja. Delovni teden predstavlja vse enote dela in celotno delo, ki bo opravljeno v določenem tednu. V integracijskem ciklu poteka dizajn, razvoj, integracija in sistemsko testiranje izdelka. Delovni dan predstavlja delo, opravljeno za en dan. Epizoda predstavlja dizajniranje, razvoj in preverjanje delca (iteracij) programske kode (Singh, 2021).

3.4 Scrum

Scrum je ena izmed najbolj priljubljenih agilnih metod. Je prilagodljiv, iterativen, hiter, fleksibilen in učinkovit okvir, zasnovan za hitro dodajanje vrednosti skozi celoten projekt. Metoda zagotavlja preglednost komunikacije in ustvarja okolje kolektivne odgovornosti in nenehnega napredka (SCRUMstudy, 2016).

Scrum izhaja iz kompleksne teorije prilagodljivih sistemov, nanj pa so vplivale najboljše prakse japonske industrije, zlasti načela vitkega razvoja, ki izhajajo iz podjetja, kot sta Toyota in Honda. Po mnenju Rubina (2012) Scrum ni samo razvojna metoda ali formalni postopek, temveč algoritem stiskanja najboljših praks na svetu, opaženih v več kot petdesetih letih razvoja programske opreme. Scrum se uporablja kot agilna praksa, ki končnim uporabnikom prinaša programsko opremo hitreje in bolje. Metoda podpira kreativni pristop

k razvoju zapletenih, inovativnih sistemov in dosega veliko število razvijalcev. To je edini postopek razvoja programske opreme, ki je pokazal linearno nadgraditev pri dodajanju virov velikim projektom. V pravilno izvedenem Scrum-u produktivnost na razvijalca ostane enaka pri dodajanju virov, ki jih pri razvoju programske opreme še nikoli nismo videli (Sutherland in Schwaber, 2011)

Scrum ni standardiziran postopek, v katerem metodično sledimo vrsti zaporednih korakov, za katere zagotovimo, da bodo pravočasno in v okviru proračuna ustvarili visokokakovosten izdelek, ki bo navdušil stranke. Namesto tega je Scrum okvir za organizacijo in management dela. Okvir temelji na naboru vrednot, načel in praks, ki zagotavljajo temelj, ki bo organizacijam dodal edinstveno izvajanje ustreznih inženirskih praks in specifični pristop za uresničevanje praks same metode. Rezultat je različica Scrum metode, ki je edinstvena za vsako podjetje. Če želimo bolje razumeti okvirni koncept, si predstavljamo, da Scrum okvir deluje kot temelj in stene stavbe. Vrednote, načela in prakse metode, bi bile ključne strukturne komponente. Ne moremo prezreti ali bistveno spremeniti vrednosti, načela ali prakse, ne da bi tvegali neuspeh. Kar pa lahko storimo, je, da prilagodimo notranjost strukture Scrum metode in dodamo napeljave in funkcije, dokler ne bomo imeli postopka, ki bo deloval nam v korist. Scrum metoda je osredotočen na ljudi oziroma okvir, ki temelji na vrednotah poštenosti, odprtosti, poguma, spoštovanja, osredotočenosti, zaupanja, opolnomočenja in sodelovanja (Rubin, 2012).

3.4.1 Sprint

Metoda Scrum kot najpopularnejši agilni okvir pri razvoju programske opreme, deluje kot iterativni pristop, ki ima v svojem jedru sprint. Sprinti imajo fiksno dolžino, ki lahko traja od enega do štirih tednov. Delo, ki je potrebno za doseg cilja izdelka, vključno z načrtovanjem sprintov, dnevnimi sestanki, se izvaja znotraj sprintov. Med izvajanjem sprinta se ne izvedejo nobene spremembe, ki bi ogrozile cilj sprinta, kvaliteta se ne zmanjša, produktna listina se po potrebi izboljša in obseg lahko postane bolj razumljiv, ko je opravljeno več dela ter se ga lahko ponovno definira skupaj z lastnikom izdelka. Za podporo tega procesa, Scrum timi uporabljajo posebne vloge, artefakte in dogodke. Da bi zagotovili, da izpolnjujejo cilje vsakega dela procesa, Scrum timi v celotnem projektu izvajajo nadzor in se prilagajajo glede na ugotovitve. V vsakem sprintu razvojni tim razvija in preizkuša funkcionalni del izdelka dnevno, dokler ga lastnik oziroma naročnik izdelka ne sprejme, funkcionalnost pa postane potencialno povečana za celoten izdelek. Ko se en sprint konča, se drugi začne. Sprostitev funkcionalnosti se pojavi na koncu več sprintov, ko lastnik izdelka ugotovi, da je dosežena dovolj visoka vrednost. Lastnik se lahko odloči, da bo sporočil funkcionalnost po vsakem sprintu ali celo tolikokrat, kot je potrebno tekom sprinta. Temeljno načelo sprinta je njegova ciklična narava, sprint se lahko tako kot procesi v njem ponavlja znova in znova (Layton, 2012).

Prednosti metode Scrum so dobro razumljive. Manj odvečnih specifikacij in manj predaj

zaradi med-funkcionalnih skupin in več prožnosti pri planiranju načrta zaradi kratkih sprintov. Prehod organizacije v uporabo Scrum-a je temeljni premik, ki je pretresel stare navade in jih spremenil v učinkovitejše (Marschall, 2015).

3.4.2 Vloge pri metodi Scrum

Vloge so pri metodi Scrum jasno določene in se razlikujejo kot pri tradicionalnem projektnem managementu. Projektni tim sestavljajo produktni lastnik (angl. Product owner), skrbnik (angl. Scrum master) in razvojni tim. Posebnost Scrum metode je samostojnost, odgovornost razvojnega tima, ki je odgovoren in se zaveže za razvoj uporabniških zgodb ob vsaki novi iteraciji. Za zagotavljanje pravilne izvedbe metode skrbi skrbnik metode. Pri tem tudi skrbi, da ima razvojni tim najboljše pogoje za delo. Za management in pregled nad celotnim projektom skrbi produktni lastnik. Pri tem je odgovoren tudi za produktno listino, kjer skrbi, da so navedene vse zahteve naročnika in da so tudi pravilno razvrščene po pomembnosti. To posebej vpliva na časovno izvedbo posamezne zahteve, tako da se izvede ob pravem času.

3.4.2.1 Skrbnik Scrum-a

Vloga skrbnika je unikatna, saj skrbi za pravilno izvajanje metode Scrum in nemoten potek dela med projektom. Vendar njegova vloga zajema tudi komunikacijske sposobnosti in s tem zagotavljati agilnost razvojnega tima oziroma imeti mentorski pristop.

Glavne naloge skrbnika do produktnega lastnika so (Paula, 2020):

- pomoč pri komunikaciji produktnega lastnika z razvojnim timom,
- zagotoviti, da produktni lastnik razume in pozna razporeditev elementov produktne listine,
- pomoč razvojnemu timu pri razumevanju produktne listine,
- zagotavlja, da produktni lastnik razume in prakticira agilnost.

Naloge skrbnika do razvojnega tima (Paula, 2020):

- pomoč pri zagotavljanju kvalitete produkta,
- mentorstvo oziroma učenje razvojnega tima samostojnosti in agilnosti,
- odstranjevanje ovir, tako, da lahko razvojni tim nemoteno nadaljuje delo,
- učenje razvojnih timov, ki delujejo v organizacijah, ki ne prakticirajo agilnih metod.

Naloge skrbnika v organizaciji (Paula, 2020):

- načrtovanje Scrum implementacije v organizaciji,
- pomoč projektnemu timu pri razumevanju in uveljavitvi Scrum metode pri razvoju

produkta,

- spodbujanje organizacije k spremembam in s tem izboljša produktivnost Scrum tima,
- vodenje in učenje celotne organizacije pri implementaciji Scrum metode,
- sodelovanje z ostalimi skrbniki znotraj organizacije in s tem izboljševanje učinkovitosti implementacije Scrum metode.

3.4.2.2 *Produktni lastnik*

Produktni lastnik je osrednja figura projekta. Njegova glavna naloga je pripravljanje načrta zgodb za vsak naslednji sprint in jih pojasni razvojnemu timu. S tem povečuje vrednost izdelka s strani razvojnega tima. Pri tem je zadolžen za management produktne listine, v kateri je treba razvrstiti uporabniške zgodbe po pomembnosti, ki jih je treba dokončati za uspešen končani projekt. Pri tem pa omogoča razvojnemu timu, da se pri opravljanju najpomembnejših zgodb sam odloči, koliko dela lahko opravi. Deluje kot strateg, ko predstavi vizijo projekta in produkta ostalim deležnikom. Glavne naloge produktnega lastnika so (Adobe, 2022):

- definiranje vizije: zagotovi strateški in vizualni povzetek vizije ciljev na visoki ravni, stranki, vodstvu podjetja, razvojnim timom,
- prioritiziranje uporabniških zgodb: ko so uporabniške zgodbe zbrane, produktni lastnik razvrsti elemente po prioritetah glede na poslovne cilje in splošno strategijo. Upošteva tudi, od česa je projekt odvisen, in v kakšen zaporedju bo potekal razvoj. Gre za dinamičen dokument, ki se razvija skupaj s projektom,
- pregled razvojnih faz: ko so vizija, strategija in prednostne naloge določene, produktni lastnik nadzira razvoj izdelka skozi vse njegove faze, vključno z načrtovanjem, izboljševanjem in pregledom. Med fazo načrtovanja produkta sodeluje z vsemi deležniki projekta, tako da se določijo vsi potrebni koraki, ki so potrebni za naslednjo iteracijo ali sprint. Sreča se z razvojnim timom preko rednih sestankov, da izboljšajo proces, odgovorijo na vprašanja in opredelijo področja za izboljšavo,
- komuniciranje: produktni lastnik je glavni komunikator med podjetjem in razvojnim timom. Dobra komunikacija zagotavlja, da deležniki v podjetju podpirajo vse pomembne odločitve, razvojnemu timu pa daje jasna navodila in prednostne naloge,
- razumevanje zahtev: z razumevanjem strankinih zahtev in pričakovanj, ima produktni lastnik popolno znanje o produktu, ki ga je treba narediti ter pomaga timu razumeti posamezne uporabniške zgodbe.

3.4.2.3 *Razvojni tim*

V samem središču delovnega procesa Scrum metode je razvojni tim. V primerjavi s tradicionalnim projektnim managementom ima tim več odgovornosti, bolj proste roke in večjo vlogo pri planiranju sprintov in razvoju izdelka. Zaželeno je, da so timi sestavljeni iz manjšega števila ljudi (od 3-9). Manj ljudi je lažje informirati, bolj so učinkoviti in

motivirani. Glavne naloge razvojnega tima so (KnowledgeHut, 2018):

- izvedba sprintov: med samo izvedbo sprinta, razvojni tim opravlja naloge načrtovanja, integracije, testiranja elementov iz seznama zgodb, ki se lahko dostavi kot kompleten izdelek po končanem sprintu. Zato se morajo samoorganizirati in medsebojno odločati, kako bodo načrtovali delo, komunicirali z ostalimi deležniki projekta in izvajali sam sprint,
- pregled in prilagoditev: vsi člani razvojnega tima morajo biti prisotni in sodelovati na dnevnih sestankih, takrat pregledujejo in preverjajo, kakšen je napredek, in prilagajajo dnevni načrt dela,
- analiziranje zgodb (angl. grooming) seznama zgodb: v vsakem sprintu mora razvojni tim nameniti nekaj časa načrtovanju naslednjemu sprintu. Pozornost je na analiziranju zgodb. To zajema ustvarjanje, izpopolnjevanje, ocenjevanje in določanje prioritet posameznih zgodb. Običajno naj bi razvojni tim dodelila deset odstotkov svojega časa za pomoč produktnemu lastniku pri vseh teh dejavnostih,
- načrtovanje sprinta: razvojni tim sodeluje pri načrtovanju sprinta, ki poteka pred začetkom vsakega sprinta. Sodeluje s produktnim lastnikom pri določanju ciljev za vsak naslednji sprint. Ko je cilj definiran, projektni tim določi prioritete elemente iz seznama zgodb, ki jih morajo narediti za doseg zastavljenega cilja. Dolžina časa načrtovanja sprinta je odvisna od dolžine samih sprintov. Daljši kot je sprint, dlje časa je treba nameniti načrtovanju. V primerjavi s tradicionalnim projektnim managementom, kjer se načrtovanje izvaja ob začetku projekta, se pri Scrum metodi načrtuje bolj pogosto, gotovo in detajlno ob začetku vsakega sprinta,
- pregled in prilagoditev izdelka in procesa: po koncu izvajanja vsakega sprinta, razvojni tim opravi dve pomembni aktivnosti: Pregled sprinta in retrospektiva. Pri pregledu sprinta, razvojni tim, lastnik produkta, skrbnik in vsi ostali deležniki projekta sodelujejo pri testiranju in preverjanju zadev in opravil, ki so bili opravljeni med končanim sprintom. Za razliko od pregleda sprinta se pri retrospektivi preverja način delovnega procesa in ali ta deluje in ali ga je treba prilagoditi, tako da zagotavlja najboljšo poslovno vrednost.

3.4.3 Potek metode

3.4.3.1 *Produktna listina*

Scrum projekt se začne s predstavo vizije končnega sistema. Sčasoma se to razvije v prečiščen in prednostni seznam del, imenovan produktni seznam del (angl. Product backlog). Ta seznam del obstaja in se razvija v celotni življenjski dobi izdelka. Predstavlja zemljevid izdelka. V vsakem trenutku je seznam izdelka enoten pogled na vse, kar mora tim storiti, in v kakšnem vrstnem redu. Obstaja samo en produktni seznam del, kar pomeni, da mora lastnik produkta sprejeti odločitve, ki so v interesu naročnika in pod vplivom razvojnega tima (Permana, 2015).

Produktni seznam vključuje različne artikle, nove funkcije, spremembe obstoječih funkcij, popravkov napak, spremembe v infrastrukturi ali druge dejavnosti, ki jih lahko projektni tim izpolni za doseganje določenega izida. Je edini avtoritativni vir za stvari, na katerih tim dela. To pomeni, da se samo izvajajo stvari, ki so del tega seznama in nič drugega. Vendar vseh elementov seznama ni treba izpolniti. Po mnenju Kumar in Sowmyavani (2012) produktni seznam predstavlja različne možnosti za doseg določenega izida in ne zaveže kako tega narediti. Elementi seznama niso predstavljeni kot uporabniške zgodbe, ampak, kot rezultat teh zgodb, primerov uporabe ali kateri koli potreben pristop, ki se timu zdi koristen. Uporabniške zgodbe veljajo bolj kot neformalna oziroma splošna razlaga programske funkcije, napisana z vidika končnega uporabnika, ali stranke (Kumar in Sowmyavani, 2012).

3.4.3.2 Seznam uporabniških zgodb razvijajoče verzije produkta

Scrum tim bo določila, katere uporabniške zgodbe bodo uporabljene in tako postanejo del seznama uporabniških zgodb razvijajoče verzije produkta (angl. Release backlog). Glavni pomen tega seznama je možnost spremljanja napredka izdaje v realnem času, s pomočjo izdajnega grafikona (angl. Release burndown chart). Tim prioritizira uporabniške zgodbe in tako točno ve, katere elemente je treba izvesti prej, katere kasneje. Razpravlja se tudi o izvedljivosti vsakega elementa, ocena in velikost navora, potrebnega za dokončanje vsake zgodbe, kar pomaga pri nadaljnjem procesu načrtovanja virov in sprintov (PremierAgile, 2021).

3.4.3.3 Uporabniške zgodbe

Pri Scrum metodi so zahteve strank ali uporabnikov pridobljene z uporabo uporabniških zgodb (angl. User stories). Cohn (2010) definira uporabniško zgodbo kot kratek, preprost opis funkcije, predstavljene z vidika osebe, ki želi nove funkcionalnosti sistema, ki ga uporablja.

Prednost uporabniških zgodb je njihova preprostost in jedrnatost. Zahteve se lahko konstantno spreminjajo med projektom. Zato je ta princip oziroma tehnika zelo učinkovita in prilagodljiva na spremembe, saj zmanjšuje nepotrebno in odvečno delo. Stranke običajno tekom projekta dodajajo nove zahteve. V Scrum metodi so uporabniške zgodbe v enostavni obliki listkov ali kartic. Pri kreiranju nove zgodbe ni potrebna nobena dokumentacija. Napisane so kratko in jedrnato, vsebujejo le pomembne informacije zahtev stranke, ki so bile pridobljene s strani produktnega lastnika. V primeru nejasnosti ali nestrinjanja glede določanja velikosti uporabniških zgodb se uporabljajo različne tehnike (npr. planiranje pokra), ki pomagajo razrešiti nejasnosti med produktnim lastnikom in ostalimi člani tima. Cohn (2010) opiše strukturo uporabniške zgodbe kot:

Kot (tip uporabnika) hočem (opraviti dejavnost), da (dosežem cilj).

Pri pisanju uporabniških zgodb niso vključeni samo člani projektnega tima, ampak tudi stranka. Pries in Quigley (2010) menita, da je najbolje, če zgodbe pišejo naročniki. Pomembno je, da zgodbe zajamejo bistvo končnega izdelka, in na kakšen način se bo uporabljalo. Uporabniške zgodbe lahko podajo vprašanja, katera se rešujejo skupaj z naročnikom, in tako omogoči resnično razumevanje zahtev in želja naročnika. Za razliko od tradicionalnega projektnega managementa, kjer je treba natančno opredeliti vse zahteve na začetku, uporaba uporabniških zgodb v Scrum metodi razbremeni vse deležnike projekta s tem, ko predpostavlja, da na začetku ni mogoče predvideti vseh zahtev in da je načrt projekta zagotovo nepopoln (Cohn, 2010). Zgodbe so na začetku zapisane v bolj splošni obliki in se bolj podrobno definirajo med projektom.

3.4.3.4 Uporabniške točke

Za oceno trajanja posameznega elementa se uporabljajo uporabniške točke (angl. Story points). To so merilne enote, ki predstavljajo količino navora, potrebnega za dokončanje posamezne uporabniške zgodbe. Pri agilnih metodah uporabniške točke nadomestijo število ur. Po mnenju Vige (2024) točke upoštevajo tri dejavnike, ki lahko vplivajo na obseg opravila in trud za dokončanje le tega. Te dejavniki so: tveganje, ponavljanje in kompleksnost. Višja je stopnja teh dejavnikov, višja je vrednost posamezne točke. Ker so zgodbe relativne, se njihova vrednost oceni na podlagi dejavnikov in medsebojne primerjave. Pomembno je vedeti, da so uporabniške točke povezane, kar pomeni, da so pomembne vrednosti in razmerja med njimi, ne pa njihova dejanska numerična vrednost. V primeru, da razvojni tim ne doseže soglasja o velikosti posamezne zgodbe, se njihova vrednost oceni po metodi poker srečanja (angl. poker meeting). Teže so uporabniškim točkam lahko dodeljene na podlagi Fibonaccijevega zaporedja. Niz števil, pri katerem je vsako število vsota dveh preteklih števil. Tako obstaja razpon med posameznimi zgodbami in so lahko ločljive pri podajanju grobih ocen trajanja zgodbe. Cilj poker načrtovanja je dodeljevanje točk uporabniškim zgodbam, spraviti tim na isto stran in razviti idejo o tem, koliko zgodb lahko tim dokonča v prihajajočem sprintu (Vige, 2024).

3.4.3.5 Planiranje sprinta

Na začetku vsakega sprinta poteka planiranje sprinta (angl. Sprint Planning). V prvem delu srečanja o načrtovanju sprinta, lastnik izdelka in tim pregledajo elemente v seznamu produktov, obravnavajo cilje in bistvo elementov. V drugem delu sestanka tim izbere elemente iz produktne seznama, ki jih bo izpolnila na koncu sprinta. Začne z elementi, ki so na vrhu produktne seznama, z drugimi besedami začnejo z elementi, ki so za lastnika izdelka najvišja prednostna zgodba. Poleg zaveze, katere elementi bodo opravljeni v sprintu, se ustvari načrt dela, ki sledi določenemu vrstnemu redu. To je ena od bolj ključnih praks v Scrum-u. Tim odloči, koliko dela bodo namenila za dokončanje, namesto da bi ga določal lastnik izdelka. To pomeni večjo zavezanost. Prvič, ker tim to dela, namesto da bi nekdo drug to opravil in drugič, ker tim sam določi, koliko dela, je potrebnega, namesto da bi imeli

nekega odločevalca, ki bi določal, koliko časa je potrebnega. Medtem ko lastnik izdelka nima nobenega nadzora nad tem, koliko je tim predana, lastnik ve, da so elementi, za katere se tim zaveže, izbrani iz vrha produktnega seznama (Deemer in Benefield, 2006).

3.4.3.6 Seznam sprinta

Ko je določen čas, ki je na voljo, tim začne z vrhnjim elementom na produktnem seznamu. Z drugimi besedami, začnejo z najvišjo prednostno zgodbo lastnika produkta in jo prenesejo v seznam sprinta (angl. Sprint backlog). Ko identificirajo elemente dela, člani podajo časovno oceno za vsako zgodbo. Posvetovali se bodo z lastnikom produkta glede točk, sprejemanja kompromisov, razbijanjem večjih elementov na manjše in na splošno zagotovili, da tim v celoti razume, kaj se zahteva od njega. Tim se bo zaporedno premikala po produktnem seznamu navzdol, dokler ne porabi vseh razpoložljivih ur. Na koncu sestanka bo tim pripravil seznam vseh zgodb in za vsako zgodbo, koliko časa bo potrebno za njeno izpolnitev. Mnogo timov uporablja vizualno orodje za sledenje zgodbam, v obliki table. Zgodbe se pomikajo po stolpcih, v vrsti, v teku, preverjanje, dokončano. Tim se zaveže, da doseže postavljeni cilj, in lastnik produkta se zaveže, da ne spremeni ciljev sprinta tekom projekta. Sprememb ni mogoče narediti, dokler se ne začne naslednji sprint. Če se splet okoliščin prednostne zgodbe začne spreminjati, ter tim izgublja čas z nadaljevanjem le te, lahko lastnik produkta prekine sprint. To pomeni, da tim prekine z delom in se ponovno začne načrtovanje sprinta (Deemer in drugi, 2012).

3.4.3.7 Dnevni stoječi sestanek

Ko se sprint začne, Scrum tim sodeluje v eni izmed ključnih praks metode Scrum, dnevni stoječi sestanek (angl. Standup meeting). To je kratek, 15-minutni sestanek, ki se izvaja vsak delovnik, ob dogovorjenem času. To je priložnost, da tim poroča sam sebi, o napredkih in ovirah. Eden za drugim, vsi člani tima, poročajo o tem kaj so opravili od zadnjega sestanka, kaj želijo storiti do naslednjega, o vseh ovirah, ki jim sledijo. Skrbnik Scrum-a naredi seznam težav in pomaga članom tima, da jih razrešijo. Med sestankom ni razprave, samo poročanje o treh stvareh. Če je razprava potrebna, ta poteka takoj po zaključku sestanka. Po srečanju člani tima posodobijo čas, ki je še potreben, da se elementi sprinta dokončajo. Te informacije so zabeležene grafično, računalniško ali vizualno, na tabli. To tablo imenujemo grafikon izgorevanja (angl. Sprint burndown chart). Prikaže, koliko dela (izmerjeno v urah ali dneh) ostane vsak dan, dokler se element ne dokonča. V idealnem primeru bi to moral biti navzdol oblikovan graf, ki bi moral biti na zadnji dan sprinta prazen. Pomembno je, da prikaže timu njihov dejanski napredek, v smeri njihovega cilja. Ne v smislu, koliko časa je bilo do sedaj porabljenega, ampak v smislu koliko dela ostaja in kaj loči tim od njihovega cilja (Schwaber in Sutherland, 2020).

3.4.3.8 *Pregled sprinta*

Po končanju sprinta poteka pregled sprinta (angl. Sprint review). Pri tem srečanju so prisotni lastnik produkta, člani razvojnega tima, skrbnik Scrum-a, stranke, uporabniki, vodilni in druge zainteresirane strani. Za dvotedenski sprint naj bi ta sestanek trajal največ dve uri. Vsak prisoten lahko sodeluje in postavlja vprašanja. Ključna ideja Scrum-a je pregled in testiranje razvitih inkrementov. Da bi odkrili možne napake z namenom popravljanje le teh. Pregled vsekakor vključuje uporabo dokončanega dela programske opreme, ki jo je tim zgradila med samim sprintom. Predstavitve programske opreme se ne tretira kot prezentacija trenutnega stanja, ampak, kot uporaba dejanskega programa v živo. Na sestanku bo prisotnih več računalnikov, ki jih udeleženci uporabljajo in testirajo programsko opremo. Ni pasivna predstavitvena seja s strani tima, ampak aktivna seja, kjer uporabniki testirajo programsko rešitev. Po pregledu sprinta lahko lastnik produkta spreminja produktni seznam. Dodaja nove, odstranjuje zastarele in pregleduje obstoječe elemente. Lastnik produkta mora zagotoviti, da so spremembe zabeležene v produktnem seznamu (Demmer in drugi, 2010).

3.4.3.9 *Retrospektiva sprinta*

Pregled sprinta vključuje pregled in prilagoditev v zvezi z izdelkom. Retrospektiva sprinta (angl. Sprint retrospective), ki sledi pregledu, vključuje pregled in prilagoditev v zvezi s postopkom, ki ga uporablja razvojni tim in okoljem. To je priložnost, da skupina razpravlja, o tem, kaj deluje in kaj ne, in se dogovorijo o spremembah, ki jih je treba poskusiti. Za razliko od pregleda sprinta pri retrospektivi sprinta ni potrebna udeležba lastnika produkta. Obstaja mnogo tehnik za izvajanje retrospektive - najbolj enostavna je vizualna predstavitev. Dva stolpca »kaj je delovalo dobro« in »kaj bi lahko delovalo bolje«. Vsak udeleženec doda enega ali več elementov v posamezen stolpec. Ponavljajoči se elementi se sortirajo, tako da skupni elementi postanejo jasni. Nato tim išče vzroke, se strinja o majhnem številu sprememb, ki jih je treba poskusiti v prihajajočem sprintu, skupaj z zavezo za pregled rezultatov na naslednji retrospektivi sprinta. Timi lahko smatrajo retrospektivo, le kot osredotočanje na težave in to lahko privede do tega, da se sestanek smatra kot nekaj negativnega. V resnici se retrospektiva osredotoča tudi na pozitivne stvari, spodbuja kreativno razmišljanje, nadgrajuje znanje in vzpostavlja boljše učno kulturo (Deemer in Benefield, 2006).

Med sprinti ni veliko časa. Timi običajno gredo iz retrospektive v planiranje sprinta takoj, ko je to mogoče. Eden od principov agilnih metod je vzdržljiv tempo in samo pri normalnem delovnem času ter stopnji dela, se lahko ta cikel izvaja v nedogled. Produktivnost skupine raste skozi razvoj praks, odstranjujejo ovire, ki vplivajo na samo produktivnost in ne z izčrpanjem skupine ali kompromisi kakovosti izdelka. Sprinti se nadaljujejo, vse dokler se lastnik produkta ne odloči, da je izdelek pripravljen na izdajo. Popolna vizija Scrum metode je, da je inkrement potencialno dobavljiv na koncu vsakega sprinta, kar pomeni, da ni potrebe po testiranju ter pripravi in izdaji dokumentacije. Obstajajo implikacije, da je v vsakem

sprintu vse popolnoma končano in je lahko izdan po pregledu sprinta. Vendar imajo številne organizacije šibke razvojne prakse, orodja, infrastrukturo ali pa obstajajo druge olajševalne okoliščine (npr. okvara strojne opreme) (Deemer in drugi, 2012).

3.4.4 Prednosti in slabosti Scrum metode

Prednosti Scrum metode (SCRUMstudy, 2016):

- z metodo Scrum je odstranjevanje napak ali njihovo odpravljanje bistveno enostavnejše kot pri drugih metodah,
- enostavna prepoznavnost vseh faz postopka v celotnem njegovem razvoju,
- spremembe, ki jih zahtevajo stranke, je lažje obvladovati, saj je Scrum metoda, sestavljena iz kratkih sprintov s stalnimi povratnimi informacijami,
- razvoj izdelka se lahko spremeni v katerikoli fazi zaradi večje fleksibilnosti metode,
- z vključevanjem strank se zagotavljajo najboljši rezultati,
- omogoča uspeh v podjetjih, v katerih je obsežna in kompleksna dokumentacija,
- hitri rezultati in preprost postopek testiranja za boljše rezultate in kakovost dela,
- to je »lahka« metoda, ki vključuje pogoste posodobitve napredka. Posodobitve so dosežene z rednimi sestanki,
- je iterativne narave in potrebuje stalne povratne informacije uporabnika za izboljšanje postopka,
- Scrum metoda vključuje vsakodnevne sestanke, kar pomaga pri prepoznavanju težav in njihovem pravočasnem reševanju,
- stranke imajo dostop do preglednosti postopka, ki jim omogoča sledenje napredka in merjenje individualne produktivnosti,
- z uporabo metode je tim vedno motivirana,
- izdelek je zelo pogosto dostavljen ob pravem času,
- metoda je dokaj ugodna za proračun in v zelo kratkem času daje zelene rezultate.

Slabosti Scrum metode (Ageling, 2020):

- postopek zahteva od Scrum mojstra, da verjame v svoj tim. Strogi management lahko privede do neuspeha celotnega projekta,
- če član tima med razvojem zapusti tim, to negativno vpliva na projekt,
- če rok za dostavo izdelka ni določen, se obseg zgodb lahko spremeni, saj bodo razvojni timi zahtevali več funkcionalnosti,
- nima predvidenih časovnih omejitev in ocene stroškov, zaradi česa se lahko razširi na več sprintov,
- zahteva močno zavzetost vseh članov tima, ali nekateri člani ne bodo kos nalogi. To lahko povzroči resne težave,
- večinoma je primeren za majhne time, ki imajo veliko kohezije in razumevanja,

- testna skupina mora opraviti regresijsko testiranje razvoja po vsakem sprintu, kar otežuje izvajanje metode Scrum,
- člani tima so pod večjim pritiskom, zato morajo porabiti veliko časa za razvoj projektov.

Scrum se ponaša s svojo preprostostjo in lahkotnostjo, vendar po nekaterih ocenah naj bi ga skoraj polovica podjetij, ki ga uporablja, izvajala napačno. V nadaljevanju bo podanih nekaj razlogov, zakaj ta navidezno nestandardna metoda izkaže za tako težko izvedljivo in kaj je mogoče sprejeti za zagotovitev njenega uspeha. Zelo pomembno je dobro razumevanje načel, strategij in pristopov te metode. Enako pomembno je, da imajo vsi člani tima skupno razumevanje Scrum okvirja, kot tudi vloge, ki jih opravljajo. Tim mora poznati različne vloge in odgovornosti, ki bi jih morali imeti lastnik izdelka, skrbnik Scrum-a in razvijalci. Običajno bo na začetku celotni tim opravil formalno usposabljanje in praktične vaje. Skrbnik Scrum-a je dolžan nenehno trenirati tim in organizirati enako usposabljanje za nove člane, ki se pridružijo timu. V večini organizacij se to ne dogaja, kar je eden od najpogostejših razlogov, da člani tima različno razumejo proces. Tako tim, kot organizacije bi morale imeti podporo za implementacijo Scrum metode. Timi, ki proaktivno sprejemajo agilno metodo, so po navadi uspešnejše od timov, katere jih vodstvo samo pooblasti, da jo uporabljajo (Liyanarachchi, 2019). Sprememba kulture je nedvomno največji izziv, s katerim se podjetja soočajo pri poskusu izvajanja kakršne koli poslovne spremembe in pri Scrum metodi ni nič drugače. Ni nobena skrivnost, da ljudje ne marajo sprememb in brez podpore ključnih deležnikov (v podjetju), bo implementacija agilne metode neuspešna. Scrum se dotika vseh področij razvoja programske opreme, zato je treba v proces vključiti vse vpletene. Scrum ne bo uspešen, če bo manjšina gonilna sila uporabe metode. Scrum uvaja več tehnik in terminologij, ki jih ljudje morda ne poznajo, in je v nasprotju z nekaterimi dolgoletnimi pravili razvoja programske opreme, zaradi česar se lahko člani tima počutijo nelagodno. To je lahko še bolj zahtevno, ali je kultura v organizaciji v nasprotju s samimi agilnimi vrednotami, zato se je treba pred implementacijo te metode prepričati oziroma imeti jasno vizijo, kakšne vrednote bo prinesla timu ter sami organizaciji. Šele ko to imamo, je potem treba prepričati zainteresirane strani. Faza določitve velikosti tima je treba opraviti v več sejah. Glavni namen le teh je, da se v člane tima vpelje agilna miselnost, da sami začnejo izvajati Scrum metodo in ne, da jim preprosto vsilimo to metodo. Tako se možnost uspeha implementacije poveča že pred samim procesom izvajanja (Hawkins, 2018).

Jedro uspešne implementacije Scrum metode pa so tri ključne vloge: skrbnik Scrum-a, lastnik produkta in razvojni tim. Ta triada je tisto zaradi česa je Scrum tako uspešen, ko deluje. Za večino neuspešnih posvojitvev te metode je odsotnost le teh. V praksi se lahko zgodi, da ena oseba opravlja dve vlogi, kar ni v skladu s Scrum metodo. Skrbnik Scrum-a ne more biti produktni lastnik, saj skrbi za izvajanje principov metode in je zato procesna avtoriteta. Vloga produktnega lastnika se pa konstantno razvija, saj mora imeti dovolj znanja o samem izdelku in razumeti potrebe in zahteve naročnika, ki so prenesene, razložene razvojnemu timu. Zagotovljena mora biti avtonomija razvojnega tima pri razvijanju samega izdelka. Scrum dodeluje poimenovane vloge, ker jim to daje najboljše možnosti, da je

prisotna, primerna miselnosti v timu. Toda na žalost jim to ne zagotavlja uspeha.

Po ugotovitvah Yorka (2016) so glavni razlogi, zakaj ta metoda ne uspe, posledica tega, da vsaka vloga ni sprejela svojega ključnega načina razmišljanja. Preveč je Scrum skrbnikov, ki razumejo pravila, a nimajo agilne miselnosti. To lahko privede do ustvarjanja efekta vagona, kjer timi sledijo pravilom, ne da bi sprejele načela te agilne metode. Tak tim ne bo nikoli razumel, zakaj sledi smernicam, zato se ne bo izboljšal. V teh okoljih se Scrum uporablja za nadzor razvoja in ne za njegovo omogočanje. Sami lastniki izdelkov, ki nimajo razmišljanja o strankah, gradijo tisto kar željo in ne nujno kar želi stranka, ali slepo zagotavljajo, kar se zahteva, ne da bi razumeli, zakaj je to potrebno, ali kar je najhuje, redno ne pregledujejo razvijajočega izdelka, kar ustvari manjkajoče priložnosti za odziv na nastajajoče potrebe in vtise strank. Rezultat je izdelek, ki zgreši svoje bistvo. Obstaja veliko razvojnih timov, ki jim manjka produkcijska miselnost in bodo preveč arhitekturne, inženirske in se zgolj dotaknile kakovosti, vzdrževanja ali celo oblikovanja, kar vse zmanjša možnost za uspeh tima. To se bo verjetno pokazalo v timu, ki opravlja nepotrebno ali nepomembno delo, da bi preprosto ostala zaposlena. Razvijalci zamudijo priložnosti za zagotavljanje prave rešitve, ker se ukvarjajo z delom, ki ima za stranko zelo malo vrednost (Yorke, 2016).

Vrednosti triade ne opredeljujejo same vloge, ampak njihova miselnost. Imeti tim, ki ima močnega člana z agilno miselnostjo, skupaj z znanjem in veščinami za podporo ter možnostjo, da se osredotoči na njeno uporabo, pomaga timu, da se osredotoči na nenehno izboljševanje in razvoj agilnih praks. Za ta namen se imenuje vloga Scrum skrbnika. Vloga podpira potrebo po osredotočenosti in izrecno išče nekoga z agilno miselnostjo kot primarno spretnostjo. To je zavestna odločitev, da se vlogo s tako miselnostjo izrecno vključi v tim (Guay, 2020). Prav tako je ključnega pomena imeti močnega člana tima z miselnostjo stranke, ki želi sodelovati s stranko in zagotoviti, da stranka dobi tisto, kar resnično potrebuje. To je namen lastnika izdelka in ta vloga omogoča osebi, da se osredotoči in odstrani kakršno koli navzkrižje interesov, do katerih bi lahko prišlo. To zagotavlja čas in prostor za dobro opravljanje dela. Lastnik določa, kaj se mora zgoditi, da se poveča vrednost izdelka. S tem je lastnik izdelka ključnega pomena pri določanju poteka dela tima in usmerja tim, kaj naj izdela ali razvije. Da bi metoda Scrum uspela, je to ključnega pomena. Vendar pa pogosto kdo drug poskrbi za to in lastnik sledi ukazom. To odvzame glavno komponento Scrum tima, sposobnost samoorganiziranja. Če nimate pravega lastnika produkta, ki bi bil odgovoren za povečanje vrednosti izdelka, bo tim zagotovo neuspešen pri uporabi te metode (Isom, 2019).

In končno mora obstajati več funkcionalni razvojni tim z močno produkcijsko miselnostjo za zagotavljanje dobro zasnovanega, visokokakovostnega in vzdrževanega izdelka, ki je pravi za stranko. Če želimo to narediti, moramo razumeti vitko in sistemsko razmišljanje in ne le biti zaposleni s pisanjem kode. Scrum je okvir za kompleksna produktna okolja. V tako zapletenih okoljih ni jasnega načina za izdelavo izdelka. V Scrum-u je namen produktnega

inkrementa pregled in možnost prilagajanja, če je to potrebno. Da bi povečali možnosti za uspeh, mora imeti razvojni tim svobodo eksperimentiranja in ustvarjanja. Številne razvojni timi morajo upoštevati stroga pravila o tem, kako in s katerimi zaporednimi koraki zgraditi svoj izdelek. To je zanesljiv način, da se onemogoča popolno delovanje razvojnega tima. Empirizem brez eksperimentiranja ne obstaja. Scrum brez empirizma ni Scrum. Ključni vidik Scrum-a je zmožnost pregleda produktnega inkrementa na koncu sprinta. To pomeni, da mora biti razvojni tim sposoben zgraditi inkrement funkcionalnosti, ki bi jo bilo mogoče zagnati, česar te zmogljivosti številni timi nimajo. Ali nimajo vseh veščin za izgradnjo končnega inkrementa ali pa potrebujejo dalj časa trajajoči sprint. V skrajnih primerih traja en mesec ali več timov, da zgradijo nekaj, kar je kvalificirano kot končano. Številni timi še vedno delujejo v silosih in gradijo del izdelka, ki deluje le, ko so združeni. To krši namen Scrum-a, brez pregleda in prilagajanja, brez empirizma, razvojni tim ne razume, ali gradi kaj vrednega. S tem se tim bliža neuspehu. Razumevanje vrednosti dela je pomembnejše od maksimalne izrabe le tega. Vse te miselnosti skupaj dosegajo cilj. Če kakšna manjka, trpi celotni tim ter posledično celotni projekt (Ageling, 2020).

