

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra systémového inženýrství



Diplomová práce

Agilní řízení projektů ve vybrané společnosti

Martin Soumar

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Martin Soumar

Projektové řízení

Název práce

Agilní řízení projektů ve vybrané společnosti

Název anglicky

Agile project management in selected company

Cíle práce

Cílem práce je popis stávajícího agilního řízení projektů ve vybrané společnosti a vypracování návrhu pro další rozvoj.

Metodika

METODIKA:

Po nastudování vybrané odborné literatury a navázání spolupráce s vybranou firmou, bude proveden sběr dat a požadavků o praxi agilního řízení projektů v této společnosti. Na základě rozboru zjištěných dat, požadavků a očekávání bude vypracován popis stávajícího agilního řízení projektů ve společnosti a následně návrh pro další rozvoj. Pro prezentaci návrhu bude v práci vhodným způsobem použito procesních modelů a vybraných nástrojů nebo přístupů systémové vědy. Vlastní návrhy budou prezentovány a diskutovány v praxi vybrané společnosti. Po vypracování praktické části práce bude sepsána literární rešerše.

HARMONOGRAM:

Leden – květen 2015: Nastudování vybrané odborné literatury.

Únor – červen 2015: Navázání spolupráce s vybranou firmou.

Červen – srpen 2015: Provedení sběru dat a požadavků.

Srpen – září 2015: Provedení rozboru zjištěných dat.

Září – listopad 2015: Vypracování popisu a návrhu pro další rozvoj.

Prosinec 2015: Prezentace a diskuze vlastí návrhů.

Leden 2016: Sepsání literární rešerše.

Doporučený rozsah práce

60 – 90 stran

Klíčová slova

Projektové řízení, agilní manifest, agilní metody a metodologie, projekty a zakázky ICT.

Doporučené zdroje informací

DOLEŽAL, Jan, MÁCHAL, Pavel a LACKO, Branislav. Projektový management podle IPMA. 2., aktualiz. a dopl. vyd. Praha: Grada, 2012, 526 s. Expert (Grada). ISBN 978-80-247-4275-5.

ISO 21500:2012 Guidance on project management. Geneva: International Organization for Standardization, 2012.

KERZNER, H. 2013. Project Management: A Systems Approach to Planning, Scheduling, and Controlling. Eleventh edition. New York: John Wiley & Sons Inc., 1264 s. ISBN 978-1-118-02227-6.

MURRAY, A. et al. 2013. Managing Successful Projects with PRINCE2®. Fifth edition. London: AXELOS Limited, 327 s. ISBN 978-0113310593.

PROJECT MANAGEMENT INSTITUTE 2013. A Guide to the Project Management Body of Knowledge (PMBOK® Guide). Fifth Edition. Newtown Square: Project Management Institute Inc., 589 s. ISBN 978-1-935589-67-9.

SVOZILOVÁ, Alena. Projektový management. 2., aktualiz. a dopl. vyd. Praha: Grada, 2011, 380 s. Expert (Grada). ISBN 978-80-247-3611-2.

Předběžný termín obhajoby

2015/16 LS – PEF

Vedoucí práce

Ing. Jan Bartoška, Ph.D.

Garantující pracoviště

Katedra systémového inženýrství

Elektronicky schváleno dne 26. 1. 2016

doc. Ing. Tomáš Šubrt, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 9. 2. 2016

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 20. 03. 2016

Čestné prohlášení

Prohlašuji, že svou diplomovou práci „Agilní řízení projektů ve vybrané společnosti“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2016

Poděkování

Rád bych touto cestou poděkoval panu Ing. Janu Bartoškovi, Ph.D. za vedení mé diplomové práce, podnětné návrhy, cenné rady a věnovaný čas. Dále chci také poděkovat panu Bc. Šimonu Vlachynskému za poskytnuté rady a diskuze z hlediska praxe. Mé díky patří také mé rodině za podporu a pomoc nejen při psaní této práce, ale po celou dobu studia.

Agilní řízení projektů ve vybrané společnosti

Souhrn

Diplomová práce je zaměřena na vypracování vlastních návrhů pro další rozvoj agilního řízení projektů ve vybrané společnosti na základě popisu a rozboru stávajících agilních postupů. Po zpracování teoretických podkladů v literární rešerši je charakterizována vybraná společnost a je popsáno agilní řízení projektů obecně i konkrétně v případové studii. Na závěr analýzy případové studie, jsou postupy společnosti porovnány s pilíři agilního manifestu a principy metody Scrum. Na základě tohoto porovnání jsou identifikovány nedostatky, pro jejichž potlačení, či eliminaci jsou v praxi společnosti nejen prezentovány, ale i diskutovány vlastní návrhy a doporučení pro další rozvoj projektového řízení.

Klíčová slova: Agilní manifest, agilní metody, agilní metodologie, metoda Scrum, projekty ICT, projektové řízení

Agile project management in selected company

Summary

This work is focused on developing of own proposals and recommendations for future growth of agile approach in selected company, based on description and analysis of existing procedures in this company. There is basic characteristic of selected company, general and specific description in case study after study of theoretical basis. Agile procedures are compared to principles of agile manifesto and Scrum method in conclusion of the case study. There are identified shortcomings of procedures in agile project management of selected company based on this comparison. For mitigation, or elimination of these shortcomings, there are presented and discussed own proposals and recommendations for future growth of agile approach in selected company.

Keywords: Agile manifesto, agile methods, agile methodologies, ICT projects, project management, Scrum method

Obsah

1. Úvod.....	12
2. Cíl a metodika práce.....	13
2.1. Cíl práce	13
2.2. Metodika.....	13
2.3. Harmonogram.....	14
3. Literární rešerše.....	15
3.1. Základní pojmy projektového řízení	15
3.1.1. Projektové řízení	15
3.1.2. Projekt	16
3.1.3. Program.....	16
3.1.4. Portfolio projektů	17
3.1.5. Cíl projektu	17
3.1.6. Trojimperativ projektového řízení	17
3.1.7. Životní cyklus projektu a fáze projektu	18
3.2. Mezinárodní standardy a metodiky projektového řízení	19
3.2.1. Mezinárodní standard PMBOK.....	19
3.2.2. Mezinárodní standard PRINCE2.....	23
3.2.3. Mezinárodní standard ICB	28
3.3. Agilní metody řízení projektů	30
3.3.1. Definice agilního přístupu.....	30
3.3.2. Manifest agilního vývoje softwaru.....	30
3.3.3. Scrum	32
3.3.4. Kanban	37
3.3.5. Lean Software Development.....	38
3.3.6. Extrémní programování (XP).....	40
3.3.7. Test Driven Development	43

4.	Praktická část	45
4.1.	Charakteristika vybrané společnosti.....	45
4.1.1.	Popis a historie společnosti	45
4.1.2.	Podnikatelské aktivity a oblast trhu	45
4.1.3.	Organizační struktura společnosti	46
4.1.4.	Komunikace ve společnosti.....	47
4.2.	Projektové řízení ve společnosti.....	47
4.2.1.	Projektová kancelář ve společnosti	47
4.2.2.	Základní principy projektového řízení ve společnosti	48
4.2.3.	Organizační struktura projektu.....	49
4.3.	Životní cyklus projektů ve vybrané společnosti	50
4.3.1.	Předprojektová fáze.....	51
4.3.2.	Plánovací fáze	52
4.3.3.	Realizační fáze	53
4.3.4.	Ukončovací fáze.....	54
4.3.5.	Poprojektová fáze.....	55
4.4.	Případová studie	56
4.4.1.	Vize projektu.....	56
4.4.2.	Předprojektová fáze.....	56
4.4.3.	Plánovací fáze	56
4.4.4.	Realizační fáze	57
4.4.7.	Shrnutí projektu.....	59
4.4.8.	Srovnání praxe společnosti s agilním manifestem.....	60
4.4.9.	Srovnání praxe společnosti s principy metody Scrum	62
4.5.	Vlastní návrhy a doporučení.....	62
4.5.1.	Včasná a průběžná dodávka software	62
4.5.2.	Denní spolupráce zákazníků a vývojářů na projektu	68

4.5.3.	Řízení komunikace.....	68
4.5.4.	Stálá pozornost návrhu a technickému řešení	70
4.5.5.	Organizace týmů a definování rolí.....	71
4.5.6.	Rozšíření dokumentace	72
5.	Výsledky a diskuze	74
5.1.	Popis agilního řízení ve společnosti	74
5.2.	Vlastní návrhy a doporučení.....	75
5.2.1.	Včasná a průběžná dodávka software	75
5.2.2.	Denní spolupráce zákazníků a vývojářů na projektu	76
5.2.3.	Řízení komunikace.....	76
5.2.4.	Stálá pozornost návrhu a technickému řešení	76
5.2.5.	Organizace týmů a definování rolí.....	77
5.2.6.	Rozšíření dokumentace	77
6.	Závěr	78
7.	Seznam použitých zdrojů	79
8.	Přílohy.....	81

Seznam obrázků

Obrázek 1:	Trojimperativ dle IPMA	18
Obrázek 2:	Typické rozložení fází životního cyklu projektu.....	18
Obrázek 3:	Interakce procesních skupin ve fázi, nebo projektu	21
Obrázek 4:	Sedm principů PRINCE2	24
Obrázek 5:	Sedm témat PRINCE2	26
Obrázek 6:	Sedm procesů PRINCE 2	28
Obrázek 7:	Tři oblasti kompetencí dle ICB	29
Obrázek 8:	Scrum.....	36
Obrázek 9:	Mantra TDD	44
Obrázek 10:	Organizační struktura společnosti	47
Obrázek 11:	Organizační struktura projektu a projektového týmu	50

Obrázek 12: Přibližný harmonogram fází popisovaného projektu	60
Obrázek 13: Návrh jednoduché WBS	65
Obrázek 14: Simulace užití nástroje Discord ve společnosti	70

Seznam tabulek

Tabulka 1: Srovnání projektového a liniového řízení	16
Tabulka 2: Kompetence ICB	29
Tabulka 3: Předprojektová fáze	52
Tabulka 4: Plánovací fáze	53
Tabulka 5: Realizační fáze	54
Tabulka 6: Ukončovací fáze	55
Tabulka 7: Poprojektová fáze	55
Tabulka 8: Základní přehled projektu	57
Tabulka 9: Stručná finanční analýza	59
Tabulka 10: Srovnání předpokládaných a skutečných termínů projektu	60
Tabulka 11: Srovnání praxe společnosti s agilním manifestem	61
Tabulka 12: Srovnání praxe společnosti s principy metody Scrum	62
Tabulka 13: Šablona Lessons Learned projektu	66
Tabulka 14: Hodnotící škály pro výskyt a dopad rizika	67
Tabulka 15: Šablona registru rizik	67
Tabulka 16: Matice rizik	68
Tabulka 17: Obecná RACI matice pro celý projekt	72
Tabulka 18: Šablona RACI matice pro jednotlivé User Story	72

1. Úvod

Rozvoj oboru projektového řízení způsobil vznik různých standardů, metodik a norem, které poskytují teoretické podklady a certifikace potřebné k úspěšnému projektovému řízení. Bohužel tyto standardy nejsou vždy vhodné k řízení IT projektů, jejichž četnost a komplexita, díky neustálému rozvoji oboru informačních technologií a systémů, stále roste. Proto se čím dál více společností snaží nalézt alternativy, jak IT projekty řídit. Obory týkající se informačních technologií a systémů, se neustále rozvíjí a jejich využití roste. Slovo „agile“ se v problematice řízení IT projektů objevuje častěji a mnozí neví, co si pod tím představit. Proto vznikla diplomová práce, která se snaží nastínit, jak se v současnosti takové projekty v české IT společnosti řídí.

Agilní metody řízení projektů jsou odpovědí na neefektivní a neúspěšné řízení IT projektů. Metody jsou založeny na předpokladu, že projekt je dynamický s nejasnými požadavky, které se v průběhu realizace projektu mohou měnit. I přes to, že jsou agilní přístupy poměrně mladé (agilní manifest byl sepsán roku 2001), vzniklo mnoho agilních metod, které jsou ve světě hojně využívány společnostmi zaměřenými na informační technologie a systémy. Mezi nejpoužívanější metody patří například Scrum, Kanban, Lean Software Development, Feature Driven Development a Extreme Programming. Těmito metodami se zabývá právě diplomová práce, konkrétně jejich využitím ve vybrané společnosti, která v současnosti své interní projekty agilně řídí. Zvolená společnost se tak řadí mezi jednu z mála českých společností, která agilním metodám důvěřuje, a využívá je.

Teoretická část práce popisuje základní pojmy problematiky projektového řízení, mezinárodní standardy a agilní metody projektového řízení. Poskytuje tak stručný přehled, potřebný ke zpracování praktické části práce. V praktické části práce je popsáno stávající agilní řízení projektů ve vybrané společnosti a vypracován vlastní návrh pro další rozvoj.

2. Cíl a metodika práce

2.1. Cíl práce

Cílem diplomové práce „Agilní řízení projektů ve vybrané společnosti“ je vypracování vlastních návrhů pro další rozvoj agilního řízení projektů ve vybrané společnosti na základě popisu a rozboru stávajících agilních postupů. Součástí práce je popis stávajícího agilního řízení projektů ve vybrané společnosti tak, aby bylo pro čtenáře srozumitelné, jak takové řízení projektů probíhá a co obnáší.

Účelem této práce je vypracování návrhu pro další rozvoj agilního řízení projektů ve vybrané společnosti, na základě popisu a rozboru její aktuální praxe. Agilní řízení bude popsáno na základě spolupráce s vybranou společností a její praxí v oboru agilního řízení projektů. Na základě studia teoretických východisek a rozboru zjištěných dat, požadavků a očekávání bude vypracován vlastní návrh pro další rozvoj řízení projektů ve společnosti.

Na základě spolupráce s vybranou společností a studie její praxe, nebude pro prezentaci návrhů užito procesních modelů a vybraných nástrojů, nebo přístupů systémové vědy. Jejich užití není pro praxi společnosti a agilní přístup řízení projektů vhodné. Záměrem diplomové práce je nalézt konkrétní praktické řešení v souladu s očekáváním firmy.

2.2. Metodika

Nejprve bude nastudována odborná literatura a internetové zdroje týkající se jak klasického projektového řízení, tak agilního. Poté bude navázána spolupráce s vybranou společností, kde následně proběhne sběr dat a požadavků o praxi agilního řízení projektů v této společnosti. Spolupráce s vybranou společností proběhne převážně formou řízených rozhovorů a schůzek se zaměstnanci společnosti.

Praktická část bude vycházet ze studia odborné literatury, internetových zdrojů a spolupráce s vybranou společností. Bude zde vypracován popis společnosti včetně jejího portfolia a organizační struktury. Dále bude v praktické části popsáno řízení projektů ve společnosti, životní cyklus zvoleného projektu. Postupy společnosti budou následně porovnány s pilíři agilního manifestu a základními principy metody Scrum.

Na závěr praktické části budou prezentovány vlastní návrhy a doporučení pro další rozvoj. Vlastní návrhy budou diskutovány v praxi vybrané společnosti.

Na základě studia odborné literatury, internetových zdrojů a interních zdrojů získaných ze společnosti bude vypracována literární rešerše. V této části diplomové práce budou uvedeny základní pojmy projektového řízení, mezinárodní standardy projektového řízení a agilní metody projektového řízení.

Posledním krokem bude společně se sepsáním závěru práce, vypracována diskuse výsledků, seznam použitých zdrojů a přílohy.

2.3. Harmonogram

Leden - květen 2015

Nastudování vybrané odborné literatury a internetových zdrojů.

Květen – září 2015

Navázání spolupráce s vybranou firmou.

Září - listopad 2015

Provedení sběru dat a požadavků.

Prosinec 2015

Provedení rozboru zjištěných dat.

Leden 2016

Vypracování popisu a návrhu pro další rozvoj.

Únor 2016

Prezentace a diskuze vlastních návrhů.

Březen 2016

Sepsání literární rešerše.

3. Literární rešerše

V této části diplomové práce jsou uvedeny základní pojmy projektového řízení, mezinárodní standardy projektového řízení a agilní metody projektového řízení.