Čeprav je Scrum odličen okvir za razvoj programske opreme, zaradi slabega poznavanja agilnih metod, ni primeren za vsakogar. Za nekatera podjetja je znano, da sprejemajo to metodo samo zato, ker to počnejo vsi ostali v panogi ali ker jim pomaga pridobiti posel, če rečejo, da so agilni. To je slaba poslovna praksa. Treba je pogledati izdelek, tim in podjetje ter nato določiti, ali je to prava metoda za podjetje. Scrum je hitro premikajoča, ponavljajoča se razvojna metoda, ki je idealna za nove ali relativno nove izdelke. To je okvir, ki pomaga hitro prinesiti vrednost, pri čemer se osredotoča na koncepte, kot je minimalno uspešen in tržen izdelek. Ko pa je izdelek že dobro razvit in pozicioniran na trgu, bi bilo treba resnično razmisliti, ali je po vsakem končanem sprintu smiselno izdati potencialno del produkta. Izdelki, ki so že v fazi zrelosti svojega življenjskega cikla, so bolj primerne druge metode, kot je na primer Kanban (Hawkins, 2018).

3.5 Kanban

Kanban je agilna metoda, ki spodbuja pretok dela in skuša preprečiti, da bi naloge, ki se opravljajo, obtičale med izvajanjem in tako bile blokirane ali bi zamujale. Ideja je, da tim dela na manj nalogah naenkrat in se osredotoča na zmanjšanje časa, porabljenega za vsako stopnjo razvoja. Namen metode je zagotoviti tekoč in stabilen pretok dela med posameznimi fazami oziroma aktivnostmi projekta (O'Malley, 2023). Kanban je metoda za izboljšanje procesov, ki jo uporabljajo agilni timi. Timi, ki uporabljajo Kanban, začnejo z razumevanjem trenutnega načina izdelovanja programske opreme in ga sčasoma izboljšujejo.

Tako kot Scrum, Kanban zahteva posebno miselnost. Po Naydenovu (2015) je Kanban metoda načrtovanja, ki jo je razvil Toyotin inženir Taiichi Ohno za vitko proizvodnjo (angl. Lean production). Zasnovan je kot sistem za načrtovanje in nadzor proizvodnje in zalog.

Kanban, poznan po sposobnosti, da omogoča pregled nad ogromnimi količinami fizičnega izdelka, je ena izmed najpomembnejših metod, s pomočjo katere lahko delovne skupine izvedejo pravočasno (angl. Just in time) proizvodnjo.

Ena najpogostejših pasti, s katero se ljudje srečajo, ko spoznajo Kanban, je poskus, da ga obravnavajo kot metodo za gradnjo programske opreme, vendar ni. Kanban je metoda za definiranje, management in izboljšanje storitev, ki zagotavlja znanja dela, kot so strokovne storitve in razvoj tako fizičnih kot programskih izdelkov (Naydenov, 2021). Pomaga razumeti metode, ki jih podjetja uporabljajo, in poiskati načine, kako jih izboljšati. Danes smo Kanban prilagodili kot metodo managementa delovnega toka, ki nam pomaga vizualizirati svoje delo, povečati učinkovitost in omogočati agilnost.

Kanban metodo managementa projektov so izumili managerji korporacije Toyota na Japonskem. Metoda je nastala v štiridesetih letih dvajsetega stoletja in jo zaradi njenega izvora nekateri zamenjujejo z vitko metodo (angl. Lean method). Vendar pa Kanban za razliko od vitke metode ni usmerjen v t. i. odpadke, to je karkoli, kar ne pripomore in ne doda vrednosti izdelka, temveč v optimizacijo proizvodnega procesa z uravnavanjem dobave surovin (Calendar-Canada, 2022). Kanban je zelo vizualna metoda managementa delovnega toka. 83 % timov, ki prakticirajo vitko metodo, uporablja Kanban za vizualizacijo in aktivni management ustvarjanja izdelkov s poudarkom na stalni dostavi, obenem pa ne dodaja večjega stresa življenjskemu ciklu razvoja programske opreme. Tako kot Scrum je tudi Kanban proces, ki je zasnovan tako, da pomaga timu, da sodelujejo učinkovitejše (Blueprint, 2022). Kanban metoda je precej manj strukturirana kot Scrum. To ni procesni okvir, ampak model za uvedbo sprememb s postopnimi izboljšavami. Načela Kanban-a se lahko uporabijo v kateremkoli postopku, ki se ga že izvaja (Marschall, 2015).

Kanban se razlikuje od ostalih agilnih metod, kot sta Scrum in XP. Scrum se v glavnem osredotoča na management projektov: obseg dela, ki bo opravljeno, kdaj bo to delo opravljeno in ali rezultati tega dela ustrezajo potrebam uporabnikov. Fokus XP je razvoj programske opreme. Načela in prakse XP temeljijo na ustvarjanju okolja, ki je ugodno za razvoj programerskih navad, ki jim pomagajo oblikovati in graditi kodo, ki je enostavna tako po strukturi kot možnosti spreminjanja. Kanban pomaga timu izboljšati način izdelave njihove programske opreme. Tim, ki uporablja Kanban, ima jasno sliko dejavnosti, ki jih uporabljajo za izdelavo programske opreme, kako komunicirajo s preostalim podjetjem, kje naletijo na prepreke zaradi neučinkovitosti in kako se sčasoma izboljšujejo z odstranjevanjem glavnih vzrokov teh preprek. Ko tim izboljša način izdelave programske opreme, to imenujemo izboljšanje procesov. Kanban je dober primer uporabe agilnih idej za ustvarjanje metod izboljšanja procesa, ki je enostavna tako za implementacijo kot uporabo (Stellman in Greene, 2014).

Procesi, kot Scrum, imajo kratke iteracije, ki v majhnem obsegu posnemajo življenjski cikel projekta in imajo ločen začetek in konec za vsako ponovitev. Kanban omogoča razvoj

programske opreme v enem velikem razvojnem ciklu, kar pomeni nemalo opravljenega dela v eni tako imenovani iteraciji. Iz seznama prednostnih zahtev se ustvari majhen delovni element ali opravilo, nato se začne razvojni postopek, običajno z določenimi zahtevami. Delovnemu elementu ni dovoljen prehod na naslednjo fazo, dokler se nekaj zmogljivosti ne sprostí. Razvijalci z nadzorom števila dejavnih opravil še vedno postopoma pristopajo k celotnemu projektu, kar poda možnost uporabe agilnih načel (Inflectra, 2014).

3.5.1 Koraki metode Kanban

3.5.1.1 Vizualizacija

Kanban je ena izmed najbolj priljubljenih metod med navdušenci agilnih metod. Zaradi tega je široko sprejet. Ponuja sposobnost vizualizacije, omejevanje dela v teku in dejansko doseganje višje ravni produktivnosti. Glede na to, da pridobitev Kanban strukture s tablo in premikanje kartic iz enega stolpca v drugega, ne pomeni, da se v resnici uporablja metodo Kanban. Kartice Kanban so preprosto vizualni nadzorni sistem, ki omogoča, da se organizira potek dela. Kanban tabla predstavlja temeljni pripomoček pri uspešnem delovanju agilne metode. Njena uporaba ponuja pomoč pri vizualiziranju dela, omejuje delo v teku in povečuje učinkovitost. Razvojnim timom omogoča celoten pregled nad opravljenim delom in si tako lažje organizirajo dnevna opravila (Nikolov, 2020).

Za preglednost in učinkovitost Kanban table, se na njej uporablja pet temeljnih elementov. Kartice ali t. i. vizualni signali, ki so lahko poljubne oblike (nalepke, listki, kartice ipd.), predstavljajo vsa opravila projekta, ki jih je treba narediti. Vsako opravilo zasede eno kartico. V digitalni obliki, kartice vsebujejo naslov, lastnika opravila oziroma kartice in vse naloge, ki so bile opravljene. Tako imajo vsi deležniki pri projektu konstanten pregled nad projektom in njegovo izvedbo. Ko so kartice dane na tablo, se jih razvrsti v stolpec produktivni seznam. Vsak stolpec predstavlja določeno aktivnost. Potovanje kartice iz ene aktivnosti v drugo se imenuje delovni tok (angl. Workflow). Običajno se uporabljajo stolpci: kaj je treba narediti (angl. To do), v teku (angl. In progress) in končano (angl. Done). Običajno projekti v realnem svetu vsebujejo bolj kompleksne aktivnosti. Določi se maksimalno število kartic v določenem stolpcu. Ko je maksimum dosežen, mora tim opraviti delo, ki ga opravlja, da lahko prenesejo kartice v nov stolpec. Tako tim hitro opazi potencialna ozka grla (angl. Bottleneck) in ustrezno urgira. Leva stran table predstavlja nabor obveznosti ali dnevnik, do katerega lahko dostopajo vsi deležniki projekta. Vsaka ideja se lahko pretvori v kartico in postavi v dnevnik obveznosti. Točka obveznosti (angl. Commitment point) predstavlja trenutek, ko je kartica iz dnevnika prestavljena v stolpec »v teku«. Takrat se tim zaveže, da bo končala opravilo. Ko kartice dosežejo konec oziroma je potek dela končan, se to imenuje točka dostave (angl. Delivery point). Običajno to pomeni, da naročnik prejme končani izdelek ali storitev. Zato tim želi v najkrajšem času priti od točke začetka do točke dostave. Čas med točkama se imenuje dobavni rok (angl. Lead time) (Meiling, 2019).

V praksi so Kanban table uporabljene v različnih oblikah. Fizične table vsebujejo listke z opravili, ki se premikajo od leve proti desno do dokončanja opravila. Največja prednost fizične table je njena dostopnost. Običajno so postavljene na vidnih mestih, kjer do nje lahko dostopajo vsi. S tem je omogočena tudi lažja komunikacija pri vsakodnevnih timskih sestankih. Ker pa v sodobnem času pri projektu sodelujejo ljudje iz celega sveta, kar omogoča sodobna komunikacijska tehnologija, fizične table niso praktične. Zato se uporabljajo digitalne table, do katerih lahko dostopajo vsi deležniki, iz celega sveta, ob katerem koli času. Ker na digitalni tabli ni omejitve prostora, lahko kartice vsebujejo več informacij o samih opravilih. Vodi se zgodovina opravljenega dela pri vsaki aktivnosti, vsi pogovori in komentarji so na voljo v pogled vsem deležnikom projekta. Dostopnost posodobljenega projekta v realnem času predstavlja največjo prednost digitalne table. Prednost pa lahko predstavlja tudi nevarnost, saj dostopnost lahko vodi do prevelike uporabe oziroma zlorabe table. Deležniki, ki niso večji uporabe, lahko zasičijo dnevnik ali nabor opravil, kar lahko drastično podaljša izvedbo projekta (Sellgren, 2014).

3.5.1.2 Omejitve števila opravil

V izogib zastojem in preveliki količini dela, razvojni timi postavijo omejitve na količino kartic v stolpcu. Kanban projekti vsebujejo omejitve števila opravil, ki so merilo zmogljivosti, ki osredotoča razvojni tim na le majhen obseg dela naenkrat. Šele ko se naloge dokončajo, se nove naloge dodajajo v cikel. Omejitve dela v teku povedo, koliko delovnih opravil je v določenem času dovoljeno biti v določenem stanju. Če določeno stanje doseže vnaprej določeno mejo dela v teku, se nobeno novo opravilo ne more dodati v to stanje. Celotni tim mora najprej pomagati očistiti napolnjeno stanje. Delovni predmeti, ujeti v stanju, bodo prikazali prepreke in ozka grla na Kanban tabli (Marschall, 2015). Pri Kanban-u omejitve dela v teku tako določajo maksimalno količino dela, ki se lahko opravi znotraj vsake aktivnosti. S pravilno postavljenimi omejitvami se projektni tim izogne potencialnim ozkim grlom in s tem doseže hitrejši in učinkovitejši delovni tok. Z razbitjem večjih opravil na manjše fragmente tim stremlji k dokončanju nalog in minimalizirati število večjih opravil, ki lahko zavzamejo nemalo časa. Naročnik projekta oziroma kupec tako dobi več končanih, manjših in uspešnih rezultatov v krajšem času.

Z omejitvami je tim prisiljen dokončati opravilo, ki je v določeni aktivnosti. S tem se onemogoča dodajanje novih nalog, ki se lahko izkažejo za težavne in naredijo projekt drastično bolj kompleksen. Z jasnimi kratkoročnimi cilji se tako razbremeni projektni tim. Kakšne omejitve ali koliko opravil se lahko izvajajo, mora biti smiselno dogovorjeno na začetku projekta. Tukaj je pomembna komunikacija med managerjem projekta in njegovim timom. Managerji želijo opraviti čim več dela v najkrajšem času in tako običajno pretiravajo s količino nalog, kar negativno vpliva na učinkovitost tima. Količina opravil oziroma nalog, ki jih je treba opraviti, mora biti smiselno določena. S tem se tim izogne stagnaciji, preobremenjenosti in večopravnosti. V praksi se velikokrat na omejitve gleda kot na oviro in je želja po večji količini opravljenega dela velika. Tim se mora upreti skušnjavi manjšanju

omejitev ali povečanju številu nalog, ki se lahko izvajajo. Priporočljivo je, da se kot alternativa, tim konstantno izobražuje in se mu omogoči pridobitev novih znanj, ki so potrebne za hitrejše reševanje ali opravljanje nalog znotraj omejitev (Nikolov, 2020).

Po mnenju Boiser (2020) se za lažje postavljanje omejitev, uporabljajo štiri smernice. Kot prvo, je pri naštevanju nalog treba upoštevati, da ni priporočljivo delati na opravi več kot 16 ur. S tem se tim izogne ozkim grlom in ji daje samozavest, da se sooči z vsako naslednjo nalogo. Druga smernica pravi, da je treba upoštevati znanje, ki ga tim ima. Za vsakega člana so lahko omejitve drugačne, saj nimajo vsi iste ravni znanja. Člani, ki so strokovnjaki na svojem področju, naj svoje znanje delijo s svojimi sodelavci in tako pohitrijo in povečajo delovni tok. Tretja smernica pravi, naj se člani, ki stagnirajo oziroma imajo manjšo količino dela, motivira, da pomagajo sodelavcem na težjih opravilih in pri tem pridobijo dodatno znanje. Kot zadnja, četrta smernica pravi, naj se znotraj organizacije ali projekta ohranja okolje, brez pritiska na člane tima, da opravijo delo hitro in površno. Cilj je razviti in izdelati kvaliteten izdelek za naročnika (Boiser, 2020).

3.5.1.3 *Management toka*

Management in izboljšanje pretoka je jedro Kanban sistema. Pomaga pri managementu pretoka s poudarjanjem različnih stopenj poteka dela in statusa dela v vsaki fazi. Glede na to kako dobro je potek dela določen in omejitve števila opravil nastavljene, bo vidno, ali bo potek potekal nemoteno v okvirju omejitve števila opravil ali se bo pa delo kopičilo in bo začelo zadrževati nadaljnji potek dela. Vse to vpliva na to, kako hitro delo poteka od začetka do konca. Kanban pomaga timu analizirati sistem in ga prilagajati za izboljšan pretok, tako da se zmanjša čas, potreben za dokončanje vsakega dela. Ključni vidik tega procesa opazovanja dela in reševanja ter odpravljanja ozkega grla je preverjanje vmesnih faz čakanja in preveriti, koliko dolgo delovni predmeti ostanejo v teh predajnih fazah (angl. Handoff stages). Zmanjšanje časa, ki se porabi v fazi čakanja, je ključnega pomena, da se zmanjša čas posameznega cikla. Ko se izboljša pretok, delo tima poteka bolj tekoče in postane bolj predvidljivo. S tem ko postane bolj predvidljivo, večja je zanesljivost, kdaj se bo kakšno delo opravilo in večja je zanesljivost, da se bo stranki dostavilo delo, ki ji je bilo obljubljeno. Večja sposobnost napovedovanja časa dokončanja je velik del izvajanj Kanban sistema (Bhaskar, 2022). Ta postopek je po mnenju Meiling (2019) usmerjen v domnevno slabo vedenje strank. Stranka ne pove timu, kdaj si je premislila. To procesno pravilo je hitro spremenilo vedenje na obeh straneh. Tako timi kot stranke. Oženje povratnih zank pospeši sodelovanje s stranko. Obe strani sta skupaj ustvarili končne izvedbene korake za nov način skupnega dela. Po ugotovitvah Meiling (2019) učinkoviti management poteka dela vključuje osredotočanje na različne procesne komponente, kot so opravila, tim, tehnologija in izdelek. Osredotočanje na potrebe strank omogoči, da se na ostale vidike osredotočimo pravilno. Prepoznavanje in odstranjevanje ovir v procesnem ciklu so znaki učinkovitega opravljanja toka dela. Nemoten potek dela je rezultat managementa sistema in ne ljudi (Meiling, 2019).

3.5.1.4 Eksplicitne procesne politike

Kot del vizualizacije procesa je smiselno tudi izrecno opredeliti in vizualizirati procesna pravila, smernice za to, kako opraviti določeno delo, ki se opravlja. Chiva (2020) ugotavlja, da ne glede na Kanban prakse, eksplicitno določanje pogojev, pravil naredi veliko razliko med visoko zmogljivo in povprečno, manj zmogljivo organizacijo. Zato med drugim poudarja pomen te Kanban prakse. V procesu je bilo vse narejeno in vse odločitve bi morale biti pregledne. Preglednost tako predstavlja pomembno razliko pri splošni uspešnosti organizacije. Posamezni člani tima opravljajo toliko nalog in se odločajo posamično. To mora biti jasno vsakemu članu ali organizaciji kot celoti, ko je to le mogoče. Za to morajo biti stvari, kot je politika omejitve števila opravil, razpoložljivost virov in kaj bo obravnavano, smatrano kot opravljeno in jasno vsem, katerih se te zadeve tičejo. Z oblikovanjem izrecnih smernic procesov se ustvari skupna podlaga za vse udeležence, da razumejo, kako narediti katero koli vrsto dela v sistemu. Politike so lahko na ravni stolpcev, so lahko v obliki kontrolnih seznamov korakov, ki jih je potrebno opraviti za vsako vrsto delovnega inkrementa, so merila za vstop in izstop iz posameznega stolpca. V bistvu je lahko karkoli, kar članom tima dobro pomaga pri managementu poteka dela na Kanban tabli (Chiva, 2020). Pravilnike je treba izrecno opredeliti in vizualizirati običajno na vrhu table ter na vsakem pasu, stolpcu. Ko so vprašanja in pravila jasna, postanejo skupinske razprave bolj racionalne in usmerjene v rezultate. Izboljševanje preglednosti deluje kot ukrep za krepitev zaupanja med ljudmi v organizaciji. Spodbuja tudi pobudo in spodbuja člane tima, naj učinkoviteje sodelujejo z drugimi. Preveč zapletene politike lahko povzročajo nesporazume in vrzeli v komunikaciji. Torej morajo biti preproste in enostavne za razumevanje. Predvsem pa morajo biti politike (procesna pravila, smernice) vsem vidne (SimpliAxis, 2019).

3.5.1.5 Hitre povratne zanke

Kanban spodbuja majhne inkrementalne in evolucijske spremembe. Vedeti, katere spremembe so potrebne je odvisno od vzpostavitve zanesljivih povratnih zank. Povratne informacije niso samo bistven, ampak sestavni del Kanban-a. Po mnenju Chiva (2020) povratne zanke predstavljajo impulz, ki ohranja organizacijo živo in povezano. Omogočajo redno povezovanje različnih ravni odločanja v organizaciji z izmenjavo informacij za stalno izboljševanje. Da bi v celoti prenesli spremembe organizaciji, nas Kanban prakse opominjajo, da je treba vse storitve, ki delajo s Kanban-om, učinkovito medsebojno povezati. Evolucijski proces ne more delovati brez povratnih zank.

Povratne informacije se zbirajo v različnih fazah projekta. Teamhood (2021) meni, da za implementiranje na nivoju dostave storitve v organizacijah, Kanban uporablja štiri posebne prakse za povratne informacije: stoječa srečanja, pregled zagotavljanja storitev, pregled operacij ter pregled tveganja. Točno, kje zbrati informacije in kako pogosto, je običajno odvisno od narave projekta in kaj bo najbolj delovalo s projektnim timom. Stoječa srečanja so dobra priložnost za dobivanje povratnih informacij na različnih ravneh. Ta kratka srečanja

potekajo redno, in kot že samo ime pove, so srečanja tako kratka, da lahko člani tima stojijo med celotnim trajanjem srečanja.

Ko ima tim informacije in Kanban tablo pred seboj, si lahko ogledajo, kakšno nalogo naj prevzamejo in kako jo je treba opraviti. Imajo vpliv tudi na socialni vidik, saj pomaga pri grajenju tima. Ne samo, da se posamezni člani tima učijo in izvejo, kaj se na splošno dogaja skozi redna srečanja, temveč imajo vpogled v delo ostalih članov tima. Na ta način bolje razumejo potrebe članov tima, komunikacija postane bolj neposredna in iskrena preko grajenja zaupanja med samimi člani tima (Teamhood, 2021). Namen povratnih zank je, da se lahko primerjajo pričakovani izidi z dejanskimi izidi in se po potrebi prilagodijo. Natančneje želimo prilagoditi procese in politike kot rezultat povratnih informacij in razumevanje trenutnih rezultatov v primerjavi s svojimi pričakovanji ali željami (Sellgren, 2014).

3.5.1.6 Izboljšuj sodelovalno, razvijaj eksperimentalno

Po mnenju Nikolov (2020) je Kanban sistem v lasti tima, ki ga uporablja za management toka njihovega dela. Zato izboljšave in spremembe najbolj opravijo ljudje v timu. V Kanban-u je tim lastnik pretoka vrednosti v svojem procesu. Člane vseh starosti in raznih vlog je treba spodbujati, da ga izboljšujejo. Pričakuje se, da bo učinkovita organizacija uporabila sodelovanje pri dobavi svojih izdelkov, storitev. V tem smislu, prvi del izboljšuj sodelovalno, pomeni, da se del kreativne energije uporabi za optimiziranje osnovnega sistema za dostavo. S stališča sodelovanja, koncept povratnih ciklov in samoorganizacije, ki spadajo v kategorijo preglednosti, prevzamejo drugačen pomen. Povratni cikli vključujejo večje število ljudi, ki sodelujejo na eni ravni. Ko tim deluje samoorganizirano, se izboljšave in inovacije razvijajo iz sodelovanja, ki izhajajo iz dejstva, da je samodejno vključena ustrezna skupina zaposlenih (Nikolov, 2020).

Meiling (2019) meni, da ni le pomembno gledati števil interakcij v delovnem procesu ali vrstnega reda, v katerem potekajo, je pomembno, da se osredotočite tudi na kakovost posameznih interakcij. To je treba razumeti kot neodvisno strategijo in ne kot čisto tehniko, ki se uporablja v okviru posamezne možnosti izboljšanja. Pomembno je opredeliti ključne procese, ki jih upočasnjujejo zamude, predelave, interakcije. Poleg tega je treba najti potencialne zaposlene ali tim, ki imajo redke priložnosti, da se vključijo v kakovostno sodelovanje, saj se s tem lahko poveča ustvarjalnost same organizacije. Če povzamemo, izboljšuj sodelovalno, pomeni, kako se je tim zmožen prilagajati na spremembe. Razvijaj eksperimentalno, po drugi strani pa pomeni, kako izvajati spremembe.

Kanban metoda temelji na konceptu procesa reševanja problemov. Ta postopek se uporablja večkrat v fazi izboljševanja. Kanban uporablja Demingov cikel znan tudi kot PDCA (angl. Plan, Do, Check, Act). Eksperiment se načrtuje (določijo se hipoteze), se izvede, preverjajo se rezultati eksperimenta ter ravna se po rezultatih tako, da se ustrezno spremenijo hipoteze

in sistem ter se končne ugotovitve delijo z drugimi (Meiling, 2019). Nikolov (2020) ugotavlja, da je za izvajanje sprememb v skladu z znanstvenimi postopki, potrebna visoka stopnja discipline. Zato je treba dati konceptu dela, usmerjenega v rezultate organizacije, nov pomen.

Vse zgoraj naštetе prakse in koraki so namenjeni izboljšanju splošne uspešnosti procesov in organizacije. Bistvo je v tem, kako izvajati spremembe in kako upravljati posamezno spremembo. Sprememba procesa ni omejena samo na sam proces, ampak vpliva na vedenje celotne organizacije in ključnega pomena je, kako se soočimo z njim. Tu imajo pomembno vlogo prakse, kot so vizualizacija, povratne informacije in nenehno izboljševanje. Torej kaj se ljudje v organizaciji naučijo iz teh osnovnih praks Kanban-a, je, kako njihovo delo in sistem delujeta, kar omogoča proaktivnost pri managementu sprememb.

3.5.2 Problemi z uporabo metode

Omejitev dela lahko ležijo globoko v temeljih metode, vendar je še nekaj, kar naredi to metodo posebno. Po mnenju Boiser (2020) je verjetno najpogostejša napaka, ki je vidna pri timih, da se ne posodablja tabla. To pomeni, da tabla ne odraža realnosti in zato tim sprejema vsakodnevne projektne odločitve na podlagi zavajajočih podatkov. Zelo preprost primer: glede na to, da se pri testiranju pojavi ozko grlo, ki pa ni prikazano na tabli, bi razvijalec pri pregledu table videl, kaj sledi in začel graditi novo funkcijo, namesto da bi pomagal pri odpravi ozkega grla. Namesto da bi stvari izboljšali, bi jih še poslabšali, zahvaljujoč neposodobljeni tabli. Boiser (2020) meni da se z isto težavo vendar na drugi ravni, soočimo, kadarkoli tim naredi tablo, ki prikazuje, kako bi radi delali, in ne kako v resnici delajo. Pravzaprav je to bolj nevarna napaka. Ne samo, da vsakodnevne odločitve sprejemamo na podlagi zavajajočih podatkov, ampak je tudi težje stvari spraviti na pravo pot. V takšnem primeru ni dovolj posodobitev samolepilnih listkov, ampak je treba popraviti tudi obliko table. Ko se ljudje naučijo teh praktičnih trikov, ki jih lahko uporabljajo pri svojih tablah, so nagnjeni k prekomerni uporabi. Pogosto prevečkrat na koncu preoblikujejo tablo, kar pomeni, da pomembne informacije zakopljejo pod kupom nesmiselnih podatkov. Kmalu se zgodi, da ljudje ne razumejo pomena listka in kateri vizualno predstavlja katero situacijo. Sčasoma nehajo skrbeti za posodobitev table, ker so izgubljeni in se vrnejo na prvotno stanje table (Boiser, 2020).

Po mnenju Naydenova (2019) je kršitev omejitev ena od težav, ki se pogosto pojavlja, seveda je to stvar politike organizacije, ali sploh dovoli zlorabo omejitev. Vendar je precej pogosta situacija, da je splošno sprejemljivo, da se omejitve kršijo zelo pogosto, ali celoten čas izvajanja. Omejitve so v Kanban-u gonilna sila za izboljšave. Če se mej ne obravnava resno, se tudi izboljšav ne bo vzelo resno. Omejevanje nedokončanega dela je mehanizem, zaradi katerega ljudje delujejo drugače, kot bi običajno. Z ignoriranjem omejitev se zamudijo priložnosti za detekcijo ozkih grl v delovnem procesu. Sčasoma se naučijo predvideti takšne situacije in prej ukrepati, preden zapolnijo čakalno vrsto za delo. Vsekakor se vse začne s

preprostim dejanji, ki jih sprožijo omejitve dela v teku. Zanimivo je, da je vse te težave mogoče opaziti na Kanban tabli. Razlog je precej preprost, saj bi tabla morala odražati resničnost, ne glede na to, kako žalostna je realnost. S slabim procesom se spopadate enako kot s katerokoli problematiko. Prvi korak je priznati, da obstajajo težave. Tudi če je proces zelo slab, ga je treba prikazati na tabli. V nasprotnem primeru je to zatiskanje oči pred realnim stanjem in se pot z nenehnim razvojem proti popolnosti ne more začeti (Naydenov, 2019).

Obstaja še en dejavnik neuspeha Kanban metode, ki pa ni viden na tabli. Obravnavanje Kanban-a kot pristop managementa projektov ali razvoja programske opreme predstavlja neuspešno uporabo agilne metode. Od Kanban metode se zahteva, da se ukvarja z nečim, za kar ni bil zasnovan. Lahko ga imenujemo kot pristop managementa sprememb, orodje za management procesov ali celo sredstvo za izboljšanje, vendar ne govori niti besede o tem, kako bi se moral potekati management projekta ali graditi programsko opremo. Kanban mora temeljiti na naboru najboljših praks in metod managementa projektov, da bo projekt uspešen. Prav zmožnost managementa sprememb in managementa procesov je glavna značilnost Kanban-a, ki omogoča integracijo v druge agilne metode.

4 PREDSTAVITEV PODJETJA RENTAVAN

Podjetje je malo družinsko podjetje, ki je bilo ustanovljeno z namenom oddaje tovornih vozil in je eno vodilnih podjetij v ljubljanski regiji. Je v manjši stanovanjski zgradbi, kjer je parkirišče voznega parka oddaljeno 100 metrov od sedeža podjetja. Zaradi večjega povpraševanja je podjetje razširilo svojo ponudbo na večje in manjše velikosti tovornih vozil. V zadnjih letih so dokupili tudi potniška vozila za najeme in prevoze oseb, za kar so morali zaposliti voznika.

Na trgu oddaje vozil je konkurenca močna, saj se vsako leto pojavijo nova podjetja, ki ponujajo oddajo po ugodnih cenah. Večina močnejših konkurentov z večletno tradicijo je vlagalo v večje število vozil, kar je predstavljalo težavo pri velikosti parkirišč, saj so to pogosto družinska podjetja in so locirana na lokacijah lastnikov podjetij. Tako je bil v zadnjih letih trend selitev ponudnikov najemov vozil na ugodnejše lokacije, ki omogočajo večje število vozil v voznem parku (Bračič, 2018). S tem niso omejeni pri modernizaciji poslovnega procesa, saj nimajo prostorske stiske, ki jo zahteva moderna tehnologija v taki panogi. Tako so se v zadnjih letih močno povečale investicije v modernizacijo poslovnega procesa. To zajema izboljšanje uporabniške izkušnje z uporabo mobilnih aplikacij in poenostavitvijo procesa najema; možnost digitalnega plačila kar omogoča podaljšanje najema vozila med samim najemom in takojšnjim plačilom rezervacije brez prihoda stranke na lokacijo podjetja; zmanjšanje dokumentacije in birokracije, kar pohitri najemni proces in zmanjša prostorsko stisko znotraj poslovnih prostorov podjetja z uporabo oblčnih storitev; povečanje varnosti z uporabo globalnih sistemov pozicioniranja (angl. Global Positioning Systems, v nadaljevanju GPS), s katerimi podjetje spremlja lokacijo vozil; evidentiranje in

elektronsko beleženje poškodb in popravil vozil, s čim je zagotovljena transparentnost stanja vozil in zmanjša možnost konflikta med najemnikom in najemodajalcem.

Digitalizacijo vzdrževanja voznega parka ponuja tehnologija telematike, ki omogoča redna popravila in nakup rezervnih delov, s čim podjetje zmanjšuje in optimizira stroške in tako povečuje donosnost vozil. Glavni cilj modernizacije poslovnega procesa je implementacija moderne tehnologije, ki ponuja uspešno obvladovanje stroškov in ponuja digitalne rešitve, ki so prijazne strankam in uporabnikom (Adarsh, 2020).

Podjetje Rentavan je omejeno pri modernizaciji, saj je locirano v četrti skupnosti v Ljubljani, kjer se nahajajo manjše stanovanjske zgradbe. Oddaljenost najetega parkirišča predstavlja oviro pri nakupu naknadnih vozil in montaži sodobnih tehnologij, ki se uporabljajo pri avtomatizaciji poslovnega procesa. Običajno podjetja investirajo v razne sisteme video nadzora, ki nadzoruje prihode in odhode vozil, domače avtopralnice itd., kar pa pri podjetju Rentavan ni mogoče. Tako se je vodstvo podjetja odločilo, da bo investiralo v diverzifikacijo podjetja na druge panoge in na posledično modernizacijo informacijskega sistema podjetja Rentavan. V procesu diverzifikacije je podjetje vstopilo na trg oddaje plovil, koles in turistične dejavnosti. Zato se je v naslednjem koraku podjetje preoblikovalo v matično podjetje, ki deluje kot krovna organizacija in vključuje več franšiz. Osrednjo poslovno dejavnost podjetja oziroma franšizo predstavlja hčerinsko podjetje Rentavan z oddajo tovornih in potniških vozil. Preostali delež poslovanja predstavljajo podjetje Rentaboat, ki ponuja storitev posrednika pri najemu čarterskih plovil po celem svetu, Rentabike, ki ponuja najem vrhunskih gorskih in cestnih koles ter turistična agencija, ki ponuja organizacijo poslovnih ekipnih dogodkov na različnih lokacijah po Sloveniji.

Glavna in najbolj dobičkonosna panoga podjetja je najem vozil, ki pa podjetju predstavlja največji logistični in finančni zalogaj, saj pri opravljanju dejavnosti nastajajo najvišji stroški. Podjetje se sooča z vzdrževanje in vodenjem voznega parka, mesečnim plačilom najemnine parkirnega prostora, stroški hranjenja letnih ali zimskih platišč, s shrambo AdBlue tekočin za novejša vozila, itd.

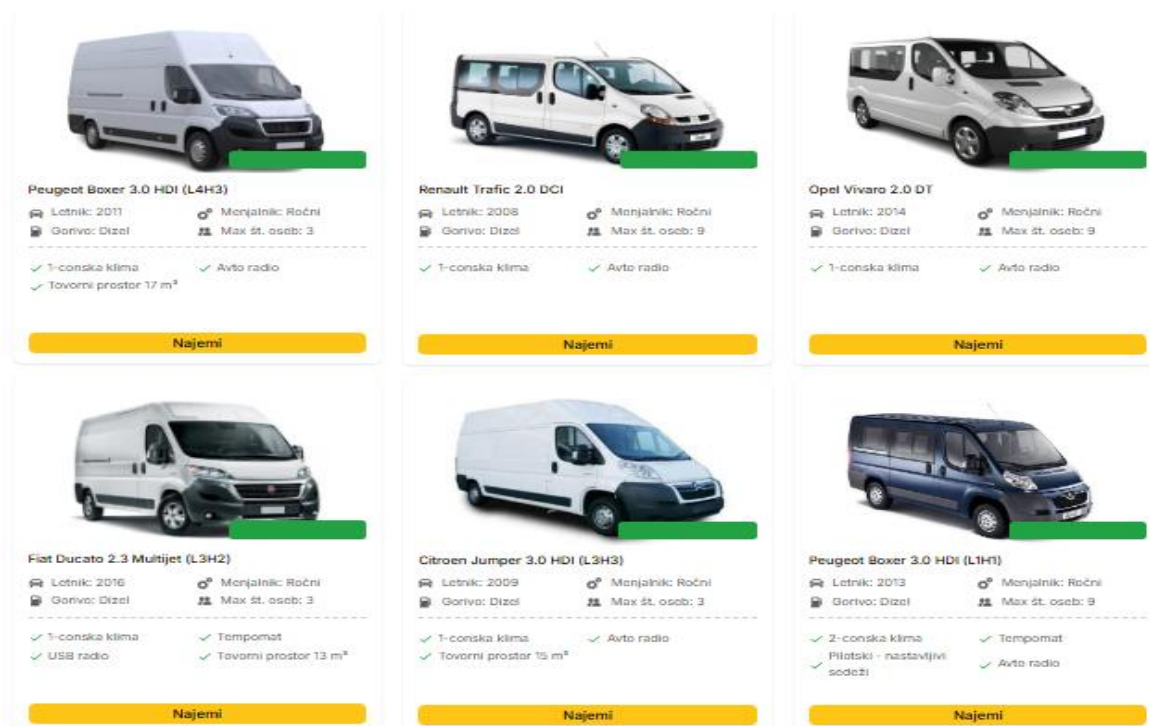
Zasičenost trga najema tovornih in potniških vozil se je v zadnjem desetletju močno povečala. V radiju dveh kilometrov od sedeža podjetja sta locirani dve podjetji, ki sta direktna konkurenta. Konkurenti imajo na razpolago tudi po trikrat več vozil kot Rentavan, zato se želi podjetje na trgu diferencirati z boljšo, hitrejšo in modernejšo ponudbo. Vozila so nadstandardna, servisirana in čista. Čeprav je v voznem parku manj vozil, podjetje poskuša vsaki stranki oddati tovarno ali potniško vozilo hitro in učinkovito, zato potrebuje učinkovit sistem sledenja in vodenja stanja voznega parka. Pri vsem tem pa poskuša ohraniti nizko ceno najema, čeprav se stroški rezervnih delov in ostali stroški poslovanja višajo. Zato se je matično podjetje pred leti odločilo diverzificirati in vstopiti na trg oddaje plovil. Tržišče najemov plovil je osredotočen predvsem na Jadransko morje, vendar so v ponudbi tudi ostale turistične lokacije, ki ponujajo idealne pogoje za jadranje oziroma vožnjo z vodnimi plovili

po celem svetu. Zaradi močne konkurence na trgu najema plovil je podjetje v preteklosti že poskusilo z digitalizacijo in modernizacijo poslovanja. S postavitvijo spletnega portala, ki prikazuje plovila, ki so na voljo, na želeni lokaciji, je podjetje želelo poenostaviti in olajšati proces najema plovila, tako za stranke kot tudi za zaposlene. Pri realizaciji projekta je razvojni tim, s pomočjo metode Kanban, razvila portal znotraj podjetja (angl. Inhouse). Zaradi slabe izkušnje, ki je privedla do zamud pri dokončanju projekta, se je podjetje odločilo najeti zunanega izvajalca za načrtovanje, management in realizacijo izboljšanja poslovnega procesa podjetja Rentavan.

4.1 Trenutno stanje (AS-IS)

Proces se začne s kontaktom stranke (preko elektronske pošte, telefonskim klicem, osebnim prihodom). Zaposleni preveri, ali je stranka v preteklosti že najemala vozila ter kakšne najeme so opravljali. V primeru, da gre za novo stranko, mora stranka priložiti osebni dokument, veljavno vozniško dovoljenje ter davčno število, če gre za pravno osebo. Zaposleni se dogovarja s stranko glede najema vozila, kakšen tovor bo prevažal, na podlagi tega se predlaga, kakšen tip vozila potrebuje (L3H2-potniški-tovorni, L3H3-tovorni-standardni, L4H3-tovorni-podaljšana različica, prikazano na sliki 2), dolžino relacije prevoza ter termin najema. Termin najema celotnega voznega parka je zaposlenemu viden v obliki table na oglasni deski v drugem prostoru, tako da mora za vsako vozilo ročno preverjati, ali je na voljo za določen termin. Na podlagi tega se poda ponudba.