3.1. Základní pojmy projektového řízení

3.1.1. Projektové řízení

Ačkoliv je projektový management definován a popisován různě, podstata je obdobná:

- **Harold Kerznera** popisuje projektové řízení jako „*souhrn aktivit spočívající v plánování, organizování, řízení a kontrole zdrojů společnosti s relativně krátkodobým cílem, který byl stanoven pro realizaci specifických cílů a záměrů*“. (citace podle [1], s. 19)
- **PMBOK** definuje projektové řízení jako „*aplikace znalostí, schopností, nástrojů a technologií na aktivity projektu tak, aby splnily požadavky projektu*“. (citace podle [1], s. 19)
- **PRINCE2** charakterizuje projektové řízení jako plánování, delegování, monitorování a kontrolu všech stránek projektu a motivování všech zúčastněných k dosažení cílů projektu v předepsaném čase, nákladech, kvalitě rozsahu, výhodách a rizicích. (vlastní zpracování podle [3])
- **IPMA** definuje projektové řízení jako „*aplikaci znalostí, dovedností, nástrojů a technik na činnosti v projektu tak, aby projekt splnil požadavky na něj kladené. Zahrnuje plánování, organizování, monitorování a předávání zpráv o všech aspektech projektu a motivaci všech zúčastněných dosáhnout cílů projektu*“. (citace podle [2], s. 315)

Projektové řízení	Liniové řízení
užití zdrojů	zajištění zdrojů
řízení v nejistotě	předvídatelnost
unikátnost	uniformita
kontrola čerpání nákladů	hospodaření s majetkem
kontrola skutečného postupu vůči plánu	kontrola v absolutních měřítcích přijatelnosti výstupů
řízení kvality prostřednictvím plánu a preventivních opatření	kvalita řízena na základě inspekce výstupů
proměnný počet pracovníků	stabilní počet pracovníků
interní hlášení	hlášení mimo podnikatelské uskupení
úspěšnost hodnocena podle míry naplnění stanovených cílů	úspěšnost měřena absolutním výkonem dle vybraných ukazatelů

Tabulka 1: Srovnání projektového a liniového řízení (vlastní zpracování podle [1], s. 42)

3.1.2. Projekt

Stejně jako pro projektové řízení, tak pro jeden z nezákladnějších pojmů, kterým je projekt, existuje řada definic, jejichž podstata je obdobná:

- **Harold Kerzner** definuje projekt jako „*jakýkoliv jedinečný sled aktivit a úkolů, který má dán: specifický cíl, jenž má být jeho realizací splněn; definováno datum začátku a konce uskutečnění; stanoven rámec pro čerpání zdrojů potřebných pro jeho realizaci*“. (citace podle [1], s. 22)
- **PMBOK** popisuje projekt jako „*dočasné úsilí vynaložené na vytvoření unikátního produktu, služby, nebo určitého výsledku*“. (citace podle [1], s. 22)
- **PRINCE 2** charakterizuje projekt jako dočasnou organizaci, která je vytvořena za účelem dodání jednoho nebo více produktů na základě odsouhlaseného obchodního případu (Business Case). (vlastní zpracování podle [3])
- **IPMA** definuje projekt jako „*jedinečný časově, nákladově a zdrojově omezený proces realizovaný za účelem vytvoření definovaných výstupů v požadované kvalitě a v souladu s platnými standardy a odsouhlasenými požadavky*“. (citace podle [2], s. 314)

3.1.3. Program

Program je složen z jednotlivých projektů, které jsou příbuzné a podléhají takové koordinaci a kontrole, aby bylo dosaženo výhody oproti samostatně řízeným

projektům. Prostřednictvím programů jsou obvykle řízené rozsáhlé systémy. Stejně jako projekt, je program definován v časovém rámci. (vlastní zpracování podle [1], s. 47)

3.1.4. Portfolio projektů

Portfolio projektů je soubor projektů (programů) se sdílenými zdroji a časovým rámcem. Nemají společný cíl, ale byly seskupeny dohromady za účelem řízení, kontroly, koordinace a optimalizace. (vlastní zpracování podle [2], s. 318)

3.1.5. Cíl projektu

Cíl projektu je „*nová hodnota – předmět, služba nebo jejich kombinace, která je výsledkem projektu a je reprezentována popisem určitého stavu, jenž má v budoucnosti existovat*“. (citace podle [1], s. 82) Představuje slovní popis účelu, jehož má být prostřednictvím realizace projektu dosaženo, měl by být dobře definován, vágní definice vede ke zvýšení pravděpodobnosti neúspěchu projektu a vysoké nejistotě.

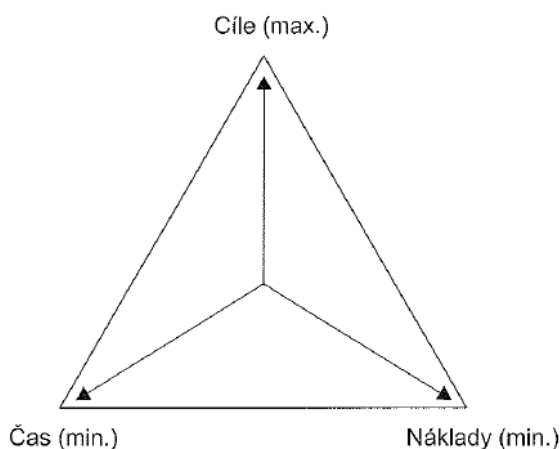
Kompetenční standard IPMA (citace podle [4], s. 62) doporučuje řídit se technikou SMART(i):

- **S** (specific) – specifický a specifikovaný – CO?,
- **M** (measurable) – měřitelný – ČEHO jsme dosáhli?,
- **A** (agreed) – akceptovaný – všichni relevantní VĚDÍ a SOUHLASÍ,
- **R** (realistic) – realistický – cíl musí být USKUTEČNITELNÝ,
- **T** (timed) – termínovaný – bez TERMÍNU postrádá cíl smysl,
- **i** (integrated) – integrovaný – INTEGROVANÝ do organizační strategie.

3.1.6. Trojimperativ projektového řízení

Trojimperativ je „*cíl projektu vyjádřený veličinami: výsledek (kvalita), čas a náklady (zdroje)*“. (citace podle [2], s. 280) Jeho účelem je optimální vyvážení a nalezení rovnováhy mezi těmito veličinami, kde cíl je bodem v daném trojúhelníkovém prostoru (viz obrázek č. 1). Tyto tři veličiny jsou vždy provázané a navzájem se ovlivňují, nikdy nelze změnit pouze jednu z nich. Optimálně chceme

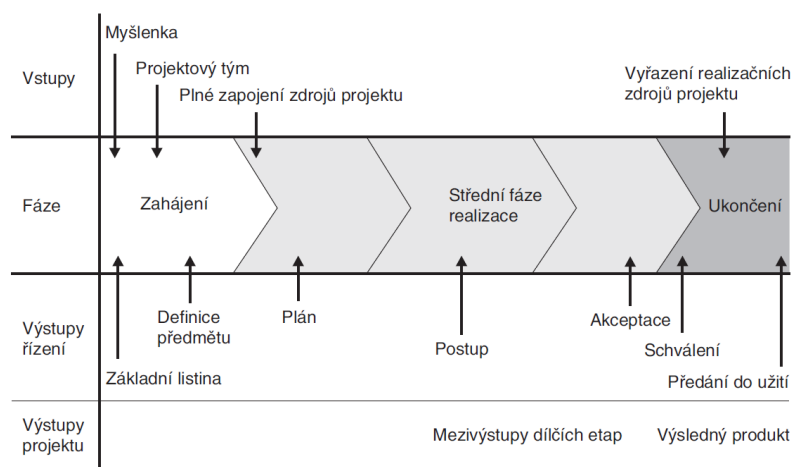
dosáhnout maximálního výsledku za minimálního času a nákladů.
(citace podle [4], s. 63)



Obrázek 1: Trojimperativ dle IPMA (převzato z [4], s. 63)

3.1.7. Životní cyklus projektu a fáze projektu

Životní cyklus projektu je „souborem obecně následných fází projektu, jejichž názvy a počet jsou určeny potřebami kontroly organizace, která je v projektu angažována“ (citace podle [1], s. 38) Fáze životního cyklu projektu definují, jaká práce má být vykonána v příslušném stupni vývoje projektu. Dále výstupy, které jsou v jednotlivých fázích generovány, jak jsou hodnoceny a ověřovány. A také kdo a kdy se zapojuje do projektu v jednotlivých úsecích. (vlastní zpracování podle [1], s. 39)



Obrázek 2: Typické rozložení fází životního cyklu projektu (převzato z [1], s. 38)

3.2. Mezinárodní standardy a metodiky projektového řízení

Rozvoj oboru projektového řízení způsobil vznik různých standardů, metodik a norem, poskytujících teoretické podklady a certifikace potřebné k úspěšnému projektovému řízení. Nejsou návodem, kterým se vedoucí pracovníci musí řídit, ale spíše inspirací, která by měla vést k zefektivnění projektového řízení v organizaci. Standardy poskytují nástroje, které lze při řízení aplikovat a kombinovat. Metodiky stanovují přímo dané postupy, které je nutné dodržovat. Součástí norem jsou zkušební metody, terminologie a značení.

V současnosti je v oblasti projektového řízení využíváno několik mezinárodních standardů. Mezi ty nejvýznamnější patří „The Guide to the Project Management Body of Knowledge (PMBOK)“ od institutu projektového řízení PMI (Project Management Institute), „Project IN Controlled Environments no. 2 (PRINCE2)“ vyvinutý britským vládním úřadem OGC (Office of Government Commerce) a kompetenční standard „IPMA Competence Baseline (ICB)“, který vytvořila mezinárodní asociace projektového řízení IPMA (International Project Managemenet Association). Mezi standardy lze také zařadit mezinárodní normu ISO 21500 Návod k managementu projektu vycházející ze standardu PMBOK.

3.2.1. Mezinárodní standard PMBOK

Procesní mezinárodní standard PMBOK od institutu projektového řízení PMI vznikl roku 1987 podle standardů armády USA. Zaměřuje se na nástroje potřebné pro projektové řízení a je rozdělen do tří oddílů, kterými jsou Rámec řízení projektu (vymezení konceptu projektu), Znalostní oblasti řízení projektu (10 oblastí) a Procesy při řízení projektu (5 hlavních skupin, které obsahují 47 procesů). (vlastní zpracování podle [5])

Hlavní procesní skupiny

PMBOK definuje 5 hlavních procesních skupin, zahrnujících 47 procesů. Procesní skupiny nejsou fázemi životního cyklu projektu, ale jsou podle něho logicky seskupené. Skupiny jsou propojené, mohou se navzájem ovlivňovat a překrývat. Procesy jsou obvykle realizovány v rámci každého projektu. Procesy se dále dělí na

hlavní a podpůrné, oba uvedené typy ale nemusí v procesní skupině vždy být. Životní cyklus je souborem fází, na které je projekt rozdělen. Fáze jsou tvořeny pro každý projekt a nemusí se s procesními skupinami shodovat. Fáze si stanovuje organizace individuálně a mohou se dělit i na sub-fáze. (vlastní zpracování podle [5])

a) Zahajovací procesní skupina

Procesy obsažené v této skupině slouží k iniciaci, přípravě a vytvoření vize projektu. Účelem této skupiny je srovnání očekávání stakeholderů s cílem projektu a zajištění toho, že jejich očekávání jsou řádně naplněna. Zahrnují popis cíle projektu, rozhodnutí o realizaci projektu, identifikaci stakeholderů, definování rolí a odpovědností, odhad nákladů a pracnosti, odhad doby trvání projektu, atd. (vlastní zpracování podle [5])

b) Plánovací procesní skupina

Plánovací procesní skupina navazuje na výstupy z předchozí fáze a zaměřuje se na podrobné plánování všech činností. Klíčovým přínosem je načrtnutí taktik a strategií, potřebných k úspěšnému dokončení projektu. Její součástí je sestavení plánu projektu a harmonogramu, stanovení projektové dokumentace, identifikace všech aktivit, požadavků, rizik, omezení projektu atd. (vlastní zpracování podle [5])

c) Realizační procesní skupina

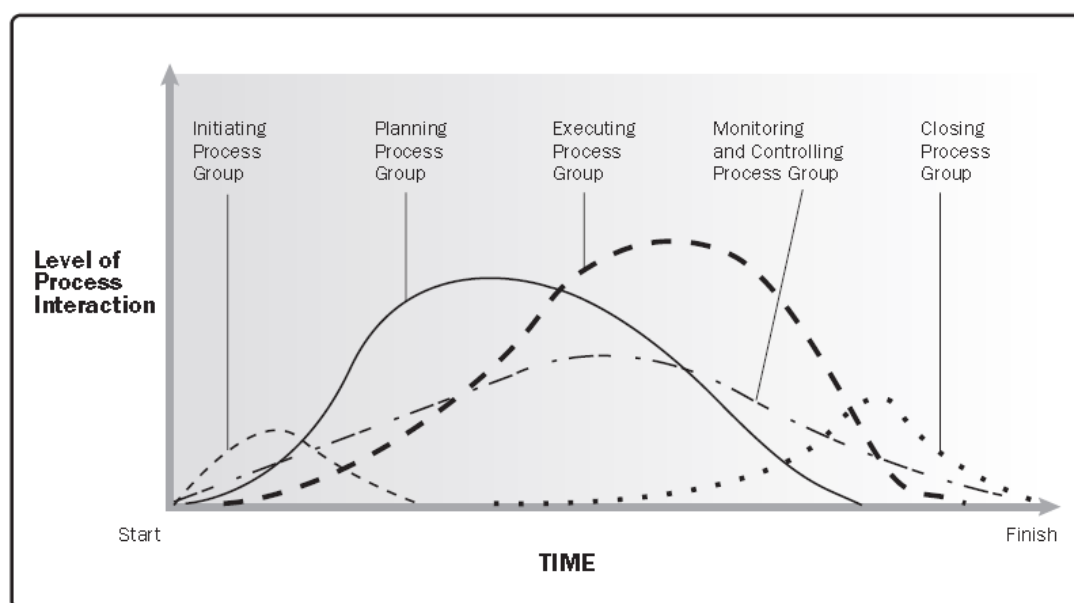
V této skupině nalezneme procesy, které souvisí s realizací projektu, naplněním jeho cíle a očekávání stakeholderů. Během těchto procesů může dojít k odchylkám a změnám, které mohou zapříčinit aktualizaci plánu, nebo jeho změnu. Do realizační procesní skupiny patří především průběh a výsledky projektu, řízení lidí, koordinace činností, dochází k čerpání zdrojů atd. (vlastní zpracování podle [5])

d) Procesní skupina monitorování a kontrola

Procesy v této skupině slouží k sledování a ověřování realizace projektu během jeho celého životního cyklu. Jsou zde nastaveny kontrolní mechanismy, sloužící k včasnému rozpoznání hrozeb a změn, díky kterým je možné včasné a patřičně zareagovat. Je to nepřetržitý proces umožňující týmům analyzovat zdraví projektu a identifikovat oblasti, které potřebují zvýšenou pozornost. Podstatnou součástí je zpětná vazba z předchozích procesních skupin. (vlastní zpracování podle [5])

e) Ukončovací procesní skupina

Ukončovací procesy slouží k uzavření projektu a také formálně ověřují, že všechny procesy v procesních skupinách byly dokončeny. Dochází k vyhodnocení projektu a srovnání dosažených cílů s plánovanými. Ukončovací procesní skupina dále zahrnuje kontrolu kvality projektu, administrativní ukončení projektu, uzavření smluv, předání zákazníkovi, rozpuštění zdrojů atd. (vlastní zpracování podle [5])



Obrázek 3: Interakce procesních skupin ve fázi, nebo projektu (převzato z [5], s. 51)

Znalostní oblasti PMBOK

Každý proces standardu PMBOK je definován ve znalostních oblastech. Znalostní oblasti představují souhrn konceptů, pravidel a aktivit, patřících určitému oboru, oblasti nebo specializaci. Znalostní oblasti neposkytují jen detailní popis vstupů a výstupů procesů, ale také popis nástrojů a technik často využívaných v projektovém řízení. Znalostní oblasti jsou stručně popsány v následujícím seznamu. (vlastní zpracování podle [5])

1. Řízení integrace projektu:

- vytvoření listiny projektu,
- vytvoření projektového plánu,

- realizace plánu projektu,
 - monitorování a kontrola prací na projektu,
 - provedení integrovaného řízení změn,
 - ukončení projektu (etapy).
- 2. Řízení rozsahu a rámce projektu:**
- inicializace projektu,
 - plánování a definice rozsahu projektu,
 - sběr požadavků,
 - vytvoření WBS,
 - validace a kontrola rozsahu projektu.
- 3. Řízení projektu v čase:**
- naplánování, vytvoření a kontrola harmonogramu projektu,
 - definice a struktura činností,
 - odhad zdrojů a pracnosti projektu.
- 4. Řízení nákladů projektu:**
- plánování a kontrola nákladů projektu,
 - odhad nákladů,
 - stanovení rozpočtu.
- 5. Řízení kvality projektu:**
- plánování, zajištění a kontrola kvality.
- 6. Řízení lidských zdrojů v projektu:**
- plánování řízení lidských zdrojů,
 - získávání lidských zdrojů,
 - tvorba, rozvoj a vedení projektového týmu.
- 7. Řízení komunikace v projektu:**
- plánování, řízení a kontrola komunikace.
- 8. Řízení rizik v projektu:**
- plánování řízení rizik,
 - identifikace rizik,
 - kvalitativní a kvantitativní analýza rizik,
 - plánování obrany,
 - kontrola a monitorování rizik.
- 9. Řízení zakázek a dodávek:**
- plánování, řízení, kontrola a uzavření dodávek.

10. Řízení zainteresovaných stran:

- identifikace zainteresovaných stran,
- plánování vztahu se zainteresovanými stranami,
- řízení závazků a kontrola vůči zainteresovaným stranám.

3.2.2. Mezinárodní standard PRINCE2

Mezinárodní standard PRINCE2 nyní vlastní společnost Axelos Limited, ale původně od roku 1989 vznikal pod křídly britského vládního úřadu OCG. Původně byl standard zaměřen na řízení IT projektů, nyní ho lze považovat za univerzální, je nejrozšířenějším na světě a lze ho použít při řízení jak malých, tak velkých projektů různých typů. PRINCE2 definuje sedm principů, sedm témat a sedm procesů. (vlastní zpracování podle [3])

Sedm principů PRINCE2

Principy jsou základními kameny standardu PRINCE2 a jsou pro něj specifické. Všech sedm principů je popsáno v následujícím seznamu. (vlastní zpracování podle [3])

1. Průběžné hodnocení odůvodnění realizace projektu:

- projekt by měl být podložen solidním obchodním případem (Business Case), který musí být řádně zdokumentovaný a schválený.

2. Učení se ze zkušeností:

- vytěžování znalostí během řízení projektů (Lessons Learned).

3. Definování rolí a odpovědností:

- projektový management není stejný jako liniový management,
- role a odpovědnosti jsou rozděleny do tří skupin (Business, User a Supplier).

4. Řízení po etapách:

- PRINCE2 rozděluje projekt do etap,
- počet etap závisí na velikosti, komplexitě a rizikovitosti projektu.

5. Řízení prostřednictvím výjimek:

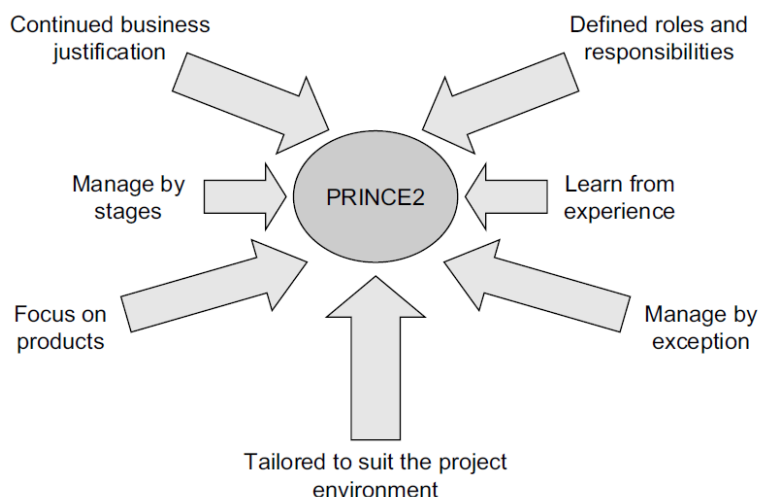
- PRINCE2 rozlišuje 4 stupně autority v projektu,
- autorita je delegována z jednoho stupně managementu do dalšího,
- každý stupeň managementu má přidělené toleranční limity (celkem 6).