Slika 2: Primer ponudbe na spletišču



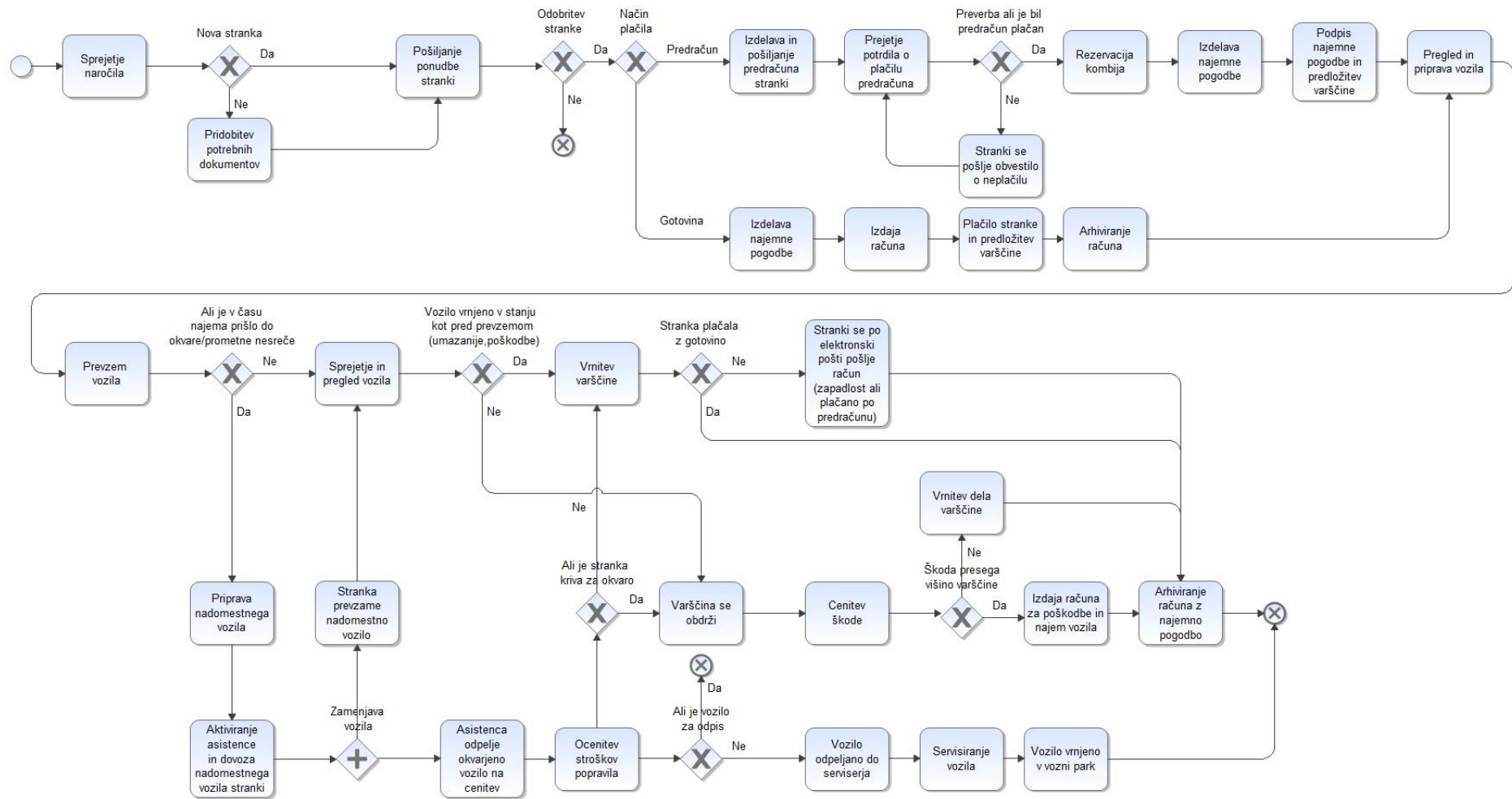
Vir: KombiCenter (2023).

Ponudba je ustvarjena kot tekstovna predloga v elektronski obliki, ki vsebuje začetek in konec najema, relacijo in dolžino kilometrov, ceno najema z in brez davka, višino varščine ter povezavo do spletnega mesta predlaganega vozila. Ker so ponudbe napisane ročno, se vsakodnevno dogajajo napake pri vnosu. Če zaposleni hoče oddati ponudbo brez napak mora čez določene podatke večkrat, kar postopek oddaje ponudbe zavleče. V primeru odobritve ponudbe se strani dogovorita glede plačila najema vozila, ali bo to preko predračuna (elektronsko) ali z gotovino. Dokler predračun ni plačan, stranka nima zagotovljene rezervacije za najem vozila in je lahko vozilo na voljo drugim strankam. Zgodilo se je, da je druga stranka prevzela dogovorjeno vozilo, in ko je poklicala stranka, ki je nameravala vozilo najeti, ostala brez tega in je zaposleni predlagal drugo vozilo istega tipa ali ne, oziroma je poklical konkurenčne ponudnike in pri njih najel isto vozilo.

Ko je vozilo najeto, zaposleni zapolni tabelo terminov za določeno vozilo in tako je vozilo smatrano kot rezervirano. Preden pride stranka prevzet vozilo, zaposleni osebno preveri stanje vozila. Trenutno stanje kilometrov, čistočo vozila, dodatno opremo, brezhibnost vozila. V prisotnosti stranke se vozilo še fotografira iz različnih zornih kotov. S tem se preprečijo neljubi dogodki, kot na primer »Te praske ni bilo tam, ko sem prevzel vozilo«. Potem gre vozilo v najem. Vedno obstaja možnost, da bo prišlo do okvare vozila ali da bo stranka udeležena v prometni nesreči. V tem primeru se obvesti podjetje za asistenco na cestah in se pripravi nadomestno vozilo. Asistenca prevzame okvarjeno vozilo, stranka pa vzame nadomestnega. Okvarjeno vozilo je odpeljeno na uradni servis, kjer se opravi cenitev škode in ocenitev stroškov popravila. Če je stranka kriva za okvaro, oziroma povzroči prometno nesrečo, se varščina obdrži in se izda še račun za popravilo vozila. Če je okvarjeno vozilo smiselno popraviti, se opravi servisiranje v primeru pa, da je škoda prevelika, gre vozilo v odpis. Po popravilu je vozilo vrnjeno v vozni park.

Ko poteka najem brez komplikacij, stranka vrne vozilo. Zaposleni preveri, ali je vozilo v istem stanju, kot je bilo pred najemom (poškodbe, umazanija). Ko je vozilo pripeljana v istem stanju, kot je bilo odpeljana, se stranki vrne varščina in pošlje račun preko elektronske pošte (če ni bilo plačila z gotovino), račun gre nato skupaj z najemno pogodbo v arhiv. Ko je pa vozilo vrnjeno v drugačnem stanju, kot je bilo oddano, se varščina obdrži in opravi cenitev škode v primeru, da varščina ne zadostuje za popravilo škode. Stranki se izda še dodatni račun za popravilo najetega vozila. Račun popravila, kot tudi najemo pogodbo, zaposleni arhivira. Slikovni prikaz trenutnega stanja poteka najema vozila se nahaja v sliki 3.

Slika 3: Trenutno stanje v podjetju (AS-IS)



Vir: lastno delo.

4.2 Organizacijska struktura podjetja

Podjetje Rentavan je manjše slovensko podjetje, ki zaposluje šest do osem ljudi. Direktor podjetja, ki je tudi lastnik in ustanovitelj Rentavana, vodi in odloča o strateških odločitvah podjetja. Po potrebi se vključuje v vse panoge, od najema vozil do najema plovil. Desna roka direktorja je procesni manager, ki skrbi za vse procese krovne organizacije. Zaradi velike obremenitve je glavni pobudnik za uvedbo informacijskega sistema.

Vodja voznega parka skrbi za zaupanje in odnos s strankami. Njegova naloga je komunicirati s strankami preko elektronske pošte, telefona ali osebnega obiska na lokaciji podjetja in dogovarjanje in izpolnjevanje strankinih zahtev.

Upravitelj voznega parka poskrbi za oddajo in prevzem vozil. Pred najemom stranko obvesti o potrebnih informacijah, ki se tičejo samega vozila in številne napotke v primeru okvare ali nezgode. Pri tem preveri stanje (npr. obstoječe poškodbe, dodatna oprema, število prevoženih kilometrov) in po oddaji, o vozilu poroča vodji voznega parka ter poskrbi za vnos ali posodobitev podatkov na pogodbi, ter da pogodbo v čakanje do konca najema. Zadolžen je za vsa vzdrževalna dela, ki so potrebna za ustrezno delovanje voznega parka in koles. Zaradi povečanja povpraševanja po tovornih prevozih je podjetje zaposlilo voznika, ki sprejema in dostavlja blago na lokacije strank.

Vodja marketinga vodi oglaševalno strategijo, ki zajema oglaševanje preko radia in časopisa ter spletno oglaševanje.

4.3 Predlog izboljšanja procesa (TO-BE)

Proces se začne s fizičnim obiskom stranke na sedežu podjetja ali obiskom stranke spletne strani, ki omogoča najem vozila. Če gre za novo stranko, mora stranka na spletni strani izpolniti vse osebne podatke, ki so potrebne za oddajo poleg tega mora priložiti slikano, skenirano vozniško dovoljenje ter davčno številko v primeru, da gre za javno osebo. Ko pride nova stranka osebno v podjetje, zaposleni vnese vse potrebne podatke v sistem namesto nje.

Stranka določi, kakšen tip vozila potrebuje za najem, kdaj ga bo najela ter kakšna je dolžino relacije. Če pride nova stranka osebno v podjetje, zaposleni vnese vse potrebne podatke v sistem namesto nje. Kolikor se stalna stranka prijavi v spletno stran, se stranki prikažejo pretekli najemi tako, da lahko najame isti kombi pod istimi pogoji. Zaposlenemu ni več potrebno pošiljanje ponudbe oziroma predračuna saj se stranki ponudba kreira avtomatsko.

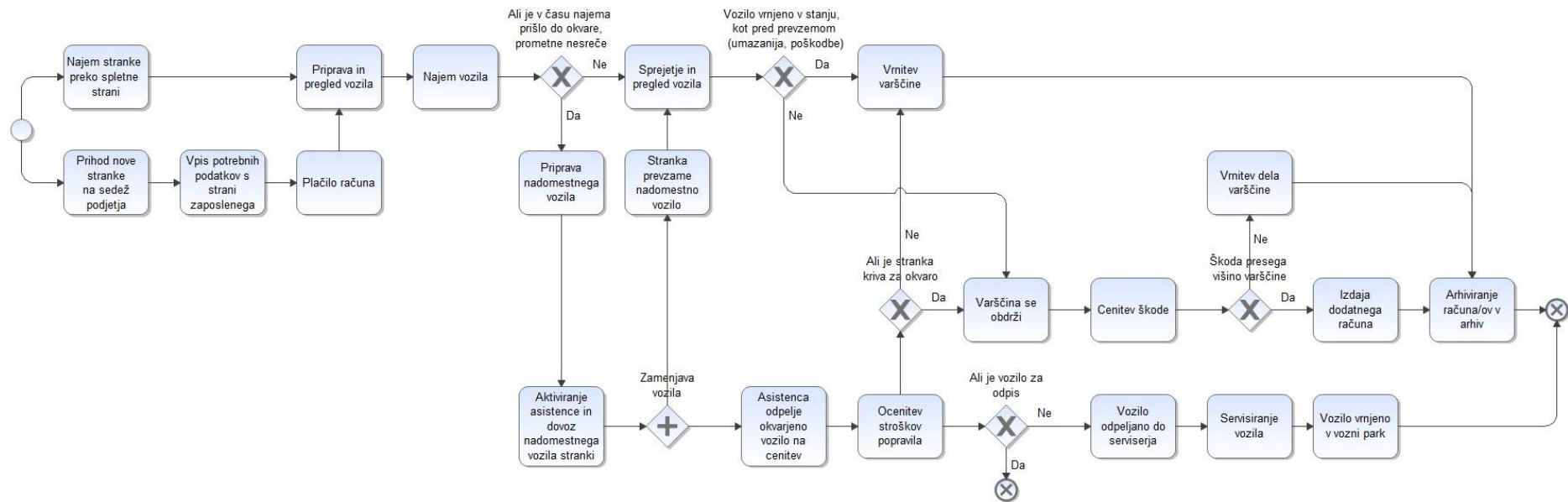
Ponudba vsebuje začetek in konec najema, relacijo in dolžino kilometrov, ceno najema z in brez davka, višino varščine, višino popusta ter specifikacije najetega vozila. Stranka lahko samo s plačilom preko spletne strani (v primeru osebnega prihoda na sedežu podjetja)

rezervira kombi, tako se podjetje izogne nepotrebnemu čakanju in omogoči najem istega vozila drugi stranki. Preden stranka pride prevzet vozilo, zaposleni osebno preveri stanje vozila. Trenutna kilometrina je vidna v sistemu po zaslugi OBD računalnika, ki je povezan z računalnikom vozila, tako da zaposlenemu ni tega potrebno preverjati. Stanje zunanosti je zaposlenemu vidno preko obrazca, ki prikazuje vse pretekle poškodbe, praske in ta obrazec se pokaže tudi stranki, da v primeru poškodbe za čas najema stranka ve, da je za poškodbo kriva sama in ne poskuša tega prikriti podjetju. Dodatna oprema vozila je prav tako vidna v sistemu, tako da zaposlenemu ni treba vedeti, kakšno opremo ima katero vozilo in katero opremo je potrebno vsakoletno pregledovati (npr. gasilni aparat).

Vozilo gre v najem. Kolikor v času najema pride do okvare ali prometne nesreče, se za stranko organizira novo vozilo. Zaposleni najeto vozilo odstrani iz možnosti najema in organizira vlečno službo. Stranki ni potrebno zaposlenemu povedati trenutno lokacijo vozila, saj jo ta lahko po zaslugi GPS-a, vidi v sistemu. Stranka prevzame nadomestno vozilo, vlečna služba pa okvarjeno. Okvarjeno vozilo je odpeljano na servis in ga na spletni strani ter sistemu ni možno najeti, dokler ne pride s popravila. Kolikor je stranka sama kriva za okvaro, prometno nesrečo, se stranki izda nov račun ter obdrži varščina. V primeru, da je strošek popravila presega vrednost vozila, gre vozilo v odpis. Stranka po izteku najema vrne vozilo v vozni park.

Zaposleni preveri stanje vozila, če je bilo vrnjeno v istem stanju, kot je bilo prevzeto. Število kilometrov je zaposlenemu vidno v sistemu, ki mu je dostopen preko tabličnega računalnika, mobilnega telefona. Če je stranka presegla dogovorjeno število kilometrov, se mu izda dodaten račun za preseženo kilometrino. V primeru, da so vidne nove poškodbe, ki jih pred izdajo ni bilo, se stranki vzame varščina in zaposleni doda poškodbo v obrazec, ki prikazuje stanje vozila. Ko strošek poškodbe presega višino varščine, se stranki izda nov račun. Dodatni račun se izda tudi v primeru, če manjka del dodatne opreme ali če je vozilo treba očistiti oziroma odpeljati v čistilni servis, in varščina tega stroška v celoti ne pokrije. Ko pa stranka vrne vozilo v istem stanju, kot je bilo pred prevzemom in je prevozila dogovorjeno število kilometrov, se stranki vrne višina varščine na njen bančni račun. Za konec proces zaposleni arhivira fizično oblika računa v arhiv. Slikovni prikaz novega stanja poteka najema vozila se nahaja v sliki 4.

Slika 4: Izboljšanje procesa (TO-BE)



Vir: lastno delo.

5 UPORABA PMBOK NA PODJETJU RENTAVAN

5.1 Začetek projekta

5.1.1 Razumevanje projekta

Podjetje Rentavan se je odločilo implementirati informacijski sistem za najem in kontrolo svojih vozil. Zaradi potrebne 24-urne dostopnosti podatkov so se odločili za rešitev, ki med seboj integrira podatke strank in vozil v proces najema, ki deluje v oblaku. Kot za primerno in stroškovno ugodno rešitev predlagamo že obstoječ sistem, ki omogoča integracijo že obstoječih programov znotraj podjetja. Prednost novega sistema je možnost najema in plačila preko spleta. Podjetje bi povečalo svoj tržni doseg in izboljšalo izkušnjo strank z avtomatizacijo postopka najema preko spletne ali mobilne aplikacije. Sistem ponuja preprosto rešitev za vodenje voznega parka. Podjetje se izogne uporabi nepreglednih in neurejenih preglednic, ki so v digitalni in papirni obliki in zaposlovanju dodatnega kadra. Sistem ponuja enostavna orodja za management razpoložljivosti, popravil in financ voznega parka. Digitalizacija je eno glavnih vodil projekta. Podjetje stremi k brezpapirnemu poslovanju. Informacijski sistem bo omogočal podpis pogodbe najema v digitalni obliki in nalaganje dodatnega slikovnega gradiva v primeru novonastalih poškodb vozila. Pregled zgodovine najemov in strank ter vpogled v trenutno in preteklo stanje vozil bo dostopno preko spleta brez uporabe papirja. To omogoča tudi avtomatizacijo plačila dolgoročnih najemov vozil. Zaradi oblačnega delovanja in shranjevanja podatkov sistem ponuja fleksibilno integracijo v podjetja različnih velikosti. Uporabniku prijazen program je lahko prilagojen različnim zahtevam podjetja, kar omogoča možnost širitve uporabe informacijskega sistema na druge panoge podjetja, kot so najem koles in jadrnic (Gorenšek, 2019).

5.1.2 Poslovni primer

5.1.2.1 Definiranje ciljev projekta

Cilj je razvoj učinkovitega informacijskega sistema, ki omogoča učinkovitejši management vseh odnosov in interakcij podjetja s strankami in potencialnimi strankami ter s tem izboljša poslovne odnose za nadaljnjo rast podjetja. To zajema nadgradnjo povezanosti s strankami, racionalizacijo procesov in izboljšavo dobičkonosnosti. Večja osredotočenost je na odnosih znotraj organizacije med posameznimi ljudmi, vključno s strankami, uporabniki sistema, serviserji ter posledično uspešnejše zagotavljanje podpore in dodatnih storitev v celotnem razmerju. Namen je tudi povečanje produktivnosti zaposlenih, omogočanje avtomatizacije ponavljajočih se del ter poenostavitev komunikacije med zaposlenimi. Fokus projekta je tudi večja vključenost stranke v sam proces najema, zagotavljanje popolnega vpogleda v stanje vozila v realnem času in samodejnega uvoza takih podatkov v sistem. S tem se zagotovi boljši vpogled v strošek vozil, na primer strošek na kilometer, strošek vzdrževanja, kakšni

so operacijski stroški za lažje določanje minimalne višine najema vozila. Cilj je tudi modernizacija poslovanja z dodatnimi funkcijami snemanja videoposnetkov, kontroliranja porabe goriva in zaznavanja trenutne lokacije vozila. Z delujočim sistemom je treba zagotoviti povečanje življenjske dobe vozila z bolj učinkovitim spremljanjem vozila, kot je trenutna kilometrina, nivo olja, obraba gum itd. S tem se poveča varnost najema z vidika plačila najema kot tudi z vidika uporabe najetega vozila. Če povzamemo, je cilj doseči boljše dostopnost, celovitost in ažurnost podatkov poslovanja z vzpostavitvijo novega informacijskega sistema.

5.1.2.2 Definiranje stroškov oziroma proračuna projekta

Podjetje Rentavan je v zadnjem desetletju vsako leto povečevalo svoj dobiček. Vendar je bilo podjetje soočeno s pandemijo Covid-19, ki je imela izjemno negativen vpliv, in je praktično ustavilo poslovanje podjetja. Zato je vodstvo prisiljeno razmišljati o investicijah prenove poslovnega procesa in tako zmanjšati vplive prihodnjih nepredvidljivih tveganj. Čeprav ni pričakovati večjih nihanj dobička v prihajajočih letih, povpraševanje po tovornih in potniških vozilih konstantno raste. Zaradi konkurenčnosti na trgu oddajanja tovornih in potniških vozil podjetje vsako leto kupi najnovejše modele kombijev, s katerimi želi razširiti svojo ponudbo in vzdrževati visoke standarde voznega parka. Zaradi modernizacije poslovanja, se je podjetje Rentavan odločilo, da se v letu 2023 ne odloči za nakup vozil, ampak investira rezervirana sredstva v nov informacijski sistem.

V tabeli 1 je prikazan dobiček zadnjih 6 let, v tabeli 2 pa načrtovan nakup vozil za leto 2023. Ker se je povpraševanje v letu 2022 močno povečalo, je pričakovati, da se bo trend uspešnosti nadaljeval, kot je bilo to pred Covid-19 pandemijo. Preneseni dobiček in zamrznitev nakupa novih vozil v naslednjem letu – cca 260.000,00 €. Stroški projekta so brez višanja 20 %: 128.989,00 € in stroški z zalogo (angl. buffer) 20 %: 154.787,00 €.

Tabela 1: Poslovni izidi 2017-2022

Poslovni izid	Dobiček	Preneseni dobiček
Dobiček leta 2017	25.776,59 €	40.317,46 €
Dobiček leta 2018	25.890,83 €	66.094,05 €
Dobiček leta 2019	1.442,38 €	91.984,88 €
Dobiček leta 2020	0 €	91.984,88 €
Dobiček leta 2021	6.622,29 €	29.151,84 €
Dobiček leta 2022	30.000,00 €	35.774,13 €

Vir: lastno delo.

Skupaj: 35.774,13 € + 30.000 € = 65.774,13 €

Tabela 2: Načrtovani nakup vozil za leto 2023

Tip vozila	Brez DDV	Z DDV
1x Renault Master Furgon RWD L4H3P3 Energy dCi 165 VI	34.190,00 €	41.712,00 €
4x Renault Master Furgon L3H2P3 Energy dCi 180 PRO+	31.190,00 €	38.052,00 €

Vir: lastno delo.

Skupaj: 41.712 € + 4 x 38.052 € = 193.920 €

Razporeditev sredstev za projekt v letu 2023: 193.920 € + 65.773,13 € = 259.693,10 €

Izvedljivost projekta: 259.693,1 € – 154.787,00 € = 104.906,10 €

5.1.3 Projektna listina

5.1.3.1 Opis projekta

Izboljšanje in digitaliziranje poslovnega procesa najema vozila ter vzdrževanje in nadzor nad voznim parkom z implementiranjem informacijskega sistema v manjše slovensko podjetje:

- nastaviti strojno opremo,
- nastaviti programsko opremo,
- integracija obstoječih sistemov (programov) v nov sistem,
- izobraževanje zaposlenih

5.1.3.2 Namen projekta

Podjetje Rentavan se hitro širi na trgu najema tovornih in potniških vozil. Zaradi pričakovanega povečanja števila vozil v voznem parku podjetje potrebuje učinkovit sistem vodenja in kontroliranja strank, zaposlenih in vozil. Uspešna implementacija sistema lahko privede do širitve uporabe na druge panoge podjetja.

Začetek projekta: 10. 10. 2022

Konec projekta: 31. 3. 2023

5.1.3.3 Mejniki

1. Konfiguracije (59 dni)
2. Konfiguracija vmesnika med vsemi paketi vključno s testiranjem skupnega dela aplikacije (24 dni)
3. Konfiguracija zalednega sistema (59 dni)

4. Uvajanje in testiranje celotnega sistema (22 dni)

5.1.3.4 Ključni deležniki

Organizacijska struktura projekta zajema vse ključne pozicije za doseganje projektnih ciljev, ki so jasno definirane. Iz slike je razvidno, da je naročnik projekta podjetje Rentavan. Projektni skrbnik je oseba, ki usmerja in zagotavlja financiranje projekta ter je idejni in gonilni motor projekta (Grigoryan, 2022).

Idejo za implementiranje informacijskega sistema v Rentavan podjetju, je direktorju podjetja predlagal procesni manager, z željo po optimizaciji procesa najema in bolj pregledno in pravočasno vodenje voznega parka. Zaradi poznavanja organizacije in njenih procesov, identificiranja problemov in težav trenutnega stanja podjetja in kakšne so potrebne rešitve, procesni manager zagotovi in določi vire projekta, izvajalca rešitve, projektnega managerja ter potrdi projektna listina. V sodelovanju s projektnim lastnikom projektni manager identificira problem predlagane rešitve in določi, kako se ga lotiti. Pri tem določi cilje in obseg projekta ter katere dejavnosti so potrebne za uspešno dokončanje projekta. Razvijalci skrbijo za izgled aplikacije (angl. front end), kako aplikacija deluje (angl. back end), vse aspekte procesa najema in procesa vodenja voznega parka. Izvajajo tudi začetno testiranje sistema. Pri začetni verziji pomagajo zunanjemu razvijalcu informacijskega sistema pri kreiranju podatkovne baze, vzpostavitvi grafičnega vmesnika, integraciji sistema za vodenje voznega parka.

Ko je osnovni sistem postavljen, je njihova naloga, da se sistem priredi uporabi podjetja. Podatkovna baza se oblikuje tako, da zajema tiste podatke, ki jih podjetje potrebuje (kupci, vozni park itd.). Integrirajo vse obstoječe programe oziroma rešitve, ki so potrebne za delovanje podjetja (npr. sistem za računovodstvo). Konfiguriranje grafičnega vmesnika zajema dve področji. Zaposleni imajo na razpolago vse podatke, ki jih potrebujejo za poslovanje. Na preprost način dostopajo do podatkov najema, strank, voznega parka, plačil. Drugo področje zajema grafični vmesnik za stranke. Stranke imajo dostop do sistema preko spletne in mobilne aplikacije. Vmesnik mora biti preprost za uporabo, s čim manj vnosi. Zunanji razvijalec sistema že obstoječo rešitev priredi in integrira v podjetje. Rešitev je v modularni obliki, kar pomeni, da ni treba vzeti celote, ampak se uporabijo moduli, ki ustrezajo zahtevam naročnika projekta. Zunanji razvijalec kreira podatkovno bazo na Amazon S3 strežnikih, katerih prostor je že vključen pri nakupu celotne rešitve. Nato se integrira sistem za vodenje voznega parka preko senzorjev, ki se vgradijo v vozila. Rešitev ponuja že narejen osnovni grafični vmesnik.

5.1.4 Obseg projekta

Cilj projekta je izdelava informacijskega sistema, ki bo deloval v oblaku in bo integriran z obstoječimi sistemi podjetja, ki so potrebni za poslovanje. S sistemov bo Rentavan podjetje

bolj učinkovito in konkurenčno na trgu najema vozil.

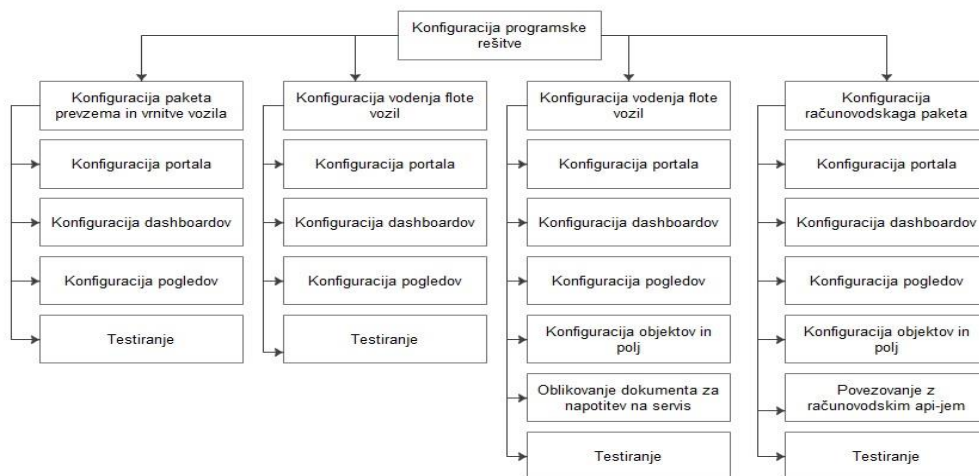
1. Konfiguracija aplikacije (81 dni)
2. Konfiguracija zalednega sistema, testiranje in zagon sistema (73 dni)

5.2 Načrtovanje projekta

5.2.1 Management obsega projekta

Struktura razčlenitve dela (angl. work breakdown structure, v nadaljevanju WBS) je hierarhična razčlenitev dela, usmerjenega v izdelek, ki ga mora projektni tim izvesti za doseganje ciljev projekta in ustvarjanja zahtevanih rezultatov. WBS je temelj učinkovitega načrtovanja, izvajanja, nadzora, spremljanja in poročanja o projektu. Vse delo, ki ga WBS vsebuje, je treba opredeliti, oceniti, načrtovati in predvideti. Zaradi boljše preglednosti smo WBS razdelili na dva dela, konfiguracija programske rešitve in ostale aktivnosti projekta. Slika 5 prikazuje delni WBS (razdeljeni WBS je v prilogah 1 in 2). Sklop konfiguracije programske rešitve je razdeljen na osem aktivnosti, ostale projektne aktivnosti so razdeljene na 7 aktivnosti. Vse aktivnosti so dodatno razdeljene na pod aktivnosti.

Slika 5: Delna Struktura razčlenitve dela (WBS)



Vir: lastno delo

5.2.1.1 Projektne aktivnosti

5.2.1.1.1 Konfiguracije aplikacije

Konfiguracija programa za management strank (angl. Customer Relationship Management, v nadaljevanju CRM): Konfiguriranje oziroma prilagoditev CRM vmesnika za potrebe zaposlenih. Vnašanje in oblikovanje potrebnih polj za vnos podatkov strank v sistem. Ustvarjanje robustnih orodij za analizo in poročanje, ki podpirajo prodajo, trženje in storitve

za stranke. Uporabniki imajo dostop do predhodnih poročil in poročil po meri, ko so na primer finančna poročila. Prilagajanje nadzornih plošč, glede na uporabnikove potrebe za zagotavljanje pogleda na ključne metrike in druge pomembne operativne podatke, v realnem času. Vključevanje funkcij, ki zaposlenim pomagajo učinkovitejše podpirati stranke. Sistem mora biti sposoben hitro pridobiti vse ustrezne podatke o kupcu vključeno s stanjem najema, računi, zgodovino najemov. Avtomatiziranje oziroma sinhronizacija določenih delov najema vozila ostalih podsistemov, kot so obvestila o iztekanju časa za najem vozila in prekoračitev kilometrine, preteklih najemov.

Konfiguracija rezervacijskega paketa: Nastavitev potrebnih podatkov, ki se bodo vnašali v sistem podjetja. To zajema podatke zaposlenih (pravice), strank (podatki, zgodovina, plačila), vozil (registracije, zavarovanja, servisi, status najema, dodatna oprema, poškodbe).

Konfiguracija paketa prevzema in vrnitve vozila: Nastavitev sistema za beleženje in pridobivanje podatkov zasedenosti vozil v realnem času.

Konfiguracija vodenja flote vozil: Namestitev strojne opreme v vozila. Nastavitev strojne opreme in sistema s tehnologijo, ki temelji na daljinskem prenosu podatkov in s pomočjo telematike vozila določene funkcije opravlja samostojno. To pomeni povezovanje GPS sledenja s sistemom, preverjanje podatkov s koordinatami vozila, podatki vozil v realnem času.

Konfiguracija vodenja popravila vozil: Nastavitev sistema komunikacije med sistemom podjetja in zunanjimi sodelavci podjetja (serviserji, zavarovalni agenti, podjetja za dodatno opremo itd.). Komunikacija poteka preko generiranih elektronskih sporočil, ki se pošiljajo zunanjim deležnikom (npr. rezervacija servisa ali izredno popravilo po poškodbi).

Konfiguracija računovodskega paketa: Integracija računovodskega programa (oz. API) v sistem podjetja. Uporabniki ali zaposleni preko sistema dostopajo do vseh potrebnih funkcionalnosti računovodskega programa (izdaja predračunov, računov, vnašanje računov podjetja itd.)

Konfiguracija plačilnega paketa: Integracija kartičnega plačila (na lokaciji, preko spleta). Namestitev terminala POS v podjetju.

Konfiguracija grafičnega vmesnika med vsemi paketi, testiranje aplikacije: Programiranje in urejanje grafičnega vmesnika sistema. Zagotavlja preglednost nad vsemi paketi informacijskega sistema. Uporabnik ima preko nadzorne plošče pregled nad najemi, stanjem vozil, strankami, financami itd. Stranka ima pregled nad vsemi preteklimi najemi in plačili. Strmi se h razvoju kvalitetnega, vendar enostavnega uporabniškega vmesnika, za lažje razumevanje in učenje uporabnikov. Testiranje integracije vseh paketov oziroma programov, ki so vključeni v sistem (računovodski program, plačilni sistem itd.).

Poizkušanje odkrivanja napak v interakciji med integriranimi programskimi moduli. Preverjanje učinkovitosti celotnega programskega sistema kot celote. Testiranje ali vse enote delujejo v skladu z opredeljenimi specifikacijami. Testiranje uporabniškega vmesnika, preizkušanje uporabniškega vmesnika aplikacije, da se zagotovi, da le ta deluje nemoteno in po pričakovanjih. Preverjanje vizualnih komponent, s katerimi uporabniki komunicirajo. Zaradi nadgradnje, spremembe aplikacije se izvede regresijsko testiranje, ker hočemo zagotoviti stabilnost programske rešitve po spremembah, ki smo jih naredili. Preverimo, da spremembe niso negativno vplivale na obstoječe funkcije aplikacije.

5.2.1.1.2 Konfiguracija zalednega sistema

Kreiranje, urejanje, optimiziranje podatkovne baze: Kreiranje podatkovne baze v oblaku in priprava baze za namestitev na lokaciji. Namestitev varnostnih varoval proti kraji in zlorabi podatkov, varnostnega kopiranja in obnovitev sistema. Določanje nivojev varnosti oziroma ustvarjanje uporabniških profilov z različnimi nivoji dovoljenja za gledanje, spreminjanje, brisanje podatkov. Ustvarjanje višjo stopenjske varnosti in integritete podatkov v sistemu. Omogočanje avtomatske posodobitve, ko bo opravljena rezervacija ali ko se bo registrirala nova stranka, bodo podatki v sistemu, posodobljeni samostojno.

Nabava strežnika in uporabnikove opreme: Nakup ustrezne strojne in programske opreme za delovanje sistema.

Namestitev strojne opreme: Namestitev strežnika na lokaciji in inštalacija podatkovne baze za varnostno kopiranje. Postavitev delovnih postaj, brezprekinitvenega napajanja UPS, tablic in usmerjevalnika. Namestitev operacijskih sistemov na delovne postaje. Prilagoditev tabličnih računalnikov za ustrezno uporabo. Ureditev povezljivosti vseh elektronskih naprav s podatkovno bazo v oblaku in na lokaciji.

Konfiguracija strežniške opreme z oblaknim strežnikom: Konfiguracija načina shranjevanja podatkov med oblaknim strežnikom in strežnikom na lokaciji. Nekateri podatki se bodo shranjevali izključno na lokaciji zaradi varstva osebnih podatkov.

Varnostno kopiranje starih datotek, dokumentov: Varnostno kopiranje podatkov na strežnik na lokaciji.

Prenos starih datotek na nov sistem: Vnos podatkov v sistem (npr. obstoječa baza strank). Urejanje kompatibilnosti podatkov, ki ne ustrezajo sistemu (dokumenti iz starih programov, ki ne delujejo več v novejših rešitvah).

5.2.1.1.3 Testiranje in zagon sistema

Obremenitveni test: Preizkus učinkovitosti sistema, pri delovni obremenitvi, ki presega

njegove zmožnosti normalnega delovanja, da se ugotovi, kako se obnaša v takih scenarijih. Pri tem se ugotavlja in prepozna ozka grla, ocenjuje odzivne čase sistema in odkriva morebitne pomanjkljivosti ali okvare, medtem ko je sistem pod stresom. Obremenitveni test velja za različne dele sistema; baze podatkov, strežnika, omrežja, aplikacije. Zagotovimo, da lahko sistem prenese pričakovano obremenitev ali promet v dejanskih razmerah, in preizkusimo njegovo sposobnost okrevanja po okvarah ali nepričakovanih dogodkih.

Funkcionalni test: Ustvarjanje nedokončane, vendar funkcionalne kopije programske opreme za manjše število uporabnikov, ki jo preizkusijo in podajo povratne informacije. Testiranje razvojne verzije sistema. Preverjanje in odpravo napak ali tako imenovanih hroščev (angl. bugs). Potrjevanje, da odpravljanje hroščev ni negativno vplivalo na funkcionalnost sistema. Preverjanje, da je programska različica stabilna za testiranje in da vse kritične funkcije delujejo, kot bi morale. Iskanje ukrepov za izboljšanje programske opreme v smislu učinkovitosti, točnosti in uporabnosti.

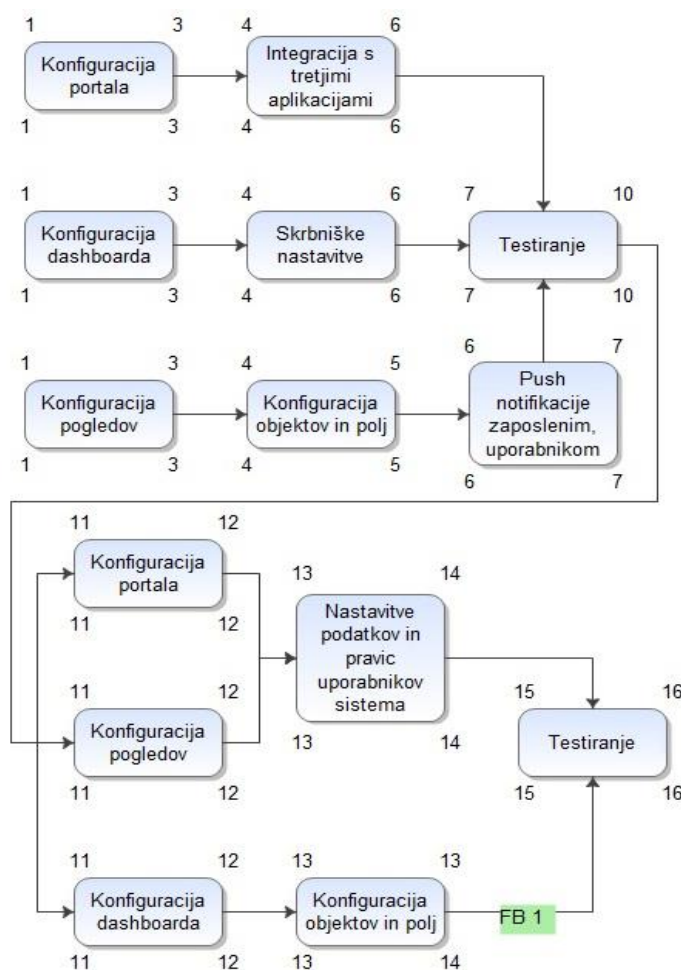
Celoten sistemski test: Čas delovanja ter odziva in omogočanje visoke uspešnosti, s čimer je sistem sposoben zagotoviti hitre povratne informacije, ko uporabnik vnaša podatke. Pri celotnem sistemskem testu se preverjata tudi odzivnost v primeru napak in spremljanje razpoložljivosti sistema v primeru večjih okvar. Preverja se tudi varnostni vidik, kar pomeni, da sistem ne dovoli nepooblaščenega dostopa do podatkov in virov.

Zagon novega sistema: Zagon sistema v poslovnem okolju.

5.2.1.2 Projektna zaloga

Na sliki 6 je prikazana projektna listina projekta (dve konfiguraciji projekta). Projektna zaloga (angl. project buffer) je 20 odstotkov dolžine celotnega projekta, zato se število dni projekta poveča iz 145 na 175 dni oziroma podaljša za 30 dni. Časovna zaloga (angl. feeding buffer) je dodeljena nekritičnim aktivnostim pred kritičnimi aktivnostmi. Kot je razvidno iz slike 3, je časovna zaloga dodana pri konfiguraciji objektov in polj.