6. Zaměření na produkt:

- PRINCE2 se soustředí na definování a dodávání produktů,
- plánování, kontrola a požadavky na kvalitu jsou na bázi produktu.

7. Přizpůsobení se projektovému prostředí v organizaci:

- standard je přizpůsoben projektovému prostředí, velikosti, rizikům, komplexitě, důležitosti a schopnostem všech zapojených lidí,
- projekt musí být vedený vzhledem jeho rozsahu, složitosti a významu.



Obrázek 4: Sedm principů PRINCE2 (převzato z [3], s. 4)

Sedm témat PRINCE2

Témata PRINCE2 slouží k popisu různých aspektů projektu, vzájemně spolu souvisí a jsou integrována do projektu. Témata jsou stručně popsána v následujícím seznamu. (vlastní zpracování podle [3])

1. Obchodní případ (Business Case):

- projekt by měl být podložen solidním obchodním případem,
- před schválením projektu by měla být ověřena existence obchodního případu,
- výsledný užitek by měl být definován v měřitelných podmínkách, aby ho bylo možné po dodávce produktu zkontrolovat.

2. Organizace (Organization):

- struktura projektového týmu s definovanými rolemi, odpovědnostmi a vztahy,
- role lze kombinovat a sdílet na základě velikosti a komplexity projektu.

3. Plány (Plans):

- PRINCE2 nabízí sérii úrovní plánů, které mohou být přizpůsobeny velikosti a potřebám projektu,
- přistupuje k plánování spíše z pohledu produktu, než aktivit.

4. Postup (Progress):

- soubor řídicích prvků, které umožňují stanovit klíčové informace pro rozhodování, díky tomu lze posuzovat vývoj projektu a určovat jeho další směr.

5. Riziko (Risk):

- jeden z hlavních faktorů, ovlivňující projekt.
- PRINCE2 definuje klíčové momenty, kde by mělo dojít k identifikaci a zhodnocení rizik,
- načrtává hlavní rysy analýzy a řízení rizik, sleduje rizika skrze všechny procesy.

6. Kvalita (Quality):

- zjištění požadavků na kvalitu od zákazníka,
- následování těchto požadavků vytvořením standardů a metod inspekce kvality.

7. Změna (Change):

- zahrnuje dvě navzájem se doplňující aktivity: řízení změn a řízení konfigurací produktů,
- stanovuje postupy, jak posuzovat a reagovat na události, které mohou mít na plán projektu a produkty dopad.



Obrázek 5: Sedm témat PRINCE2 (převzato z [3], s. 10)

Sedm procesů PRINCE2

PRINCE2 je procesně orientovaný standard, jednotlivé kroky projektového řízení jsou popsány sedmi procesy, chronologicky popisujícími průběh projektového řízení v projektu. Všechny procesy PRINCE2 jsou popsány v následujícím seznamu. (vlastní zpracování podle [3])

1. Směrování projektu (Directing a Project):

- autorizace příprav projektového plánu a obchodního případu,
- schválení projektu,
- kontrola proveditelnosti projektu,
- monitorování postupu a poskytování rad,
- zajištění kontrolovaného uzavření projektu.

2. Zahájení projektu (Starting up a Project):

- stanovení cíle projektu,
- návrh a jmenování řídicího týmu projektu,
- rozhodnutí způsobu, jak bude práce v projektu prováděna,
- potvrzení požadavků na kvalitu vyjádřených zákazníkem.

3. Nastavení projektu (Initiating a Project):

- vytvoření detailního plánu projektu.

4. Kontrola etapy (Controlling a Stage):

- autorizace práce, která má být provedena,
- shromažďování informací o postupu práce,
- monitorování a kontrola změn,
- posuzování situace,
- podávání zpráv,
- provádění nutných opravných akcí.

5. Řízení dodání produktu (Managing Product Delivery):

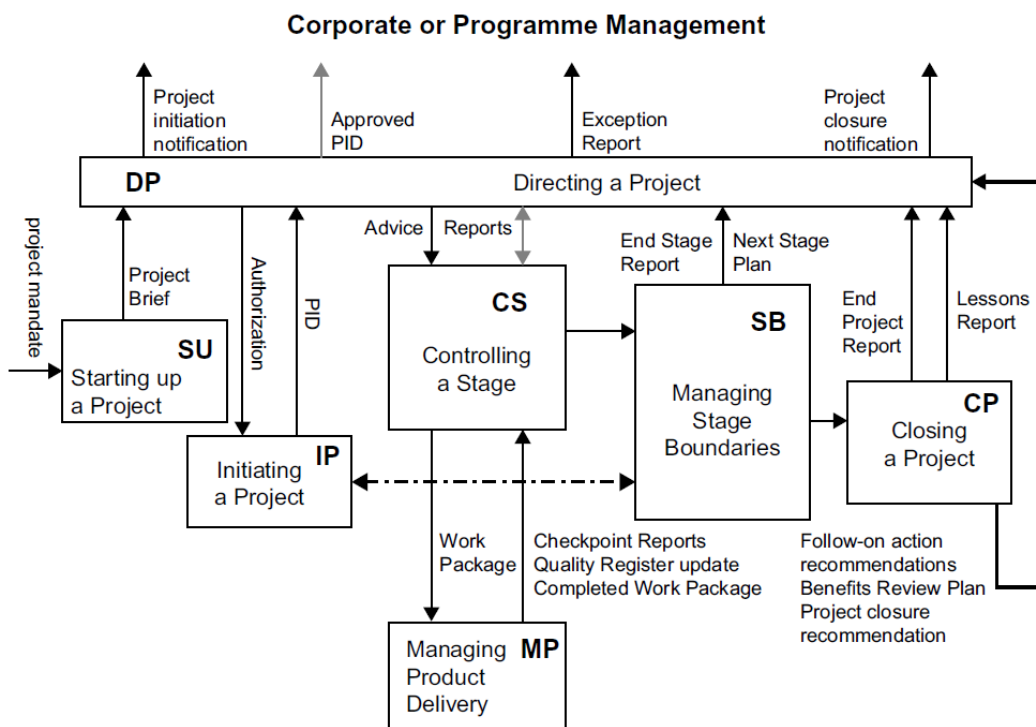
- plánování pracovních balíčků týmu,
- zajištění, že přiřazené pracovní balíčky jsou autorizované a potvrzené,
- zajištění toho, že je práce odpracována.

6. Řízení přechodu mezi etapami (Managing Stage Boundaries):

- plánování dalších etap,
- aktualizace projektového plánu a obchodního případu,
- aktualizace průzkumu rizik,
- podání zprávy o výsledku a výkonu právě ukončené etapy,
- získání schválení od řídicího výboru k započetí další etapy.

7. Ukončení projektu (Closing a Project):

- posouzení splnění cílů projektu,
- potvrzení zákaznickovy spokojenosti,
- potvrzení, že naplánovaná podpora a údržba je na svém místě,
- vypracování doporučení pro následující akce,
- zajištění toho, že aby všechny znalosti vytěžené během projektu byly uloženy a vhodně okomentované pro potřeby dalších projektů,
- podání zprávy, jestli aktivity projektového řízení byly úspěšné, nebo ne,
- připravení plánu pro kontrolu úspěchu finálního produktu.

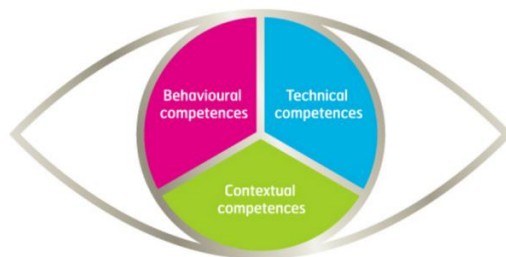


Obrázek 6: Sedm procesů PRINCE 2 (převzato z [3], s. 12)

3.2.3. Mezinárodní standard ICB

Standard ICB od mezinárodní asociace projektového řízení IPMA, je na rozdíl od předešlých standardů, standardem kompetenčním a ne procesním. (podle [4]) První verze certifikace vznikla roku 1998, v současné době už je dostupná třetí verze této certifikace. (podle [5]) Poslední aktualizovaná verze (ICB 4) tohoto kompetenčního standardu byla vydána na podzim roku 2015. Při vypracování této diplomové práce je však čerpáno z jeho předešlé verze (ICB 3.2).

Standard není zaměřen na přesnou podobu definovaných procesů a jejich konkrétní aplikaci, ale na schopnostech a dovednostech manažerů a členů jejich týmů. Kompetencí je celkem 46 a jsou rozděleny na odborné, behaviorální a kontextové. (podle [4])



Obrázek 7: Tři oblasti kompetencí dle ICB (převzato z [2], s. 5)

Kompetenční oblasti ICB:

1. Odborné (Technical competences) – 20 kompetencí,
2. Behaviorální (Behavioural competences) – 15 kompetencí,
3. Kontextové (Contextual competences) – 11 kompetencí.

Technické kompetence	Behaviorální kompetence	Kontextové kompetence
Úspěšnost řízení projektu	Vůdčovství	Orientace na projekt
Zainteresané strany	Zainteresanost a motivace	Orientace na program
Požadavky a cíle projektu	Sebekontrola	Orientace na portfolio
Rizika a příležitosti	Asertivita	Realizace projektu, programu a portfolio
Kvalita	Uvolnění	Trvalá organizace
Organizace projektu	Otevřenost	Byznys
Týmová práce	Kreativita	Systémy, produkty, technologie
Řešení problémů	Orientace na výsledky	Personální management
Struktury v projektu	Výkonnost	Zdraví, bezpečnost, ochrana života a životního prostředí
Rozsah a dodávané výstupy projektu	Diskuze	Finance
Čas a fáze projektu	Vyjednávání	Právo
Zdroje	Konflikty a krize	
Náklady a financování	Spolehlivost	
Obstarávání a smluvní vztahy	Porozumění hodnotám	
Změny	Etika	
Kontrola, řízení a podávání zpráv		
Informace a dokumentace		
Komunikace		
Zahájení		
Ukončení		

Tabulka 2: Kompetence ICB (převzato z [4], s. 29)

3.3. Agilní metody řízení projektů

Agilní metody řízení projektů jsou odpovědí na neefektivní a neúspěšné řízení IT projektů. Metody jsou založeny na předpokladu, že projekt je dynamický s nejasnými požadavky, které se v průběhu realizace projektu mohou měnit. Proto je také využíváno termínu „agilní“, vycházejícího z anglického slova „agile“ (hbitý, mrštný). Součástí této kapitoly jsou i vybrané metody agilních přístupů. Z důvodu rozsahu této diplomové práce zde nejsou uvedeny všechny, jen ty, které se ve vybrané společnosti vyskytují, nebo využívají. Nejsou zde uvedeny metody Feature Driven Development, Dynamic System Development a Crystal metody.

3.3.1. Definice agilního přístupu

Agilní metody jsou využívány převážně při řízení softwarových projektů, jejichž vývoj bývá dynamický a požadavky se často mění.

PMBOK definuje agilní přístupy řízení projektu, jako adaptivní životní cyklus, který očekává a umožňuje změny. Vyžaduje úzkou a trvající spolupráci se stakeholdery. Adaptivní životní cykly jsou iterativní a inkrementální. Iterace jsou velmi krátké, obvykle 2-4 týdny, jsou fixované v čase a zdrojích. (vlastní zpracování podle [1])

3.3.2. Manifest agilního vývoje softwaru

Snaha zefektivnit vývoj softwaru, přinést strukturu a jednotný přístup do systémového inženýrství, vedla k sepsání Manifestu agilního vývoje softwaru. Ten slouží k popsání základních principů agilního přístupu.

Manifest definuje dvanáct základních pilířů agilního vývoje softwaru a byl sepsán zakladateli Aliance pro agilní vývoj roku 2001. Mezi jeho nejznámější představitele patří zejména Alistair Cockburn, Jeff Sutherland, Ken Schwaber, Kent Beck, Martin Fowler, a Ward Cunningham. Pilíře agilního vývoje softwaru jsou popsány v následující podkapitole. (vlastní zpracování podle [6])

Dvanáct pilířů agilního vývoje

- 1) Prioritou je včasná a průběžná dodávka funkčního softwaru, který přinese zákazníkovi hodnotu.
- 2) Změny požadavků a zpětná vazba v pozdních fázích vývoje je vítaná. Agilní metodiky předpokládají změny, soustředí se jen na důležité a potřebné.
- 3) Iterativní vývoj je podporován, jsou upřednostňovány krátké iterace (2 týdny až 2 měsíce) a tudíž časté dodávky softwaru zákazníkovi.
- 4) Denní spolupráce zákazníků a vývojářů na projektu. Na začátku spolupráce jsou definovány pouze hrubé požadavky, které se upřesňují během vývoje podle potřeb zákazníka.
- 5) Rozhodování provádí kompetentní a pozitivně motivovaní jedinci. Důvěra ve schopnosti zaměstnanců a jejich podpora. Zaměstnanci, kteří jsou správně motivováni a mají dobré podmínky, jsou důležitým faktorem pro úspěšné řízení projektů.
- 6) Pro porozumění problému je nejdůležitější verbální a přímá komunikace, nikoliv psaní a čtení.
- 7) Prioritou je fungující software a ne naplnění specifikace. Splnění specifikací, neznamená, že byl projekt úspěšně dokončen.
- 8) Agilní procesy podporují udržitelný rozvoj. Přetěžování zaměstnanců je jen krátkodobým řešením, z dlouhodobého hlediska může mít za následek sníženou produktivitu práce.
- 9) Důležitá je stálá pozornost kvalitnímu návrhu a technickému řešení. Návrh je souvislá, každodenní činnost, která zasahuje do všech vývojových fází projektu.
- 10) Důraz na jednoduchost – umění maximalizovat množství nevykonané práce. Je kladen důraz na jednoduché postupy. Snaha zahrnout do řešení jen to, co je prokazatelně potřebné.
- 11) Nejlepších výsledků dosáhnou samo-organizující se týmy. Podpora kreativity a důvěra přináší výjimečné návrhy a výsledky práce.
- 12) Týmy se v pravidelných intervalech zabývají tím, jak nejefektivněji odvádět svou práci. Stejně jako agilní metodiky se týmy následně vyvíjejí a přizpůsobují.

3.3.3. Scrum

Význam anglického slova „scrum“ má původ ve sportu rugby a doslova znamená „mlýn“, „skrumáž“. Scrum je inkrementální, iterativní a empiristická metoda, poprvé představená roku 1995 Jeffem Sutherlandem a Kenem Schwaberem. Autoři čerpali z publikace „The New New Product Development Game“ od autorů Takeuchiho a Nonaky (1986), kteří popsali úspěšnost řízení projektů v japonských společnostech jako je Honda, Canon a Fuji-Xerox. Původně byla metoda Scrum formalizována pro potřeby vývoje softwarových projektů, ale je vhodná pro jakékoliv jiné, stejně komplexní a inovativní projekty. Cílem této metody je optimalizovat předvídatelnost a průběžně řídit rizika. (vlastní zpracování podle [7])

Principy metody Scrum

Metoda Scrum čerpá přímo z principů Agilního Manifestu (vlastní zpracování podle [8]):

- Product Owner vytváří seznam přání, který se nazývá Product Backlog (prioritizovaný),
- během plánování sprintu vybírá tým z Product Backlogu malou část, tvoří tak Sprint Backlog a rozhoduje, jak tuto část bude implementovat,
- tým má k dispozici na dokončení práce určitý čas, který se nazývá sprint (běžně dva až čtyři týdny), tým se ale schází na denních schůzkách (denní standupy),
- Scrum Master se snaží udržet tým soustředěný na určený cíl,
- na konci každého sprintu by projekt měl být potencionálně předatelný zákazníkovi, nebo představitelný stakeholderovi,
- sprint končí posouzením sprintu a retrospektivou,
- na začátku každého sprintu se tým rozhoduje, jakou část Product Backlogu zpracuje a začíná znovu pracovat.

Základní pilíře implementace metody Scrum

Jak již bylo řečeno, Scrum je inkrementální, iterativní a empiristická metoda, jejíž implementace je založena na následujících třech pilířích, a to transparentnosti, kontrole a adaptaci. (vlastní zpracování podle [9])

1) **Transparentnost (Transparency)**

- Všechny důležité aspekty procesů musí být zřetelné a jednotně definovány (standardizovány). Všichni, kdo se na projektu zúčastní, musí používat stejné výrazy a jazyk.

2) **Kontrola (Inspection)**

- Uživatelé Scrumu musí často kontrolovat artefakty a průběh Scrumu vzhledem k cíli sprintu, aby mohlo dojít k včasnému rozpoznání nežádoucích změn a odchylek. Ke kontrole by mělo docházet jen tak často, aby to nestálo v cestě průběhu práce na projektu.

3) **Adaptace (Adaptation)**

- Pokud kontrolor zjistí, že se jeden nebo více aspektů procesu odchyluje od akceptovatelných limitů tak, že výsledný projekt bude neakceptovatelný, musí být proces přizpůsoben. Přizpůsobení procesu musí být provedeno co nejdříve, aby byla minimalizována jakákoliv další odchylka.