Slika 6: Projektna listina z zalogo

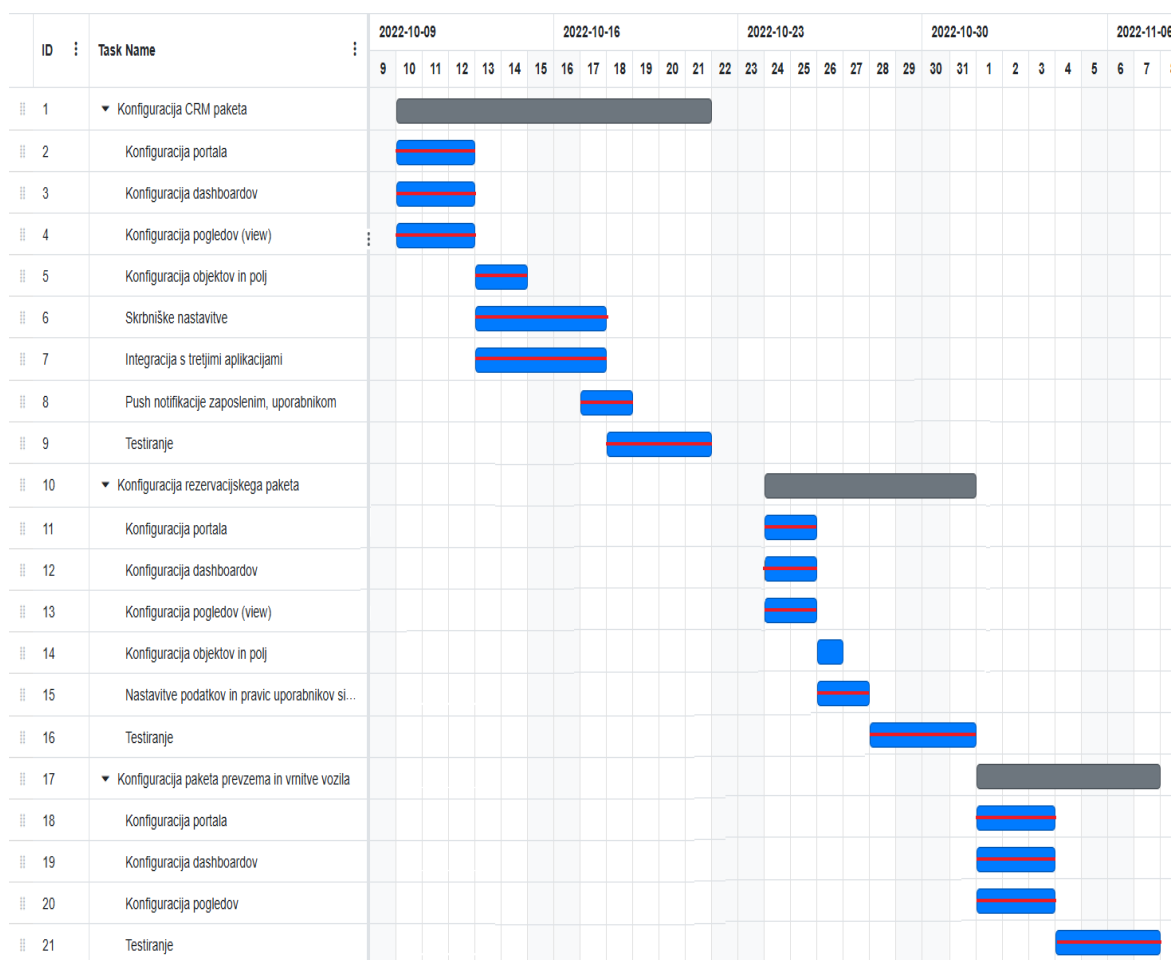


Vir: lastno delo.

5.2.2 Management časa

Čas je pri projektnem managementu bistvenega pomena za učinkovito razdelitev virov, da se zagotovi, da tim opravi visokokakovostno delo v dogovorjenem roku. To zajema razčlenjevanje in dodeljevanje nalog, zagotavljanje okvirja za razvoj zaporedja dejavnosti, trajanja dejavnosti, ocene virov in kako se ti prilegajo celotnemu načrtu managementa projekta (Nutchache, 2019). Delni potek projekta je prikazan na sliki 7. Gantogram je med bolj priljubljenimi diagrami, ki se uporabljajo za prikaz časovnega poteka aktivnosti. Predstavljene so kot vrstice, katerih položaj prikazuje začetni in končni datum ter trajanje aktivnosti. Diagram omogoča pregled opravljenih aktivnosti in ki se morajo še izvesti ter njihovo pomembnost. Pomembnost je prikazana kot kritična pot, ki prikazuje tiste aktivnosti, ki bi vplivale na končni datum projekta, če bi pri teh prišlo do zamud (Duke, 2010). Z rdečo barvo so obarvane vse aktivnosti, ki so na kritični poti, s temno modro barvo so označene aktivnosti, ki so bile končane.

Slika 7: Gantogram



Vir: lastno delo.

Tabela 3 prikazuje okvirno izvajanje projekta, brez dodane projektne zaloge.

Tabela 3: Okvirno izvajanje projekta

Opravila	Začetek	Konec
<i>Konfiguracije</i>	10. 10. 2022	30. 12. 2022
Konfiguracija CRM paketa	10. 10. 2022	24. 10. 2022
Konfiguracija rezervacijskega paketa	24. 10. 2022	3. 11. 2022
Konfiguracija paketa prevzema in vrnitve vozila	2. 11. 2022	8. 11. 2022
Konfiguracija vodenja flote vozil	8. 11. 2022	15. 11. 2022
Konfiguracija vodenja popravila vozil	15. 11. 2022	22. 11. 2022
Konfiguracija računovodskega paketa	22. 11. 2022	29. 11. 2022
Konfiguracija plačilnega paketa	29. 11. 2022	6. 12. 2022
Konfiguracija plačilnega vmesnika med vsemi paketi	6. 12. 2022	30. 12. 2022

se nadaljuje

Tabela 3: Okvirno izvajanje projekta (nad.)

Opravi	Začetek	Konec
<i>Konfiguracije</i>	10. 10. 2022	30. 12. 2022
<i>Ostale aktivnosti projekta</i>	20. 12. 2022	3. 3. 2023
Nabava strežnika in uporabnikove opreme	20. 12. 2022	3. 1. 2023
Kreiranje, urejanje, optimiziranje podatkovne baze	29. 12. 2022	16. 1. 2023
Namestitev strojne opreme	16. 1. 2023	25. 1. 2023
Namestitev programske opreme	25. 1. 2023	1. 2. 2023
Konfiguracija strežniške opreme in oblachnega streznika	1. 2. 2023	6. 2. 2023
Prenos starih datotek na nov sistem	6. 2. 2023	9. 2. 2023
Uvajanje novega sistema	9. 2. 2023	3. 3. 2023

Vir: lastno delo.

5.2.3 Management kakovosti

Načrtovanje kakovosti projekta je bistven del pri zagotavljanju uspešnosti projekta. Pridobivanje potreb in zahtev je zabeleženo v elektronski obliki. Programska rešitev mora temeljiti na najnovejši in najustreznejši programski opremi, ki zagotavlja nemoteno in varno delovanje sistema. Zagotoviti je treba ustrezen projektni tim, ki bo opravil delo strokovno in kvalitetno. Načrtovanje zajema tudi ustrezen način komuniciranja med vsemi deležniki projekta. To je omogočeno z različnimi komunikacijskimi orodji (npr. telefon, elektronska pošta, programi, kot so Zoom Video Communications in Microsoft Teams) (Rochford in Mitchell, 2019). Kakovost projekta zagotavlja strokovnost projektnega tima, standardi za varnost in ustrezno delovanje informacijske infrastrukture, fleksibilnost projekta na morebitne spremembe, ki bi lahko vplivale na podaljšanje projekta in ustrezno izobraževanje zaposlenih podjetja, za pravilno in najbolj učinkovito delovanje sistema (Rochford in Mitchell, 2019). Nadzor nad kvaliteto se bo izvajal preko stalnega nadzora projekta s strani projektnega tima in projektnega skrbnika. Način nadzora in kakšni ukrepi se bodo izvajali v primeru padca kvalitete, se določijo v začetku projekta. Odziv zaposlenih med izvedbo projekta se bo spremljal preko skupnega odlagališča za dokumente, kjer bodo imeli možnost komuniciranja s projektnim timom in podajanja komentarjev. Podpiše se sporazum o ravni storitev (angl. Service Level Agreement, v nadaljevanju SLA), ki je zaveza med ponudnikom storitev in stranko. O posameznih vidikih storitve, kakovosti, razpoložljivosti, odgovornosti se dogovorita ponudnik in uporabnik storitve (Overby in drugi, 2024). Podroben opis je zajeti v tabeli 4.

Tabela 4: Management kakovosti

Načrtovanje kvalitete	Zagotavljanje kvalitete	Nadzor nad kvaliteto
Ugotovitev ali projekt izpolni naročnikove želje, S pravilnim načrtovanjem se izognemo težavam oziroma se z njimi lažje soočimo, Prepoznavanje tveganj je temelj kvalitetnega managementa, Zagotavljanje ustrezne strojne in programske opreme, Zaposlitev ustrezno kvalificiranega kadra, Ustrezno spremljanje in management posameznih faz projekta, Ustrezna dokumentacija pogodb s stranko, Preprosto in razumevajoče izobraževanje ljudi (zaposlenih)	Redno spremljanje aktivnosti za zagotavljanje standardov kvalitete, V primeru zaznave težav pri kvaliteti projekta se uporabijo ustrezni ukrepi, V primeru dodatnih zahtev po izboljšavah je treba zagotoviti dovolj časa, da se te izpolnijo, Zagotavljanje kakovosti pomaga spremljati uspešnost zaposlenih, V primeru sprememb pri zagotavljanju kakovosti se lahko projektni načrt posodobi, Zagotavljanje ustrezne usposobljenosti vseh članov projektnega tima, Zagotavlja ustrezno izobraževanje vseh zaposlenih pred izdajo projekta oziroma zagonom celotnega sistema	Način rednega nadzora, ki ne dovoli padca standardov projekta, Določi ustrezne korektivne ukrepe, Primerjava rezultatov s kvalitetnimi standardi, »Orodja za kontrolo kvalitete: Pareto diagrami, kontrolni grafikoni, diagram poteka ipd.)«, Ponavljajoča se dela naj bodo minimalizirana zaradi možnosti prekoračitve časovne omejitve projekta, Zagotavljanje kvalitete pregleduje in zagotavlja ustrezno komunikacijo med vsemi deležniki projekta, Testiranje posamezne enote aplikacije, Integracijsko testiranje enot aplikacije, Sistemske testiranje celotnega informacijskega sistema, Testiranje sistema s strani uporabnika, ki zmanjšuje težave in zagotavlja zadovoljstvo stranke in uporabnikov, Sporazum o ravni storitve (SLA) (penali), Implementacija sprememb kadarkoli v času projekta za zagotavljanje zadovoljstva stranke

Vir: lastno delo.

5.2.4 Management komunikacij

Učinkovita komunikacija tekom projekta lahko služi kot katalizator za preoblikovanje dojemanj deležnikov glede projekta, iz dobrega v zelo uspešnega. Nasprotno pa lahko slaba komunikacija tekom projekta spremeni dojemanje deležnikov, iz odličnega v srednjeročnega. Ena od težav, s katero se je soočalo podjetje v preteklosti, je bila prevelika količina informacij. Vsakodnevni sestanki so trajali predolgo časa, pri katerih se je želelo

zajeti in predelati preveč tekočih in načrtovanih nalog ter opravil, med katerimi so bile nekatere tudi nesmiselne oziroma nepotrebne. Zato se pogosto dogaja, da niso podana točna navodila, kar lahko povzroči zmedenost med zaposlenimi. Opravila so se kopičila in so bila opravljana z namenom, da se dokončajo in ne z namenom, da se naredi tisto, kar je bilo potrebno, kaj je bilo zahtevano (Martins, 2024). Prevelika vključenost vodstva v sam proces formiranja, dodeljevanja opravila je onemogočila normalno funkcioniranje zaposlenih. Potrebno je večje vključevanje samih zaposlenih in onemogočiti, da vodstvo ostane kanal za širjenje nadaljnjih informacij. Srečanja so izjemno pomembna za razumevanje napredka projekta, zagotavljanje posodobitev projekta in razumevanje, da so vsi člani na isti strani v smislu razumevanja zahtev projekta. Bistvenega pomena je, da so sestanki kratki in vsebujejo ključne informacije. Pomembno je prejemanje povratnih informacij od vseh deležnikov s strani podjetja, zato je treba med njimi spodbujati proaktivnost. Ko je oblikovan in dogovorjen celovit komunikacijski načrt, mora projektni manager določiti, kako lahko zagotovi, da vse strani, vključene v projekt, učinkovito uporabljajo standardizirane komunikacijske prakse. Najbolj zanesljiva in učinkovita sredstva za doseg tega cilja bi bila izbira združljive programske rešitve, ki vso komunikacijo shranjuje na eni osrednji lokaciji. Udeleženci so usmerjeni skozi sprejete komunikacijske procese, ki jih je manager vnaprej določil. Uporaba videokonferenčnih programov (Zoom, Microsoft Teams) pripomore k povečanju produktivnosti in vključenosti v projekt. Uporaba skupnega odložišča (Google Drive) omogoča vsem udeležencem v projektu, da pridejo do informacij o stanju projekta, kadarkoli in kjerkoli (Martins, 2024). Podrobnejši načrt managementa komunikacij je prikazan v prilogi 4.

5.2.5 Načrt nakupa opreme

Načrt nakupa opreme je nujen proces za nakup ali pridobitev izdelkov, storitev ali rezultatov, ki jih potrebuje projektni tim za uspešno izvedbo projekta (Bit.AI, 2022). Podjetje potrebuje celotno prenovo strojne in programske opreme, ki mora ustrezati zahtevam nemotenega poslovanja podjetja. Oprema mora zagotavljati klasičen kot mobilni dostop do sistema. Načrt nakupa opreme vključuje štiri sodobne delovne postaje, dva tablična računalnika za dostop do podatkov, ko je zaposleni na terenu in sodobni strežnik za varnostno kopiranje in shranjevanje občutljivih podatkov. V primeru izpada električnega toka je pomožno napajanje zagotovljeno z brezprekinitvenim napajanjem UPS, ki omogoča dvourni nemoteno delovanje po izpadu električnega toka. Dodatno varnost zagotavlja brezžični usmerjevalnik s požarnim zidom, ki je primeren za manjše domače pisarne. Večina nakupa strojne opreme vključuje tudi strošek programske opreme, ki je potrebna za ustrezno delovanje in poslovanje. Načrt nakupa opreme zajema tudi nakup senzorjev za vozila, ki so potrebni za komunikacijo med sistemom in samim vozilom. To vključuje 32 senzorjev za diagnostiko in prav toliko avto kamer s pripadajočimi spominskimi karticami.

5.2.6 Management tveganj

Vsak projekt se sooča z različnimi tveganji, ki se jim ni mogoče izogniti. Nekatera tveganja imajo lahko močan vpliv na potek projekta, zato jih je treba identificirati, jih obvladovati in predvidevati verjetnost njihovega pojava. Načrt obvladovanja tveganj predstavlja pregled tveganj, kakšen imajo vpliv oziroma posledice na projekt, s kakšno verjetnostjo se lahko pojavijo, kakšni so primerni ukrepi in kako se jim lahko izognemo (Schwartz, 2021). Ker gre za manjše podjetje, ki deluje na trgu popolne konkurence in se je v preteklosti soočalo s težavami pri modernizaciji poslovanja, je treba dobro identificirati možna tveganja in definirati primerne ukrepe za zmanjševanje možnosti pojavljanja le teh. Obvladovanje tveganj in določitev omejitev in smernic omogoča transparentnost, ki ponuja vsem deležnikom, ki so bili v preteklosti soočeni z neuspešnimi projekti podjetja, motivacijo, sledljivost, sodelovanje in ustrezno seznanitev s potekom celotnega projekta.

Glavni vidiki tveganj pri projektu so:

- tveganja pri razumevanju poslovanja podjetja,
- tveganja pri pridobivanju informacij poslovanja podjetja in razumevanju zahtev informacijskega sistema,
- tveganja, ki so povezana z zamikom časovnice projekta, prekoračitve stroškov projekta,
- tveganja pri dobavi opreme, ki je potrebna za uspešno dokončanje projekta.

Načrt obvladovanja tveganj se nahaja v tabeli 5.

Tabela 5: Načrt obvladovanja tveganj

Področje tveganja	Tveganje	Verjetnost pojavitve	Vpliv dogodka	Posledice dogodka	Omilitveni ukrepi	Monitoring
Potrebe	Potrebe se lahko spreminjajo med projektom	Visoka	Močno	Zelo visoke	Pravilni management obsega projekta	Držati se obsega projekta
Zahteve	Ne zadostuje zahtevam	Visoka	Močno	Zelo visoke	Obsežna analiza pred določitvijo zahtev	Brez popuščanja pri določitvi standardov zahtev
Zadovoljstvo uporabnika	Nezadovoljstvo naročnika zaradi počasnega projekta in premalo ažurnih rezultatov	Visoka	Močno	Zelo visoke	Izpolnjevanje dogovora stranke poveča zadovoljstvo	Veliko povratnih informacij

se nadaljuje

Tabela 5: Načrt obvladovanja tveganj (nad.)

Področje tveganja	Tveganje	Verjetnost pojavitve	Vpliv dogodka	Posledice dogodka	Omilitveni ukrepi	Monitoring
Strošek	Zaradi spreminjanja potreb in zahtev lahko privede do povečanja stroškov projekta	Visoka	Močno	Zelo visoke	Beleženje spreminjanja stroškov in potrditev s strani stranke o povečanju stroškov	Monitoring stroškov
Projektni Management	Slaba komunikacija in sinergija med vodstvom in timom je slaba	Nizko	Močno	Srednje	V timu so visoko kvalificirani ljudje	Primerna komunikacija in primerno delovno okolje
Obseg	Sprememba obsega projekta je težavna, saj so določene omejitve, ki vplivajo na čas, izvajanje, stroške prilagajanje	Nizko	Močno	Srednje	Izobraževanje zaposlenih ublaži spremembe obsega projekta	Definirana so pravila in omejitve obsega projekta
Čas	Čas za dokončanje projekta	Visoka	Močno	Zelo visoke	Jasno definirani mejniki	Doseganje ciljev
Zunanji vplivi	Zamude, težave zunanjih izvajalcev, Težave s programsko opremo	Nizko	Močno	Srednje	Redno spremljanje dela	Redno obveščanje zunanjih izvajalcev

Vir: lastno delo.

5.3 Izvajanje in kontrola projekta

Izvajanje projekta se začne z izvedbo konfiguracij informacijskega sistema. Razvijalci hkrati razvijajo konfiguracijo CRM paketa, rezervacijskega paketa, paketa prevzema in vrnitve vozil, vodenja popravil vozil, računovodskega paketa in plačilnega paketa. Razen računovodske in plačilne konfiguracije vsi razvijalci hkrati razvijajo pakete. Potencialna težava takega pristopa razvoja je medsebojna odvisnost posameznih komponent in zasedenost razvijalcev na posameznih paketih. V primeru, da prvi razvijalec obični na konfiguraciji paketa vodenja flote, lahko zastane delo na konfiguraciji paketov vodenja popravila vozil in prevzema ter vrnitve vozila. Takim težavam smo se izognili z uporabo projektnih zalog na posameznih aktivnostih in tako alocirali dovolj časa za njihovo izvedbo. Znotraj razvoja se po vsaki končani delujoči komponenti aplikacije opravi prva faza testiranja posamezne enote, ki jo opravijo sami razvijalci. Pri tem se preveri ustrezno

delovanje programske kode. Med izvedbo projekta projektni tim razvija prvo alfa verzijo informacijskega sistema. Ko se razvijejo vse komponente konfiguracij paketov, se projektni tim osredotoči na postavitve informacijskega sistema na lokaciji podjetja. Postavi se podatkovna baza, namesti se programska in strojna oprema, konfigurira se dostop do oblačnega strežnika in konfigurira se grafični vmesnik z vsemi paketi informacijskega sistema. Tukaj se izda beta verzija informacijskega sistema, ki je namenjena stranki in zaposlenim v podjetju. S tem se prvič preveri, ali informacijski sistem izpolnjuje strankine zahteve.

Pri projektu se je nadzor kakovosti spremljal preko testiranja. Testiranje se je izvajalo v štirih fazah – testiranje posamezne enote, integracijsko testiranje, sistemsko testiranje in testiranje sprejemljivosti uporabnika. Testiranje posamezne enote izvaja razvijalec, medtem ko integracijsko in sistemsko testiranje izvajajo vsi člani projektnega tima in delujejo kot testni tim. Test sprejemljivosti opravi stranka oziroma zaposleni in se s tem dokončno preveri ali rešitev izpolnjuje vse zastavljene zahteve.

Tekom projekta se preverja skladnost projekta s časovnim in finančnim načrtom. S pomočjo gantograma se preverja, sledi časovni potek projekta ali je razvojni tim v okvirjih načrtovanega dela, delo poteka hitreje, počasneje, kot je bilo načrtovano ali obstaja možnost komplikacij aktivnosti, ki so na kritični poti.

5.3.1 Izvajanje stroškov

Tabela 6 prikazuje planirane stroške projekta. Višine urnih postavk so določene iz splošne prakse. Urna postavka projektne managerja je postavljena v višini 80 €, zunanji izvajalec ima urno postavko postavljeno v višini 100 €. Projektni tim, ki ga sestavljata dva razvijalca, je plačana različno, zaradi različne stopnje znanja, prvi programer ima urno postavko 70 €, drugi programer 60 €. Stroški strojne opreme so razvidni iz tabele 7 in skupaj znašajo 10.722,55 €. Izdatek za vzdrževanje strojne opreme in usmerjevalnika je fiksno določen, 10 % od celotnega stroška nakupa strojne opreme. Strošek internetne povezave zajema storitev poslovnega paketa Poslovni DUO (doplačilo 2Gbit linija) Telekoma Slovenije, kar znaša prvih šest mesecev 33,61 €, nato pa 50,1 €. Ker bo podjetje kupilo že obstoječo rešitev, ki je na voljo na trgu, se smatra kot strošek nakupa programske opreme sama aplikacija, katere vrednost je 1500 €. Izdatek za licenco aplikacije znaša 225 € mesečno. Vrednost konfiguracije sistema zajema celoten strošek obdobja konfiguracije, implementacije in dela, v skupnem znesku 115.800 €, kot je razvidno iz tabele 6. Stroški za vzdrževanje strojne opreme, internetno povezavo, licenčno programsko opremo se smatrajo kot dolgoročni strošek, saj se bodo nadaljevali po zaključku projekta. Končni strošek projekta je postavljen z 20 % zalogo, ter znaša 154.787 €.

Tabela 6: Strošek projekta

	Oktober	November	December	Januar	Februar	Marec	
Osebj							
Projektni manager	2400 €	3200 €	3200 €	3200 €	3200 €	800 €	
Zunanji izvajalec	7.900 €	11.900 €	8.900 €				
Razvijalec 1	6.090 €	8.330 €	7.840 €	8.120 €	7.910 €	770 €	
Razvijalec 2	4.980 €	7.140 €	6.060 €	6.600 €	6.600 €	660 €	
Strojna oprema							
Delovne postaje				2.516 €			
Delovna postaja UPS				536 €			
Tablični računalnik				500 €			
Strežnik				1.754 €			
Usmerjevalnik				492 €			
Namestitev interneta				58 €			
Telematika (OBD, GPS,...)				4.924 €			
Vzdrževanje strojne opreme				30 €	30 €	30 €	
Vzdrževanje usmerjevalnika				15 €	15 €	15 €	
Internet				34 €	34 €	34 €	
Programska oprema							
Strošek nakupa				1.500 €			
Licenčna programska oprema				225 €	225 €	225 €	
Mesečni strošek	21.370 €	30.570 €	26.000 €	30.503 €	18.013 €	2.533 €	
Skupni strošek							128.989 €
20 % zvišanje stroškov							154.787 €

Vir: lastno delo.

Tabela 7 prikazuje strošek nakupa strojne, programske opreme.

Tabela 7: Stroški nakupa opreme

Strojna oprema	Tip	Znesek
Delovna postaja in operacijski sistem	HP Z440 delovna postaja, 1x Intel Xeon E5-1620 v3, 3,5 GHz, 32 GB RAM DDR4, 512 GB SSD, NVidia Quadro K2200, Win 10 Pro	4 x 629,11 € = 2.516,44 €
Napajalnik	Rebel UPS Micropower 1000 (offline, 1000VA / 600 W, 230V, 50Hz)	4 x 133,90 € = 535,6 €
Tablični računalnik	Lenovo Tab M10 FHD Plus tablični računalnik (ZA5W0189BG) + polnilna postaja	2 x 249,99 € = 499,98 €
Strežnik in operacijski sistem	IBM strežnik xSeries x3200M3 + Microsoft Windows SmallBusinessServer 2011 Standard Edition +	1114,50 € + 639,15 € = 1.753,65 €
Usmerjevalnik	ASUS RT-AX89X brezžični usmerjevalnik (90IG04J1-BM3010)	492,40 €
Senzor za telematiko	4G OBD GPS Tracker HS-4000G	32 * 80 € = 2560 €
Avto kamera	SENCOR SCR 5000GS avto kamera	32 * 64,99 € = 2.079,68 €
Spominska kartica	SanDisk Ultra MicroSDXC spominska kartica, 64 GB, UHS-I + SD adapter	32 * 8,90 € = 284,8 €
Skupaj		10.722,55 €

Vir: lastno delo.

5.3.2 Kontrola stroškov

Management stroškov je dokument, ki opisuje, na kakšen način se bo v projektu planiralo, strukturiralo in kontroliralo stroške. Proces zajema pridobitev denarnih sredstev za dokončanje projektnih aktivnosti. Običajno se uporablja pristop od spodaj navzgor (angl. bottom up) za določitev potrebnih virov posameznih aktivnosti, katerih vsota je ocena stroškov celotnega projekta (Roseke, 2016). Za kontrolo stroškov projekta se uporablja metoda prislužene vrednosti (angl. Earned Value Management, v nadaljevanju EVM). EVM je tehnika merjenja uspešnosti projekta, ki združuje podatke o času in stroških. Zajema stroške, načrt in obseg projekta ter tveganja, s katerimi se primerja trenutno stanje s planiranim potekom projekta. Tehnika EVM omogoča projektnemu managerju oceno napredka, napoved stanja stroškov ob zaključku in prepoznavanje možnih težav projekta (Breuker, 2021).

5.3.2.1 Osnovna analiza prislužene vrednosti - PV, AC in EV

EVM vključuje tri elemente za kalkulacijo posamezne aktivnosti. To so planirana vrednost (angl. Planned Value, v nadaljevanju PV), dejanski stroški (angl. Actual Cost, v nadaljevanju AC) in prislužena vrednost (angl. Earned Value, v nadaljevanju EV). Planirana vrednost je

celoten planiran strošek za posamezno aktivnost (v urah, denarnih enotah ali drugih merljivih enotah). EVM opredeli količinsko vrednost napredka čez čas trajanja projekta z uporabo iste merske enote, ki je bila uporabljena pri načrtovanju proračuna projekta. Planirana vrednost se izračuna z enačbo $PV = \text{strošek na uro} * \text{število načrtovanih ur}$.

Dejanski stroški so celotni stroški za posamezno aktivnost. Izračunajo se z enačbo $AC = \text{strošek na uro} * \text{število ur}$ (Acqnotes, 2021).

Prislužena vrednost (angl. Budgeted Cost of Work Performed) je kazalnik uspešnosti, ki prikazuje ocenjeno vrednost opravljenega dela do določenega datuma. Prislužena vrednost se izračuna po enačbi $EV = PV * \text{odstotek opravljenega dela}$ (Breuker, 2021). Izračuni se nahajajo v tabeli 8.

Tabela 8: Kumulativne vrednosti PV, AC, EV

	PV	AC	EV	EAC
Oktober	20.973,85	21.368,86	20.973,85	
November	53.721,90	53.415,00	53.721,90	
December	80.290,12	78.361,43	80.290,12	
Januar	110.817,57	105.603,44	104.205,86	
Februar	128.027,30	105.603,44	104.205,86	
Marec	130.481,84	105.603,44	104.205,86	
				132.157,76

Vir: lastno delo.

5.3.2.2 Načrtovano in stroškovno odstopanje (SV in CV)

Z uporabo matrike EV projektni manager lahko opravi analizo prislužene vrednosti. Preprosti izračuni nudijo vpogled v različna odstopanja projekta in pomagajo projektному managerju pri odločitvah o najpomembnejših in problematičnih področjih projekta. S primerjavo prislužene vrednosti in planirane vrednosti projektni manager pridobi merljivo vrednost obsega zamudnega dela. Razlika med prisluženo vrednostjo in planirano vrednostjo se imenuje načrtovano odstopanje (angl. Schedule variance, v nadaljevanju SV). Načrtovano odstopanje se izračuna po enačbi $SV = EV - PV$. Negativen SV označuje, da projekt opravlja manj dela, kot je bilo planirano, medtem ko pozitivna vrednost SV označuje, da projekt opravlja več dela, kot je bilo planirano. Visoka vrednost SV lahko nakazuje na slabo načrtovanje projekta in težave z viri projekta, ki jih je treba takoj obravnavati. Izračun stroškovnega odstopanja (angl. Cost variance, v nadaljevanju CV) omogoča projektному managerju uporabno merilo ali projekt dosega načrtovane proračunske cilje s primerjavo vrednosti prislužene vrednosti in dejanskih stroškov. Enačba za izračun je $CV = EV - AC$. Negativna vrednost CV pomeni, da projekt potrebuje več sredstev, kot je bilo planirano za dokončanje opravil, medtem ko pozitivna vrednost CV pomeni, da je poraba sredstev na projektu manjša kot planirana. Projektni manager lahko z uporabo CV hitro ugotovi, ali

projekt presega načrtovan proračun.

5.3.2.3 Indeks načrtovanega in stroškovnega odstopanja (SPI in CPI)

Indeks načrtovanega odstopanja (angl. Schedule Performance Index, v nadaljevanju SPI) in indeks stroškovnega odstopanja (angl. Cost Performance Index, v nadaljevanju CPI) so odstopanja, izražena v odstotkih. Če je, vrednost SPI enaka 1 ali 100 %, projekt poteka po načrtu. V primeru zamud, bo vrednost SPI manjša kot 1. V nasprotnem primeru, bo vrednost SPI večja kot 1, kar pomeni, da je projekt pred rokom dokončanja. Tako kot pri SPI, je lahko vrednost CPI enaka 1 ali 100 %, kar pomeni, da so stroški projekta enaki planiranim stroškom. Ko je CPI večji kot 1, so trenutni stroški manjši, kot je bilo planirano. V primeru, ko je CPI manjši kot 1, se projekt izvaja nad planiranim proračunom. Z analizo trendov CV in SV tekom projekta, projektni manager ugotovi, kakšna je uspešnost projekta. Največja prednost napredne analize EV je njena zmožnost napovedovanja stroškov in načrtovanega zaključka projekta. Ocena časa za dokončanje (angl. Estimate Time to Complete, v nadaljevanju ETC) pridobimo z deljenjem planiranega časa projekta (angl. Original Time Estimate, v nadaljevanju OTE) in SPI, in tako napovemo, koliko časa bo trajal projekt. Enačba za izračun je $ETC = \text{planiran čas projekta} / SPI$. Rezultati načrtovanega in stroškovnega odstopanja ter indeksi se nahajajo v tabeli 9. V našem primeru so vrednosti indeksov pod 1, kar pomeni, da je projekt v zamudi in so stroški višji kot je bilo načrtovano.

Tabela 9: Izračun CV, CPI, SV, SPI

CV	-1397,59
CPI	0,99
SV	-2514,35
SPI	0,98

Vir: lastno delo.

5.3.2.4 Napredna analiza prislužene vrednosti - EV (ETC in EAC)

Medtem ko preprosti izračuni osnovne analize EV nudijo projektному managerju trenuten vpogled v različna odstopanja projekta, se napredna analiza EV uporablja kot orodje za napovedovanje stanja ob zaključku projekta. Ocenjeni strošek ob zaključku (angl. Estimated Cost at Completion, v nadaljevanju EAC) oceni višino stroškov, za uspešno dokončanje projekta. Za izračun sta potrebna proračun ob zaključku projekta (angl. Budget at Completion), ki je vsota planiranih vrednosti ter CPI, ki prikazuje razmerje vrednosti EV in AC celotnega projekta. Izračuni se nahajajo v tabeli 10.

Tabela 2: Proračun ob zaključku projekta (BAC)

BAC	130.481,84
PV	111.325,86
AC	110.209,09
EV	108.811,50

Vir: lastno delo.

Enačba za izračun ocene stroška ob zaključku projekta je $EAC = BAC / CPI$. Rezultati se nahajajo v tabeli 11.

Tabela 3: EAC in ETC

EAC	132.157,76
ETC	148,63
ETC	149 dni

Vir: lastno delo.

Rezultati vseh izračunov se nahajajo v tabeli 12.

Tabela 4: Prikaz EMV ob končanem sistemskem testu

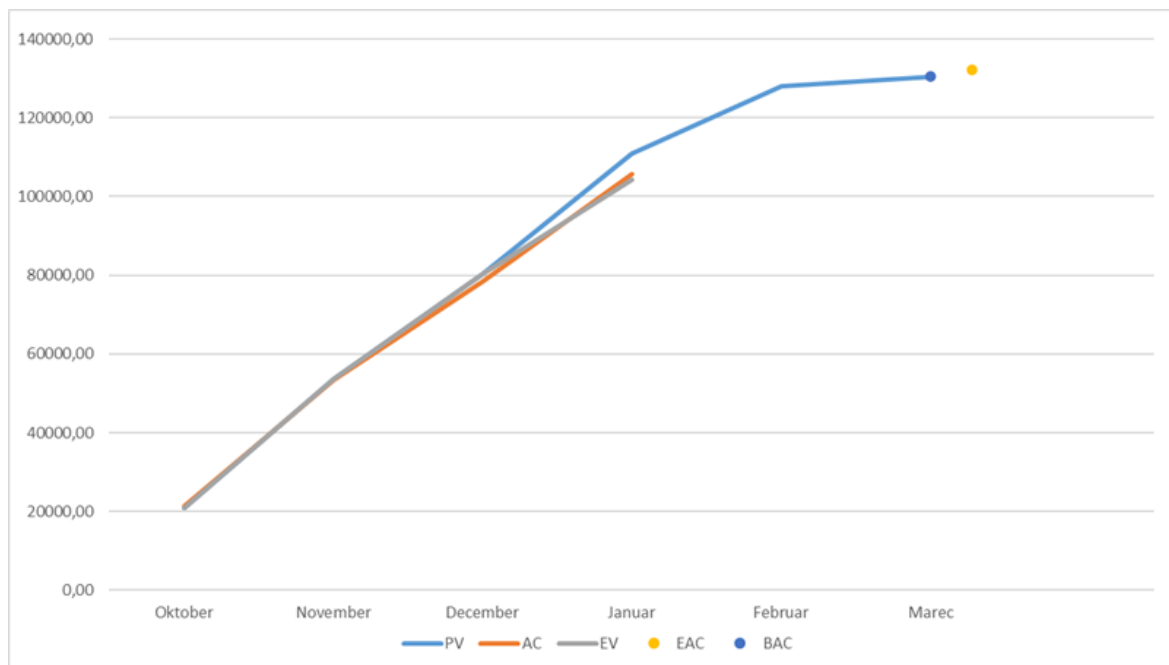
Proračun	260.000
Izračun	
BAC	130.481,84
EV	108.811,50
PV	111.325,86
AC	110.209,09
ETC	148,63
EAC	132.157,76
$SV = EV - PV$	-2514,35
$CV = EV - AC$	-1397,59
$SPI = EV / PV$	0,98
$CPI = EV / AC$	0,99

Vir: lastno delo.

Slika 8 prikazuje graf prislužene vrednosti za polletni projekt po štirih mesecih. Krivulji prislužene vrednosti in dejanskih stroškov se januarja končata. To je razvidno iz tabele 10, kjer so razlike vrednosti aktivnosti po sistemskem testu AC, EV, CV, SV, CPI in SPI enake nič, kar pomeni, da še niso bile izvedene. Krivulja planirane vrednosti prikazuje kumulativo planiranih vrednosti vseh aktivnosti po mesecih in se konča v točki planiranega proračuna ob zaključku projekta. Krivulji dejanskih stroškov in prislužene vrednosti prikazujeta kumulativo vseh stroškov in prisluženih vrednosti po mesecih. Točka ocenjenega stroška ob

zaključku projekta se nahaja nad in desno od točke planiranega proračuna, kar pomeni, da bo projekt po štirih mesecih predvidoma stal več in trajal dlje, kot je bilo načrtovano, ampak znotraj 20 % zalog (projektne, stroškovne).

Slika 8: Graf prislužene vrednosti za projekt po 4 mesecih



Vir: lastno delo.

6 UPORABA SCRUM-A NA PODJETJU RENTAVAN

Razlogi za uporabo Scrum metode v podjetju Rentavan je prevelika kompleksnost izvedbe tradicionalnega managementa projekta za tako majhno podjetje, pri katerem je preveč tveganj, da bi projekt bil izveden pravilno in pravočasno. Učinkovito rabo agilnih metod omogočajo nove napredne tehnologije, posebej za razvoj programske opreme. Celoten projekt sestavljajo sklopi programske opreme z uporabo računalniške opreme za pridobivanje podatkov v realnem času. Za razvoj posameznega sklopa je najbolj učinkovit pristop in uporaba agilnih metod, s katerimi so vsi deležniki vedno seznanjeni z razvojem oziroma napredkom projekta, kar ustreza vodstvu podjetja Rentavan. Scrum metoda pride v poštev zaradi modularne narave projekta, saj se lahko posamezni deli sistema ločeno razvijajo in ne vplivajo na druge segmente poslovanja podjetja Rentavan. S tem se prihrani na stroških uvajanja novega sistema in stroških dela. Časovno je tudi bolj ugodna uporaba agilne metode Scrum, saj ima projektna skupina boljši pregled nad časovnico projekta in ga hitreje in učinkoviteje konča. Pri vsem tem so vključeni tudi vsi deležniki projekta, od katerih se neprestano preverja in upošteva njihovo mnenje in zahteve. Seznanjeni so z vsakim naslednjim korakom razvoja in testiranja novega sistema, kar pripomore tudi pri uvajanju in učenju zaposlenim z uporabo bolj kompleksnega od prejšnjega, že obstoječega

sistema. Scrum metoda omogoča stalno preverjanje stanja projekta, kar zmanjša, hitreje identificira in kontrolira tveganja, s katerimi je soočena projektna skupina. V primeru deviacije posameznega sklopa projekta, Scrum metoda omogoča hitro in preprosto reševanje problema, ki drastično ne vpliva na potek celotnega projekta.