Scrum tým

Tým ve Scrumu je multifunkční a organizuje se sám. Je tvořen Product Ownerem, vývojovým týmem a Scrum Masterem. Týmy pro dosažení stanovených cílů mají dostupné všechny potřebné kompetence a schopnosti. Samy si vybírají, co je nejvhodnější pro dosažení stanovených cílů. Týmy ve Scrumu jsou navrženy tak, aby byly co nejvíce flexibilní, kreativní a produktivní. (vlastní zpracování podle [9])

Product Owner je jedna osoba, zodpovědná za maximalizaci hodnoty produktu a práce vývojového týmu. Tvoří Product Backlog, který popisuje vizi a ideu produktu. Na základě Product Backlogu je prováděna práce vývojovým týmem. Aby mohl být Product Owner úspěšný, celá organizace musí podporovat jeho rozhodnutí. (vlastní zpracování podle [9])

Vývojový tým je složen z profesionálů, kteří pracují na zhotovení produktu, který by měl být připraven k dodání na konci každého sprintu. Vývojové týmy jsou strukturované tak, aby se samy organizovaly a řídily svou práci. Výsledná synergie optimalizuje celkovou účinnost a efektivitu týmu. Týmy jsou většinou malé, protože velké týmy generují příliš velkou komplexitu pro empirické procesy. (vlastní zpracování podle [9])

Scrum Master je zodpovědný za to, že je Scrum správně pochopen a přijat. Toho je v praxi dosaženo tím, že se Scrum tým drží Scrum teorie, praktik a pravidel. Pomáhá a poskytuje služby nejen Product Ownerovi, vývojovému týmu, ale i celé organizaci (vlastní zpracování podle [9]):

- služby pro Product Ownera:
 - o nalezení technik potřebných pro efektivní tvorbu Product Backlogu,
 - o pomáhání týmu pochopit potřebu jasných a stručných částí Product Backlogu,
 - o pochopení produktového plánování a empirického prostředí,
 - o zajištění toho, že Product Owner ví, jak maximalizovat hodnotu Product Backlogu,
 - o pochopení a praktikování agilního přístupu,
 - o umožňování Scrum aktivit, jak je žádáno, nebo jak je potřeba.
- služby pro vývojový tým:
 - o koučování vývojového týmu v sebe organizaci,
 - o pomáhání vývojovým týmům vytvořit kvalitní produkty,
 - o odstraňování překážek průběhu práce vývojového týmu,
 - o umožňování Scrum aktivit, jak je žádáno, nebo jak je potřeba,
 - o koučování vývojového týmu v organizačních prostředích, kde není Scrum plně adaptovaný a pochopený.
- služby pro organizaci:
 - o vedení a koučování organizace v adaptaci Scrumu,
 - o plánování implementace Scrumu v organizaci,
 - o pomáhání zaměstnancům a stakeholderům pochopit, jak se držet Scrumu a empirického vývoje produktu,
 - o způsobování změn, které napomáhají zvýšení produktivity Scrum týmu,
 - o spolupracování z ostatními Scrum Mastery tak, aby byla aplikace Scrumu v organizaci co nejefektivnější.

Scrum aktivity

Předepsané aktivity jsou v metodě Scrum využívány k vytvoření pravidelnosti a minimalizaci potřeby porad, které nejsou v této metodě definovány. Všechny aktivity

jsou časově vymezené a mají stanovenou maximální dobu trvání. Jakmile sprint začne, jeho doba trvání je fixní a nelze ho zkrátit ani prodloužit. Sprint obsahuje všechny ostatní aktivity, které jsou speciálně navrženy tak, aby umožňovaly transparentnost a kontrolu. Nezahrnutí těchto aktivit do vývoje má za následek snížení transparentnosti, ztráty příležitosti ke kontrole a přizpůsobení se. (vlastní zpracování podle [9])

Sprint je jádrem Scrumu, je obvykle dlouhý dva až čtyři týdny a má na základě Product Backlogu stanovený cíl. Začíná hned poté, co byl předešlý sprint ukončen. Každý sprint zahrnuje plánování sprintu, denní standupy (denní Scrum), vývoj, posouzení a retrospektivu sprintu. Každý sprint lze chápat jako projekt s dobou trvání kratší, než jeden měsíc. Sprint může být zrušen jen před uplynutím své doby trvání, jediný kdo k tomuto má pravomoc je Product Owner. (vlastní zpracování podle [9])

Plánování sprintu probíhá na začátku sprintu a definuje vše, co a jakým způsobem má být v jeho rámci dodáno. Doba plánování by neměla překročit osm hodin a je řízena Scrum Masterem. (vlastní zpracování podle [9])

Denní standupy jsou maximálně patnáct minut dlouhá setkání, sloužící vývojovému týmu k jeho synchronizaci. Během standupu jsou synchronizovány aktivity týmu a je vytvořen plán na dalších 24 hodin. Toho je dosahováno na základě kontroly provedené práce, prezentace toho, co bude prováděno a problémů, které zasahují do průběhu práce. (vlastní zpracování podle [9])

Posouzení (vyhodnocení) sprintu je prováděno na jeho konci, prostřednictvím spolupráce Scrum týmů a všech stakeholderů. Vyhodnocení slouží ke kontrole dosaženého přírůstku během sprintu, a pokud je potřeba, tak k přizpůsobení Product Backlogu. Vyhodnocení sprintu jsou informativní schůzky, které slouží hlavně k prezentaci dosaženého, vytěžení zpětné vazby a podpoře spolupráce. Jejich výsledkem je revidovaný, nebo pozměněný Product Backlog, který je podkladem pro další sprint. (vlastní zpracování podle [9])

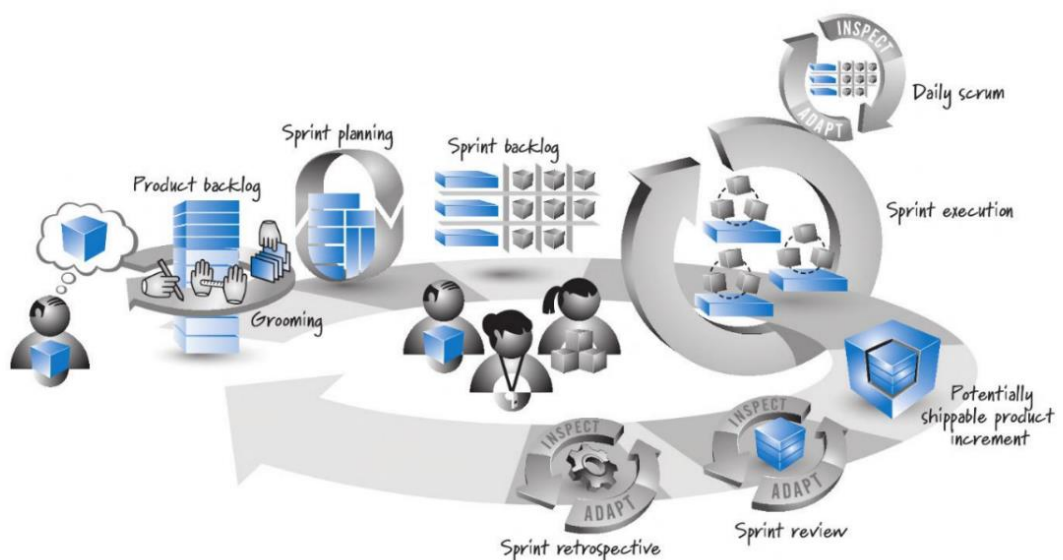
Retrospektiva sprintu je příležitostí ke zhodnocení práce Scrum týmu a vytvoření plánu, jak se následně zlepšit. Retrospektiva probíhá mezi vyhodnocením a plánováním sprintu a je přibližně tři hodiny dlouhá. Dohlíží na ni Scrum Master, který zajišťuje, že jsou všichni přítomni a rozumí důvodu, proč se schůzka koná.

Dochází ke kontrole toho, jak sprint proběhl z hlediska vztahů, procesů a nástrojů.
(vlastní zpracování podle [9])

Scrum artefakty

Scrum artefakty jsou nástroje speciálně navržené k maximalizaci transparentnosti klíčových informací tak, aby je každý člen Scrum týmu i stakeholdeři chápali stejně. Mezi scrum artefakty patří (vlastní zpracování podle [9]):

- **Product Backlog:**
 - prioritizovaný seznam požadavků a potřebných částí, vlastností, vylepšení a funkcionalit produktu,
 - vytvářený Product Ownerem,
 - obsahuje prioritu, popis a odhad pracnosti,
 - není nikdy finální, je dynamický (mění se mezi sprinty).
- **Sprint Backlog:**
 - tvořen vybranou částí s nejvyšší prioritou z Product Backlogu, která je poté specifikována, rozdělena na aktivity a přehodnocena z hlediska pracnosti.



Obrázek 8: Scrum (převzato z [9], s. 17)

3.3.4. Kanban

Význam japonského slova „kanban“ je doslova „cedule“, „tabule“. (vlastní zpracování podle [10])

Metoda Kanban byla vyvíjena společností Toyota, jako součást jejího produkčního systému od 40. let 20. století a sloužila k naplnění výrobního přístupu „Just in Time“. Metoda je nejčastěji používána formou papírových lístků, zprostředkávajících zpětný tok informací mezi procesy, které lze roztrždit do tří kategorií: sběr informací, transfer informací a produkce informací. Lze tak monitorovat tok jednotek v produkčním systému, spotřebu a řídit produkci. (vlastní zpracování podle [11])

Kanban je v současnosti využíván různými způsoby. Kanban je nejen systémem vývoje softwaru, založeným na inkrementálním a evolučním vývoji, ale také jako „karta“, sloužící k zobrazení a spravování seznamu úkolů, formou tabule, nebo softwaru. (vlastní zpracování podle [12])

Pět klíčových vlastností systému Kanban

1. Vizualizace průběhu prací:

- Vizualizace usnadňuje pochopení systému práce, komunikaci a koordinaci týmu. Je realizována prostřednictvím tabule s lístky, představujícími jednotlivé úkoly a požadavky, umístěnými na tabuli dle stavu jejich zpracování. Tabule je průběžně aktualizována a tvořena na základě předem definovaných pravidel. (vlastní zpracování podle [13])

2. Omezení množství rozpracované práce:

- Omezení množství rozpracované práce probíhá z důvodu jejího omezení, protože každý systém má svou kapacitu. Systém je nastaven tak, že vždy když je dostupná volná kapacita, tak nový úkol, nebo požadavek přijme. Nedochází tím tak k zahlcení systému ani členů týmu úkoly a požadavky. (vlastní zpracování podle [13])

3. Řízení toku úkolů:

- Řízení toku úkolů slouží k monitorování a podávání zpráv o jejich rychlosti a plynulosti. Hlavním důvodem tohoto sledování je dosažení maximální rychlosti zpracování při maximální plynulosti, a tak snížení rizika

zvýšených nákladů způsobených prodlením. (vlastní zpracování podle [12])

4. Explicitní popis zásad procesu:

- Každý proces, který chceme vylepšit je nutno nejprve pochopit a explicitně popsat. Výsledkem toho je vyšší objektivita diskuzí a tím pádem snazší spolupráce uvnitř týmu. (vlastní zpracování podle [12])

5. Společná spolupráce na zlepšení:

- Systematický přístup k zlepšování procesů a samoučící organizace je základem pro vytváření předpovědí dopadu případných a navrhovaných změn. Předpovědi by měly být ověřeny empirickými daty. (vlastní zpracování podle [12])

3.3.5. Lean Software Development

Význam anglického výrazu „lean“ je doslova „štíhlý“. Stejně jako metoda Kanban, tak Lean Manufacturing má původ ve společnosti Toyota ve 40. letech 19. století v jejím produkčním systému. Lean Software Development má původ v knize stejného názvu od manželů Poppendieckových, kteří z filozofie Lean Manufacturing čerpají a využívají ji pro vývoj softwaru. (vlastní zpracování podle [14])

Sedm principů Lean Software Development

Sedm principů, které systém vymezují a popisují, definovali již zmínění manželé Poppendieckové ve své publikaci „Lean Software Development: An Agile Toolkit“ (vlastní zpracování podle [14]):

1. Eliminovat odpad:

- Vše co nevytváří v danou chvíli hodnotu pro zákazníka je odpadem, který je nutno eliminovat.
- Nástroje využívané pro eliminaci odpadu:
 - o **Sledování odpadu:**
 - Nástroj je tvořen seznamem oblastí, ve kterých se může během výroby, nebo vývoje odpad tvořit. Uvedený seznam obsahuje inventář funkcionalit, extra procesy, extra

funkcionality, přecházení pracovníků mezi úkoly, čekání, přesun požadavků, chyby a aktivity managementu.

○ **Mapování toku hodnoty produkty:**

- Nástroj slouží k identifikaci odpadu a je založen na mapování procesů v čase. Sleduje se, kdy dochází k časovým prostožům a kdy se produkt stává pro zákazníka hodnotným.

2. Zesilovat učení:

- Vývoj softwaru je průběžný učící se proces založený na iteracích, během kterých je psán kód. Učící proces je usnadněn a zrychlen využitím zkrácených iterací, které obsahují testování a konzultování se zákazníkem. Díky tomu je šíření informací mezi vývojovým týmem a zákazníkem nejen snazší, ale také rychlejší.

3. Rozhodovat se co nejpozději:

- Vývoj softwaru je vždy spojen s určitou mírou nejistoty, lze dosáhnout lepších výsledků, pokud rozhodnutí odkládáme, tak dlouho, že je provádíme na základě faktů a ne nejistých předpokladů. Princip je opět podporován iterativním systémem vývoje, kdy adaptujeme změny a opravujeme chyby.

4. Dodávat co nejrychleji:

- Vyšší flexibilita je zajištěna rychlými dodávkami produktu zákazníkovi. Čím rychleji je produkt dodán bez větších chyb a odchylek, tím rychleji může být obdržena zpětná vazba a následně začleněna do další iterace. Vytěžování znalostí a komunikace je efektivnější, pokud jsou iterace co nejkratší. Pokud je produkt dodáván rychle, jsou naplňovány současné požadavky zákazníka a klesá tak pravděpodobnost, že by během vývoje změnil názor, nebo měnil požadavky.

5. Posilovat tým:

- Posilování týmu je založeno na tom, že týmy za sebe rozhodují samy, mají přístup k zákazníkovi a dělají svou práci, jak jim to nejvíce vyhovuje. Manažeři jsou vedeni k tomu, aby dávali vývojovým týmům prostor se vyjádřit nejen ohledně toho, jaké akce je možné provést, ale i jak je zlepšit. Vedoucí týmu poskytuje podporu, pomoc a motivaci v obtížných situacích.

6. Vytvářet integritu:

- Zákazník musí vnímat celkovou integritu systému, nejen kolik stojí, jak je přístupný a intuitivní, ale také jak je oznamován, dodáván, nasazen a jak funguje. Integritu lze rozdělit na konceptuální a vnímanou. Konceptuální integrita představuje, jak funguje vzájemná spolupráce všech částí systému. Systém by měl být flexibilní, udržitelný, konzistentní, efektivní a mělo by na něj být pohlíženo, jako na celek. Vnímaná představuje, jak je systém vnímán uživatelem, nebo zákazníkem. To znamená jak je systém srozumitelný, spolehlivý, použitelný a jaká je jeho cena.

7. Vidět celek:

- Softwarové systémy nejsou jen součtem jejich jednotlivých částí, ale i jejich interakcí s okolím. Interakce je nutno testovat a u zjištěných problémů se musí řešit prvotní příčina a ne jen symptomy její části. Čím větší systém je, tím více je potřeba organizace jeho vývoje a více částí je vyvíjeno různými týmy. Každý vývojový tým by však měl být schopen na softwarové systémy nahlížet jako na celek a shromažďovat informace o jejich dopadu nejen na okolní systémy, ale i na celý obchodní systém.

3.3.6. Extrémní programování (XP)

Extrémní programování, jinak známé pod zkratkou „XP“, je snadnou, účinnou, flexibilní, vědeckou a nízkorizikovou metodou, jak vyvíjet software. Byla poprvé popsána v roce 2000 Kentem Beckem. Metoda je vhodná pro menší vývojové týmy (2-10 členů) a je založena na dvou základních předpokladech. Užití této metody přinese efektivitu, schopnost sledovat průběh vývoje projektu a provádět během jeho vývoje změny. Vývojový tým může pracovat na důležitých úkolech, nemusí se soustředit na nic jiného. Extrémní programování závisí na zautomatizovaných testech, které jsou naprogramovány za účelem monitorování průběhu vývoje, aby se mohl systém nadále rozvíjet a chyby i odchylky byly zavčas identifikovány. Dále je důležitá komunikace a úzká spolupráce programátorů. Podle knihy „Extreme Programming Explained“ od Kenta Becka, stojí extrémní programování na čtyřech hodnotách, a to komunikaci, jednoduchosti, zpětné vazbě a odvaze. Dále Kent Beck definuje pět hlavních principů a dvanáct praktik extrémního programování. Tyto definice jsou stručně definovány v následujících seznamech a textu. (vlastní zpracování podle [15])

Čtyři hodnoty XP

1. Komunikace:

- Ke špatné komunikaci nedochází náhodou, existuje mnoho okolností, které vedou k defektu v komunikaci. Extrémní programování se snaží zajistit plynulý a dostatečný tok informací mezi vývojovým týmem, vedením a zákazníkem. O monitorování, údržbu a obnovování informačních toků se stará týmový kouč.

2. Jednoduchost:

- Je optimální vypracovat jednodušší řešení a následně ho přepracovat do komplexnějšího, než vypracovat komplexní řešení, které nemusí být ve finále využito. Jednoduchost a komunikace mají vzájemný a podporující se vztah. Pokud dochází k dobré komunikaci, je všem jasnější, jaká práce musí být odvedena a jaká naopak být odvedena nemusí. Čím jednodušší systém je, tím méně je potřebné komunikovat. K co nejjednodušším řešením se snaží vývojový tým motivovat týmový kouč.

3. Zpětná vazba:

- Konkrétní zpětná vazba o aktuálním stavu systému je absolutně klíčová a k nezaplacení. Aktuální stav systému lze zjistit testováním a zpětnou vazbu je možné rozčlenit do dvou časových soustav, v jejichž rozsahu funguje. V řádu minut a dnů slouží zpětná vazba k srovnání skutečně provedené práce s plánem. Zpětná vazba je získávána prostřednictvím testů a odhadů nových požadavků zákazníka. V řádu týdnů a měsíců zpětná vazba poskytuje informace o skutečné rychlosti vývojového týmu, což je potřebné při úpravách plánu. Další zpětná vazba je poskytována po určitých časových úsecích (například na konci iterace) zákazníkem.