Projektni tim sestavljajo produktni lastnik, Scrum skrbnik, dva notranja razvijalca in zunanji razvijalec, specializiran za implementacijo sistemov za vodenje voznega parka. Produktni lastnik je odgovoren za spremljanje stroškov in za kreiranje in dopolnjevanje produktnega seznama z uporabniškimi zgodbami, v katerem prioritizira bolj pomembne zgodbe projekta, ki se morajo izvesti in je zadolžen, da so zgodbe in delo, ki mora biti opravljeno jasne razvojnemu timu. Skupaj z razvojnimi timom razčlenjuje večje zgodbe. Prednost razvojnega tima je njegova agilnost. To doseže tako, da ima proste roke znotraj dela, ki jim ga določi produktni lastnik. Velikost tima mora biti določena primerno zahtevnosti projekta. Manjši tim je lažje vodljiv in bolje komunicira, vendar mora biti dovolj velik, da se dosežejo vse zgodbe posameznega sprinta. Znanje razvojnega tima je na visokem nivoju, vendar se med trajanjem projekta znanje z vsakim novim sprintom povečuje, kar vpliva na hitrost izvedbe projekta. Pri tem jim pomaga Scrum skrbnik, ki deluje kot svetovalec razvojnemu timu na področju Scrum metode. Njegova naloga je, da v primeru kršenja metode posreduje in pomaga razvojnemu timu, da se drži okvirjev Scrum-a. Scrum skrbnik ima tudi dobro tehnično znanje, s katerim deluje kot procesna avtoriteta. Predvsem pa je glavna naloga Scrum skrbnika, da vsi člani projektne skupine razumejo in pravilno implementirajo Scrum metodo pri izvedbi celotnega projekta.

Projekt je sestavljen iz podsistemov informacijskega sistema. Posamezni podsistemi so bili tako razdeljeni v posamezne sprinte, ki so bili ob dokončanju predstavljeni naročniku kot samostojna rešitev. Ob vsakem dokončanem sprintu je bil novi sklop dodan k že opravljeni rešitvi kot dodatek, tako da naročnik in vsi zaposleni spremljalo razvoj celotnega sistema postopoma. S tem si projektni tim drastično zmanjša delo pri uvajanju in učenju zaposlenih o delovanju novega sistema. Projekt je sestavljen tako, da se najprej pridobi vse zahteve in ugotovi, katere rešitve so najbolj ugodne za naročnika, implementira in prilagodi sistem za management strank. Nato se implementira programska opremo za vodenje voznega parka z vso računalniško opremo, ki jo sistem in uporabniki potrebujejo, nato se prilagodi računovodski sistem, ki beleži in vodi finančni in transakcijski del celotnega plačilnega sistema. Ko so implementirane glavne komponente sistema, se prilagodi grafična podoba, preko katere dostopajo zaposleni in uporabniki sistema.

Uporabnost in preglednost vmesnika mora biti dovolj preprosta za uporabo, tako da celoten proces naročanja, najema in vrnitve vozil poteka nemoteno in z najmanj težavami. Za uporabnika aplikacije je posebej pomembno, da čas od obiska portala do prejema vozila in plačila storitve mine čim manj časa. Razvojni tim postavi bazo podatkov, ki je v oblaku in na lokaciji, kjer bodo vsi podatki poslovanja. Treba je tudi zagotoviti dostop do že obstoječih podatkov podjetja, kar pomeni digitalizacija predhodnega stanja podjetja. Zaradi varnostnih

vidikov in novejših tehnologij, ki je na voljo, projektni tim tudi posodobi računalniške opreme v podjetju. To zajema tablice, računalnike, modeme oziroma vso računalniško opremo, ki jo potrebujejo za uspešno uporabo novega sistema.

6.1 Potek metode

Projekt implementacije programske rešitve z uporabo metode Scrum poteka okvirno od 10. 10. 2022 do 3. 3. 2023 in traja 145 dni oziroma slabih pet mesecev. Za izpolnitev projekta je potrebnih deset sprintov. Vsak sprint traja deset delovnih oziroma štirinajst dni. Projekt sestavlja šestnajst aktivnosti, ki so nato še posamično razdeljene na več pod aktivnosti. Tabela 13 prikazuje predvideno projektno zalogo projekta. Vsakemu sprintu je bilo dodeljeno določeno število uporabniških točk, ki so bile dodeljene na podlagi ocen projektnega tima. Konfiguracija zalednega, čelnega dela aplikacije in testiranje, uvajanje sistema veljata za najbolj kompleksni aktivnosti v našem projektu, zato imata največjo vrednost uporabniških točk in ju v enem sprintu ni mogoče do konca narediti.

Tabela 5: Projektna zaloga

Sprint	Aktivnost	Število točk
1	Konfiguracija CRM paketa	38
2	Konfiguracija rezervacijskega paketa, Konfiguracija paketa prevzema in vrnitve vozila	42
3	Konfiguracija vodenja flote vozil Konfiguracija vodenja popravila vozil	43
4	Konfiguracija računovodskega paketa Konfiguracija plačilnega paketa	42
5 in 6	Konfiguracija čelnega, zalednega izgleda, grafičnega vmesnika med vsemi paketi	78
7	Kreiranje, urejanje, optimiziranje podatkovne baze Namestitev strojne opreme	36
8	Namestitev strojne opreme Namestitev programske opreme Konfiguracija strežniške opreme z oblaknim strežnikom	44
9	Varnostno kopiranje starih datotek, dokumentov Prenos starih datotek na nov sistem Testiranje sistema	38
10	Testiranje sistema Zagon novega sistema	40

Vir: lastno delo.

Začetek projekta obvezno zajema določitev uporabniških zgodb. Pridobivanje zahtev in potreb je zajemalo postavljanje vprašanj v obliki »Kot (tip uporabnika) hočem (opraviti dejavnost), da (dosežem cilj)«. Na ta vprašanja so odgovarjali direktor podjetja, zaposleni v podjetju in končni uporabniki. Po opredelitvi vseh zgodb so se razdelile v pet okvirnih področij. To so program CRM, sistem voznega parka, aplikacija, plačilni ter računovodski

sistem. Uporabniške zgodbe so tako postale temelj določitve aktivnosti projekta. Tabela 14 prikazuje primere uporabniških zgodb.

Tabela 6: Primer uporabniških zgodb

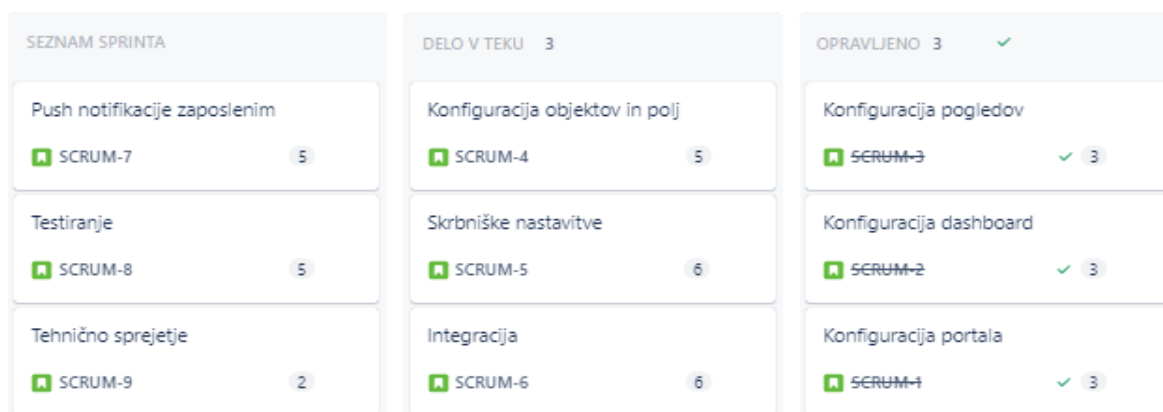
Kot <tip uporabnika>	Hočem <opraviti dejavnost>	Da <dosežem cilj>
Direktor	imeti vpogled nad najemi	vidim mesečni prihodek iz najemov
Zaposleni	imeti pregled nad preteklimi najemi strank (caller id)	v primeru klica znam svetovati stranki
Končni uporabnik	imeti na voljo seznam kombijev	se lažje odločim med njimi
Direktor	imeti vpogled nad najemi kombijev	vem, katere se splača kupovati v prihodnje
Direktor	imeti vpogled nad zaposlenimi	vidim, koliko kombijev, so dali v najem
Zaposleni	imeti enoten sistem dokumentacije	mi ni potrebno istih papirjev trikrat delati
Končni uporabnik	imeti seznam preteklih najemov	mi ni treba iskati istih kombijev

Vir: lastno delo.

Uporabniške točke so bile določene z metodo poker planiranja (angl. planning poker). Vrednosti so določali vsi člani tima. Ta vključuje produktnega lastnika, Scrum skrbnika in razvijalni tim. Metoda preko kartic, na katerih so številke Fibonaccijevega zaporedja, dodeljuje vrednosti aktivnosti po zahtevnosti, časovni obremenitvi, količini dela ipd. V primeru, da so bile vrednosti med aktivnostmi velike in različne ali pa se nekateri člani tima niso strinjali z dodeljenimi točkami, je potekala diskusija, kjer je vsak predstavil in objasn timer svojo oceno. Nato se je metoda planiranja pokra ponovno izvedla. Ta postopek se je ponavljal, dokler se celotni projektni tim ni strinjal z vrednostmi točk. Tako je lahko projektni tim določila časovne okvirje in planirala potek celotnega projekta in to predstavila naročniku.

V prvem sprintu se začne razvoj programske rešitve s konfiguracijo CRM-ja. Poleg oblikovnih sprememb (konfiguracija portala, nadzornih plošč, pogledov, objektov in polj) se dodajo še zaščitni nivo ter dodatne funkcionalnosti, ki nadgradijo osnovno aplikacijo. Dodelijo se skrbniške nastavitve, aplikacijo se integrira s tretjimi aplikacijami (npr. Gmail), oblikujejo in dodelijo se notifikacije, obvestila, ki bodo poslana, prikazana zaposlenim, uporabnikom. Po razvoju in prilagoditvi CRM paketa se opravi pregled in tehnično sprejetje. Na koncu sprinta se poleg pregleda izvaja tudi retrospektiva sprinta. Slika 9 prikazuje tablo poteka prvega sprinta. Levi del table, seznam sprinta, prikazuje vse uporabniške zgodbe, ki morajo biti opravljene, za dokončanje sprinta, ter število uporabniških točk, ki je potrebno za njihovo dokončanje. Delo v teku predstavljajo uporabniške zgodbe, ki se trenutno izvajajo. Ko so naloge končane, se prestavijo v stolpec opravljeno. Ko so vse naloge v končnem delu table, se izvede pregled sprinta.

Slika 9: Potek prvega sprinta



Vir: lastno delo.

Tabela 15 prikazuje dejansko delo razvojnega tima v prvem sprintu. Razvidno je, koliko uporabniških točk so opravili v posameznem dnevu in ali je bilo potrebno, pri kakšni aktivnosti opraviti več ali manj dela, kot je bilo prvotno načrtovano.

Tabela 7: Uporabniške točke v prvem sprintu

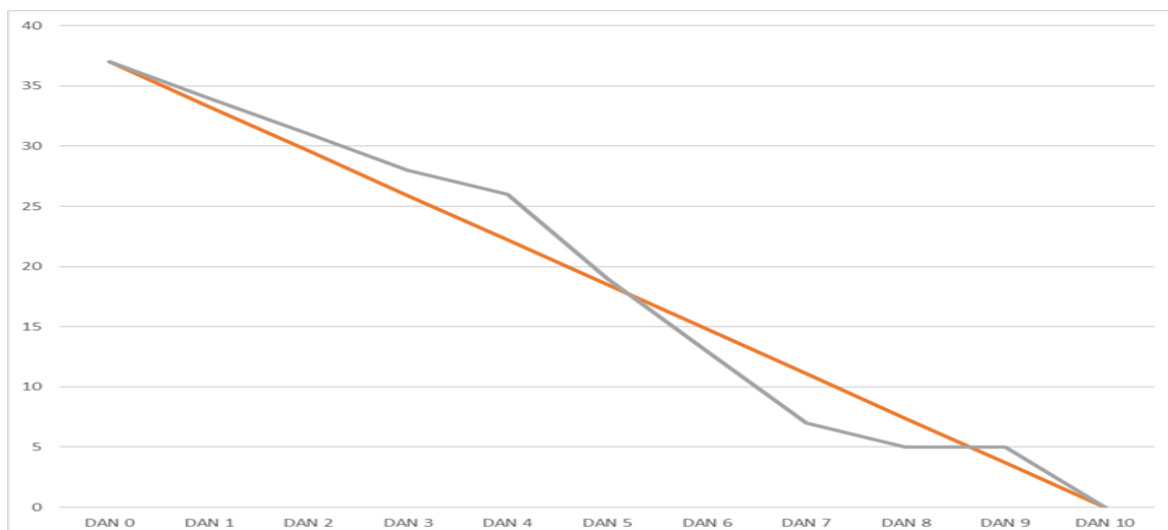
Uporabniške zgodbe	Dan 1	Dan 2	Dan 3	Dan 4	Dan 5	Dan 6	Dan 7	Dan 8	Dan 9	Dan 10
Konfiguracija portala	3	2	1							
Konfiguracija dashboard	3	2	1							
Konfiguracija pogledov (view)	3	2	1							
Konfiguracija objektov in polj	5	5	5	4	2	1				
Skrbniške nastavitve	6	6	6	5	3	1				
Integracija s tretjimi aplikacijami	6	6	6	6	4	3	2	1		
Push notifikacije	5	5	5	5	5	3	1			
Testiranje	5	5	5	5	5	5	4	3	3	1
Tehnično sprejetje	2	2	2	2	2	2	2	2	2	2
Skupaj	38	35	32	27	21	15	9	6	5	3
Opravljen		3	3	5	6	6	6	3	1	2

Vir: lastno delo.

Graf izgorevanja na sliki 10 prikazuje razliko med optimalnim in dejanskim številom uporabniških točk v prvem sprintu. Celotno število uporabniških točk za sprint konfiguracije

CRM paketa, je znašalo 38. Iz grafa je razvidno, da je v začetnih dneh sprinta razvojni tim bil v zamudi, ker se dejanska krivulja nad optimalno krivuljo. Konfiguracija objektov in polj ter skrbniške nastavitve so se začele kasneje, kot je bilo predvideno. Ko je optimalna krivulja pod dejansko, razvojni tim opravlja naloge hitreje, kot je to bilo predvideno, kar je pri nas zgodilo na začetku petega dne sprinta. Na koncu sprinta se obremenitev poveča, ker se je postopek testiranja začel z zamikom.

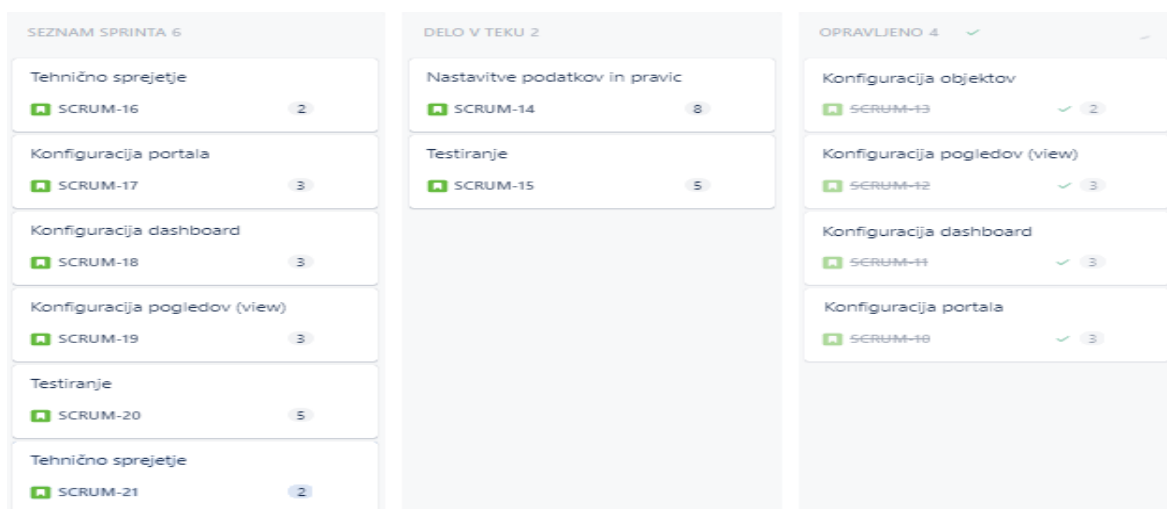
Slika 10: Graf izgorevanja za prvi sprint



Vir: lastno delo.

V drugem sprintu se opravljata dve aktivnosti (prikazano na sliki 11). Konfiguracija rezervacijskega paketa, konfiguracija paketa prevzema in vrnitve vozila. Pri teh aktivnostih se opravljajo oblikovne spremembe obstoječe programske rešitve, testiranja ter tehnična sprejetja.

Slika 11: Potek drugega sprinta



Vir: lastno delo.

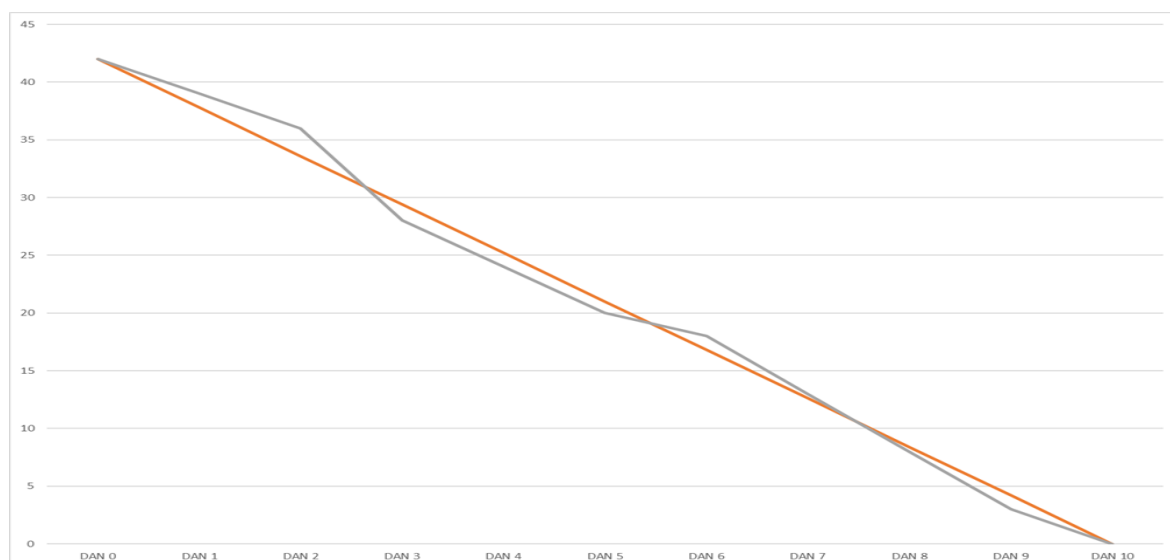
Tabela 16 prikazuje skupno 42 uporabniških točk, ki so bile potrebne za izvedbo drugega sprinta. Tim je večino sprinta izvedla skoraj tako, kot je bilo planirano, vendar je iz slike 12 razvidno, da je razvojni tim naletel na težave in zamude ob začetku sprinta, kjer je bilo treba vzpostaviti programsko opremo rezervacijskega sistema in v šestem ter sedmem dnevu sprinta, kjer se je izvajalo testiranje rezervacijskega paketa. Pri testiranju je bilo treba reševati različne probleme, ki nastajajo ob rezervaciji tovornega vozila. To vključuje od menjave vozila pred najemom, menjave med samim najemom in dostavo novega vozila v primeru okvare ali prometne nesreče. Vse situacije mora rezervacijski sistem beležiti in to predstaviti zaposlenim in uporabnikom na enostaven in prijazen način.

Tabela 86: Uporabniške točke v drugem sprintu

Uporabniške zgodbe	Dan 1	Dan 2	Dan 3	Dan 4	Dan 5	Dan 6	Dan 7	Dan 8	Dan 9	Dan 10
Konfiguracija portala	3	2	1							
Konfiguracija dashboard	3	2	1							
Konfiguracija pogledov (view)	3	2	1							
Konfiguracija objektov in polj	2	2	2							
Nastavitev podatkov in pravic uporabnikov sistema	8	8	6	4	2	1				
Testiranje	5	5	5	5	3	1				
Tehnično sprejetje	2	2	2	2	2	2	2			
Konfiguracija portala	3	3	3	3	3	3	3	2	1	
Konfiguracija dashboard	3	3	3	3	3	3	3	2	1	
Konfiguracija pogledov (view)	3	3	3	3	3	3	3	2	1	
Testiranje	5	5	5	5	5	5	5	5	3	1
Tehnično sprejetje	2	2	2	2	2	2	2	2	2	2
Skupaj	42	39	34	27	23	20	18	13	8	3
Opravljen		3	5	7	4	3	2	5	5	3

Vir: lastno delo.

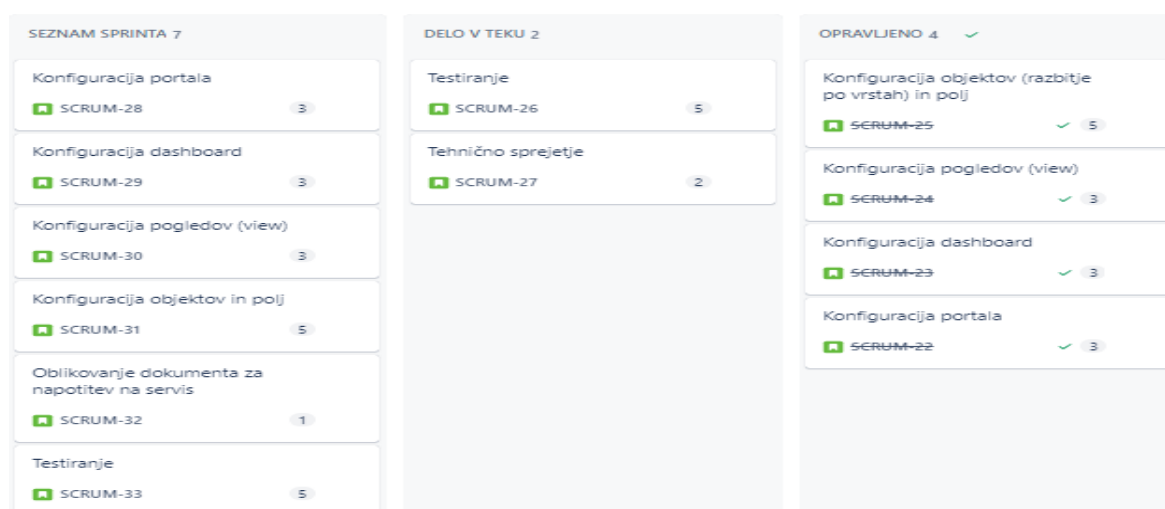
Slika 12: Graf izgorevanja za drugi sprint



Vir: lastno delo.

V tretjem sprintu se opravljata dve aktivnosti (prikazano na sliki 13). Konfiguracija vodenja flote vozil, konfiguracija vodenja popravila vozil. To zajema vzpostavitev sistema za pridobivanje podatkov iz senzorjev, ki bodo vgrajena v vozila. Zaposleni preko sistema lahko redno preverjajo trenutno stanje vozil (prevoženi kilometri, količina goriva, stanje dodatne opreme ipd.). Sistem obvešča o obnovi avtomobilskega zavarovanja, rednega servisa, menjavo pnevmatik ipd. V primeru servisa ali nujnega popravila zaposleni preko systemskega portala kontaktira in rezervira termin pri pooblaščenem servisu. V sprintu se opravi še oblikovne spremembe, testiranje ter tehnično sprejetje.

Slika 13: Potek tretjega sprinta



Vir: lastno delo.

Tabela 17 prikazuje skupno število 43 uporabniških točk, ki jih je bilo treba opraviti za

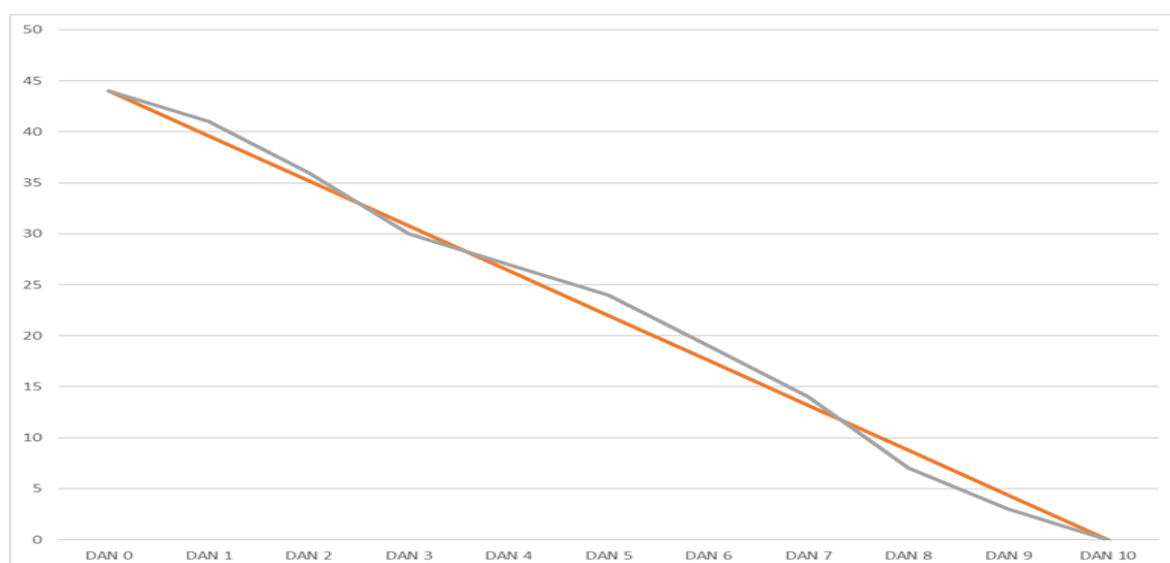
dokončanje sprinta. Iz slike 14 je razvidno, da se je tim seznanjal s podatki, ki se pridobijo iz senzorjev, tako da se pravilno implementirajo v sistem. Saj je bila montaža senzorjev planirana v kasnejšem sprintu. Po poteku grafa je tudi razvidno, da je bilo treba pripraviti podatke za nadaljnjo uporabo pri komunikaciji s serviserji. Pojavil se je rahel zamik v sredini sprinta, ker je moral tim usklajevati dokument z zahtevami serviserjev in ali poteka proces naročanja nemoteno.

Tabela 17: Uporabniške točke v tretjem sprintu

Uporabniške zgodbe	Dan 1	Dan 2	Dan 3	Dan 4	Dan 5	Dan 6	Dan 7	Dan 8	Dan 9	Dan 10
Konfiguracija portala	3	2	1							
Konfiguracija dashboard	3	2	1							
Konfiguracija pogledov (view)	3	2	1							
Konfiguracija objektov in polj	5	5	3	1						
Testiranje	5	5	5	4	2	1				
Tehnično sprejetje	2	2	2	2	2	1				
Konfiguracija portala	3	3	3	3	3	2	1			
Konfiguracija dashboard	3	3	3	3	3	3	1			
Konfiguracija pogledov (view)	3	3	3	3	3	3	1			
Konfiguracija objektov in polj	5	5	5	5	5	5	5	3	1	
Oblikovanje dokumenta za napotitev na servis	1	1	1	1	1	1	1	1	1	
Testiranje	5	5	5	5	5	5	5	5	3	1
Tehnično sprejetje	2	2	2	2	2	2	2	2	2	2
Skupaj	43	40	35	29	26	23	16	11	7	3
Opravljenost		3	5	6	3	3	7	5	4	4

Vir: lastno delo.

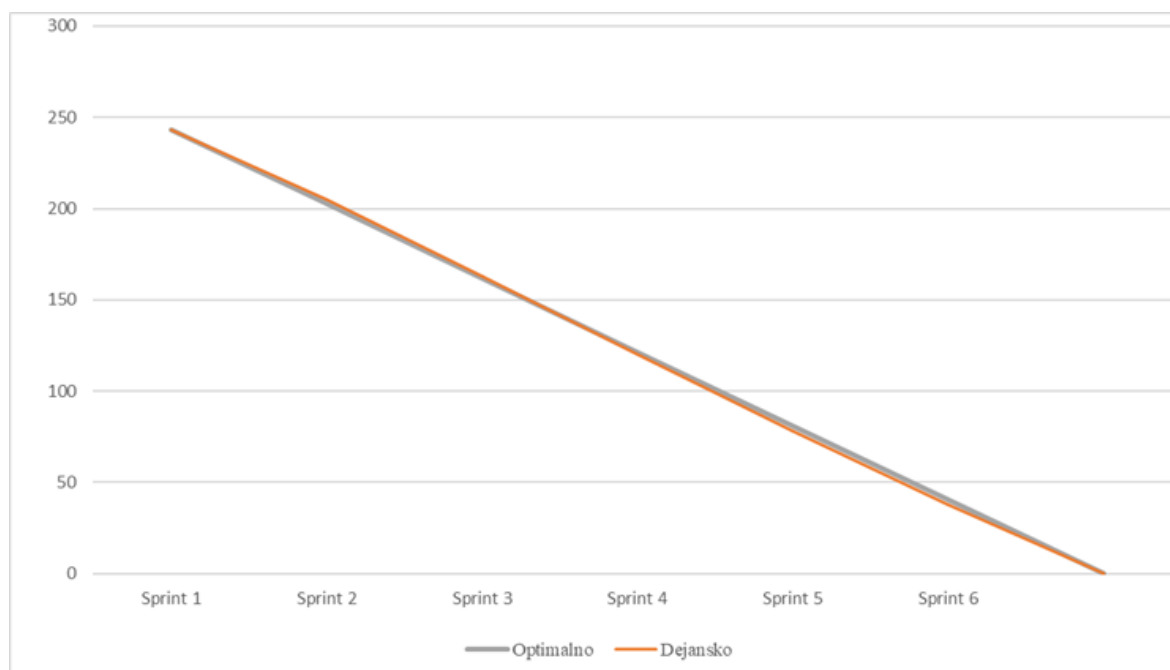
Slika 14: Graf izgorevanja za tretji sprint



Vir: Lastno delo

Nadaljnji sprinti 4, 5, 6 se nahajajo v prilogi. Podrobni potek sprinta 4 v prilogah 4, 5, 6, potek sprinta 5 v prilogi 7, 8, 9 in potek sprinta 6 v prilogi 10, 11 in 12.

Slika 15: Graf izgorevanja do šestega sprinta



Vir: lastno delo.

Grafikon izgorevanja grafično prikaže, koliko uporabniških točk je bilo treba opraviti do sedmega sprinta (prikazano na sliki 15). Vertikalna os prikazuje skupno število opravljenih točk (243 točk), horizontalna os prikazuje število sprintov. Graf vsebuje dve premici, idealno

in dejansko. Dejanske točke rahlo presežejo idealno premico na začetku projekta, kar pomeni, da smo na začetku projekta v rahli zamudi. Eden izmed glavnih razlogov, zakaj smo v začetku projekta v zamudi, je, ker projektni tim še ni dobro usklajen in je potrebno obdobje prilagajanja ter učenja. Vendar se v nadaljevanju projekta dejanske ure spustijo pod idealnimi urami, kar pomeni, da je delo bilo opravljeno hitreje, kot je bilo ocenjeno.

6.2 Stroški projekta

Skupni strošek z izvajanjem metode Scrum naj bi znašal okoli 124 tisoč evrov. Od tega strošek dela predvideno znaša okoli 112 tisoč evrov in strošek opreme okoli 12 tisoč evrov. Urna postavka Scrum skrbnika in zunanega razvijalca je 100 evrov, produktnega lastnika 80 evrov, urna postavka prvega razvijalca znaša 70 evrov na uro, drugega razvijalca 60 evrov na uro. Najcenejša aktivnost je nabava strojne in programske opreme, kjer je v strošek všteta samo cena opreme. Najdražja aktivnost je konfiguracija čelnega ter zalednega dela aplikacije, grafičnega vmesnika med vsemi paketi, kar je smiselno glede na to, da je potrebno obstoječo programsko rešitev, aplikacijo prilagodi potrebam zaposlenim, končnim uporabnikom in vse izdane verzije rešitve sestaviti v smiselno celoto. Stroški projekta se nahajajo v tabeli 18.

Tabela 18: Stroški projekta

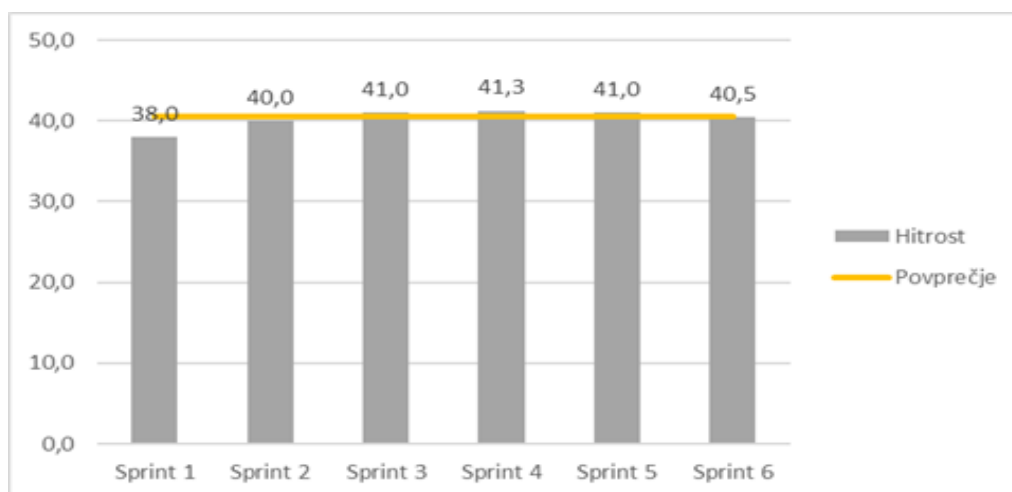
Aktivnost	Strošek
Konfiguracija CRM paketa	12.180 €
Konfiguracija rezervacijskega paketa	8.070 €
Konfiguracija paketa prevzema in vrnitve vozila	5.880 €
Konfiguracija vodenja flote vozila	6.900 €
Konfiguracija vodenja popravila vozila	7.400 €
Konfiguracija računovodskega paketa	6.490 €
Konfiguracija plačilnega paketa	5.700 €
Konfiguracija front, back enda, grafični vmesnik med vsemi paketi	20.240 €
Urejanje, optimiziranje podatkovne baze	7.300 €
Nabava strežnika in uporabniške opreme	1.590 €
Namestitev strojne opreme	5.910 €
Namestitev programske opreme	3.650 €
Konfiguracija strežnika s cloud serverjem	1.950 €
Varnostno kopiranje starih datotek, dokumentov	1.890 €
Testiranje novega sistema	12.560 €
Daily scrum	4.050 €
Nakup programske opreme	1.500 €
Strošek dela	111.760 €
Strošek opreme	12.223 €
Skupni strošek	123.983 €

Vir: lastno delo.

6.3 Hitrost poteka

Hitrost poteka (angl. Velocity) je vsota uporabniških točk, ki poda celotno delo posameznega sprinta. Graf hitrosti poteka prikazuje vrednosti, ki so bile opravljene oz. dostavljene pri vsakem sprintu. Na podlagi zadnjih treh opravljenih sprintih, smo lahko določili, koliko uporabniških točk smo opravili v posameznem sprintu. S tem nam je omogočeno planiranje nadaljnjega dela za prihodnje sprinte. To tudi zajema, kako se je tim soočala z bolj kompleksnimi ali ponavljajočimi opravili. Z vsakim sprintom se tudi razvojni tim nekaj nauči in to znanje pripomore k hitrejšemu in učinkovitejšemu opravljanju nadaljnjih sprintov (Drumond, 2013). Iz slike 16 je razvidno, kje in kdaj je projekt najbolj kompleksen za projektni tim. V šestih sprintih se postavi celotna infrastruktura poslovanja, kar zajema aplikacijo, sistem za vozni park ter računovodski in plačilni sistem. Pri vsakem sprintu se izvaja tudi testiranje posamezne komponente.

Slika 16: Hitrost poteka po šestih sprintih



Vir: lastno delo.

6.4 Napoved

Uporabili smo grafikon večanja (angl. Burn up chart) za spremljanje napredka projekta in celotnega tima. Z grafikonom smo razvili model napovedi, ki nam je pomagal pri nadaljnjih odločitvah ali iteracijah, ter v kakšno smer projekt mora iti. Princip delovanja temelji na grafikonu ameriške agencije NOAA za napovedovanje orkanov v ZDA (Frohman, 2016). Grafikon je bilo treba posodablјati po vsaki iteraciji zato, da smo imeli vedno najnovejšo napoved. V njem je zajeta količina opravljenega celotnega dela v nekem časovnem okvirju. Opravljeno delo lahko predstavljajo uporabniške točke na posamezen sprint. V tabeli 19 je prikazana napoved ob končanem tretjem sprintu. Ko smo zaključili tretjo iteracijo, smo želeli izvedeti, kakšna je napoved za celoten projekt. Zajeli smo zadnje tri končane sprinte, kjer so se izvajale kompleksne konfiguracije sistema najema vozil. Tabela vsebuje vse opravljene in planirane sprinte, njihovo točkovanje v uporabniških točkah in kumulativo točk. S

preprostim izračunom smo dobili povprečje točk zadnjih treh opravljenih sprintov. Ker napoved ni zanesljiva oziroma »stoodstotna«, in grafikon temelji na modelu napovedovanja vremena, je vedno treba zajeti najnovejše podatke in se zato vrednost povprečja treh sprintov uporabi kot pravilo palca, s katerim želimo izvedeti, kako tim opravlja svoje delo, ali je produkt projekta pravilno narejen, ali se projektni tim lahko uči med tekom projekta in kakšni so odnosi znotraj tima. Ker pa gre za manjšo razvojni tim, tega aspekta nismo pokrivali.

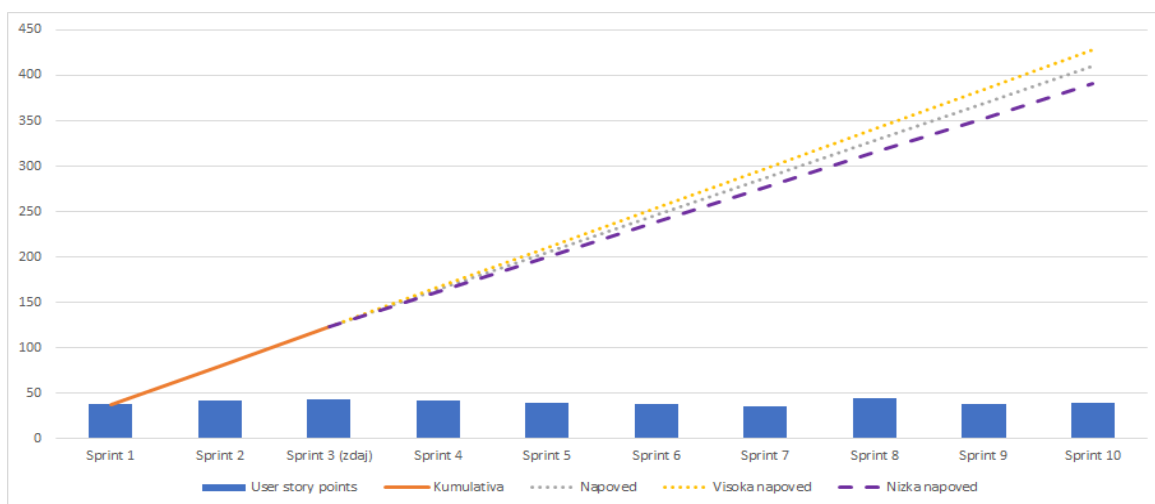
Tabela 19: Napoved po opravljenem tretjem sprintu

Sprint	Uporabniške točke	Kumulativa	Napoved	Visoka napoved	Nizka napoved
Sprint 1	38	38			
Sprint 2	42	80			
Sprint 3 (zdaj)	43	123	123	123	123
Sprint 4	42	165	164	166,64	161,35
Sprint 5	40	205	205	210,29	199,70
Sprint 6	38	243	246	253,93	238,06
Sprint 7	36	279	287	297,58	276,41
Sprint 8	44	323	328	341,22	314,77
Sprint 9	38	361	369	384,87	353,12
Sprint 10	40	401	410	428,52	391,47
Skupaj	401				
Povprečje Optimistično	40,1				
Povprečna hitrost	41				
Standardna deviacija	15,56	2,64			

Vir: lastno delo.

Na sliki 17 so na vertikalni osi uporabniške točke, na horizontalni osi pa vsi opravljeni in planirani sprinti. Oranžna črta je kumulativa uporabniških točk, opravljenih v treh sprintih, siva črta pa predstavlja napovedano količino uporabniških točk, ki bi jih opravili, če bi nadaljevali z isto povprečno hitrostjo kot v prejšnjih sprintih. Vrednost standardnega odklona za zadnje tri sprinte nam je pomagala določiti zgornje in spodnje meje napovedi. V stolpcih visoke in nizke napovedi so navedene vrednosti enake napovedani vrednosti z upoštevanim standardnim odklonom, kjer je odstopanje aditivno, saj se napoved razteza v prihodnost. Prikazuje, koliko dela bo dokončano v nekem časovnem obdobju. Po tretjem sprintu lahko iz napovedi sklepamo, da bo tim lahko opravila od 391 do 428 uporabniških točk do sprinta 10.

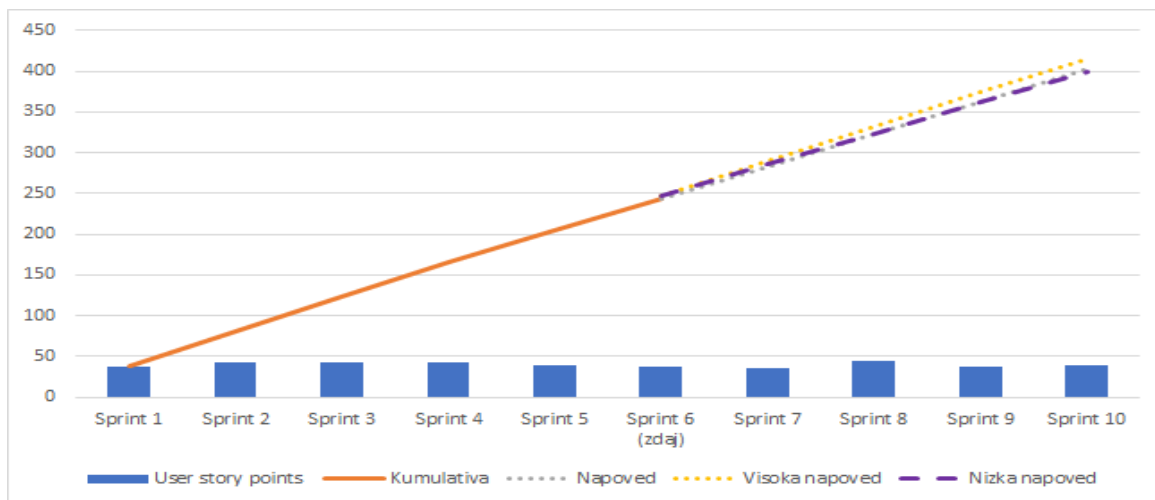
Slika 17: Grafikon naraščanja po treh sprintih



Vir: lastno delo.

Priporočljivo je preverjati napoved po vsaki iteraciji. V sliki 18 je prikazana napoved po končanem šestem sprintu. Kot pri tretjem sprintu smo v šestem upoštevali povprečje uporabniških točk zadnjih treh končanih sprintov. Razlika zgornje in spodnje meje napovedi sta se drastično zožile. Standardni odklon se je zmanjšal z 2,65 na 2, kar pomeni boljši model napovedi. Tudi povprečna hitrost se je povečala iz 40 na 41. Po napovedi lahko predvidevamo, da je projektni tim postal bolj utečen in hitrejši pri izvajanju samih sprintov.

Slika 18: Grafikon naraščanja po šestih sprintih



Vir: lastno delo.

7 UPORABA KANBAN-A NA PODJETJU RENTAVAN

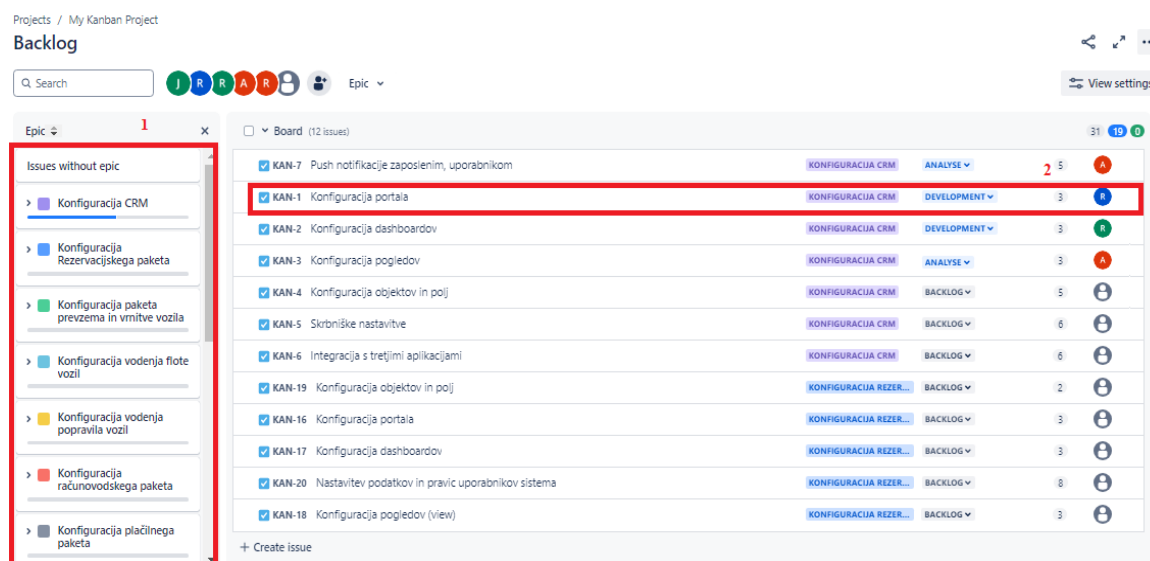
7.1 Kanban tabla

Učinkoviti timi se morajo nenehno odzivati na dogodke skupaj in usklajevati njihove dejavnosti. V hitro razvijajočem okolju se tradicionalni sestanki o stanju projekta monotona, preobsežna z nepopolnimi, netočnimi informacijami. Želimo, da so informacije posodobljene, enostavno dostopne vsem zainteresiranim stranem v času, ko je to potrebno. Zato smo se odločili, da bo Kanban tabla predstavljena virtualno. Univerzalno dostopna in zato v veliko pomoč članom tima, ker niso vsi locirani skupaj. Da bi imeli še fizično tablo, se nam ne zdi smiselno, ker bi morali posodabljati obe tabli hkrati in ker pri digitalni tabli obstaja zelo majhna možnost izgube podatkov, medtem ko lahko s fizične table kartice lahko padejo s table. Virtualna tabla omogoča tudi samodejni izračun meritev, ter lahko shranjujejo informacije in razprave o posameznem elementu, nalogi.

Orodje JIRA omogoča podroben pogled seznama uporabniških zgodb in trenutnega stanja projekta. Seznam uporabniških zgodb je bil narejen na podlagi intervjujev z zaposlenimi in strankami. Zaposleni so podajali mnenja, kaj bi bilo potrebno spremeniti pri že obstoječem poslovanju oziroma kako bi te pozitivne spremembe vključili v novejši sistem. Stranke so podajale mnenja, kaj bi bilo treba spremeniti v postopku najema, oziroma kako bi omogočili enostavnejši postopek najema vozila. Vse zahteve so bile nato vključene v uporabniške zgodbe, ki so omogočile nadaljevanje projekta.

Slika 19 prikazuje seznam uporabniških zgodb. V levem delu so prikazane uporabniške zgodbe ter njihova stopnja dokončanja (1). V desnem delu so prikazani delovni elementi ki se trenutno opravljajo ali se bodo v bodoče, njihova šifra, del katere uporabniške zgodbe so, v kateri aktivnosti se trenutno nahajajo, koliko uporabniških točk imajo in kdo jih opravlja. Delovni element konfiguracija portala, ki je del konfiguracije CRM-ja, je v aktivnosti razvoj (angl. development), ima 3 uporabniške točke in ga razvija prvi programer (2).

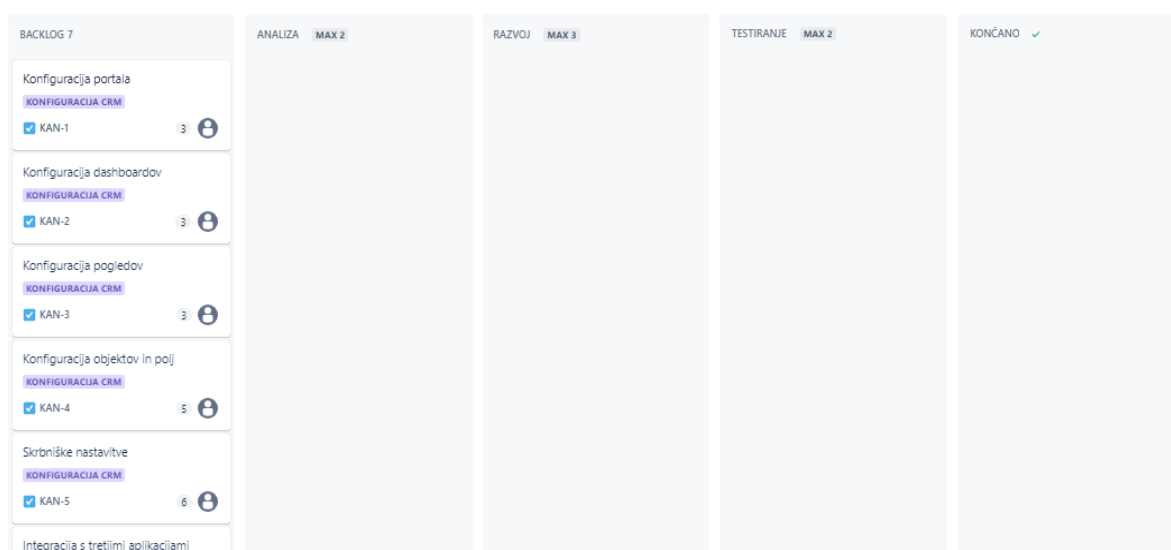
Slika 19: Seznam uporabniških zgodb



Vir: lastno delo.

Slika 20 prikazuje Kanban tablo. Tabla je razdeljena na pet zavihkov: seznam uporabniških zgodb, analiza, v razvoju, testiranje, končano. V zavihku seznam uporabniških zgodb so postavljene kartice, ki se bodo morale izvajati med projektom. Te kartice predstavljajo elementi dela, ki bodo opravljeni. Kartice vsebujejo osnovne podatke delovnega elementa in del, katere uporabniške zgodbe je, na primer: Konfiguracija portala, ki je del uporabniške zgodbe konfiguracija CRM-ja. V našem primeru so v seznamu uporabniške zgodbe že razdeljene na posamezne delovne elemente.

Slika 20: Kanban tabla



Vir: lastno delo.

Velikost posameznega elementa dela je ocenjena na podlagi uporabniških zgodb in je razdeljena na tri velikosti: majhno (angl. Small, v nadaljevanju S), srednjo (angl. Medium, v nadaljevanju M), veliko (angl. Large, v nadaljevanju L). Sistem ocenjevanja je preprost. To je relativna ocena velikosti delovnega elementa. Večje število vrednosti kaže na večjo natančnost v naših ocenah in celotna ideja je samo navesti relativne velikosti dela. Kot pri uporabniških točkah je prednost relativnih ocen o tem, da lahko sčasoma začnemo delati napovedi na podlagi uspešnosti tima. Če tim potrebuje šest dni za majhno aktivnost, konfiguracijo, ter smo za njeno izpolnitev potrebovali okoli sedem delovnih nalog, potem naslednja konfiguracija, ki je bila ocenjena z isto vrednostjo in ima podobno število nalog, bo isto trajala okoli šest dni.

Bolj obsežna je uporabniška zgodba, večjo je število nalog, ki je potrebnih za njeno izpolnitev. Kartice vizualno predstavijo, kdo bo sodeloval pri določeni uporabniški zgodbi, ker se s tem želimo preprečiti prekomerno nalaganje zadolžitev na posameznega člana tima in ker vsak član ima različen nivo znanja. V projektnem timu imamo štiri člane. Analitik skrbi za analizo projekta, dva razvijalca skrbita za razvoj aplikacije in enega testerja, ki preverja, ali so bile dodane spremembe vplivale na delovanje aplikacije.

Pred samim začetkom projekta se člani projekta seznanijo s strukturo table, z omejitvami dela posamezne aktivnosti. Vsaka kartica ima naslednje lastnosti: za kateri delovni element gre, del katere uporabniške zgodbe je, kdo ga opravlja, kakšen je status elementa v teku (angl. doing), končano (angl. done), koliko ima dodeljenih uporabniških točk.

7.2 Omejitve števila opravil

Omejitve števila opravil (angl. Work in progress, v nadaljevanju WIP) v posameznem stolpcu so postavljene na podlagi Brechnerjevega pristopa določanja omejitve dela. WIP se določi na podlagi najpočasnejše aktivnosti in prilagajanju ostalih aktivnosti na le to. Najpočasnejši aktivnosti se doda 50 % zaloga (glede na število oseb), kar omogoča, da so člani tima vedno zasedeni. Tako analitik kot tudi tester lahko okvirno opravita po petnajst delovnih elementov na mesec, medtem ko lahko razvijalec opravi sedem delovnih elementov. Glede na to, da imamo dva razvijalca, se lahko v aktivnosti razvoj opravi štirinajst delovnih elementov na mesec. Iz tega sledi, da je najpočasnejša aktivnost razvoj. Tako da ni potrebe po zaposlovanju dodatnega števila analitikov, testerjev, ker pri nas oba imata večjo zmožnost kot oba razvijalca skupaj. Število delovnih elementov na mesec je bilo predpostavljeno, glede na to, da je rešitev, ki jo bomo implementirali že narejena in je potrebno samo spreminjati, dodajati določene elemente sami aplikaciji. Zato je tudi mesečno število delovnih elementov večje in sam razvoj aplikacije poteka hitreje, kot bi potekal, če bi morali razvoj začeti od samega začetka, koncepta rešitve. Ko je določena količina dela, se vsakemu članu tima doda 50 % (k številu oseb) in zaokrožimo navzgor. Na primer imamo enega analitika, ga pomnožimo z 0,5 in zaokrožimo, dobimo 2. Tako je lahko največje število elementov dela, ki se nahaja v stolpcu analiza dva. Če uporabimo isto tehniko

določanja omejitev, pri ostalih stolpcih imata lahko razvijalca največ tri, tester pa dva delovna elementa. Aktivnosti analiza, razvoj, testiranje imajo določene omejitve WIP, in sicer 2, 3, 2.

7.3 Potek metode

Stoječi sestanek dan 1:

Slika 21 prikazuje delovni element Kan-1, ki ga je analist povlekel iz seznama uporabniških zgodb na začetku prvega dne.

Slika 21: Tabla na začetku prvega dne

SEZNAM UPORABNIŠKIH ZGODB 8	ANALIZA 2 MAX 2	RAZVOJ 2 MAX 3	TESTIRANJE MAX 2	KONČANO ✓
Konfiguracija dashboardov KONFIGURACIJA CRM ☒ KAN-2 3	Konfiguracija portala KONFIGURACIJA CRM Doing ☒ KAN-1 3			
Konfiguracija pogledov KONFIGURACIJA CRM ☒ KAN-3 3				

Vir: lastno delo.

Stoječi sestanek dan 2 (prikazan na sliki 22):

Za analizo elementa Kan-1 je analitik v prvem dnevu potreboval 2 uri. Nato ga je prvi razvijalec potegnil v aktivnost Razvoj. Kar je omogočilo, da je analitik potegnil Kan-2 iz seznama uporabniških zgodb v aktivnost Analiza. Ta element je dokončal v 3 urah, katerega je drugi razvijalec povlekel v aktivnost Razvoj. Tako je lahko analist povlekel nov element Kan-3, ki ga je dokončal v 1 uri. Po končanju Kan-3 je analist iz seznama uporabniških zgodb v aktivnost Analiza povlekel Kan-4, ki ga je analiziral v 2 urah.

Slika 22: Tabla na začetku drugega dne

SEZNAM UPORABNIŠKIH ZGODB 8	ANALIZA 2 MAX 2	RAZVOJ 2 MAX 3	TESTIRANJE MAX 2	KONČANO ✓
Skrbniške nastavitve KONFIGURACIJA CRM ☒ KAN-5 6	Konfiguracija pogledov KONFIGURACIJA CRM Done ☒ KAN-3 3	Konfiguracija portala KONFIGURACIJA CRM Doing ☒ KAN-1 3		
Integracija s tretjimi aplikacijami KONFIGURACIJA CRM ☒ KAN-6 6	Konfiguracija objektov in polj KONFIGURACIJA CRM Done ☒ KAN-4 5	Konfiguracija dashboardov KONFIGURACIJA CRM Doing ☒ KAN-2 3		

Vir: lastno delo.

Stoječi sestanek dan 3 (prikazan na sliki 23):

Na stoječem sestanku 3. dne smo ugotovili, da v drugem dnevu ni bilo sprememb na tabli, ker se delo v razvoju še zmeraj opravlja. Analitik ne more vzeti novih kartic, zaradi dosega omejitve dela in ker sta oba razvijalca še zmeraj zasedena.

Slika 23: Tabla na začetku tretjega dne

SEZNAM UPORABNIŠKIH ZGODB 8	ANALIZA 2 MAX 2	RAZVOJ 2 MAX 3	TESTIRANJE MAX 2	KONČANO ✓
Skrbniške nastavitve KONFIGURACIJA CRM ✓ KAN-5 6	Konfiguracija pogledov KONFIGURACIJA CRM Done ✓ KAN-3 3 A	Konfiguracija portala KONFIGURACIJA CRM Doing ✓ KAN-1 3 R		
Integracija s tretjimi aplikacijami KONFIGURACIJA CRM ✓ KAN-6 6	Konfiguracija objektov in polj KONFIGURACIJA CRM Done ✓ KAN-4 5 A	Konfiguracija dashboardov KONFIGURACIJA CRM Doing ✓ KAN-2 3 R		

Vir: lastno delo.

Stoječi sestanek dan 4 (prikazan na sliki 24):

Tim je na začetku četrtega dne povlekla delovni element Kan-1 v aktivnost Končano. Razvoj delovnega elementa Kan-1 je bil tretji dan dokončan, zato ga je tester potegnil v aktivnost Testiranje in ga v istem dnevu dokončal. Nato je lahko tester iz razvoja vzel element Kan-2. V aktivnosti Razvoj prvi razvijalec razvija Kan-3, drugi razvijalec pa Kan-4. V aktivnosti Analize je bilo prostora za dva elementa, zato je analitik iz seznama uporabniških zgodb potegnil Kan-5 po končanju le tega pa Kan-6.

Slika 24: Tabla na začetku četrtega dne

SEZNAM UPORABNIŠKIH ZGODB 6	ANALIZA 2 MAX 2	RAZVOJ 2 MAX 3	TESTIRANJE MAX 3	KONČANO 1 ✓
Integracija s tretjimi aplikacijami KONFIGURACIJA CRM ✓ KAN-7 6	Push notifikacije zaposlenim, uporabnikom KONFIGURACIJA CRM Done ✓ KAN-6 5 A	Konfiguracija pogledov KONFIGURACIJA CRM Doing ✓ KAN-3 3 R	Konfiguracija dashboardov KONFIGURACIJA CRM Doing ✓ KAN-2 3 TT	Konfiguracija portala KONFIGURACIJA CRM Done ✓ KAN-1 3
Konfiguracija portala KONFIGURACIJA REZERVACIJSKEGA PAK... ✓ KAN-8 3	Skrbniške nastavitve KONFIGURACIJA CRM Done ✓ KAN-5 6 A	Konfiguracija objektov in polj KONFIGURACIJA CRM Doing ✓ KAN-4 5 R		
Konfiguracija objektov in polj				

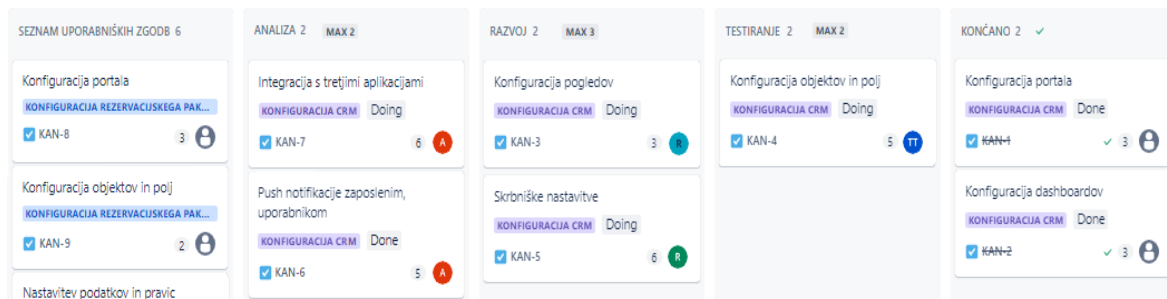
Vir: lastno delo.

Stoječi sestanek dan 5 (prikazan na sliki 25):

Testiranje delovnega elementa Kan-2 je bilo končano, zato je bil potegnjjen v aktivnost Končano. Razvoj elementa Kan-4 je končan, zato ga je tester potegnil v aktivnost Testiranje, medtem ko razvoj Kan-3 še ni končan, zato ostane v Razvoju. Posledično drugi razvijalec

povleče element Kan-5 v aktivnost Razvoj, kar omogoči, da analitik iz seznama uporabniških zgodb potegne delovni element Kan-7 v Analizo.

Slika 25: Tabla na začetku petega dne

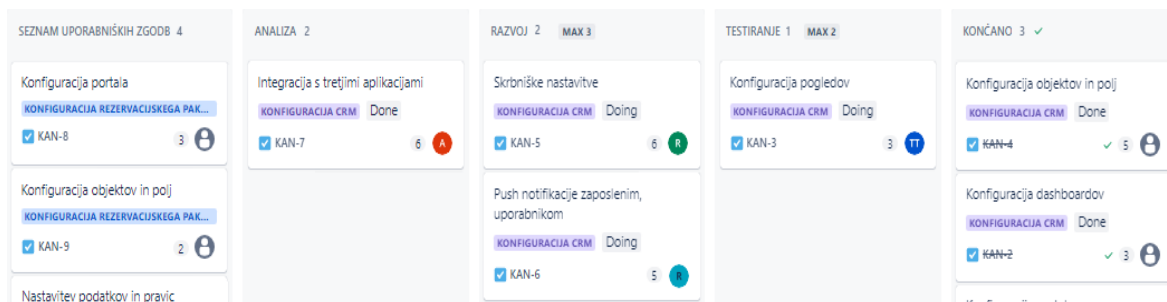


Vir: lastno delo.

Stoječi sestanek dan 6 (prikazan na sliki 26):

Testiranje elementa Kan-4 je bilo končano in potegnjeno v aktivnost Končano. Razvoj elementa Kan-3 je bil končan, zato ga je tester potegnil v Testiranje. To je omogočilo prvemu razvijalcu, da je iz Analize potegnil Kan-6, medtem drugi razvijalec še vedno dela na Kan-5. Sprostitve analize je omogočilo analistu, da je iz seznama uporabniških zgodb vzel nov delovni element Kan-8.

Slika 26: Tabla na začetku šestega dne



Vir: lastno delo.

7.4 Nadzor nad metodo

Vsakodnevno potekajo kratki 15-minutni sestanki, kjer si tim ogleduje trenutno stanje Kanban table in določijo načrt izvajanja delovnega elementa za tisti dan. Sestanek se osredotoča na tok, torej začne se z branjem table od desne proti levi, debatirajo pri posameznih elementih, ki so mogoče blokirani in zagotovijo, da bodo vsi delali, tako da bodo se stvari čim prej premaknile proti koncu, ob upoštevanju zaključenega dela na posameznem koraku in postavljenih omejitev na naslednjem koraku. Ko pridejo na skrajno levo stran table, preverijo, ali je na voljo prostor za jemanje novih aktivnosti. Ni pomembno, kaj je kdo naredil in kaj namerava storiti. Namen je spoznati, kateri delovni elementi so v

teku in kaj se lahko naredi, da se bodo hitreje premaknili proti koncu table. Motiv je opazovati in spremljati stanje dela, opazovati potek dela. Sestanki pridejo do izraza pri na primer konfiguraciji, montaži strojni opremi, senzorjev telematike v vozila, ki poteka hkrati z namestitvijo programske opreme le teh in je nujno treba izvajati sestanke za boljše usklajevanje projektnega tima in pregleda nad delom.

Tabela 20 prikazuje tabelo končanih zgodb v različnih aktivnostih za prvih deset dni projekta. Vsak stolpec prikazuje seštevek vseh opravljenih zgodb za posamezno aktivnost. Pod stolpcem Seznam je prikazana, količina vseh uporabniških zgodb, ki morajo biti opravljene za uspešno dokončanje projekta. Na prvi dan projekta so vrednosti v stolpcih aktivnosti ničelne vrednosti, saj ni bil opravljen noben delovni element. Iz tabele je razvidno, da je na primer do petega dne projekta še ostalo petinosemdeset delovnih elementov, analist do tistega dne opravil šest elementov, razvijalca tri in tester dva elementa. Ob začetku petega dne projekta sta bila tako dokončana dva delovna elementa.

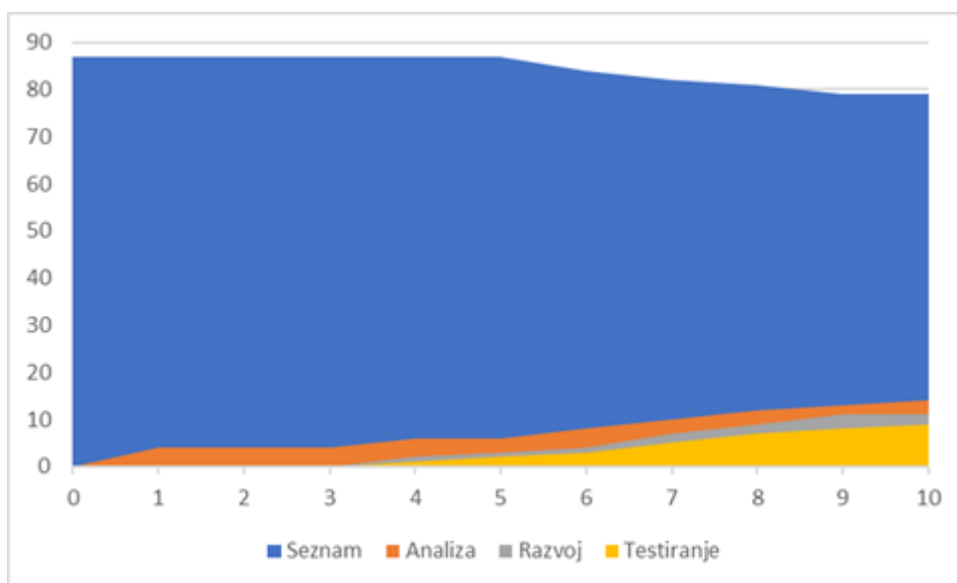
Tabela 9: Število zgodb v različnih aktivnostih

Dan	Seznam	Analiza	Razvoj	Testiranje	Končano
1	87	0	0	0	0
2	87	4	0	0	0
3	87	4	0	0	0
4	86	6	2	1	1
5	85	6	3	2	2
6	84	7	4	3	3
7	82	11	7	5	5
8	81	12	9	7	6
9	79	13	11	8	8
10	79	14	11	9	8

Vir: lastno delo.

Kumulativni graf (angl. Cumulative Flow Diagram, v nadaljevanju CFD) poteka je eno najbolj uporabljenih in praktičnih orodij za vizualiziranje treh pomembnih komponent ali metrik Kanban metode. Prikazuje čas cikla, prepustnost in delo v teku. Namen grafa je prikaz stabilnosti toka in razumevanje, kje se je treba osredotočiti, da bo proces bolj predvidljiv. Tim lahko ugotovi, kje so nastajala ozka grla in ali je bil način reševanja primeren. Graf se uporablja tudi pri ponovni določitvi WIP omejitev tekom projekta, če je to potrebno (Naydenov, 2021). Iz slike 27 je grafično razvidno, da se potek dela opravljanja delovnih elementov po posameznih aktivnostih vsak dan projekta konstantno povečuje, kar nakazuje stabilen pretok dela. To prikazujejo enakomerni razmiki med krivuljami aktivnosti, saj bi v nasprotnem primeru to signaliziralo na potencialna ozka grla oziroma težave pri izvajanju poteka projekta.

Slika 27: Kumulativni graf poteka



Vir: lastno delo.

7.5 Trajanje projekta

Projekt implementacije programske rešitve z uporabo Kanban metode naj bi okvirno potekal od 10. 10. 2022 do 6.3. 2023 in trajala 148 dni. Časovnica projekta je bila določena po 22 dnevih projekta. Projekt je sestavljen iz šestnajstih večjih uporabniških zgodb, ki so nato razdeljene na več manjših uporabniških zgodb. Skupno število vseh manjših uporabniških zgod, ki naj bi bile dokončane tekom projekta je 87.

Enostavnejši elementi dela S velikosti so: konfiguracija paketa prevzema, vrnitve vozila, nabava strežnika uporabniške opreme, namestitev strojne opreme, namestitev programske opreme, konfiguracija strežnika z oblaknim serverjem, varnostno kopiranje starih datotek, dokumentov, prenos starih datotek na nov sistem, učenje uporabnikov.

Bolj kompleksni elementi dela M velikosti so: konfiguracija CRM paketa, konfiguracija rezervacijskega paketa, konfiguracija vodenja flote vozil, konfiguracija vodenja popravila vozila, konfiguracija računovodskega paketa, konfiguracija plačilnega sistema, kreiranje optimiziranje podatkovne baze.

Najobsežnejši delovni elementi L velikosti so konfiguracija čelnega dela aplikacije, zalednega dela aplikacije, grafični vmesnik, uvajanje novega sistema. Velikosti so dodeljene na podlagi zahtevnosti, časa, ki so potrebni, da se določena aktivnost konča.

7.6 Stroški projekta

Skupni strošek implementacije programske rešitve predvideno znaša okoli 127 tisoč evrov. Od tega znaša predvideni strošek dela okoli 115 tisoč evrov in strošek opreme okoli 12 tisoč evrov. Strošek opreme vsebuje stroške delovne postaje z vključenim operacijskim sistemom, napajalnik UPS, tablične računalnike, strežnik, usmerjevalnik, senzorje za telematiko in avto kamere s pripadajočimi spominskimi karticami. Urna postavka analista je 80 €, enega razvijalca 70 €, drugega razvijalca 60 €, testerja 100 €.

Od agilnih metod je v našem primeru Kanban dražji, vendar dejanski pretok dela poteka bolj tekoče kot pri Scrum-u, ker je delo bolj razčlenjeno. Večja je sposobnost odzivanja na nenadne spremembe, če se pojavi nova uporabniška zgodba in ima tako manjši vpliv na sam potek metode. Metoda je oblikovana tako, da je tester prisoten do konca projekta, v primerjavi z ostalimi metodami, kjer tester (v ostalih metodah testiranje opravlja zunanji izvajalec) opravlja delo, ki se tiče samega razvoja aplikacije in konča sodelovanje s projektnim timom na sredini projekta.

8 PRIMERJAVA TRADICIONALNEGA Z AGILNIM PROJEKTNIM MANAGEMENTOM

Po mnenju Singh (2021) je tradicionalna metoda najbolj primerna za manjše ali manj zapletene projekte, saj sledi linearnemu pristopu. Nenadne spremembe v projektu ali katerekoli drugi zapleti, lahko blokirajo celoten postopek in prisilijo tim, da se vrne na začetek koraka in začne znova. Agilnost je najboljša metoda, ki jo je treba uporabiti v primeru zapletenih projektov. Zapleten projekt ima lahko različne medsebojno povezane faze in vsaka faza je lahko odvisna od mnogih drugih in ne od ene same, kot pri preprostih projektih. Zato so za večje zapletene projekte agilne metode bolj primerne, saj se na takšno strukturo projekta bolj odzovejo. Tradicionalni pristop deluje s prepričanjem, da ko je faza končana, je ne bomo več pregledovali. Torej ni prilagodljiv hitrim spremembam delovnega načrta. V primeru, da se pojavi nenadna situacija ali kakšna sprememba zahtev s strani stranke, se tradicionalni pristop ne spremeni. Edina izbira je to, da se vse začne znova. To zapravi mnogo truda in časa. Pri agilnem pristopu je faktor prilagodljivosti zelo visok, saj ni linearen. Kompleksni projekti so sestavljeni iz več med seboj povezanih stopenj, kjer lahko sprememba v eni fazi vpliva na drugo. Managerji projektov lahko v takšnem scenariju vključijo tudi potencialno tveganje, saj obstaja velika možnost prilagodljivosti (Singh, 2021).

Pri tradicionalni metodi je vsak postopek na začetku projekta jasno podrobno opredeljen in definiran. Ne more se spoprijeti z nobenimi velikimi spremembami ali povratnimi informacijami, ki bi lahko zahtevale spremembe v postopku. Čas in proračun za izvedbo projekta sta večinoma določena, kar omogoča le redke spremembe. Pri agilnem pristopu je veliko sprejemanja povratnih informacij in sprememb. Postopek je zelo prilagodljiv in

omogoča stalne povratne informacije, ki lahko pomagajo do boljšega učinka v določenem času izvajanja projekta (Narasimman, 2024).

Kot pri vsaki drugi metodi, tudi agilna ni primerna za vsak projekt, zato se vedno priporoča dovolj velika skrbnost, da se določi najboljša metoda za vsako edinstveno situacijo. Agilnost morda ne deluje, kot je bilo predvideno, če ni stranki in managerju projekta jasno glede ciljev, tim je neizkušen ali ne deluje dobro pod močnim pritiskom. V celotnem razvojnem procesu je agilnost, naklonjena razvijalcem, projektnim skupinam in ciljem strank, ne pa nujno tudi končnim uporabnikom. Zaradi manj formalnih in fleksibilnejših procesov agilnost ne bo vedno zlahka absorbirana v večjih, bolj tradicionalnih organizacijah, kjer je v procesih, politikah ali skupinah, značilna večja togost ali fleksibilnost. Lahko se sooči tudi s težavami pri uporabi samih odjemalcev, ki imajo podobno toge procese ali načine delovanja (Alexander, 2023).

Agilnost vključuje različne metode in okvirje za management projektov. Te metode so popolna izbira za preprečevanje nepotrebne porabe časa, povečanje zadovoljstva stranke in spodbujanje odločanja na vsakem koraku razvoja izdelka. Sprva so bile namenjene za industrijo programske opreme, vendar se v zadnjem času uporabljajo tudi v drugih panogah, kot so arhitektura, trženje, finančne storitve.

V nadaljevanju bo podanih nekaj razlogov in vidikov, zakaj se vse več projektnih managerjev in razvijalcev odloča za uporabo agilnih metod.

Fleksibilnost:

Tradicionalni management projektov ponuja malo sprememb v izdelku. To je postopek, ki sledi samo pristopu od zgoraj navzdol. Ko je načrt dokončan, ga managerji sporočijo svojim skupinam in poskrbijo, da se ga bodo vsi držali na najboljši možni način. Veliko je odpora proti kakršnim koli spremembam, ki lahko porušijo časovni načrt projekta. Agilna metoda je bolj prilagodljiva in ponuja večjo ter hitrejšo odzivnost pri spreminjanju izdelka. Članom tima omogoča eksperimentiranje in iskanje nekaterih najboljših možnosti. Prosto sporočijo katerokoli idejo, za katero verjamejo, da bo pripomogla k nadaljnjemu izboljšanju izdelka. Ker se agilnost osredotoča na celostni pristop, se osredotoča, da se najde pravi izdelek, kot da se sledi togi strukturi (Singh, 2021).

Lastništvo in preglednost:

V tradicionalnem managementu lastništvo pripada managerju projekta. Odgovornost managerja je, da načrtuje in dokumentira celotno pot izdelka. Poleg managerjev in nekaterih, s strani projektnega managerja povabljenih članov tima, so v fazi načrtovanja vključene tudi stranke, ko pa se začnejo faze implementacije, se vpletenost strank zaključi. Ker imajo managerji vse vaje projekta v svojih rokah, člani tima po navadi nimajo besede o rezultatu

svojih prizadevanj ali o tem, kako projekt napreduje.

Medtem si pri agilni metodi člani skupine delijo lastništvo projekta. Skupaj sodelujejo in pripravijo načrt zaključitve vseh del ob predvidenem času ter stroških. Napredovanje izdelka vidijo že od samega začetka pa vse do konca projekta. Takšna preglednost ima pomembno vlogo pri ohranjanju produktivnega in zelo angažiranega delovnega mesta.

Reševanje težav:

V primeru nepričakovanih ovir morajo posamezniki ta vprašanja prenesti na svoje managerje. Vendar to ni vedno mogoče. To lahko povzroči neupravičene zamude in preseže predvideno časovno omejitev, poleg tega pa poveča skupne stroške.

Agilni timi imajo to pristojnost, da same odločajo. Vsa vprašanja poskušajo rešiti znotraj tima, da ne bi zapravili preveč časa. Ker so tesno vključeni v postopek, jim znanje pomaga pri reševanju večine težav, ki ovirajo njihov napredek. Razen, če ni treba sprejemati skrajnih odločitev, člani tima redko prestavijo svoje trivialne zadeve na svojega managerja (Narasimman, 2024).

Kontrolne točke in spremljanje napredka:

Tradicionalne metode zagovarjajo detajlno načrtovanje v fazi analize in načrtovanju projekta. Njihov poudarek je bolj na racionalizaciji procesov kot na samem izdelku. Ko je postopek končan, se pričakuje, da mu bo tim sledil, korak za korakom z minimalnimi napotki. Napredek je določen po zaključku postopka. Ni pogostega preverjanja.