4. Odvaha:

- Její hodnota nejvíce závisí na dodržování předešle uvedených hodnot. Během vývoje musí tým čelit rozhodnutím, která mohou vést ke zdržení, nebo kompletnímu zrušení a opuštění vývoje.

Principy XP

1. Rychlá zpětná vazba:

- Zpětná vazba musí být vytěžována a vyhodnocena, aby mohly být změny implementovány co nejrychleji.

2. Předpoklad jednoduchosti:

- Problémy by měly být řešeny jednoduše, znovu použitelnost kódu by neměla být brána v potaz.

3. Přírůstková změna:

- V projektu by neměly probíhat velké změny, a ty by měly být rozděleny do menších částí a implementovány postupně.

4. Využití změny:

- Akceptovaná změna by měla být soustředěna vždy na hlavní problém a současně poskytnout co nejvíce alternativ.

5. Kvalitní práce:

- Všichni pracovníci by měli chtít odvádět kvalitní práci.

Praktiky XP

1. Plánovací hra:

- Rychlé určení rozsahu další verze kombinováním obchodních priorit a technických odhadů. Plán je nutno aktualizovat, pokud ho skutečnost předbíhá.

2. Malé verze:

- Rychlé vydávání jednoduchého systému do produkce, poté vydávat nové verze v krátkých intervalech.

3. Metafora:

- Řízení vývoje prostřednictvím jednoduchého příběhu, který popisuje, jak celý systém funguje.

4. Jednoduchý design:

- Systém by měl být navržen co nejjednodušeji, jak je to možné. Jakákoliv objevená komplexita, která je v systému navíc, je ihned odstraněna.

5. Testování:

- Programátoři průběžně píší testy, které musí být bezchybné, aby mohl vývoj pokračovat. Zákazníci píší testy, které popisují, jaké funkce systému jsou hotovy.

6. Refactoring:

- Programátoři restrukturalizují systém, aby se zbavili duplikací, zlepšili komunikaci, systém zjednodušili, nebo přidali flexibilitu. To vše bez změny chování systému.

7. Párové programování:

- Veškerý kód je programován dvěma programátory na jednom počítači.

8. Společné vlastnictví:

- Kdokoliv může změnit jakoukoliv část kódu, kdy se mu zachce.

9. Neustálá integrace:

- Systém je integrován a stavěn pokaždé, když je dokončen úkol (i několikrát denně).

10. Čtyřicetihodinový týden:

- Nikdy by nemělo být odpracováno více než čtyřicet hodin týdně. Nikdy by nemělo být pracováno přesčas nepřetržitě po dva týdny.

11. Zákazník na pracovišti:

- Začlenění uživatele, nebo zákazníka do týmu, aby mohl poskytovat cennou zpětnou vazbu a odpovědi na otázky.

12. Standardizované psaní zdrojového textu:

- Programátoři píší kód, tak aby se shodoval se stanovenými pravidly a postupy.

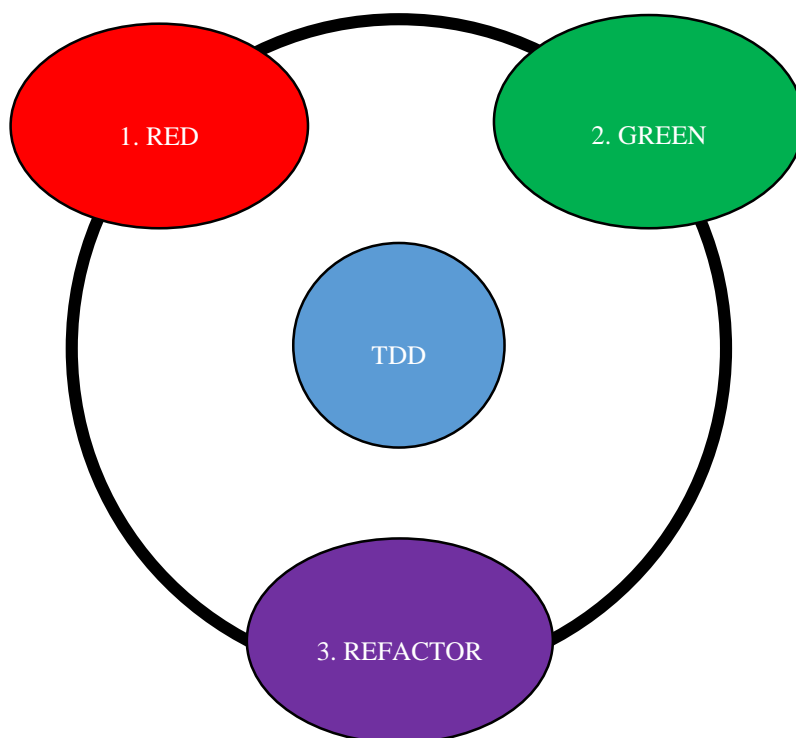
3.3.7. Test Driven Development

Do češtiny je Test Driven Development překládán jako vývoj řízený testy. Cílem této metody je čistý kód, to znamená fungující kód bez jakýchkoliv nadbytečných součástí. Takový kód je velmi předvídatelný, poskytuje jistotu a uspokojení nejen programátorovi, ale i uživateli. Čistý kód je prostřednictvím této metody vyvíjen pomocí zautomatizovaných testů. To znamená, že dochází k testování ještě před implementací. Základními pravidly TDD je eliminování duplicity a vyvíjení nového kódu, jen pokud testy selhaly. Aby tohoto bylo dosaženo, je nutné tvořit návrhy

organicky, v běžícím kódu, který poskytuje zpětnou vazbu mezi jednotlivými rozhodnutími. Testy si musí vývojový tým programovat sám, je to snadnější a efektivnější. (vlastní zpracování podle [16])

Mantrou TDD je rozlišení pořadí úkolů v programování do tří částí (tzv. rychlý rytmus) a to (vlastní zpracování podle [16]):

- RED – Napište malý test, který nefunguje (neprojde).
- GREEN – Test rychle zprovoznit, je dovoleno cokoliv.
- REFACTOR – Eliminovat duplicity.



Obrázek 9: Mantra TDD (vlastní zpracování podle [16])

4. Praktická část

Vlastní práce popisuje stávající stav agilního řízení projektů ve vybrané společnosti a prezentuje vlastní návrh pro další rozvoj. Snaží se tak rozebrat a popsat, jak probíhá agilní řízení projektů v praxi velké české společnosti. Vlastní návrhy jsou prezentovány a diskutovány v praxi vybrané společnosti. Společnost je v diplomové práci anonymizována.

Pro popsání stávajícího stavu agilního řízení projektů, využívá práce řízených rozhovorů se zaměstnanci vybrané společnosti a vlastního šetření.

4.1. Charakteristika vybrané společnosti

4.1.1. Popis a historie společnosti

Popisovaná akciová společnost působí na českém trhu od 90 let, je zaměřena na IT odvětví, převážně na obsahové generátory, webové portály a tvorbu aplikací. Společnost podporuje inovace a je známá svou otevřenou a neformální podnikovou kulturou. Projektové řízení bylo ve společnosti po většinu její existence, nyní se řídí převážně dle metody Scrum, která je ve společnosti aplikována přibližně třetím rokem. Projekty však neřídí jen dle této metody, ale využívá i různých kombinací ostatních agilních metodik podle potřeb jednotlivých týmů a řízených projektů. Ve společnosti tak lze nalézt prvky Kanbanu, Leanu, Lean startupu, Design sprintu, TDD (Test Driven Development) a XP (Extreme Programming). Řadí se tak mezi jednu z mála českých společností, která v projektovém řízení využívá agilních metodik. Ve společnosti pracuje přibližně tisíc zaměstnanců, z čehož třetinu tvoří vývojáři. (vlastní zpracování podle [17])

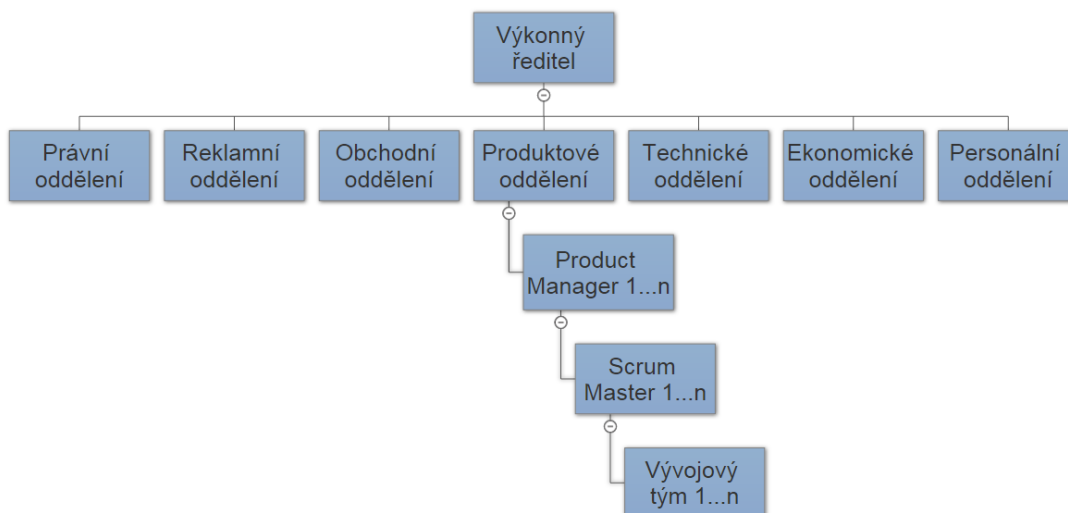
4.1.2. Podnikatelské aktivity a oblast trhu

Společnost řídí a zadává převážně „in-house“ projekty, to znamená, projekty iniciované i realizované pro její vlastní potřeby. Projekty, nebo jejich části jsou outsourcovány jen ve výjimečných případech. Jedná se o IT projekty, převážně interní vývoj softwaru, obsahové generátory a webové portály, včetně technologického

řešení, grafické a obchodní stránky. Svým zákazníkům poskytuje technické zázemí (servery, hosting, testování) a pomoc s vývojem (poradenství). Dále poskytuje služby týkající se marketingové podpory a podpory prodeje, kde pomáhá s návštěvností, tvorbou vlastní marketingové strategie a je schopna zprostředkovat obchod s desítkami tisíc zákazníků v B2B segmentu. Společnost při vývoji využívá vlastních a opensource technologií. Významnými aktivitami společnosti je i charitativní činnost a mediální partnerství. Získává nejen finanční podporu například pro Nadaci Naše dítě, Nadaci Terezy Maxové dětem, Linku bezpečí, sbírku Bílá Pastelka, sdružení ADRA, sdružení Zdravotní klaun aj. (vlastní zpracování podle [17])

4.1.3. Organizační struktura společnosti

Organizační struktura vybrané společnosti je maticová, na úrovni týmu a řízení projektu je více plochá. Plochá organizační struktura vyhovuje agilnímu řízení projektů, protože je díky rychlému toku informací velmi pružná v rozhodování. Hlavním představitelem společnosti je výkonný ředitel, jemuž podléhá vrcholový management. Vrcholový management společnosti zahrnuje ředitele jednotlivých oddělení, mezi které patří například produktový ředitel, který je Product Ownerem projektu a spadá pod něj vše, co se týká projektového řízení v organizaci, včetně Scrum Masterů a jednotlivých týmů. Je nutné zmínit, že projektové týmy v agilním řízení nepodléhají organizační struktuře tolik, jako ve společnostech, kde jsou projekty řízeny tradičně, mají větší volnost a měly by za projekt nést zodpovědnost. To závisí na vrcholovém managementu a jeho důvěře, jak ve své zaměstnance, tak v agilní přístupy obecně. Do vrcholového managementu dále spadá reklamní, obchodní, personální, technické, ekonomické a právní oddělení. Celá organizační struktura společnosti je stručně prezentována na následujícím obrázku. (vlastní zpracování podle [17])



Obrázek 10: Organizační struktura společnosti (vlastní zpracování podle [17])

4.1.4. Komunikace ve společnosti

Komunikace ve společnosti probíhá v několika fázích. Týmy realizující projekty spolu komunikují na denní bázi, formou standupů. Dále se schází na plánovacích a předplánovacích schůzích, které se zabývají úkoly daného sprintu (iterace). Na konci každého sprintu probíhá představení dema projektu, jehož úkolem je prezentovat odvedenou práci Product Managerovi.

Dále ve společnosti komunikace probíhá prostřednictvím celofiremního komunikátoru „Jabber“ a ostatních prostředků, které se liší podle potřeb týmů, nebo jednotlivců (telefon, Skype, Slack, TeamSpeak). (vlastní zpracování podle [17])

4.2. Projektové řízení ve společnosti

4.2.1. Projektová kancelář ve společnosti

Projektová kancelář jako taková ve společnosti není, její funkci částečně přebírá Product Manager, který má vizi finálního produktu, plán jak má vypadat, určuje jeho účel a jedná s externími partnery. Také určuje prioritu projektů a je oprávněný přerozdělovat členy týmů, což může mít za následek problémy se zaučováním nového člena v průběhu projektu a tak projekt pozdržet. Product Manager má, kromě předprojektové fáze a iniciace projektů, na starosti kontrolu, monitorování a ověřování výstupů. Na základě dohody a konzultace se Scrum Masterem a týmem

určuje, jaká dokumentace je při řízení projektu tvořena a jak. Nejčastěji Backlog projektu (idea, vize), Demo (představení výsledků iterace), Mindmapy (milníky a vstupy třetích stran), někdy i dokumentace kódu a průběhu práce na projektu. Jako jediný má pravomoc zrušit projekt (sprint).

Dále existuje tým Scrum Masterů, kteří vedou vývojářské týmy pracující na projektech. Scrum Master také funguje jako zástupce Product Managera, pokud je nemocný, nebo je na dovolené. Znalosti a manažerskou praxi zaměstnancům poskytují interní lektori, projektovou kulturu šíří jednotliví Scrum Masteři. (vlastní zpracování podle [17])

4.2.2. Základní principy projektového řízení ve společnosti

Jak už bylo uvedeno, společnost řídí projekty dle agilních metodik a jejich kombinací. K tomuto rozhodnutí vedla skutečnost, že společnost vyvíjí převážně IT projekty a agilní metodiky ji poskytují významné benefity. Product Manager (Product Owner) má průběžný přehled o dokončení projektu, vidí výsledky práce a může rychle reagovat na změny. Vývojáři mají pocit stability, motivaci dokončit zadanou práci ve sprintu (na konci sprintu musí představit Demo) a rychle se odhalí úzká hrdla (Velocity, Retro).

Společnost je projektově orientovaná, ale nepoužívá žádnou vlastní směrnici ani mezinárodní standard projektového řízení. Každý tým je řízen trochu jinak a produkuje jiné výstupy (včetně dokumentace). Týmy se tvoří samy, každý pracuje na tom, co ho baví a na čem pracovat chce.

Ve společnosti probíhá srozumitelná, častá a cílená komunikace. Agilní metodiky v sobě zahrnují základní principy řízení a komunikace se zákazníkem. Každý den jsou uskutečňovány „stand up“ meetingy, na základě kterých každý člen týmu ví, kdo na čem právě pracuje, co má za úkol a jaké má problémy. Avšak komunikace v týmu není vždy úplně přímá a jednotná. Alokace zdrojů a kapitálu v projektech není řešena, společnost má dostatek prostředků na experimentování. I přes to, že se projektové řízení ve společnosti neřídí žádnými směrnicemi ani mezinárodními standardy a má jen částečnou podporu od top managementu, lze hovořit o tom, že ve společnosti existuje projektová kultura. (vlastní zpracování podle [17])

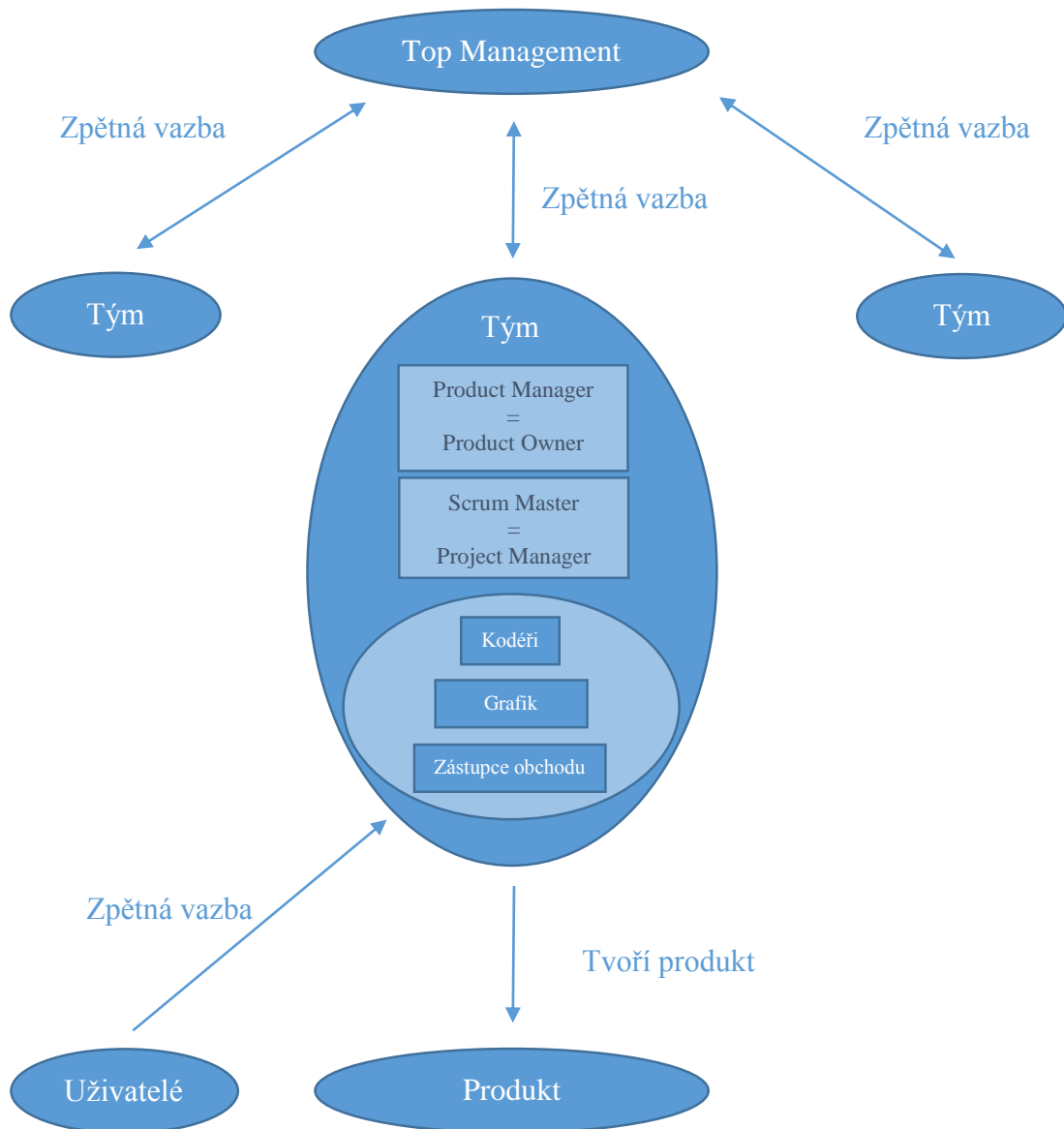
Společnost se při řízení projektů řídí heslem „Jak chceme pracovat? Tak, aby to dávalo smysl a bavilo nás to!“. Uvedená skutečnost je definovaná i v krátkém interním komiksu pro své zaměstnance, který popisuje základní principy řízení projektů ve společnosti. (viz příloha č. 1):

- zaměstnanci si sami vybírají, na čem chtějí pracovat,
- projekt řeší konkrétní potřebu (víme k čemu je),
- pro projekt se stanovují metriky úspěchu,
- projekt je řízen v iteracích (jedna iterace maximálně do tří týdnů)
- změny jsou očekávány a akceptovány,
- není nutné se bát chyb, protože se z nich učíme,
- každý v týmu ví, kdy jsme s projektem spokojeni,
- pokud tým projekt přijme, musí ho dotáhnout až do konce,
- tým sedí pohromadě, úzká komunikace vítající i ostatní oddělení,
- distribuované týmy se potkají alespoň jednou za 14 dní osobně,
- před realizací projektu je nutné vytvořit prototyp a backlog (oceněný, seřazený podle priorit), nastavit pravidla fungování, definovat použité technologie a termíny milníků,
- aplikace staví v nezávisle provozovatelných datových centrech, která jsou robustní a odolná vůči technickým výpadkům,
- projekty jsou permanentně a průběžně testovány (uživatelské, bezpečnostní a zátěžové testy)
- při feature freeze ladíme službu jako celek (před cílem zmrazíme),
- projekty jsou vydávány postupně po malých částech, prostřednictvím beta testů (interní beta test, veřejný beta test)
- projekty jsou konečné, po ukončení projektu přechází do údržbového systému.

4.2.3. Organizační struktura projektu

Vrcholový management autorizuje jednání týmů, a drží tak zodpovědnost ve svých rukou. Vlastníkem produktu je Product Manager, který má vizi finálního produktu (projektu) a tvoří jej. Procesy řídí Scrum Masteři, kterých je ve společnosti celý tým, ti jsou v roli Project Managerů. Týmy se skládají z kodérů (web masteři a programátoři), grafika a zástupce obchodu. Web masteři tvoří user experience

a grafik se stará o vizuální podobu. Zástupce obchodu do projektu zasahuje nejméně. Týmy vždy sedí a řeší problémy pohromadě, distribuované týmy se setkávají nejméně jednou týdně. Organizační struktura projektu a projektového týmu je vyobrazená na následujícím obrázku. (vlastní zpracování podle [17])



Obrázek 11: Organizační struktura projektu a projektového týmu (vlastní zpracování)

4.3. Životní cyklus projektů ve vybrané společnosti

V této části je na základě řízených rozhovorů vypracován obecný životní cyklus agilně řízeného projektu ve vybrané společnosti. Projekty řízené agilně mají

odlišný životní cyklus než projekty řízené tradičně. Každá iterace v sobě zahrnuje plánování, vývoj, retrospektivu a uzavření, takže každá má svůj životní cyklus, který se může trochu lišit.

Obecný životní cyklus agilně řízeného projektu:

- **Předprojektová fáze**
 - Backlog (User Story)
 - Volba Scrum Mastera
 - Tvorba týmu
- **Plánovací fáze** (mění se)
 - Nastavení spolupráce a pravidel
 - Plán projektu (milníky, předpokládané termíny)
 - Posloupnost User Story
 - Přibližné termíny pro Top Management
- **Realizační fáze** (opakované realizace iterací)
 - Plánování iterace (plánovací fáze iterace)
 - Vývoj
 - Demo
 - Retrospektiva (Lessons Learned)
 - Feedback
 - Beta verze (ideálně výstupem iterace)
- **Ukončovací fáze**
 - Vydání ostré verze
 - Vyhodnocení užítku
 - Feedback
 - Zapracování bugů
 - Retrospektiva (Lessons Learned)
- **Poprojektová fáze**
 - Údržbový systém

4.3.1. Předprojektová fáze

V předprojektové fázi projektu je vytvořen Backlog a jednotlivé User Story, jejichž účelem je definovat co by mělo být výstupem projektu (definování minimálně životaschopného produktu). Backlog je živý dokument, který zahrnuje road mapu, stručný harmonogram (předpokládané termíny realizace milníků) a obecnou identifikaci rizik. Akutní problémy jsou řešeny až během sprintu, k analýze rizik dochází během retrospektiv. Je zvolen Scrum Master, který projekt bude řídit. User Story jsou seřazené podle priorit a celý Backlog je naceněný. Poté je podle priority vybrána User Story a zařazena do Sprint-Backlogu. Náročnost User Story je

naceňována poměrem již ke známým úkolům v minulosti (poměrové vs. absolutní naceňování).

	Činnost	Vlastník	Vstupy	Výstupy
Předprojektová fáze	Tvorba Backlogu a jednotlivých User Story	Product Manager, Scrum Master	Idea/Vize (ověřená - opřená o hrubý výzkum)	Backlog (definování minimálně životaschopného projektu)
	Zvolení Scrum Mastera	Product Manager	Iniciace Product Managera (Backlog), sladění týmu	Scrum Master
	Tvorba Scrum týmu	Product Manager a Scrum Master	Backlog	Definovaný Scrum tým s nastavenými rolemi a zodpovědnostmi
	Prioritizace Backlogu (seřazení User Story podle priorit)	Scrum tým	Backlog	Prioritizovaný Backlog
	Nacenení Backlogu	Vývojový tým	Backlog	Nacenený Backlog

Tabulka 3: Předprojektová fáze (vlastní zpracování)

4.3.2. Plánovací fáze

Plánovací fáze projektu se zabývá tvorbou týmu, plánu projektu (milníky, předpokládané termíny), nastavením spolupráce a pravidel. Dalo by se říci, že týmy se tvoří samy, každý potencionální člen se vyjádří, na jakém projektu by chtěl pracovat a pokud je to možné, je na ten projekt přidělen. Nastavení spolupráce a pravidel zahrnuje definování délky iterací, plánovacích dnů a termínů dema. Krátké iterace dovolují dobře reagovat na drobné změny a napomáhají soustředění týmu. Je naplánován termín, kdy se na projektu začne pracovat a odhadnut termín konce projektu. Milníky plánuje většinou tým sám, někdy do tvorby zasahují i zainteresované strany, převážně se týkají beta testů softwaru (uzavřená beta, otevřená beta) a spuštění ostré verze.

Plánovací fáze	Činnost	Vlastník	Vstupy	Výstupy
	Zařazení User Story do Sprint-Backlogu	Scrum tým	Backlog	Sprint-Backlog
	Nastavení spolupráce a pravidel projektu	Scrum Master	Backlog	Definované délky iterací, termíny plánovacích dní a dema
	Stanovení termínů (začátek a předpokládaný konec projektu)	Scrum Master	Backlog	Předpokládaná doba trvání
	Stanovení milníků projektu	Scrum tým	Backlog	Předpokládané termíny testovacích verzí softwaru (veřejných, neveřejných) a jiných milníků
	Sestavení plánu projektu	Product Manager a Scrum Master	Nastavená spolupráce a pravidla, definované termíny a milníky	Plán projektu (záměr projektu)

Tabulka 4: Plánovací fáze (vlastní zpracování)

4.3.3. Realizační fáze

Realizační fáze projektu se týká realizací jednotlivých iterací. Iterace je plánována na základě dohody mezi týmem a Product Managerem. Product Manager stanovuje, co by chtěl a tým stanovuje, co je schopen v daném termínu udělat. Ke změnám iterací dochází jen ve výjimečných případech. Následuje realizace iterace, kdy dochází k samotnému vývoji, to zahrnuje i vydávání neveřejných a veřejných beta verzí. Na závěr iterace je představeno takzvané demo, zde tým prezentuje Product Ownerovi a zainteresovaným stranám odvedenou práci a rozhoduje se, zda se bude pokračovat, něco se změní, nebo se projekt ukončí. Dále proběhne retrospektiva, kde je zhodnoceno vše, co se povedlo i nepovedlo v iteraci naplnit.

Realizační fáze	Činnost	Vlastník	Vstupy	Výstupy
	Plánování iterace	Scrum Master	Sprint-Backlog	Plán iterace
	Vývoj (realizace iterace)	Vývojový tým	Plán iterace	Funkční část projektu
	Demo projektu	Product Owner (Product Manager)	Funkční část projektu	Schválená/neschválená realizace proběhlé iterace
	Sběr zpětné vazby	Product Owner (Product Manager, nebo uživatelé)	Funkční část projektu	Zpětná vazba
	Retrospekti va iterace	Product Manager, Scrum tým	Zážitků iterace (průběh práce, vnější vlivy a zpětná vazba)	Zhodnocení úspěchu a užitku
Beta verze (testovací verze)	Vývojový tým	Funkční část projektu	Zpětná vazba	

Tabulka 5: Realizační fáze (vlastní zpracování)

4.3.4. Ukončovací fáze

V ukončovací fázi projektu dochází k vydání ostré verze, u které je následně vyhodnocován užitek. Vyhodnocení užitku slouží ke zhodnocení celkového přínosu a užitku projektu. Je sbírána zpětná vazba od Product Managera a zainteresovaných stran, zejména uživatelů, na základě které se tým snaží zapracovat většinu bugů, nalezených v předložené ostré verzi. Výsledkem je čistý kód a funkční produkt připravený k nasazení. Následuje další retrospektiva, týkající se celého projektu, zde je zhodnocen úspěch a slabá místa vývoje projektu.

Ukončovací fáze	Činnost	Vlastník	Vstupy	Výstupy
	Vydání ostré verze	Scrum tým	Finální verze projektu	Dodaná ostrá verze projektu
	Vyhodnocení užítka	Product Manager a Scrum tým	Ostrá verze projektu	Zhodnocen přínos a užitek celého projektu
	Sběr zpětné vazby	Product Owner (Product Manager, nebo uživatelé)	Zpětná vazba uživatelů	Zpětná vazba
	Zpracování bugů	Vývojový tým	Zpětná vazba	Projekt na 99% funkční
	Retrospektiva celého projektu (Lessons Learned)	Product Manager, Scrum tým	Přímá zkušenost vývojovéh o týmu s celým průběhem projektu	Zhodnocení vývoje (úspěch, užítka a slabá místa)

Tabulka 6: Ukončovací fáze (vlastní zpracování)

4.3.5. Poprojektová fáze

Po ukončovací fázi následuje poprojektová fáze projektu, kde je projekt převeden do údržbového systému. Údržbový systém slouží ke sběru reportů, na jejichž základě jsou tvořeny tasky, které jsou pak podle priority postupně řešeny.

Poprojektová fáze	Činnost	Vlastník	Vstupy	Výstupy
	Předání projektu od údržbového systému	Vývojový tým	Zpětná vazba od uživatelů	Reporty o chybách (tasky)
Odbavování reportů	Vývojový tým	Reporty o chybách	Opravené chyby	

Tabulka 7: Poprojektová fáze (vlastní zpracování)

4.4. Případová studie

4.4.1. Vize projektu

Ke vzniku vize došlo na základě již existující webové služby společnosti, která nebyla využívána podle předpokladů, začala ztrácet uživatele, a tudíž došlo k i poklesu výnosů. Díky tomu, že se uvedený trend zrychloval, vedení společnosti společně s Product Managerem rozhodlo, že je nutno tuto službu inovovat. Webová služba je katalogem produktu „x“, základní vize projektu proto zněla jednoduše: „Najdu nejvhodnější produkt „x“ pro mé potřeby, na první pokus.“. Cílem projektu nebyl upgrade existující verze, ale webová služba byla tvořena znovu od základů „na zelené louce“. Hlavním problémem bylo vyvážit technickou, designovou a uživatelskou stranu projektu, což vedlo k potřebě vytvoření prototypu.

4.4.2. Předprojektová fáze

Předprojektová fáze je vždy spojená s vizí, nebo základní ideou projektu, což v tomto případě převážně zahrnovalo definování Backlogu a jednotlivých User Story. V Backlogu je také stručně a obecně provedena analýza rizik. Na základě Backlogu je rozhodováno, jestli projekt bude, nebo nebude realizován. Dále je v této fázi zvolen Scrum Master, který projekt bude řídit. Analýza proveditelnosti nebyla nutná, jedná se o inovaci existujícího projektu. Studie příležitostí, potřeb trhu, očekávání (atraktivnější, kvalitnější) byla převzata z předešlé „verze“ projektu. Finanční analýza nebývá ve společnosti prováděna, šance se dává skoro všem projektům, které mají podle top managementu nějakou naději, finance se neřeší. Pokud je projekt neúspěšný, tedy negeneruje dostatečný zisk, je buď ukončen, nebo jako v tomto případě inovován, o těchto problémech rozhoduje finanční a analytické oddělení.

4.4.3. Plánovací fáze

Nejprve byl sestaven projektový tým, zahrnující jednoho Scrum Mastera, jednoho grafika, pět vývojářů a redakci. Redakce je stakeholderem, který do výsledné služby dodává obsah a poskytuje analýzu potřeb trhu. Většina členů týmu byla na projekt nasazena na plný úvazek.

Nastavení spolupráce a pravidel zahrnuje definování délky sprintů, stanovení plánovacích dnů a termínů dema. Nejprve byly definovány délky sprintů, tým chtěl být flexibilní, soustředěný a rychle reagovat na změny, proto byly zvoleny týdenní sprinty. Plánovací dny byly stanoveny na pondělky a termíny dema na pátky, nebo pondělky dle potřeby. Naceňování zatím neproběhlo, tým ani Product Manager ho nevyžadoval.

Jako začátek projektu bylo stanoveno datum 25. května 2015, se závazkem, že první veřejná testovací verze služby bude spuštěna 31. srpna a ostrý provoz bude zahájen do 30. října téhož roku. Uvedené časové údaje jsou stanovovány pouze orientačně, doba trvání u některých projektů není ve společnosti fixní. Odhad doby trvání se zpřesňuje s rozvojem vývojového týmu, přepočítávání velocity týmu (počet dokončených Story Pointů za iteraci), velocita týmu následně slouží i ke kontrole doby trvání projektu.

Základní přehled projektu			
Délka projektu do ostrého spuštění (měsíce)	Délka projektu do ostrého spuštění (MD _I)	Projektový tým (počet členů)	Průměrná velocity týmu
5	313	7 + redakce	7,5 Story Pointů/týdně

Tabulka 8: Základní přehled projektu (vlastní zpracování)

4.4.4. Realizační fáze

V realizační fázi probíhají týdenní iterace, každá z těchto iterací zahrnuje plánování iterace, její realizaci a následné představení Product Managerovi (Demo). Všechny iterace z důvodu rozsahu této práce, zde nejsou podrobně rozepsány, pozornost je věnována jen důležitým situacím (milníkům), které během projektu nastaly. První problém, který nastal, se týkal redakce. Ta nedodávala dostatek obsahu podle představ Product Managera, který rozhodl, že v projektu bude dána přednost kvalitě. Redakce tak dostala více času, konkrétně jeden měsíc, aby mohla doladit technickou a obsahovou stránku. Došlo tak ke zpřesnění odhadu, ne odložení termínu. Dalším problémem v projektu byla nedostatečně těsná spolupráce. Komunikace byla slabá, nedocházelo k technickým konzultacím a výsledná práce nebyla dostatečně předváděna vedení. K tomu docházelo zejména mezi grafiky a Scrum týmem,

následně mezi Scrum týmem a Top Managementem. Problém vedl ke zdržování práce a byl částečně vyřešen tím, že na retrospektivu (demo), byly přizvány i zainteresované strany, konkrétně grafici a Top Management.

První veřejná beta tak byla spuštěna až 30. září 2015 s rezervovanou dobou trvání tří měsíců. Týdenní iterace byly zachovány, nyní však do projektu svou zpětnou vazbou nezasahoval jen Product Manager, ale také uživatelé, kteří se zúčastnili veřejné bety. Tato zpětná vazba je nejdůležitější a vede k finalizaci projektu.

Poté co bylo vyřešeno dostatečné množství nalezených chyb a kód byl z 99% čistý, Product Manager definoval, že je spokojený a lze spustit ostrou verzi služby. Před spuštěním ostré verze však musela být vyvíjená webová služba napojena na další již existující systémy. Jednalo se jak o vnitropodnikové systémy, tak o jeden externí, konkrétně iOS Affiliate Program. Napojení se týkalo převážně monetizace a podpory ostatních služeb společnosti. Vnitropodnikový systém se z technických důvodů zdržel a tak došlo k další menší úpravě doby trvání projektu. Ke spuštění ostré verze tak došlo až 25. listopadu 2015, přibližně 25 dní po původně předpokládaném termínu. Dne 14. prosince 2015 proběhla jedna z posledních retrospektiv (Lessons Learned), kde byla zhodnocena Feedback Collection, komunikace a celkový průběh vývoje projektu. Je nutné podotknout, že Lessons Learned nejsou ve společnosti nijak standardizovány, získané znalosti se sice vytěžují, ale nedochází k jejich uchování, předávání ani rozvíjení. Existují však Best Practices a interní komiks, které se snaží nastínit problematiku agilního řízení a vývoje ve společnosti.