Ker obstajajo krajše in hitrejše iteracije, agilna metoda spodbuja člane tima, da imajo kontrolne točke v rednih presledkih. Napredek je enostavno določiti in je lažje pomagati posameznikom, da ohranijo odgovornost pri svojem delu. Pri agilnih metodah se timi vsak dan sestanejo, da ugotovijo, kaj je bilo opravljeno včeraj, kakšen je dnevni red za današnji dan in če obstajajo kakršnekoli ovire (Pawar, 2017).

Izbira:

Glavni razlog, da managerji ali razvijalci izberejo agilno usmeritev, je v prilagodljivosti, ki jo ponuja. Razvijalci, ki delujejo po načelih agilnega managementa, se lahko hitro odzovejo na zahteve strank, saj naenkrat naslavlajo le manjše dele projekta in kupec pred dokončanjem potrdi vsako ponovitev ali sprint.

Obstaja možnost, da se prilagodi drugim metodam, kot je metoda slap (angl. waterfall), da se ustvari hibridna rešitev. Model slap je postopek za razvoj programske opreme. Ker sledi logičnemu zaporedju korakov skozi celoten življenjski cikel razvoja programske opreme,

podobno kot se voda spušča po slapu navzdol, je od tod dobila metoda ime slap (Chi, 2018). Podjetja včasih uporabijo slap tehniko za obvladovanje ene ali več faz, na primer načrtovanje, kadar ti ne zahtevajo hitrih ali ponavljajočih se korakov. Zlasti načrtovanje zahteva celovit, metodičen, pogosto počasnejši pristop k definiranju, analiziranju in dokumentiranju vidikov projekta. Po ugotovitvah Alexander (2023) je zaradi tega slap tehnika primernejši pristop. Ko projekt vstopi v razvojno fazo, hitre in ponavljajoče se spremembe zahtevajo drugačen pristop in tu se pojavi agilnost, ki v najkrajšem času doseže najboljše rezultate. Ta hibridni pristop pomaga, da agilnost postane še bolj prilagodljiva v različnih panogah ali pa bolj ustreza unikatni naravi projekta, izdelka ali storitve. Ponovno je potrebna skrbnost za določitev ustreznosti in zmogljivosti različnih razpoložljivih metod in procesov. V dinamičnem okolju, v katerem zdaj živimo, je zelo malo situacij, ko se spremembe ne zgodijo. V takem času bo upoštevanje agilnih metod podjetjem služilo veliko bolje kot tradicionalni projektni management (Alexander, 2023).

Funkcijske skupine:

Tradicionalni razvoj, ki ima funkcijske skupine s skupnim strokovnim znanjem oziroma enofunkcijskimi skupinami, zapoznele ali šibke povratne zanke in zaporedni pretok od analize do testa, v današnjem nestabilnem svetu ni zelo uspešen. Takšen pristop povzroča zamude v povratnih informacijah, učenju in potencialne donosnosti naložb zaradi odsotnosti prave delujoče programske opreme, katere funkcionalnosti se pokažejo šele proti koncu projekta, kar povzroči pomanjkanje preglednosti, pomanjkanje sposobnosti za izboljšanje, zmanjšanje fleksibilnosti in povečanje poslovnih in tehničnih tveganj. Druga možnost so medfunkcionalne skupine z iterativnim razvojem. Obstajajo že desetletja, vendar niso bila tako široko uporabljena kot pri tradicionalnem modelu enofunkcijske skupine (Deemer in drugi, 2012).

Agilne metode vsebujejo preizkušene koncepte razvoja izdelkov v preprostem okvirju, ki vključuje: prave time, medfunkcijske skupine, samoupravne time, kratke iterativne povratne zanke v celotnem ciklu in znižuje stroške sprememb. Ti koncepti povečujejo okretnost in povratne informacije, omogočajo zgodnejšo donosnost naložbe in zmanjšujejo tveganje (Deemer in drugi, 2012). Agilni projekt vključuje sodelovanje pri ustvarjanju novega izdelka, storitve ali drugega rezultata, kot je opredeljeno v izjavi o viziji projekta. Na projekte vplivajo omejitve časa, stroškov, obsega, kakovosti, virov, organizacijskih zmognosti in druge omejitve, ki otežujejo načrtovanje, izvajanje, management in na koncu uspejo. Vendar pa uspešna izvedba rezultatov končnega projekta podjetju prinaša pomembne poslovne koristi. Zato je pomembno, da organizacije izberejo in izvajajo ustrezen pristop za management projektov (Pawar, 2017).

Uveljavitev metod:

Obstajajo različne okoliščine, zaradi katerih se agilne metode težko uveljavljajo v številnih

organizacijah. Najpogostejša težava v zvezi s sprejetjem je nerazumevanje številnih agilnih praks in tehnik, katerih ključna komponenta je razumevanje temeljnega razloga, zakaj je treba nekatere prakse izvajati. Več drugih težav se po navadi pojavi zaradi pritiska vodstva, ki išče status ali hitrejšo dostavo. Agilne metode se vodstvu pogosto prodajajo kot cenejše in hitrejša metode za izpolnjevanje nalog, medtem ko je realnost pogosto nasprotna. Agilnost lahko naredi več škode kot koristi, če povzroči nesporazum, zlasti glede stroškov ali dostave. Ta nesporazum na koncu povzroči trenja in kar je še pomembneje, pomanjkanje zaupanja med vodstvom in agilnimi timi (Pawar, 2017). V praksi lahko agilni timi napačno razumejo nekatere prakse, ki so se jih naučili, kar povzroči neučinkovito ali celo napačno uporabo teh metod. Timi morajo zagotoviti, da dosledno razumejo svoje agilne prakse in da postanejo učinkovitejše. To razumevanje skozi celotni tim potrjuje, zakaj je praksa strukturirana tako, kot je, in celo zakaj se šteje za del splošne metode. Ko bodo timi postali bolj dosledni pri zagotavljanju uspešnih izdaj ali minimalno izvedljivih izdelkov, bodo začele graditi globlje odnose z vodstvom. Potem, ko se bo okno zaupanja med razvojnimi timi in vodstvom povečalo, bodo timi bolj dovzetne za eksperimentiranje in bodo bolj verjetno uvajale inovacije, medtem ko bodo še naprej izpolnjevale svoje razumevanje in izvajanje agilnih praks. Poleg tega se bo vodstvo z globljim zaupanjem počutilo bolj udobno, če bo timom omogočilo, da izvajajo svoj agilni proces in bodo tako bolj pripravljeni sprejeti postopno izdajo minimalno izvedljivih izdelkov z zaupanjem, da najboljše šele prihaja. Tukaj se pojavlja vprašanje problema krožnega zaupanja, kjer morajo timi vzpostaviti zaupanje s svojim vodstvom, preden jim vodstvo podeli več zaupanja. Da bi dosegli to stanje zaupanja, je potrebno boljše razumevanje agilnih praks. Pri tem razumevanju ne gre niti za to, kaj storiti niti za to, kako to storiti, temveč gre za resnično razumevanje čemu je agilnost namenjena (Verma, 2018).

Ko se stvari odvijajo po načrtih, tradicionalni management projektov zasije in cilji bodo doseženi brez težav. Vendar v realnosti stvari redko potekajo, kot so bile načrtovane. Ker tradicionalni management temelji na skrbnem načrtovanju in izvrševanju lahko, v primeru težav, projekt obtiči. V takšnih primerih je treba iti izven začrtanega postopka in prilagoditi načrt novim okoliščinam. Treba je tudi zaupati vsakemu članu tim, kaj dela in kaj je sposoben narediti v dodeljenem času. Vse faze procesa se pojavijo v zaporedju, in ko se doseže mejnik, se lahko nadaljnje faze projekta nadaljujejo. Omejena prilagodljivost proračuna in čas izvajanja projekta sta pogosto lahko velik problem. Če projekt potrebuje kakšno spremembo, tudi če gre za manjšo ali manj pomembno spremembo, je lahko zaradi faznega pristopa projekta prilagoditev težavna. Ustvarjanje nove programske opreme mora biti odzivno na tehnološke spremembe, vse težave in ideje razvijalcev ter na nenehno spreminjajoče potrebe strank. Pri tradicionalni metodi lahko končamo z načrtovanim izdelkom, ki pa zaradi tehnoloških novitet in zahtev postane irelevanten (Dehoyos, 2020).

Pomen manifesta:

Javna izjava agilnega manifesta iz leta 2001 je bistveno spremenila način, kako so mnogi

pristopili k razvoju programske opreme in je temelj za vse različice agilnega razvoja programske opreme. Ideali manifesta so namenjeni oblikovanju delovanja skupin za dostavo programske opreme. Manifest je temeljil na dvanajstih načelih, ki pomagajo pri ustvarjanju okolja skupne produktivnosti. Ta načela naj bi bile smernice za time, ko vpeljujejo in izvajajo princip agilnosti. Da bi zagotovili nekaj strukture, je več agilnih metod (Scrum, XP, Lean itd.) opredelilo prakse ali tehnike, ki poudarjajo te ideale (Eby, 2017). Zakaj bi se organizacija odločila za agilne metode namesto tradicionalne metode. Nekatere organizacije lahko sprejemajo agilno metodo, da bi pospešile njihovo trženje, medtem ko poskušajo zadovoljiti nenehno spreminjajoče se povpraševanje strank ali pa preprosto želijo povečati produktivnost tima. Poenostavljeno povedano, organizacije želijo razvijati programsko opremo bolje, hitreje in ceneje. Vendar pa je hiter vzpon in komercializacija agilnih metod ustvarila tisto, kar bi nekateri lahko poimenovali kot degradacijo prvotne filozofije. Eden od soustvarjalcev manifesta Arien van Bennekum je mnenja, da je v današnjih časih vedno več agilnih trenerjev, ki ne vedo o čem govorijo, drugi soustvarjalec, Jon Kern pa meni, da ljudje so agilni, ampak samo po imenu in ne po sledenju smernicam (Singh, 2021).

Zaupanje vodstva:

Organizacije se legitimno poskušajo izboljšati, razvijalci pa poskušajo izpopolniti svojo obrt, torej zakaj smo deležni takih izjav. Agilne metode se predpisujejo brez razumevanja temeljnih korakov in dejanj. Morda je razlog medijska mantra, ki pravi, da usvojitev agilnosti, zagotoviti boljšo programsko opremo, ceneje in hitreje. Milne (2022) ugotavlja, da je to morda zato, ker višje vodstvo vidi agilnost kot način zniževanja stroškov in s tem povečanje prihodkov. Morda je to zato, ker se timi izobražujejo v naglici in se zaposlujejo učitelji in trenerji, ki popolnoma ne razumejo filozofije, na kateri temelji agilnost, in ki svoje nesporazume prenašajo na svoje učence, ki se potem trudijo spoznati vrednosti in obljube agilnih metod. Tako se postavlja splošno vprašanje, zakaj ima toliko članov timov težav pri sprejemanju in ponotranjenju agilnih metod. Precej objavljenih člankov obravnava težave, ki ji imajo timi pri prehodu na agilno kulturo. Skupno nekaterim tem člankom je, da se strinjajo, da so najpomembnejši dejavniki, ki vodijo do uspešnega sprejemanja agilnih metod kultura, ljudje in komunikacija (Milne, 2022).

Podporna kultura okoli tima je ključnega pomena za uspešno izvajanje katere koli agilne metode. Razvijalcem je treba dati svobodo pri sprejemanju odločitev, ki bodo vodile do pogostih delujočih izdaj kod, in o njih med projektom ne bi smeli dvomiti. Poleg stopnje zaupanja v razvojni tim bi morala agilna okolja spodbujati hitro in pogosto komunikacijo med člani tima, deležniki in višjim managementom. Te izjave se lahko zdijo očitne in to so stvari, ki jih morajo vse agilne metode upoštevati. Konec koncev naj bi vsakodnevni sestanki to omogočili. Morda pa vrsta pogovorov, ki se dogajajo v resnici, ne podpirajo prizadevanja tima za zagotavljanje vrednosti v rednih intervalih. Na primer, ena najmočnejših ovir za uspešno uvajanje agilnih metod, je pomanjkanje zaupanja, ki ga vodstvo nehoti pokaže svojim timom. Takšno vedenje lahko postane začaran krog (Eby, 2017). Eby (2017)

ugotavlja, da ko poskuša tim izpopolniti svoje veščine, hkrati pa razume potrebe po hitrem odzivu na neuspeh (uveljavitev sprememb, kakšen bo njihov vpliv) in začne vodstvo dvomiti o njih zaradi zamud oziroma nedoseganja rezultatov, se začne zaupanje med vodstvom in razvojnim timom rušiti. Ker to zaupanje izginja, se tim počuti manj varno pri sledenju agilnih praks in je bolj prisiljena, da ponudi nekaj, kar pokaže napredek. Manj zaupanja kot vodstvo daje, manj učinkovita postaja agilni tim. Po drugi strani pa več kot vodstvo omogoči timu oziroma ponudi manevrskega prostora, da še naprej postaja učinkovitejša pri vadbi in pridobivanju veščin z agilnimi metodami, bolj bo tim, ki ve, da vodstvo kaže potrpežljivost in mu daje čas, ustvarila želene rezultate z uporabo iterativnih metod. Drugi vodilni vzrok za neuspeh pri adaptaciji agilne metode je nerazumevanje, kaj bi morali timi dejansko početi. Treba se je zavedati, da mora biti tim za učinkovito izvajanje dela podkrepjen z znanjem. V primeru vpeljevanja agilnosti imajo nekateri timi preprosto pomanjkljivo izobrazbo ter posledično slabšo predstavo, kako naj bi bila videti agilna metoda (Eby, 2017).

Glede na raziskavo, ki so jo opravili Ganis in drugi (2020), objavljeno v letnem poročilu o stanju agilnosti, ima 36 % timov nezadostno usposabljanje in izobraževanje, 35 % jih ima nedosledne procese in prakse v skupinah, 40 % pa nima dovolj spretnosti ali izkušenj z agilnimi metodami. Timi očitno ne pridobivajo znanja, ki ga potrebujejo za ustrezno izvajanje agilnih praks. Takšni timi nimajo ustrezne izobrazbe, ker ko preidejo na agilne metode, to pogosto storijo zelo hitro, morda zato, da zadovoljijo svoje vodstvo, pri tem pa ustvarjajo nedoslednosti in zgrešijo osnovne filozofije agilnosti. Namesto da bi prehod na agilnost omogočil timom, da delajo na agilni način in naredijo boljše delo, so zdaj omejeni v svojih zmožnostih dela. To je zanka, ko se timi ujamejo, ko se sprašujejo, kako naj delajo, namesto da dejansko ustvarijo dobro delo. S tem ko hitijo neposredno v razvojni projekt, timi v bistvu delajo najboljše s tem, kar imajo. Pomanjkanje ključne komponente izobraževanja vodi v frustracije in nezadovoljstvo z agilnimi metodami.

Vodstvo nosi prav toliko odgovornosti kot timi, ko gre za napačno razumevanje agilnosti. Ko se organizacije odločijo sprejeti, se zavežejo svojim strankam, timom in sebi, da bodo spremenile način svojega dela. Glede na rezultate raziskave, ki so jo opravili Ganis in drugi (2020), 52 % timov še vedno meni, da je organizacijska kultura v nasprotju z agilnimi vrednotami, 44 % pa meni, da ni ustrezne podpore in sponzorstva vodstva. Ker več kot polovica timov potrjuje, da vodstvo ne igra svoje vloge pri sprejemanju agilnosti, lahko očitno razumemo, zakaj to sprejetje oziroma vpeljava ne uspe. Vodstvo, tako kot timi, nimajo znanja, potrebnega za izvajanje. Čeprav je vodstvo navdušeno nad izvajanjem tega novega načina, ki je domnevno hitrejši, cenejši in bo prinesel boljše delo, se ne zavezuje, da bi spoštovalo prakse, ki omogočajo razcvet agilnosti. Da bi timom omogočili agilno delo, morajo tako vodstvo kot timi razumeti metodo od zgoraj navzdol (Ganis in drugi, 2020).

9 SKLEP

Podjetje se je v preteklosti že seznanilo z agilnimi metodami. Poleg najema vozil ponujajo

najem koles, organizacijo ekipnih (angl. teambuilding) dogodkov in najem plovil. Zaradi izboljšanja procesa najema plovil je podjetje investiralo v prenovu spletnega portala, preko katerega lahko stranke najamejo plovila na lokacijah po celem svetu. Podjetje Rentavan je za izdelavo portala najelo manjše razvojno IT podjetje z dvema razvijalcema. Zahteve lastnika oziroma direktorja podjetja so bile zabeležene na kratkih enotedenskih sestankih, ki so potekali na lokaciji podjetja ali preko video klica. Vse zahteve, komunikacija med sestanki in spremljanje poteka projekta so potekale preko spletnega programa Trello, ki je vodilna spletna aplikacija za Kanban tablo. Tak pristop se je na začetku projekta izkazal kot zelo učinkovit, vendar so se med projektom pojavile tipične pomanjkljivosti, težave in zlorabe metode. Čeprav je v teoriji za Kanban tablo odgovoren tim, se je v praksi izkazalo, da so dostop do table imeli vsi deležniki projekta. Nove zahteve so se dodajale brez predhodnih sestankov oziroma ustrezne komunikacije. Tako je nastala zmešnjava, kjer razvojni tim ni mogel dohajati novim karticam, ki so bile dodane v projektno listino s strani direktorja ali managerja procesov, saj omejitve dela niso bile določene. Dodajale so se tudi nesmiselne zahteve, ki so bile po sestanku odstranjene. Ni se sledilo navodilom strukture samih kartic. Med projektom se je struktura poenostavila. Niso se več dodajali izvajalci kartice in ni se pravočasno premikalo kartic v ustrezne zavihke. Zato je pomembno, da se v začetni fazi projekta določijo stroga pravila, ki se jih morajo držati vsi deležniki projekta, in da točno določena oseba, v našem primeru projektni lastnik, to izvajanje nadzoruje.

Metoda PMBOK zahteva zelo dobro predvidevanje in je zelo neprilagojena na nepredvidene spremembe. Procesi, ki jih ponuja, so generični. To ne pomeni, da niso učinkoviti, ker metoda je v bistvu skupek najboljših praks tradicionalnega projektnega managementa in je zasnovana tako, da jo lahko uporablja čim večje število podjetij, ampak da so procesi preveč splošni in premalo specifični za določeno podjetje ali panogo. To lahko povzroči nepredvidene težave med načrtovanjem ali izvajanjem projekta. Metoda je bolj primerna za obsežnejše projekte, saj je usmerjena v poenostavitev procesa velikih projektov in je bolj potratna z viri namenjenih manjšim nalogam. Zahteva zelo detajlno načrtovanje na začetku projekta, med projektom pa ni predvideno dodatno načrtovanje. Mogoče delo, ki je bilo predvideno za izpolnitev določene aktivnosti poteka hitreje, kot je bilo načrtovano in namesto da bi se začelo delati na drugi aktivnosti, se mora čakati na določen datum. Sam postopek izvedbe mora potekati, tako kot je bilo načrtovano in čeprav bi lahko povratne informacije članov tima pripomogle k hitrejšemu izvajanju nalog ali aktivnosti ni veliko manevrskega prostora za večje spremembe oziroma boljše izvajanje projekta. PMBOK je po zasnovi bolj generična metoda, ki se osredotoča na učinkovitost in uspešnost, hkrati pa predvideva, da se bo večina članov tima držala predpisanega izvajanja projekta. S tem ko se odstopi od načrtnega izvajanja projekta, lahko to povzroči težave vsem članom tima. Zato je treba to metodo uporabljati za projekte, ki bodo čim bolj upoštevali, zasledovali načrtovano izvajanje projekta.

Veliko časa se preživi v zgodnjem delu projekta, v zbiranju zahtev, oblikovanju, ocenjevanju poteka. Razlog, zakaj lahko ta metoda deluje pri implementaciji programske opreme, je v

tem, da stranka na splošno želi vnaprej vedeti stroške projekta. Edini pravi način za določitev stroškov projekta je sledenje določenemu zaporedju izvajanja. Komunikacija je zelo pomembna pri vodenju organizacije, še toliko bolj pa pri izvajanju projekta. Metoda PMBOK temelji na učinkovitosti in produktivnosti, kar pusti manj prostora za komunikacijo. Projekt je bolj omejen z vidika stroškov kot zahtevane funkcionalnosti. Dodatne spremembe tekom projekta, ki imajo velik vpliv na končni strošek projekta, lahko prisilijo, da se sredstva preusmerijo iz pomembne stvari na potrebno spremembo. Z metodo poizkušamo narediti čim več oziroma načrtujemo preveč stvari naenkrat, preden se je projekt zares začel. Poudarek na dokumentaciji privede do pretirane birokracije in upravnega bremena. PMBOK je niz industrijskih standardnih procesov, ki jih je mogoče upoštevati za izvedbo kateregakoli projekta. Obstaja veliko delov PMBOK-a, ki so koristni v svetu razvoja programske opreme, kot so: načrtovanje komunikacije, načrtovanje tveganj. Pri metodi je poudarek na managementu tveganj in zagotavlja smernice za proaktivno zmanjševanje tveganj in znižanje negativnih vplivov na cilje projekta. Točno je določeno, katere težave lahko nastanejo med projektom in katerim ukrepom je treba slediti.

Načrtovanje projekta pri Scrum-u poteka enostavnejše kot pri PMBOK-u. Za vsak sprint poteka načrtovanje, tako da se lahko lažje pripravimo oziroma predvidimo tveganja in zmanjšamo njihov vpliv. Delo, ki je bilo predvideno, da bo trajalo določen čas pri enem sprintu, njegovo realno izvajanje lahko prenesemo v naslednji sprint, če bo vrsta dela enaka. S tem je metoda bolj prilagodljiva, saj lahko krivulja učenja tekom projekta narašča in se čas izvajanja nalog zmanjšuje in tako se lahko v naslednjem sprintu lotimo večjega števila nalog. Ker Scrum vključuje krajša obdobja intenzivnega dela, morajo biti vsi člani tima sposobni uspešno izvajati svoje naloge. To na splošno zahteva dovolj izkušenj, da se lahko to izvede. Vsi v timu morajo izvajati naloge in zagotavljati kakovostne povratne informacije o procesu in rezultatih. Ker je cilj vsakega konca sprinta izdelati del programske rešitve, je bolj smiselno uporabljati metodo pri rešitvah, ki so sestavljene iz bolj kompleksnega, večjega števila nalog. Načeloma, ko ima rešitev veliko nalog, to pomeni, da jo lahko razčlenimo na večje število manjših delujočih delov programa in vemo, ko se lotimo naslednjega koraka, da je pretekla faza razvoja potekala v skladu s pričakovanji, zahtevami. Namesto da se testira celotna rešitev, kjer je možnost večjega števila napak, programskih hroščev, poteka testiranje med projektom, ob vsakem koncu posameznega dela programa ter se sama rešitev razvija hitreje, kot če bi to delali na koncu celotne razvite rešitve. Metoda Scrum spodbuja pripadnost in predanost projektnemu timu, saj si vsi člani tima prizadevajo, da se v sprintu opravijo stvari, ki so bile načrtovane. Z dnevnimi sestanki tim spremlja napredek vsakega sprinta in opazi potencialne težave hitreje. Preglednost nad projektom je boljša in vse zainteresirane strani vedo, kaj se dogaja s projektom in kje trenutno tim je.

Zahteve so dodeljene s strani vseh uporabnikov. Ker nimajo vsi istih zahtev, tako smiselnih kot nesmiselnih, je lažje določiti, kakšna rešitev je v resnici potrebna. Sam način zbiranja zahtev je bolj usmerjen v uporabnike in ni tako splošno kot pri PMBOK-u. Z razdelitvijo zahtev po nivojih nujnosti imajo uporabniki širšo predstavbo končne rešitve. Sami med seboj

se dogovorijo, kaj naj bi nova programska rešitev vsebovala. Tudi če se potrebe in zahteve spremenijo, je spremembe lažje vpeljati, saj se dela po manjših sklopih in bo to lahko vplivalo samo na izvajanje naslednjega sprinta. Velik problem pri tej metodi je ocenitev končnih stroškov, saj je metoda tako zasnovana, da naj ne bi imela konca. Vedno se bodo našle stvari, ki jih bo mogoče popraviti, izboljšati. Bolj je usmerjena v kakovostno izdelavo programske rešitve kot pa strogemu zasledovanju omejitve cene oziroma proračuna projekta. Stroški projekta postajajo bolj jasni šele med izvajanjem projekta, tako da je lahko začetna ocena stroškov podcenjena. Seveda začetni oceni stroškov dodamo določen odstotek presežka (na primer 20 %), vendar je treba vedeti, da pri tej metodi obstaja verjetnost, da bo del presežka porabljen.

Prednost Kanban metode je to, da vsi deležniki vidijo stanje celotnega projekta kadarkoli, kjerkoli. Čeprav je uporaba table enostavna rešitev, zelo dobro obrazloži trenutno stanje projekta. Točno je vidno, kdo dela, kaj se dela, kje so potencialna ozka grla. Z večjo preglednostjo tim postane učinkovitejši, saj bodo vsi usmerjeni v prave stvari ob pravem času. Trenutno stanje projekta se vidi brez potrebe po pretiranem sestankovanju. Tako čas trajanja projekta kot tudi končni strošek je težko določiti na začetku projekta, saj vloge niso porazdeljene kot pri Scrum-u (na primer Scrum skrbnik). Načeloma vsak lahko dela karkoli, za posamezne delovne naloge je težko določiti, koliko se bodo v resnici izvajale oziroma koliko časa bo posamezni nalog potreboval, da bo potoval po nadaljnjih zavihkih. V primeru pojavitve ozkega grla oziroma zastoja spodbuja sodelovanje, saj vsi člani tima usmerijo pozornost v nalogo, ki povzroča težave. Spremembe in nove zahteve, ki se pojavijo med izvajanjem projekta, se preprosto lahko dodajo v čakalno vrsto table in ne vplivajo na potek dela razvoja izdelka. Vendar ta prednost vsebuje pomanjkljivost, saj se poveča tako čas kot cena projekta. Kanban je metoda, katere podobnost je delu v proizvodnem podjetju, kjer se ves čas opravljajo naloge. Podobno kot pri Scrum-u so zahteve glede programske rešitve določene s pomočjo uporabniških zgodb, uporabniki določijo, kaj potrebujejo. Stopnja podrobnosti posameznih aktivnosti je pri tej metodi, v primerjavi z ostalima dvema metodama, največja in najbolj podrobna. Saj se aktivnosti porazdelijo po velikosti, kompleksnosti. Za določanje velikosti posameznih uporabniških zgodb se uporablja relativni način meritve. Namesto uporabe uporabniških točk, kot pri Scrum-u se pri Kanban-u uporabljajo velikosti: S, M, L. Večja, kot je velikost, večja je stopnja razbitja. Aktivnosti morajo biti razdeljene čim bolj podrobno, saj je cilj Kanban-a, da se posamezna aktivnost opravi čim bolj učinkovito in to se doseže s smiselnim številom majhnih nalog.

Agilne metode delujejo veliko bolje za gradnjo programske opreme. Z gradnjo mislimo na celoten proces gradnje od zasnove, razvoja, testiranja. Pri agilnosti gre predvsem za iterativno gradnjo ter za vnaprejšnje oblikovanje. Želimo hitre iteracije, zbiranje dodatnih zahtev, načrtovanje, izdelavo in testiranje. To zmanjšuje količino nepotrebno zapravljenega denarja, časa in omogoča razvoj uporabne programske opreme že v začetnih fazah projekta. Na ta način agilnost ponuja številne prednosti za projekte razvoja programske opreme.

Pri izbiri ustrezne metode za izboljšanje poslovnega procesa najema vozila bi se odločila za uporabo agilnih metod, saj so nama rezultati projekta pokazali, da izvajanje v primeru uporabe Scrum metode traja 145 dni in strošek celotnega projekta je 123.983 evrov. Kanban metoda se izvaja 146 dni in strošek znaša 126.433 evrov. V primerjavi z agilnimi metodami je PMBOK metoda dražja za 30.000 evrov in traja 30 dni dlje, vendar je treba upoštevati projektno zalogo, ki je všteta pri samem načrtovanju. Čeprav sva na podlagi tehnike EVM ugotovila, da projekt PMBOK ne bo prekoračil projektne zaloge, bo vseeno dražji in trajal dlje kot pri agilnih metodah. Glede na rezultate bi se odločila za metodo Kanban, čeprav je malenkost dražji od Scrum metode. Pri tem sva upoštevala dejstvo, da je implementacija Scrum metode lahko otežena, posebej s strani razumevanja oziroma pomanjkanja ustrezne miselnosti, ki je potrebna pri uporabi metode. Kanban metodo je lažje implementirati v projekt, tako iz tehničnega vidika (uporaba ustreznih orodij) kot s strani razumevanja stranke oziroma vseh deležnikov projekta. Prednost, zaradi katere bi izbrala Kanban metodo, je enostavnejša zmožnost prilagoditve na spremembe. Tukaj bi predvsem izpostavila reševanje ozkih grl, kjer v primeru zastoja, lahko projektni tim takoj urgira s premeščanjem članov tima, ki so brez dela, na naloge, zaradi katerih projekt stoji. Projektni tim ima na voljo tudi začasno zaposlitev dodatnega člana za reševanje ozkega grla.

Izvedba projekta izboljšanja poslovnega procesa temelji na primeru majhnega slovenskega podjetja. Priložnosti za nadaljnje raziskave magistrske naloge vidiva v samem vzorcu. Agilne metode so v najinem primeru bolj primerna izbira. Rezultati izvedbe prikazujejo, da se projekt izvede hitreje in ceneje kot z uporabo tradicionalne metode. Vendar pa ne moremo trditi, da bodo rezultati, pridobljeni na podlagi izvedbe enega projekta, zagotovili uspešnost izvedbe pri večjih organizacijah ali pri organizacijah, ki poslujejo v drugih panogah. Pomemben dejavnik uspešnosti je samo razumevanje delovanja agilnih metod znotraj projektnega tima in podjetja. Ne glede na velikost podjetja in s čim se ukvarja, je pomembno, da imajo vodstvo in zaposleni pravo miselnost in se projekt uspešno izvede s pomočjo agilne metodologije. Prav na tem področju bi lahko bolj podrobno zajeli fazo pridobivanja zgodb s strani uporabnikov in zagotovili načine dobrega razumevanja delovanja agilnih metod, tako da se projekt uspešno opravi.

LITERATURA IN VIRI

1. Acqnotes. (2021, 11. julij). *Earned Value Management*. <https://acqnotes.com/acqnote/tasks/estimate-at-completion>
2. Adarsh, K. (2020, 20. september). *8 Innovative Digitization Solutions Transforming the Car Rentals Industry* [objava na blogu]. <https://inspektlabs.com/blog/8-innovative-digitization-solutions-transforming-the-car-rentals-industry/>
3. Adobe. (2022, 18. marec). *Scrum Product Owner* [objava na blogu]. <https://business.adobe.com/blog/basics/product-owner#:~:text=In%20the%20Scrum%20methodology%2C%20the,marketplace%2C%20competitors%2C%20and%20trends.>)

4. Ageling, W.-J. (2020, 14. junij). Medium. *10 Reasons Why You Are Doomed to Fail With Scrum*. <https://medium.com/p/d130953c4e4f>
5. Alby, T. (2008, 28. julij). Project management knowledge. *Project Management Body of Knowledge (PMBOK)*. <https://project-management-knowledge.com/definitions/p/project-management-body-of-knowledge-pmbok/>
6. Alexander, M. (2023, 28. avgust). Cio. *Agile project management: 12 key principles, 4 big hurdles*. <https://www.cio.com/article/237027/agile-project-management-a-beginners-guide.html>
7. AresPrism. (2023, 1. oktober). *Progress Management & Performance Measurement Guide*. <https://info.contruent.com/hubfs/Using%20Progress%20and%20Performance%20Measurement%20to%20Successfully%20Manage%20Projects%2010272023.pdf>
8. ASANA (2024, 15. februar). *What is the project management triangle and how can it help your team?*. <https://asana.com/resources/project-management-triangle>
9. Bhaskar, S. (2022, 18. avgust). Nimble. *What is Kanban? A Beginner's Guide!*. <https://www.nimblework.com/kanban/what-is-kanban/#:~:text=%E2%80%93%20A%20Brief%20History%20on%20Kanban%20It%20all,Engineer%20and%20Businessman%29%20for%20Toyota%20automotive%20in%20Japan.>
10. Bit.AI. (2022, 30. marec). *Procurement Management Plan: What is it & How to Create? (Steps Included)* [objava na blogu]. <https://blog.bit.ai/procurement-management-plan/>
11. Blueprint. (2022, 18. junij). *Agile Methodologies*. <https://www.blueprintsys.com/learning-center>
12. Boiser, L. (2020, 14. april). Kanbanzone. *Why Your Kanban System Implementation Failed: Having a Kanban Board is NOT Enough*. <https://kanbanzone.com/2020/having-a-kanban-board-is-not-enough/>
13. Bračič, A. (2018, 19. september). Outsider. *Tatjana Kerčmar: Poslovni prostori Borger-Gremo*. <https://outsider.si/tatjana-kercmar-poslovni-prostori-borger/>
14. Breuker, M. (2021, 19. oktober). *Earned value management: an introduction* [objava na blogu]. [https://www.pinnaclemanagement.com/blog/earned-value-management-an-introduction#:~:text=Earned%20Value%20Management%20\(EVM\)%20is,extent%2C%20schedule\)%20at%20completion](https://www.pinnaclemanagement.com/blog/earned-value-management-an-introduction#:~:text=Earned%20Value%20Management%20(EVM)%20is,extent%2C%20schedule)%20at%20completion)
15. Calendar-Canada. (2022, 10. januar). *What is the difference between Lean and Kanban?*. <https://www.calendar-canada.ca/frequently-asked-questions/what-is-the-difference-between-lean-and-kanban>
16. Carlos, T. (2011). *Why Projects Fail: Avoiding the Classic Pitfalls*. California: Oracle.
17. Chi, C. (2018, 30. december). *Everything You Need to Know About Using the Waterfall Methodology* [objava na blogu]. <https://blog.hubspot.com/marketing/waterfall-methodology>
18. Chiva, G. (2020, 26. januar). Aktia Solutions. *Kanban Practices*. <https://aktiasolutions.com/kanban-practices/>

19. Cohn, M. (2010). *Succeeding with Agile: Software Development Using Scrum*. Addison-Wesley.
20. Davenport, T. (1993). *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business Press.
21. Deemer, P. in Benefield, G. (2006). *Scrum primer* [objava na blogu]. https://t1.daumcdn.net/tistoryfile/fs2/2_12_4_20_blog50577_attach_0_2.pdf?original
22. Deemer, P., Benefield, G., Larman, C. in Vodde, B. (2012). *The Scrum Primer*. https://scrumprimer.org/scrumprimer20_small.pdf
23. Dehoyos, M. (2020, 26. februar). AgileConnection. *The Differences between Traditional and Agile Project Management*. <https://www.agileconnection.com/article/differences-between-traditional-and-agile-project-management>
24. Drumond, C. (2013, 26. avgust). Atlassian. *What is scrum?*. <https://www.atlassian.com/agile/scrum>
25. Duke, R. (29. december 2010). Gantt.com. *What is a Gantt Chart?..* <https://www.gantt.com/>
26. Eby, K. (2017, 15. februar). Smartsheet. *What's the Difference? Agile vs Scrum vs Waterfall vs Kanban*. <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>
27. Egovernment. (2011, 1. julij). *Project Management guidelines*. https://www.dpac.tas.gov.au/__data/assets/pdf_file/0029/108992/Tasmanian_Government_Project_Management_Guidelines_V7_0_July_2011_2.pdf
28. Espirito, D. (2022, 28. marec). *Top 5 main Agile methodologies: advantages and disadvantages* [objava na blogu]. <https://www.xpand-it.com/blog/top-5-agile-methodologies/>
29. Farrar, M. (2021, 18. februar). Technicali. *What are Triple Constraints of Project Management*. <https://technicali.com/triple-constraints-of-project-management/>
30. Frohman, R. (2016, 4. december). *How to Create a Release Burn-Up Chart* [objava na blogu]. <https://www.robfrohman.com/blog-agilewhys/2016/11/28/how-to-create-a-release-burn-up-chart>
31. Ganis, M., Ackerbauer, M. in Cariello, N. (2020, 23. julij). Cutter. *Why Does Agile Fail?*. <https://www.cutter.com/article/why-does-agile-fail>
32. Ghahrai, A. (2017, 4. februar). Testing excellence. *Most Common Agile Development Methodologies*. <https://web.archive.org/web/20170630200711/http://www.testingexcellence.com/common-agile-development-methodologies/>
33. Gorenšek, T. (2019, 28. februar). Homopolitikus. *Digitizacija, digitalizacija in digitalna transformacija*. <https://homopolitikus.si/digitizacija-digitalizacija-in-digitalna-transformacija/>
34. Grigoryan, A. (2022, 17. januar). Wrike. *Project Manager vs. Project Owner*. <https://www.wrike.com/product-management-guide/faq/what-is-the-difference-between-a-project-manager-and-project-owner/>

35. Guay, C. (2020, 27. januar). *5 reasons why your scrum project fails in 2020* [objava na blogu]. <https://const.fr/blog/agile/5-reasons-why-your-scrum-project-fails-in-2020/>
36. Haughey, D. (2010, 10. oktober). ProjectSmart. *A Brief History of Project Management*. <https://www.projectsmart.co.uk/history-of-project-management/brief-history-of-project-management.php>
37. Hawkins, J. (2018, 27. september). Medium. *Top 5 Reasons Why Scrum Fails and What You Can Do About It*. <https://jay-hawkins.medium.com/5-reasons-why-scrum-fails-and-what-you-can-do-to-overcome-them-932e863b43c9>
38. Inflectra. (2014, 1. junij). *Methodology Guide & Explainer*. <https://www.inflectra.com/Methodologies/Kanban.aspx>
39. Isom, E. (2019, 28. september). PM Guaranteed. *Why Scrum Fails: The 2 Main Reasons*. <https://web.archive.org/web/20210801141354/https://www.pmguaranteed.com/why-scrum-fails-the-2-main-reasons/>
40. ITBusinessEdge. (2010, 9. marec). *PMBOK's Five-Phase Project Lifecycle: An Overview*. <https://www.itbusinessedge.com/it-management/pmboks-five-phase-project-lifecycle-an-overview/>
41. Kashyap, V. (2022, 1. januar). ProofHub. *Ten Strong Reasons Why Projects Fail (And How To Kick Them Out)*. <https://www.proofhub.com/articles/reasons-why-projects-fail>
42. KnowledgeHut. (2018, 5. december). *Development team*. <https://www.knowledgehut.com/tutorials/scrum-tutorial/development-team>
43. KombiCenter. (2023, 5. januar). *Vozni park*. <https://kombicenter.si/vozila/>
44. Kothari, A. (2024, 22. maj). Tallyfy. *PMBOK in Project Management: Definition and Best Practices*. <https://tallyfy.com/pmbok/>
45. Kukhnavets, P. (2018, 18. marec). *Advantages and disadvantages of Extreme Programming (XP)* [objava na blogu]. <https://hygger.io/blog/disadvantages-and-advantages-of-extreme-programming/>
46. Kumar, V. in Sowmyavani, M. (2012). Agile Software Development: A Case Study of Web Application. *Global Journal of Computer Science and Technology*, 12(C14), 19–23.
47. Layton, M. C. (2012). *Agile Project Management For Dummies*. Wiley.
48. Ledig, A. (2021, 11. marec). *What are the main causes of project failure?* [objava na blogu]. <https://web.archive.org/web/20210311204200/https://www.wimi-teamwork.com/blog/main-causes-project-failure/>
49. Lester, A. (2014). *Project Management, Planning and Control*. Elsevier Ltd.
50. Lewis, S. (2018, 8. september). WhatIs.com. *Project Management Body of Knowledge (PMBOK)*. <https://whatis.techtarget.com/definition/Project-Management-Body-of-Knowledge-PMBOK>
51. Liyanarachchi, R. (2019, 1. Marec). DZone. *5 Reasons Why Scrum Fails in Software Development*. <https://dzone.com/articles/five-reasons-why-scrum-fails-in-software-developme>
52. Marschall, M. (2015, 27. julij). Agile Web Operations. *Kanban vs Scrum vs Agile*. <https://agileweboperations.com/2015/07/27/scrum-vs-kanban/>

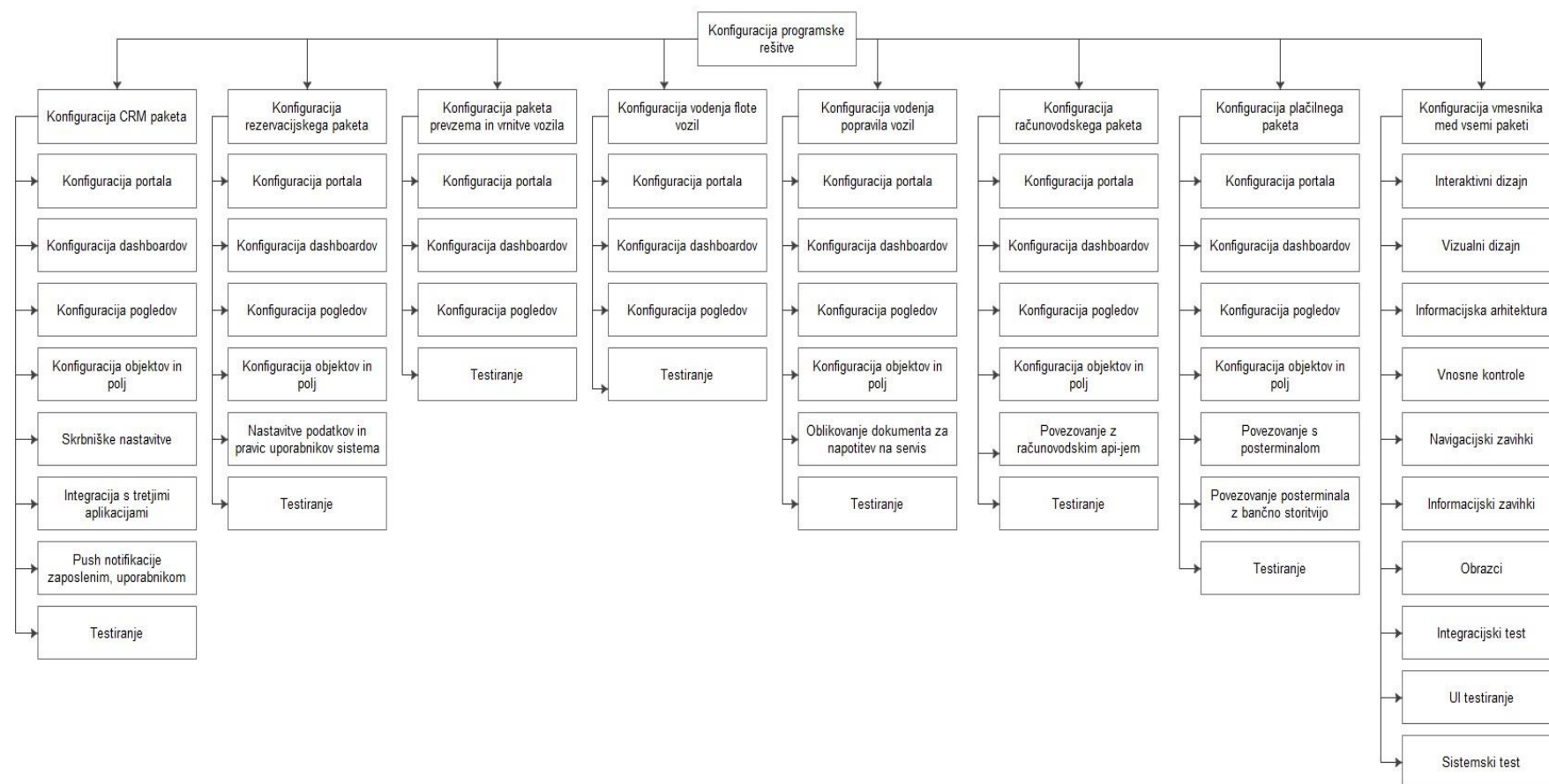
53. Martins, J. (2024, 27. marec). Asana. *12 tips for effective communication in the workplace*. <https://asana.com/resources/effective-communication-workplace>
54. Matta, N. in Ashkenas, R. (2003, 1. september). Harvard Business Review. *Why Good Projects Fail Anyway*. <https://hbr.org/2003/09/why-good-projects-fail-anyway>
55. Meiling, J. (2019, 15. januar). *Kanban in Organizations Part II* [objava na blogu]. <https://www.projectwizards.net/en/blog/2019/01/kanban-corepractices#explicit-policies>
56. Microsoft. (2014, 20. maj). *History of project management*. <https://support.microsoft.com/en-us/office/history-of-project-management-a2e0b717-094b-4d1e-878a-fcd0978891cd>
57. Milne, A. (2022, 8. avgust). Net solutions. *10 Reasons Why Agile Transformations Fail and how yours can succeed*. <https://www.netsolutions.com/insights/how-to-prevent-agile-transformation-failure/#:~:text=A%20lopsided%20and%20underutilized%20methodology%20of%20communication%20and,favors%20individuals%20%26%20interactions%20over%20processes%20%26%20tools>.
58. Narasimman, P. (2024, 17. maj). *Agile vs Traditional Project Management [Top Differences]* [objava na blogu]. <https://www.knowledgehut.com/blog/agile/agile-project-management-vs-traditional-project-management>
59. Naydenov, P. (2015, 23. november). Businessmap. *The Best 6 Agile Methodologies: Which Type Fits Best Your Needs*. <https://businessmap.io/agile/different-agile-methodologies>
60. Naydenov, P. (2019, 9. februar). *6 Reasons You May Fail with Kanban Implementation* [objava na blogu]. <https://web.archive.org/web/20190510005717/https://kanbanize.com/blog/problems-with-kanban-implementation/>
61. Naydenov, P. (2021, 1. julij). Businessmap. *What Is Kanban? A Simple Guide to Improve Efficiency*. <https://businessmap.io/kanban-resources/getting-started/what-is-kanban>
62. Nikolov, D. (2020, 6. december). Get Agile Right. *How Kanban Works (The Only Introduction You'll Ever Need)*. <https://getagileright.com/how-kanban-works/>
63. Nutcache. (2019, 22. junij). *Six Main Processes in PMBOK Time Management* [objava na blogu]. [https://www.nutcache.com/blog/six-main-processes-pmbok-time-management/#:~:text=Time%20management%20in%20PMBOK%20\(Project,can%20ensure%20adherence%20to%20schedule](https://www.nutcache.com/blog/six-main-processes-pmbok-time-management/#:~:text=Time%20management%20in%20PMBOK%20(Project,can%20ensure%20adherence%20to%20schedule)
64. O'Malley, P. (2023, 10. september). Openclassrooms. *Discover Three Different Types of Agile Project Management Frameworks*. <https://openclassrooms.com/en/courses/4544621-learn-about-agile-project-management-and-scrum/5080431-discover-three-different-types-of-agile-project-management-frameworks>
65. Overby, S., Greiner, L. in Gibbons, L. (2024, 20. junij). CIO. *What is an SLA? Best practices for service-level agreements*. <https://www.cio.com/article/274740/outsourcing-sla-definitions-and-solutions.html>

66. Paula. (2020, 24. avgust). PremierAgile. *Understanding Project Success Criteria with Examples*. <https://premieragile.com/scrum-master-responsibilities/>
67. Paula. (2023, 26. februar). PremierAgile. *Life cycle of extreme programming*. <https://premieragile.com/life-cycle-of-extreme-programming/>
68. Pawar, Y. (2017, 12. februar). *What is the difference between Traditional and Agile Project Management* [objava na blogu]. <https://web.archive.org/web/20210726172118/https://upraise.io/blog/traditional-vs-agile-project-management-3/>
69. Permana, P. A. (2015). Scrum Method Implementation in a Software. *International Journal of Advanced Computer Science and Applications*, 6, 198-204.
70. PMI. (2013). *A Guide to the Project Management Body of Knowledge*. Project Management Institute.
71. PMI. (2020). *Project Management Institute*. <https://www.pmi.org/>
72. Pries, K.H. in Quigley, J.M. (2010). *Scrum Project Management*. CRC Press.
73. PremierAgile. (2021, 17. avgust). *Sprint Backlog Vs Release Backlog*. <https://premieragile.com/sprint-backlog-vs-release-backlog/>
74. Reh, J. (2019, 25. junij). The balance careers. *Basic Project Management 101*. <https://www.thebalancecareers.com/project-management-101-2275338>
75. Richards, K. (2023, 20. februar). Agilekrc. *What is DSDM and the 8 principles*. <https://agilekrc.com/resource/168/what-dsdm-and-8-principles>
76. Roberts, J. in Furlonger, J. (2000). *Successful IS Project Management*. Gartner.
77. Rochford, D. in Mitchell, I. (2019, 18. januar). PMP. *Project Quality Management*. <https://web.archive.org/web/20221208024156/https://pmp-practitioners.com/project-quality-management/>
78. Rodriguez, J. (2018, 4. december). Plan Academy. *5 Reasons Why A Project Management Plan Fails*. <https://www.planacademy.com/reasons-project-management-plan-fails/>
79. Roseke, B. (2016, 27. maj). ProjectEngineer. *Project Cost Management According to the PMBOK*. <https://www.projectengineer.net/project-cost-management-according-to-the-pmbok/>
80. Rosencrance, L. (2019, 15. april). *Top 5 agile software development methodologies* [objava na blogu]. <https://web.archive.org/web/20220809194048/https://www.insureon.com/blog/pros-and-cons-of-the-top-5-agile-software-development-methodologies>
81. Rubin, K. S. (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley.
82. Rudder, A. in Main, K. (2023, 20. julij). What Is The Project Management Triangle?. *Forbes*. <https://www.forbes.com/advisor/business/project-management-triangle/#:~:text=It%20is%20a%20project%20management,be%20adjusted%20to%20maintain%20quality>

83. Schwaber, K. in Sutherland, J. (2020, 1.november). *The Scrum guide*. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf#zoom=100>
84. Schwartz, B. (2021, 21. februar). *The Risk Management Process in Project Management* [objava na blogu]. <https://www.projectmanager.com/blog/risk-management-process-steps>
85. SCRUMstudy. (2016). *Scrum body of knowledge (Sbok guide)*. VMEdU.
86. Sellgren, A. (2014, 5. september). Kanban University. *Principles and General Practices of the Kanban Method*. <https://web.archive.org/web/20240519195309/https://kanban.university/principles-general-practices-kanban-method/>
87. SimpliAxis. (2019, 1. oktober). *Core Practices of Kanban*. <https://www.simpliaxis.com/resources/core-practices-of-kanban>
88. Simplilearn. (2023, 18. maj). *Understanding Project Success Criteria with Examples*. <https://www.simplilearn.com/tutorials/project-management-tutorial/project-success-criteria>
89. Singh, N. K. (2021, 22. marec). Agilemania. *6 Reason to choose Agile Project Management over Traditional Project Management*. <https://agilemania.com/agile-project-management-vs-traditional-project-management/>
90. Singh, V. (2021, 7. julij). ToolsQA. *Crystal Method in Agile*. <https://www.toolsqa.com/agile/crystal-method/>
91. Stellman, A. in Greene, J. (2014). *Learning Agile: Understanding Scrum, XP, Lean, and Kanban*. O'Reilly.
92. Sutherland, J. in Schwaber, K. (2011). *The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework*. ScrumInc.
93. Teamhood. (2021, 1. maj). *4 Core Kanban Principles and 6 Practices*. <https://teamhood.com/kanban-resources/kanban-principles/#:~:text=The%20%20Kanban%20practices%201%201.%20Visualize%20%28the,...%206%206.%20Improve%20collaboratively%2C%20evolve%20experimentally%20>
94. Verma, A. (2018, 6. december). *7 Reasons Why Agile fails* [objava na blogu]. <https://www.functionize.com/blog/7-reasons-why-agile-fails>
95. Verzuh, E. (2021). *The Fast Forward MBA in Project Management*. John Wiley & Sons.
96. Vige, W. (2024, 17. maj). Asana. *What are story points? Six easy steps to estimate work in Agile*. <https://asana.com/resources/story-points>
97. Weber, J. L. (2018, 12. december). *Project management* [objava na blogu]. <https://www.projectmanager.com/blog/history-of-pmi>
98. Williams, M. (2008). *The principles of project management*. SitePoint Pty.
99. Yorke, J. (2016, 19. oktober). Agileconnection. *The Reason Scrum So Often Fails Agile Teams*. <https://www.agileconnection.com/article/reason-scrum-so-often-fails-agile-teams>

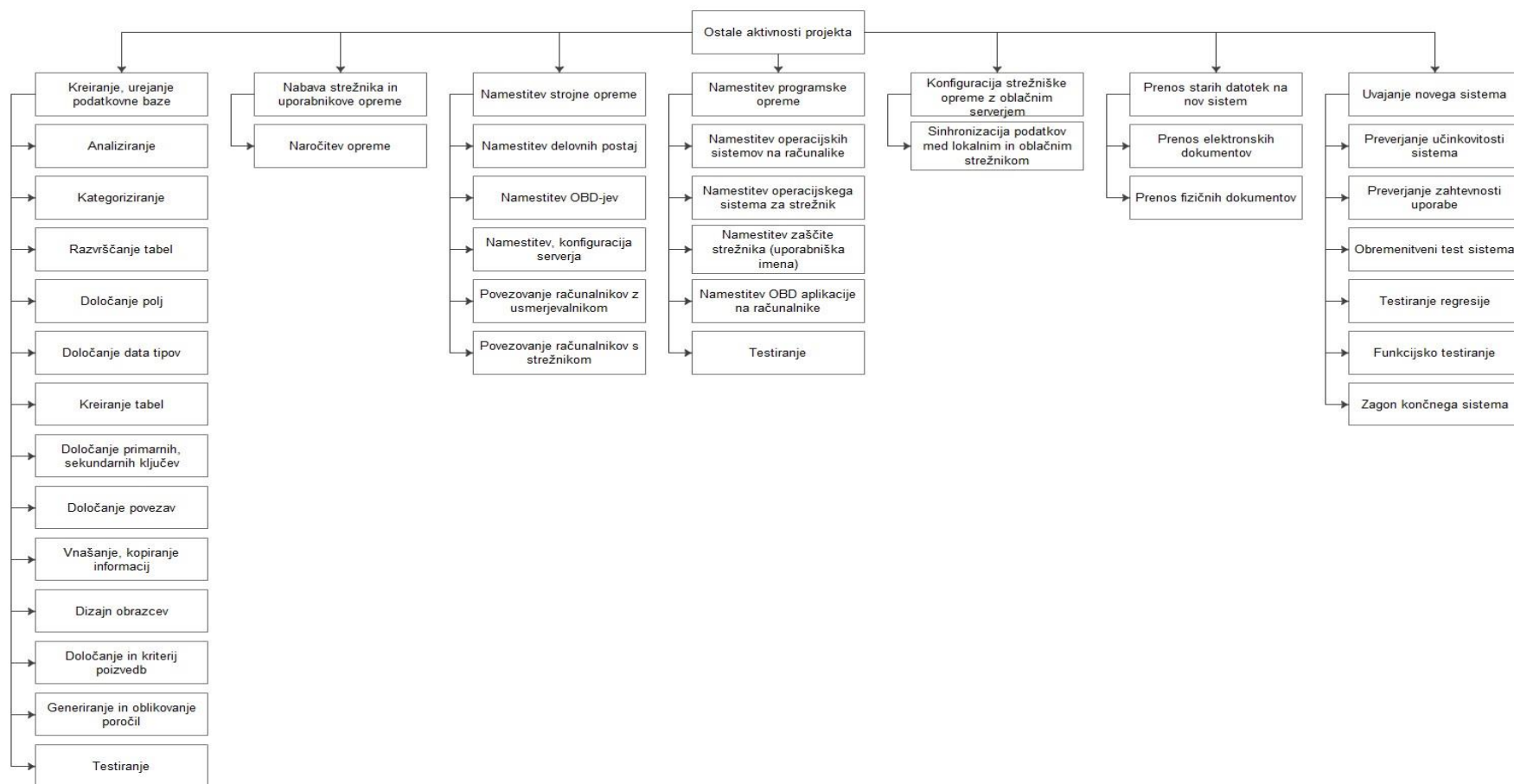
PRILOGE

Priloga 1: WBS konfiguracije programske opreme



Vir: lastno delo.

Priloga 2: WBS ostale aktivnosti projekta



Vir: Lastno delo

Priloga 3: Management komunikacij

Priloga 4 prikazuje, kako poteka komunikacija in kdo je lastnik posameznega dela (projektni manager-PM, timski manager-TM, projektni skrbnik-PS, zunanji razvijalec-ZR).

Tip komun.	Cilj komun.	Način komun.	Pogostost	Deležniki	Lastnik	Končni rezultat	Oblika
Začetni sestanek	Predstavitev projekta in projektne tima, Pregled projektne ciljev in način izvedbe	V živo	Enkrat	Projektne skrbnik, Projektne tim Deležniki	PM	Dnevni red, Zapisnik sestanka	Poročilo začetnega sestanka v elektronski obliki naloženo na skupno odlagališče
Sestanki projektne tima	Pregled stanja projekta s timom	V živo, Konferenčni klici (Zoom, Teams)	Tedensko	Projektne tim	PM	Dnevni red, Zapisnik sestanka, Urnik projekta	Poročilo v elektronski obliki naloženo na skupno odlagališče
Tehnično načrtovanje projekta	Razpravljati in razvijati tehnične rešitve projekta	V živo	Po potrebi	Razvijalci	PM	Dnevni red Zapisnik sestanka	Poročilo v elektronski obliki naloženo na skupno odlagališče
Mesečni pregledi projekta	Mesečno poročilo o stanju projekta	V živo, Konferenčni klici (Zoom, Teams)	Mesečno	PMO, Projektne tim	PM	Projektne načrt	Poročilo v elektronski obliki naloženo na skupno odlagališče
Poročilo stanja projekta	Poročilo stanja projekta z vsemi aktivnostmi, stroški, težavami in napredkom	Elektronska pošta	Mesečno	Projektne skrbnik, Projektne tim, Deležniki	PM	Poročilo statusa projekta, Projektne načrt	Poročilo v elektronski obliki naloženo na skupno odlagališče in deljeno med vsemi deležniki projekta

se nadaljuje

Tip komun.	Cilj komun.	Način komun.	Pogostost	Deležniki	Lastnik	Končni rezultat	Oblika
Sistemske zahteve	Pridobivanje potreb projekta	Elektronska pošta, V živo, Konferenčni klici	Po potrebi	Razvijalci	PM	Dokumentacija (grafika, predstavitve, prve različice oz. verzije)	Poročilo v elektronski obliki naloženo na skupno odlagališče
Programske zahteve	Pridobivanje programskih zahtev	V živo, Elektronska pošta, Konferenčni klici	Po načrtu projekta, pred implementacijo programske opreme	Razvijalci, Projektni tim	PM	Dokumentacija (grafična predstavitev, različne verzije)	Poročilo v elektronski obliki naloženo na skupno odlagališče
Seznam opravil	Dodeljevanje seznama opravil	V živo, Elektronka pošta, Konferenčni klici,	Po potrebi	Projektni tim, Razvijalci	PM	Razdelitev vlog in opravil	Dokumentacija opravil
Testiranje Sistema	Testiranje stabilnosti sistema, Testiranje sprejemljivosti uporabnika	Programsko, V živo, Elektronska pošta	Po načrtu projekta	Testni tim (Projektni tim)	PM	Testiranje glede na zahteve uporabnika, Testno okolje (UI)	Testno okolje, Poročila o testiranju
Finance	Proračun projekta	V živo	Mesečno	Projektni skrbnik	PM	Pregled mesečnih financ projekta	Finančna poročila
Verzije programa	Novosti in posodobitve (prototipi) projekta	Konferenčni klici, V živo	Mesečno	Uporabniki (zaposleni)	PM	Predstavitve novosti zaposlenim	Dokumentacija za pridobivanje povratnih informacij zaposlenih

se nadaljuje

Tip komun.	Cilj komun.	Način komun.	Pogostost	Deležniki	Lastnik	Končni rezultat	Oblika
Uporabniško učenje, treniranje	Učenje in prilagajanje uporabnikov na nov sistem	Preko spleta, Na lokaciji	Po načrtu projekta, Po vsakem mejniku	Uporabniki	PM	Znanje o novem sistemu	Učno gradivo o novem sistemu za uporabnike
Stanje kritičnih del projekta	Stanje napredovanja projekta, Integracija različnih sistemov v oblaku	V živo, Elektronska pošta	Mesečno, Po načrtu projekta	Projektni skrbnik, Projektni tim	PM	Redne posodobitve projektnih aktivnosti in možnih težavah	Dokumentacija oziroma poročilo
Sestanek pregleda projekta	Tedenski pregled projekta in možnosti izboljšav	V živo, Elektronska pošta	Tedensko	Projektni tim, Zunanji razvijalec za integracijo IS Paketa	TM	Dokumentacija o stanju projekta in implementiranih rešitev	Dokumentacija
Vsakodnevni sestanek	Sestanek o vsakodnevnih opravilih, njihovih težavah, Obveščanje koliko časa še ostane za dokončanje opravila	V živo, Elektronska pošta	Dnevno	Razvijalci, Zunanji razvijalec za integracijo IS Paketa (po načrtu)	TM	Projektna naznanila	Ustni ali pisni dogovor o opravilih
Timbuilding	Morala tima	V živo	Vsaka dva meseca	Projektni skrbnik, Projektni manager, Projektni tim	PS, PM	Visoka morala tima, boljša povezljivost	Timbuilding dogodki

se nadaljuje

Tip komun.	Cilj komun.	Način komun.	Pogostost	Deležniki	Lastnik	Končni rezultat	Oblika
Poročilo o kakovosti projekta	Redno izboljševanje kakovosti projekta	Elektronska pošta	Tedensko	Zunanji razvijalec za integracijo IS Paketa (po potrebi), Projektni skrbnik Projektni tim	PM	Posodabljanje in pregled nad kvaliteto izvedbe projekta	Poročilo o kvaliteti
Prošnje za systemske spremembe	Dodajanje ali odstranjevanje dodatnih zahtev projekta	Elektronska pošta	Po potrebi	Projektni skrbnik Projektni tim	ZR	Prošnja za dodajanje ali odstranitev zahtev projekta v pisni elektronski obliki	Dokument oziroma Prošnja

Vir: lastno delo.

Priloga 4: Potek četrtega sprinta

SEZNAM SPRINTA 9	DELO V TEKU 3	OPRAVLJENO 3 ✓
Tehnično sprejetje SCRUM-41 2	Konfiguracija objektov in polj SCRUM-38 2	Konfiguracija pogledov (view) SCRUM-37 ✓ 3
Konfiguracija portala SCRUM-42 3	Povezovanje z računovodskim apijem (minimax) SCRUM-39 2	Konfiguracija dashboarda SCRUM-36 ✓ 3
Konfiguracija dashboard SCRUM-43 3	Testiranje SCRUM-40 8	Konfiguracija portala SCRUM-35 ✓ 3
Konfiguracija pogledov (view) SCRUM-44 3		
Konfiguracija objektov in polj SCRUM-45 3		
Povezovanje s postterminalom SCRUM-46 1		

Vir: lastno delo.

V četrtem sprintu se opravljata dve aktivnosti. Konfiguracija računovodskega paketa, konfiguracija plačilnega paketa. Aplikacija se poveže z računovodskim programom Minimax, ki ga podjetje trenutno uporablja za računovodske storitve, tako da ni potrebe po spremembi ponudnika računovodske rešitve. Implementacija programa poteka z Minimax API, ki je RESTful spletna storitev in omogoča povezavo z aplikacijami, ki niso del računovodskega programa. Pri konfiguraciji plačilnega paketa se poleg oblikovnih sprememb omogočijo še finančne transakcije med strankami in podjetjem. Obe aktivnosti se testirata in tehnično sprejme.

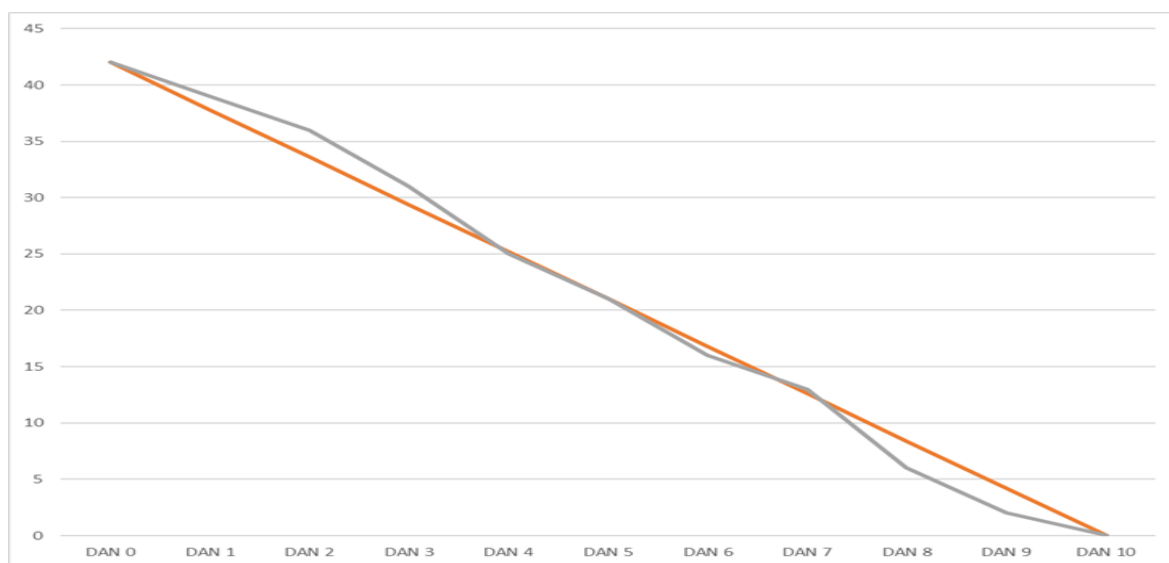
Priloga 5: Uporabniške točke v četrtem sprintu

Uporabniške zgodbe	Dan 1	Dan 2	Dan 3	Dan 4	Dan 5	Dan 6	Dan 7	Dan 8	Dan 9	Dan 10
Konfiguracija portala	3	2	1							
Konfiguracija dashboard	3	2	1							
Konfiguracija pogledov (view)	3	2	1							
Konfiguracija objektov in polj	2	2	2	1	1					
Povezovanje z računovodskim apijem (minimax)	2	2	2	1	1	1				
Testiranje	8	8	8	8	4	4				
Tehnično sprejetje	2	2	2	2	2	2				
Konfiguracija portala	3	3	3	3	3	3	2	1		
Konfiguracija dashboard	3	3	3	3	3	3	2	1		
Konfiguracija pogledov (view)	3	3	3	3	3	3	2	1		
Konfiguracija objektov in polj	3	3	3	3	3	3	3	3		
Povezovanje s posterminalom	1	1	1	1	1	1	1	1		
Povezovanje postterminala z bančno storitvijo	2	2	2	2	2	2	2	2	2	
Testiranje	2	2	2	2	2	2	2	2	2	
Tehnično sprejetje	2	2	2	2	2	2	2	2	2	2
Skupaj	42	39	36	31	27	26	16	13	6	2
Opravljeno		3	3	5	4	1	10	3	7	4

Vir: lastno delo.

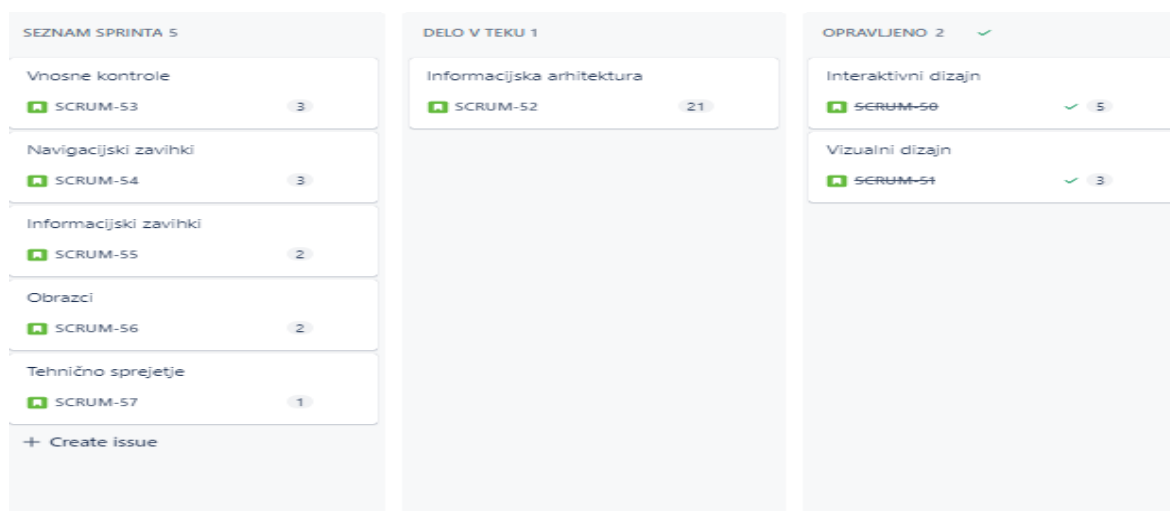
Ob začetku sprinta se razvojni tim ni držal predvidene časovnice, saj aplikacija in spletni portal nista omogočala enostavne integracije z zunanjimi aplikacijami, kot je bilo na začetku predvideno. V drugi polovici sprinta je integracija elementov Minimax programa potekala kasneje kot planirano, kljub temu da je večino elementov aplikacije podjetje že uporabljalo pri poslovanju. Razvijalci so poskrbeli, da so funkcionalnosti računovodskega programa ostale iste oziroma podobne in grafično prijazne uporabnikom.

Priloga 6: Graf izgorevanja za četrti sprint



Vir: lastno delo.

Priloga 7: Potek petega sprinta



Vir: lastno delo.

Pri petem sprintu konfiguracije grafičnega vmesnika se dokončata samo dve aktivnosti, interaktivni dizajn in vizualni dizajn rešitve, aplikacije ter spletne strani. Razvijalci so morali poskrbeti, da je grafična podoba aplikacije sodobna, privlačna in preprosta za uporabnike. Vse glavne funkcionalnosti morajo biti hitro dostopne in jasno prikazane na prvi strani aplikacije. S stališča zaposlenega mora nadzorna plošča pravilno prikazovati trenutno stanje voznega parka, stanje najemov (pretekli, trenutni in planirani).

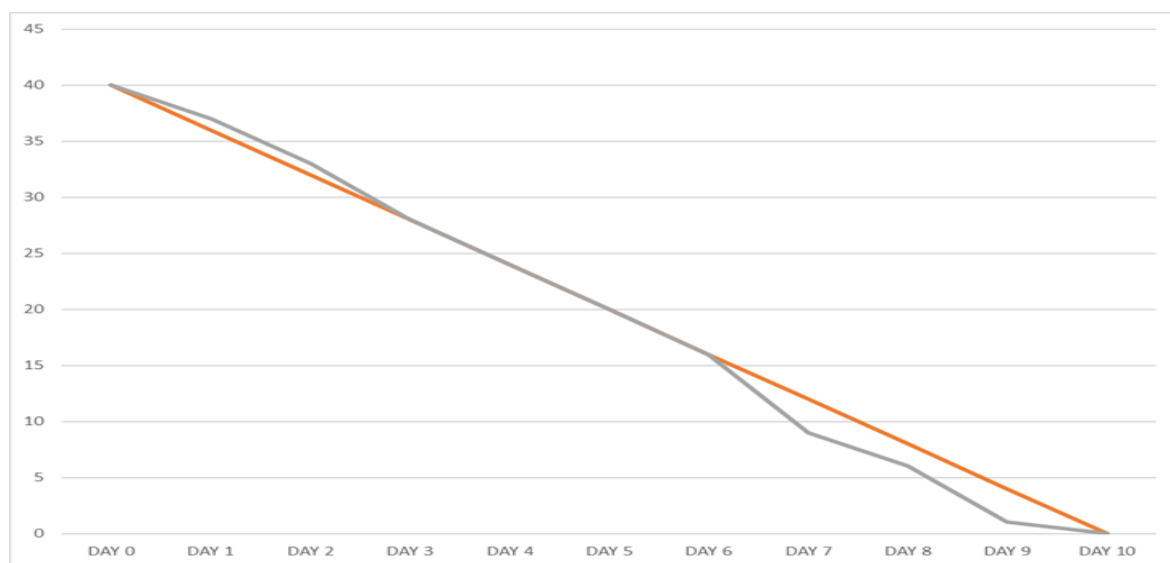
Priloga 8: Uporabniške točke v petem sprintu

Uporabniške zgodbe	Dan 1	Dan 2	Dan 3	Dan 4	Dan 5	Dan 6	Dan 7	Dan 8	Dan 9	Dan 10
Interaktivni dizajn	5	2								
Vizualni dizajn	3	3	1							
Informacijska arhitektura	21	21	21	17	13	9	5			
Vnosne kontrole	3	3	3	3	3	3	3	1		
Navigacijski zavihki	3	3	3	3	3	3	3	3	1	
Informacijski zavihki	2	2	2	2	2	2	2	2	2	
Obrazci	2	2	2	2	2	2	2	2	2	
Tehnično sprejetje	1	1	1	1	1	1	1	1	1	1
Skupaj	40	37	33	28	24	20	16	9	6	1
Opravljenost		3	4	5	4	4	4	7	3	5

Vir: lastno delo.

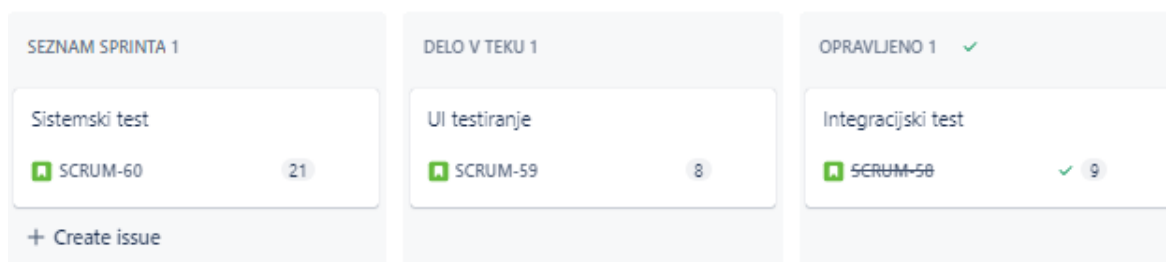
Zaradi že opravljenih sprintov smo predvidevali, da celotne konfiguracije grafičnega vmesnika ne bo mogoče končati v desetdnevem sprintu. Ker obstoječa rešitev ponuja prilagajanje in spreminjanje vseh modulov, ki jih vsebuje (od programske kode do grafične podobe), razvijalci niso pričakovali težav pri sami izvedbi konfiguracije. Zamudo smo pričakovali zaradi obsega dela, ki ga zajema izdelava celotne grafične podobe aplikacije. Prekoračitev smo predvidevali in ga prenesli v naslednji sprint.

Priloga 9: Graf izgorevanja za peti sprint



Vir: lastno delo.

Priloga 10: Potek šestega sprinta



Vir: Lastno delo

V šestem sprintu se preveri grafična skladnost celotne rešitve, aplikacije. V tem sprintu se zaključi konfiguracija ponujene obstoječe rešitve in se preveri, kako celotna rešitev in njeni moduli funkcionirajo. Izvedejo se integracijski test, testiranje uporabniškega vmesnika in regresijski test.

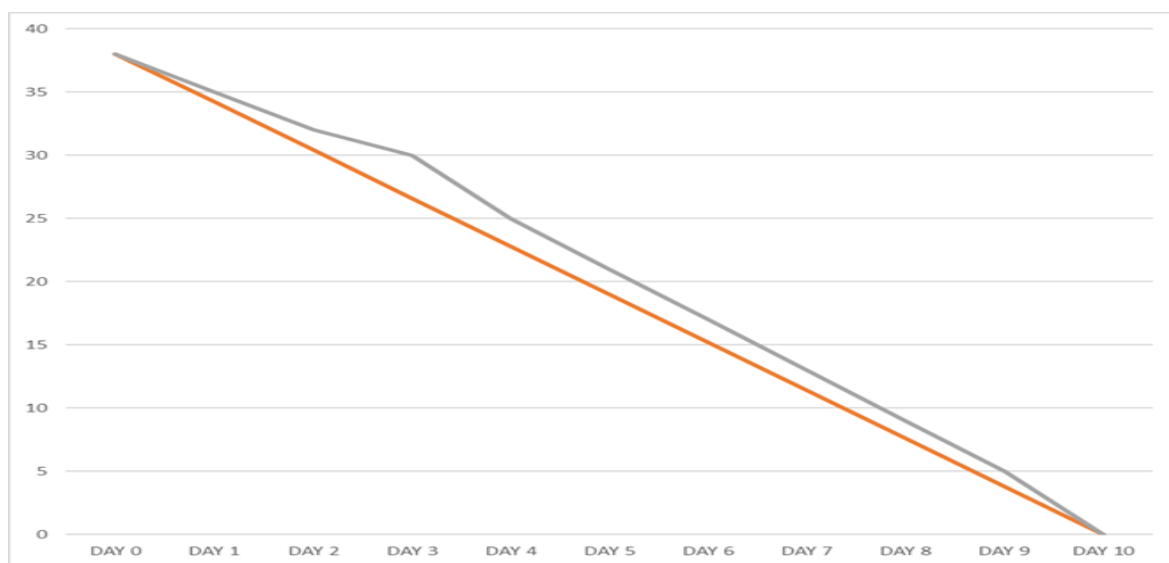
Priloga 11: Uporabniške točke v šestem sprintu

Uporabniške zgodbe	Dan 1	Dan 2	Dan 3	Dan 4	Dan 5	Dan 6	Dan 7	Dan 8	Dan 9	Dan 10
Integracijski test	9	6	3	1						
UI testiranje	8	8	8	8	4					
Sistemiški test	21	21	21	21	21	21	17	13	9	4
Skupaj	38	35	32	30	25	21	17	13	9	4
Opravljeno		3	3	2	5	4	4	4	4	5

Vir: lastno delo.

Testiranje je potekalo mnogo počasneje, kot je bilo predvideno, ker je bil sam obseg potrebnega testiranja bolj zamuden in kompleksen.

Priloga 12: Graf izgorevanja za šesti sprint



Vir: lastno delo.