Během realizace projektu zpracoval Scrum Master pro představu a motivaci všech zainteresovaných stran finanční analýzu, kde jsou stručně naznačeny náklady, zároveň zde jsou uvedeny zisky a počet uživatelů před přepisem projektu (výsledky původní verze projektu).

Náklady (Kč)	
Mzdy	1 025 000
Obsah	600 000
Hardware	500 000
Celkem	2 125 000
Zisk před přepisem (Kč/měsíc)	
	500 000
Uživatelé před přepisem (měsíční)	
	65 000

Tabulka 9: Stručná finanční analýza (vlastní zpracování)

4.4.5. Ukončovací fáze

31. prosince 2015 byla práce na projektu úspěšně ukončena, řízení i průběh projektu byl ze strany vedení pochválen za rychlost a informovanost. Během projektu nedošlo k žádným velkým překvapením a jeho cíl byl naplněn. Webová služba byla dále předána do údržbového systému, tedy poprojektové fáze životního cyklu projektu.

4.4.6. Poprojektová fáze

Nakonec byl projekt předán do údržbového systému, kde jsou průběžně doladovány chyby a nedostatky v projektu, na základě zpětné vazby běžných uživatelů. Během března 2016 dojde k vyhodnocení užítku, které zhodnotí, zdali projekt generuje zisk a neztrácí uživatele.

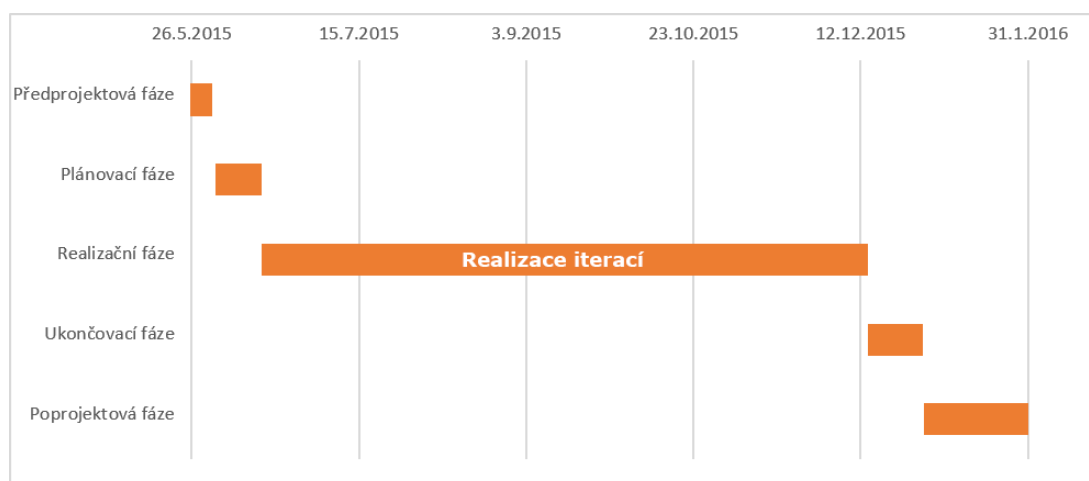
4.4.7. Shrnutí projektu

V této části je stručně shrnut projekt, jak dlouho jeho realizace trvala a zda naplnil stanovené cíle.

Projekt byl realizován úspěšně a trval přibližně o měsíc déle, než se na začátku předpokládalo (159 pracovních dní). Top Management, Product Owner i zbytek Scrum týmu byli s výsledkem projektu nadmíru spokojeni. Projekt byl realizován bez větších problémů a zdržení. I přes to, že se Top Management cítil být informovaný, došlo ke komunikačním problémům, zejména mezi grafiky a Scrum týmem, následně

mezi Scrum týmem a Top Managementem. K tomu došlo zejména z toho důvodu, že komunikace není ve společnosti vždy úplně přímá a jednotná.

Časový průběh popisovaného projektu v případové studii je přibližně zachycen v následujícím diagramu:



Obrázek 12: Přibližný harmonogram fází popisovaného projektu (vlastní zpracování)

Porovnání odhadovaných termínů se skutečnými je znázorněno v následující tabulce:

Důležité součásti projektu	Odhadovaný termín	Skutečný termín	Rozdíl
Začátek projektu	25.5.2015	-	-
Veřejná testovací verze	31.8.2015	30.9.2015	zpoždění 30 dní
Ostrý provoz	30.10.2015	25.11.2015	zpoždění 26 dní
Retrospektiva projektu	-	14.12.2015	-
Konec projektu	-	31.12.2015	-
Celková délka projektu	159 pracovních dní		

Tabulka 10: Srovnání předpokládaných a skutečných termínů projektu (vlastní zpracování)

4.4.8. Srovnání praxe společnosti s agilním manifestem

V této části je srovnána praxe společnosti s pilíři agilního manifestu, které poskytují základ agilnímu řízení projektů.

Společnost úplně splňuje sedm pilířů z dvanácti, čtyři pilíře splňuje pouze částečně a jeden vůbec. Částečně splňuje včasnou a průběžnou dodávku software, ne vždy však tým na konci sprintu dodá funkční část projektu. Dále ve společnosti částečně nefunguje přímá komunikace. Tým sice navzájem komunikuje přímo, ale komunikace se zainteresovanými stranami není vždy ideální a jednotná. Stálá pozornost návrhu a technickému řešení také není vždy věnována, ne u všech projektů dochází k prototypování. Týmy bohužel nemají Top Managementem poskytnutou plnou svobodu, nelze je tak proto hodnotit, jako zcela samo-organizující. Ne všechny týmy by měly převzít plnou odpovědnost za projekt, ale ty týmy, které už jsou dostatečně rozvinuté a zkušené, si odpovědnost a svobodu rozhodně zaslouží.

V praxi společnost úplně nesplňuje pouze jeden princip agilního manifestu, konkrétně ten, že vývojáři nespolupracují se zákazníky (Product Owner a zainteresované strany) na denní bázi. Spolupráce se zákazníky závisí na předem určené délce sprintu, většinou však ke vzájemné spolupráci dochází na týdenní bázi.

Srovnání aktuální praxe společnosti s pilíři agilního manifestu je znázorněno na následující tabulce:

Srovnání aktuální praxe společnosti s pilíři agilního manifestu		
Pilíř	Stručný popis	Ve vybrané společnosti
1	Včasná a průběžná dodávka SW	částečně
2	Změny v požadavcích na výstup	ano
3	Iterativní vývoj	ano
4	Denní spolupráce zákazníků a vývojářů na projektu	ne
5	Dostatečná motivace řídicích pracovníků	ano
6	Přímá komunikace	částečně
7	Naplnění specifikace	ano
8	Podpora udržitelného rozvoje	ano
9	Stálá pozornost návrhu a technickému řešení	částečně
10	Důraz na jednoduchost	ano
11	Samo-organizující se týmy	částečně
12	Postupný rozvoj týmu	ano

Tabulka 11: Srovnání praxe společnosti s agilním manifestem (vlastní zpracování)

4.4.9. Srovnání praxe společnosti s principy metody Scrum

Stejně jako v předešlé části i v této dochází ke srovnání praxe řízení projektů společnosti, nyní však přímo s principy metody Scrum.

Na základě případové studie a řízených rozhovorů se dá říci, že se společnost plně řídí principy metody Scrum. Výčet těchto principů je znázorněn na následující tabulce:

Principy metody Scrum ve společnosti		
Pilíře	Transparentnost	ano
	Kontrola	ano
	Adaptace	ano
Aktivity	Plánování sprintu	ano
	Denní standupy	ano
	Vyhodnocení sprintu (Demo)	ano
	Retrospektiva sprintu	ano
Artefakty	Product Backlog	ano
	Sprint Backlog	ano
Scrum tým	Product Owner	ano
	Vývojový tým	ano
	Scrum Master	ano

Tabulka 12: Srovnání praxe společnosti s principy metody Scrum (vlastní zpracování)

4.5. Vlastní návrhy a doporučení

Na závěr předcházející kapitoly bylo provedeno srovnání praxe řízení projektů společnosti s pilíři agilního manifestu a principy metody Scrum, kde byly identifikovány hlavní nedostatky projektového řízení ve společnosti. Bylo konstatováno, že projektové řízení společnosti úplně nenaplnuje pět pilířů agilního manifestu, ale naopak splňuje všechny principy metody Scrum.

4.5.1. Včasná a průběžná dodávka software

V agilním manifestu je definováno, že prioritou je včasná a průběžná dodávka funkčního software, který přinese zákazníkovi hodnotu.

Projektové týmy však ne vždy tento pilíř splní. Problém se netýká až tak včasnosti, ale průběžného dodávání funkčního software. Včasnost dodávání funkčního software je ve společnosti již vyřešena předem definovanými dny, kdy dochází k plánování a prezentování dema projektu. K tomu dochází během předprojektové fáze projektu, konkrétně během nastavování spolupráce a pravidel.

S dodáváním funkční části software by mohlo pomoci řízení rozsahu projektu, konkrétně tvorba jednoduché WBS provázané s Product Backlogem projektu. Důležité je jasně definování User Story a jednotlivých Story Pointů tak, aby se vždy týkaly funkčních částí a součástí. Tvorba podrobné WBS ani logického rámce při řízení rozsahu a integrace projektu v agilních přístupech není doporučováno. Tyto postupy odporují agilním přístupům a snižují jeho flexibilitu a efektivnost.

Řízení rozsahu

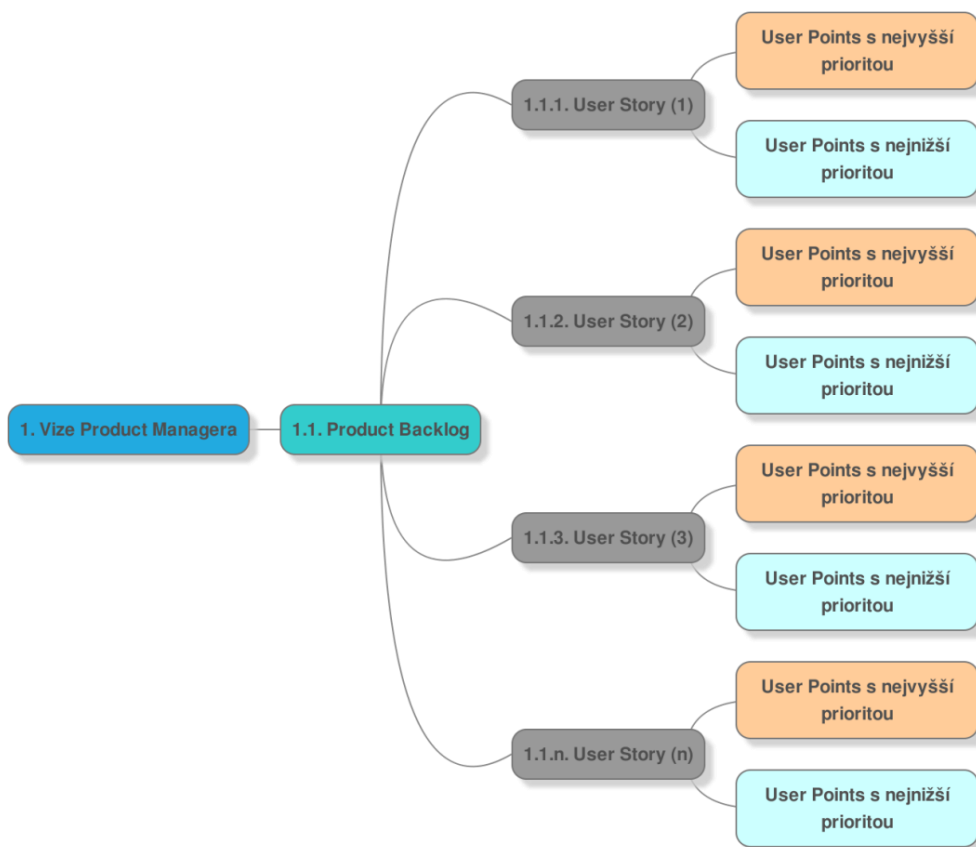
Na počátku řízení rozsahu projektu by měl Product Backlog obsahovat všechny základní informace o projektu. Product Backlog by měl být snadno dostupný celému Scrum týmu. Jelikož se během agilního řízení projektu nejen očekávají, ale také vítají změny, měl by být Product Backlog také snadno editovatelný.

Vhodným nástrojem, který by projektové týmy mohly využívat, je online služba jménem **JIRA Software** vyvíjená společností Atlassian. Služba poskytuje jednoduché a praktické nástroje speciálně pro agilní řízení projektů. Zahrnuje nástroje pro plánování (tvorba Product Backlogu), sledování (virtuální Kanban, zaznamenávání změn), dodávání a podávání zpráv (velocity, burndown chart atp.). Tvorba, organizace a prioritizace Product Backlogu je jednoduchá, intuitivní a přehledná. Službu lze upravit a přizpůsobit projektovému týmu pomocí dostupných rozšíření. Využitím této služby by byly všechny informace a data týkající se projektu dostupná online, nejen projektovému týmu, ale i zainteresovaným stranám.

Týmy ve společnosti nyní něco podobného dělají tabulovou formou s lístečky, podobnou metodě Kanban. Na tabuli je vyznačená časová osa, která definuje milníky a předpokládané termíny jejich realizace. Tabule tak vizuálně definuje základní informace o projektu, dokumentuje jeho průběh a stručně zahrnuje identifikaci rizik.

Product Backlog by dále mohl být podložen a provázán se základním dokumentem **WBS (Work Breakdown Structure)**, který vymezuje veškeré činnosti v projektu a jejich posloupnost. WBS by tak definovala jednotlivé User Story (sprinty), způsob jejich dosažení a rozlišení úkolů ve sprintu podle priority (Story Pointy), to by mohlo posloužit ke snazšímu řízení prací na projektu a celkově přispět k vyšší přehlednosti. User Story by měla být definována tak, aby její realizací vznikla funkční součást finální verze software. To znamená, že i Story Pointy by měly být definovány jako malé funkční součásti User Story. Tak by bylo zajištěno, že tým na konci iterace vždy dodá funkční část projektu. WBS by bylo možné tvořit buď na tabuli, nebo v cloudové formě, např. jednoduchou mindmapou. Tabule by mohla sloužit k brainstormingu Scrum týmu a následně být stručně digitalizována do formy mindmapy. Vhodným nástrojem je například cloudová verze mindmap, která je jednoduchá, přívětivá a snadno editovatelná, nebo již zmíněná služba JIRA Software.

Na následující mindmapě je stručně načrtnuta struktura jednoduché WBS, která jasně vymezuje, jak z vize vzniká Backlog, který je následně rozčleněn na User Story a dále na prioritizované úkoly (jednotlivé Story Pointy). Úkoly mohou být řešeny různě a demo jednotlivých sprintů se může lišit. Dále se může mindmapa členit na rizika, příležitosti, předpoklady, náročnost a jiné faktory, které Scrum tým uzná za vhodné. Výslednou mindmapu je také možné zachytit tabulkovou formou. Tvorbou takového dokumentu by došlo k dokumentaci změn, ke kterým během realizace projektu dochází a tvorbě Lessons Learned, které v jakékoliv formálnější podobě ve společnosti chybí. Dále lze takový výstup, díky jeho jednoduchosti, prezentovat během dema a retrospektiv zainteresovaným stranám.



Obrázek 13: Návrh jednoduché WBS (vlastní zpracování)

Jednoduchost je největší prioritou, agilní řízení z jednoduchosti těží. Pokud by tvorba takových dokumentů byla moc složitá, nebo zdlouhavá, odporovalo by takové řízení základním principům agilního řízení i metody Scrum konkrétně. Důležité je, aby veškeré informace byly snadno dostupné celému Scrum týmu a snadno změnitelné v průběhu realizace projektu. Proto doporučuji využít službu JIRA Software, která je intuitivní, praktická a poskytuje všechny důležité nástroje pro agilní řízení projektů.

Lessons Learned

Lessons Learned v praxi společnosti v jakékoliv formální formě chybí. Existují manažerské Best Practices, ale k žádnému vytěžování znalostí, jejich rozvíjení a předávání nedochází. Vytváření jednoduchého a stručného Lessons Learned, které by zahrnovalo důležité situace a problémy, ke kterým u jiných projektů došlo a dochází, by mohlo napomoci nejen projektovému týmu, ale projektovému řízení ve společnosti obecně.

Stručné a jednoduché Lessons Learned lze vytvářet například pomocí tabulky nebo formuláře. Mělo by zahrnovat alespoň název problému, jeho popis a jak byl problém vyřešen. V následující tabulce je navrženo, jak by k vytěžování znalostí ve společnosti a projektech mohlo docházet. Hodnoty v tabulce jsou pouze ilustrační, ale snaží se vycházet z popisovaného projektu v případové studii.

ID	Problém/Situace	Popis problému	Vlastník	Doporučená změna	Předáno (komu)
1	Nový vývojář v projektovém týmu	Během realizace projektu došlo k úpravě projektového týmu, přidáním klíčového vývojáře.	Scrum Master	Povinná dokumentace kódu.	Projektový tým
2	Technické problémy zainteresované strany	Kvůli technickému problému, který nastal u jiné služby, došlo k posunutí předpokládaného termínu ukončení projektu.	Scrum Master	Užší komunikace se zainteresovanými stranami.	Product Manager
3	iOS verze služby nebyla dokončena	Vývojář iOS dlouhodobě onemocněl a nebyl nadále schopen pracovat na verzi služby pro iOS.	Scrum Master	Delegování práce na iOS verzi Android vývojáři.	Projektový tým
n	-	-	-	-	-

Tabulka 13: Šablona Lessons Learned projektu (vlastní zpracování)

Řízení rizik

Na základě analýzy rizik vytvořené v Product Backlogu (pomocí brainstormingu v mindmapě) by měl být vytvořen registr rizik. Řízení rizik v praxi společnosti probíhá pouze v předprojektové fázi a není dále rozvíjeno. Tvorba takového podpůrného nástroje by mohla řízení projektů prospět převážně v předvídání problémových situacích a jejich řešení. V následujících tabulkách je prezentován ilustrační příklad, jak by takový registr mohl vypadat.

V první tabulce jsou stanoveny hodnotící škály pro výskyt a dopad rizika (v našem případě 1-5) a jejich význam.

Výskyt	Význam	Dopad	Význam
1	Nepravděpodobné	1	Velmi malý
2	Méně pravděpodobné	2	Malý
3	Středně pravděpodobné	3	Střední
4	Spíše pravděpodobné	4	Vysoký
5	Velmi pravděpodobné	5	Extrémní

Tabulka 14: Hodnotící škály pro výskyt a dopad rizika (vlastní zpracování)

Registr rizik je z důvodu potřeb agilního řízení stručný a jednoduchý, jsou zaznamenávány jen důležité informace. Obsahuje název riziko, jeho stručný popis, výskyt, dopad a vypočtenou očekávanou hodnotu rizika (výskyt x dopad). Na základě OHR a matice rizik je vybrána obranná strategie (přijetí, omezení, přenesení) a preventivní opatření. V registru rizik je nejen zaznamenáno, kdo riziko identifikoval, ale také kdo je jeho řešitelem.

ID	Riziko	Scénář	Výskyt	Dopad	OHR	Obranná strategie	Preventivní opatření	Vlastník	Řešitel
1	Ztráta vývojáře	Klíčový vývojář dlouhodobě onemocněl.	2	4	8	Omezení	Dokumentace kódu.	Projektový tým	Scrum Master
2	Ztráta práce na projektu	Díky technickému selhání došlo ke ztrátě kódu.	1	5	5	Omezení	Pravidelná záloha kódu a práce.	Projektový tým	Vývojář
3	Iterace zahozena	Výstup iterace neodpovídal požadavkům.	1	4	4	Omezení	Podrobnější specifikace cíle v Product Backlogu.	Scrum Master	Product Manager
n	-	-	-	-	-	-	-	-	-

Tabulka 15: Šablona registru rizik (vlastní zpracování)

Následuje tabulka reprezentující matici rizik zobrazující čtyři rizikové skupiny, na základě kterých je vybírána obranná strategie. Pro červené oblasti připravujeme krizový scénář.

Dopad	5b	5				
	4b	4	8			
	3b					
	2b					
	1b					
Neobáváme se		1b	2b	3b	4b	5b
Obáváme se minimálně		Výskyt				
Obáváme se						
Nejsme schopni podstoupit riziko						

Tabulka 16: Matice rizik (vlastní zpracování)

4.5.2. Denní spolupráce zákazníků a vývojářů na projektu

Dalším nenaplněným pilířem v praxi společnosti je denní spolupráce zákazníků a vývojářů na projektu. Podle pilířů agilního manifestu by měly být na začátku spolupráce definovány pouze hrubé požadavky, které se upřeshňují během vývoje podle potřeb zákazníka, s kterým je jednáno na denní bázi.

Projektové týmy ve společnosti se zákazníkem, konkrétně Product Managerem, spolupracují a podřizují se jeho požadavkům. Spolupráce však neprobíhá denně, ale odvisí na délce iterací (týdenní až třítýdenní). Chování týmů je v souladu s metodou Scrum, kdy jsou iterace plánovány předem a jejich výstupy jsou prezentovány Product Ownerovi na jejich konci.

Denní spolupráce by bylo možné docílit přizváním Product Managera a zainteresovaných stran na denní standupy (alespoň na některé), které v praxi společnosti probíhají. Product Manager i zainteresované strany by tak mohli poskytovat zpětnou vazbu už v průběhu iterace a tento pilíř by tak byl naplněn.

4.5.3. Řízení komunikace

Dalším pilířem agilního manifestu, který společnost úplně nesplňuje je přímá komunikace. V projektu komunikace probíhá podle preferencí členů projektového

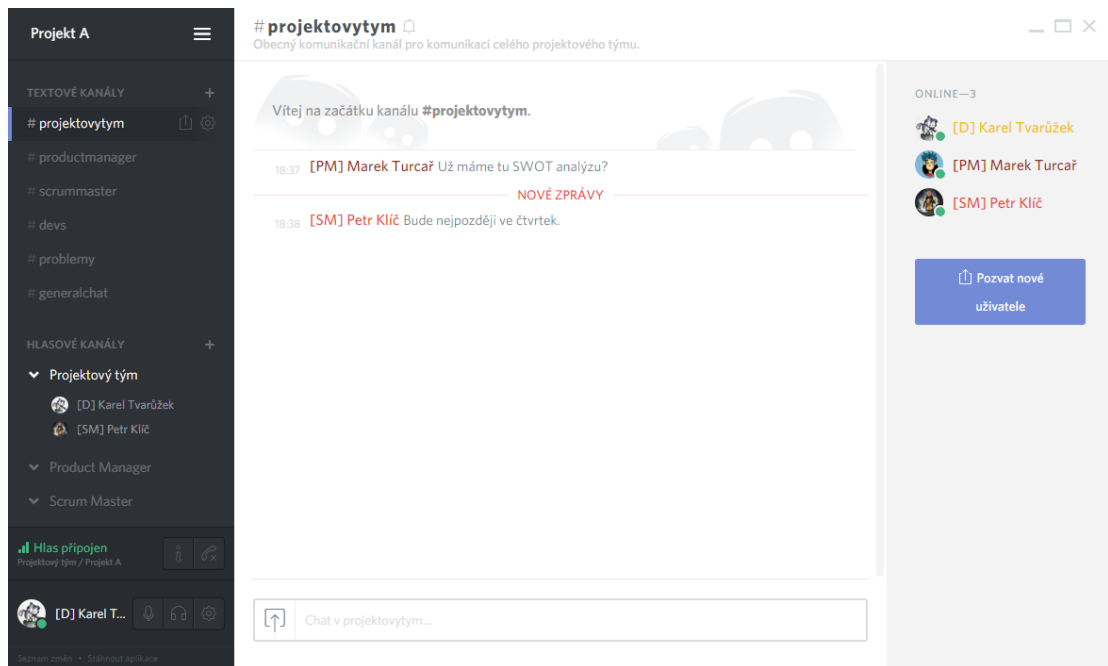
týmu. Probíhají sice denní standupy, kde spolu členové týmu komunikují přímo, ale mimo ně žádný jednotný nástroj pro celou společnost není.

Ke zlepšení komunikace, jak v týmu, tak v celé společnosti, navrhuji implementovat jednotný online nástroj „Discord“. Nástroj je zcela zdarma a nezahrnuje jen textový, ale také hlasový komunikátor. Discord funguje na všech operačních systémech a není potřeba žádná instalace, lze jej spustit přímo v prohlížeči (www.discordapp.com). Je velmi nenáročný na hardware i internetové připojení, také zahrnuje IP a DDoS zabezpečení.

Discord poskytuje také mobilní aplikaci na operační systémy Android a iOS. Použití je díky přívětivému uživatelskému prostředí jednoduché a služba disponuje mnoha možnostmi, jak spravovat více komunikačních kanálů pro různé úrovně řízení.

Komunikátor byl vybrán na základě jeho jednoduchosti a uživatelské přívětivosti, která je pro agilní řízení velmi důležitá. Zavedení jednotného komunikátoru alespoň v rámci projektového řízení ve společnosti, by mohlo vést k optimalizaci komunikace.

Na následujícím obrázku je simulovaný kanál pro „Projekt A“, každý projekt může mít svůj kanál a uživatelé mohou být připojeni ke všem zároveň. Také lze vytvořit kanály pro jednotlivé stupně řízení a oddělení ve společnosti. Jednotlivé role v projektovém týmu, nebo společnosti, mohou mít nastaveny různé pravomoci, barvu i ikonku. Role uživatelů mohou být definovány i jednoduchou zkratkou v prefixu u jejich jména např. [SM], jako Scrum Master, [D], jako vývojář (developer) atd.



Obrázek 14: Simulace užití nástroje Discord ve společnosti (vlastní zpracování)

4.5.4. Stálá pozornost návrhu a technickému řešení

Tento pilíř agilního manifestu, který společnost ve své praxi úplně nesplňuje, souvisí převážně s prototypováním projektů. Je nutné nejprve testovat koncept a poté použitelnost. K prototypování IT projektů je převážně používána tzv. „Deníková studie“, která spočívá v tom, že uživatelům (testerům), předáme prototyp a necháme je po dobu testu zapisovat do deníku, jak prototyp používají, co prožívají a jak vše funguje. Nevýhodou takové studie je, že je zdlouhavá a nemusí tak být pro agilní řízení projektů vhodná.

S řešením přišla BBC Research and Development, která rozvíjí svou metodu „Lean Experience Mapping“. Metoda využívá hned několika nástrojů, prvním je již zmíněná deníková studie (neveřejný test v uzavřené skupině), která však probíhá online (v případě BBC prostřednictvím služby Google +). Je to rychlejší způsob, než klasická deníková studie, interaktivní a intuitivní, jak pro uživatele, tak pro celý projektový tým. Dále využívá metody „Wizard of Oz“, její princip je založen na situaci, kdy uživatelé nevědí, že automatické funkce služby nejsou automatické, ale jsou prováděny manuálně „za oponou“ (dochází k simulaci automatických funkcí). Následuje zpracování nasbíraných dat, do kterého je zapojen celý tým. Data jsou zpracována do dvou map, kontextové mapy a experience mapy (zkušenostní mapa).

Mapa kontextu zahrnuje sloupce, které popisují, kdy je služba využívána a řádky, které jsou inspirovány empatickým mapováním, tedy co uživatelé při užívání prototypu dělají, co si myslí a co cítí. Projektový tým tak ví, kdy se co děje, dochází k popisu uživatelské aktivity v kontextu.

Experience mapa mapuje cestu uživatele skrze testovaný prototyp. Je složená z uzlů, které se týkají jednotlivých činností a úrovní. Úrovně se týkají user experience od nejlepší, po nejhorší. Výstupem tak je hodnocení user experience prototypu a může tak následně dojít k jeho úpravám.

Na základě těchto výstupů projektový tým vidí kontext a user experience prototypu, hlavně jeho pozitiva a negativa. Může tak následně upravovat a prioritizovat činnosti v iterativním vývoji.

4.5.5. Organizace týmů a definování rolí

Posledním z pilířů, které řízení projektů ve společnosti nenaplnuje, se týká samo-organizujících se týmů, které v praxi dosahují nejlepších výsledků. Podpora kreativity a důvěra přináší výjimečné návrhy a výsledky práce. Projektové týmy ve vybrané společnosti rozhodně mají nadstandartní svobodu, tvoří se samy a využívají nástrojů, které jim vyhovují. Top Management však ještě plně nevěří agilním přístupům, autorizuje jednání týmů, a drží tak odpovědnost ve svých rukou. Týmy bohužel nemají Top Managementem poskytnutou plnou svobodu, nelze je tak proto hodnotit, jako zcela samo-organizující. Ne všechny týmy by měly převzít plnou odpovědnost za projekt, ale ty týmy, které už jsou dostatečně rozvinuté a zkušené, si odpovědnost a svobodu rozhodně zaslouží. Odpovědnost za projekt a jeho dokončení tak nese většinou Product Manager a ne tým, který ho realizuje.

S tím souvisí i to, že někdy není úplně jasné, kdo je za co v projektu odpovědný, tedy, jaká je jeho role. Role jsou navrhnuté a rozvrženy v následující RACI matici:

- R – odpovědný za vykonání úkolu,
- A – odpovědný za celý úkol,
- C – s kým je úkol konzultován,
- I – kdo má být o průběhu úkolu informován.

RACI Matice (pro celý projekt)	Product Manager	Scrum Master	Vývojový tým	Zainterесované strany
Product Backlog	A	R	R	R
Denní Standup	I	R, C	A	I
Plán sprintu	C, I	A, R	R, A	R, I
Demo	R	C	A	R
Retrospektiva	R	R	A	I

Tabulka 17: Obecná RACI matice pro celý projekt (vlastní zpracování)

RACI matici lze vytvořit i pro jednotlivé User Story (iterace) a přesně tak definovat role jednotlivých členů týmu v iteraci. RACI matice pro jednotlivé User Story i Story Pointy je ilustračně zobrazena v následující tabulce:

RACI Matice (pro jednotlivé User Story)		User Story (1..n)			
Projektový tým	Člen	Story Point (1)	Story Point (2)	Story Point (3)	Story Point (n)
	Jan Novák	I	I	I	I
	Josef Starý	R	C	A	C
	Karel Jarůšek	C	R	C	R
	n	A	A	R	A

Tabulka 18: Šablona RACI matice pro jednotlivé User Story (vlastní zpracování)

V projektu není úplně zřetelné, kdo zodpovídá za kompletní dokončení činnosti, proto jsem jako nástroj zvolil RACI matice, kde je přehledně znázorněno, kdo za činnost zodpovídá, kdo ji kontroluje, s kým je konzultována a kdo je o jejím průběhu informován.

4.5.6. Rozšíření dokumentace

Při řízení projektů je tvořen Product Backlog projektu (idea, vize), Demo (představení výsledků iterace), mindmapy (milníky a vstupy třetích stran), někdy i dokumentace kódu a průběhu práce na projektu. Což je v souladu s agilním manifestem i metodou Scrum.

Doporučuji však, aby dokumentace kódu, alespoň ve stručné formě, byla povinná. Pokud v projektu dojde ke vstupu nových vývojářů, je pro ně velmi těžké se

zorientovat a navázat na práci v projektu. Dochází tak k větší prodlevě, než jsou vývojáři schopni na projektu pracovat. Pokud by dokumentace kódu byla povinná, noví vývojáři by měli být schopni se rychleji zorientovat. Docházelo by tak i k pravidelné dokumentaci průběhu vývoje projektu.

Dokumentaci kódu mohou projektové týmy provádět způsobem, který jim nejvíce vyhovuje a odpovídá programovacímu jazyku, kterého se týká. Proto také není uvedena žádná šablona dokumentace.

Pro zachování agility v řízení projektů není další rozšíření dokumentace navrhováno.

5. Výsledky a diskuze

Diplomová práce se věnovala popisu agilního řízení projektů ve vybrané společnosti, která řídí převážně IT/IS projekty. Agilní přístupy v projektovém řízení vybrané společnosti jsou využívány přibližně tři roky. Konkrétně se nejvíce soustředí na agilní metodu Scrum.

5.1. Popis agilního řízení ve společnosti

Vybraná společnost byla v úvodu praktické části popsána obecně, tedy její charakteristika, organizační struktura a průběh komunikace. Postupy agilního řízení projektů ve vybrané společnosti, byly nejprve popsány prostřednictvím pěti fází životního cyklu projektu dle mezinárodních standardů:

- předprojektová fáze,
- plánovací fáze,
- realizační fáze,
- ukončovací fáze,
- poprojektová fáze.

Každá fáze obsahuje tabulku zobrazující všechny činnosti, které v ní probíhají, jejich vlastníka, vstupy a výstupy.

Poté byl vybrán a popsán jeden konkrétní IT projekt, který byl prostřednictvím těchto fází analyzován. Na závěr analýzy případové studie, byly postupy společnosti porovnány s pilíři agilního manifestu a principy metody Scrum. Z tohoto porovnání lze usoudit, že vybraná společnost splňuje všechny principy metody Scrum, naopak plně nesplňuje pět pilířů agilního manifestu, konkrétně níže uvedené:

- včasná a průběžná dodávka software,
- denní spolupráce zákazníků a vývojářů na projektu,
- přímá komunikace,
- stálá pozornost návrhu a technickému řešení,
- samo-organizující se týmy.

I přesto projektové řízení ve společnosti velmi dobře funguje, projekty jsou úspěšně realizovány a agilní přístup zaměstnancům vyhovuje.

5.2. Vlastní návrhy a doporučení

5.2.1. Včasná a průběžná dodávka software

Pro docílení včasné a průběžné dodávky software, byla navržena implementace vhodného nástroje pro řízení integrace a rozsahu projektu. Jedním z takových nástrojů, který by projektové týmy mohly využívat, je online služba jménem JIRA Software. Služba poskytuje jednoduché a praktické nástroje speciálně pro agilní řízení projektů.

Dle zpětné vazby společnosti o implementaci tohoto nástroje dříve uvažovala. Na základě nepřiměřené ceny nástroje vůči přidané hodnotě však bylo rozhodnuto, že implementován nebude. Společnost preferuje využití Trello / CML Trac systému a GitLab jako jeho náhrady.

Dále byla navržena tvorba jednoduché WBS provázané s Product Backlogem projektu. Důležité je zřetelné definování User Story a jednotlivých Story Pointů tak, aby se vždy týkaly funkčních částí a součástí.

Uvedený návrh byl společností částečně přijat. Dle Scrum Mastera společnosti není WBS a vymezení posloupnosti v konkrétních návaznostech pro agilní přístup řízení projektů vhodné. Na druhou stranu by bylo možné tyto nástroje využít při startu projektu jako „plán“, který se bude zpřesňovat v návaznosti na zpětnou vazbu klienta (Product Manager). Prioritizace úkolů je nedílná a nejdůležitější práce s Product Backlogem. Bez ní je to jen seznam co má být uděláno, ale WBS může sloužit jako jednoduchý pomocný ukazatel pro jeho správu.

V návaznosti na WBS byla dále navržena tvorba Lessons Learned tabulkovou formou. Ve společnosti nyní fungují na úrovni týmů retrospektivy a existují manažerské Best Practices, které jsou jednou ročně doplňovány.

Tabulkový systém Lessons Learned by dle zpětné vazby společnosti nemusel být dlouhodobě udržitelný.

Jako poslední nástroj, který by napomáhal včasné a průběžné dodávce software, byla navržena tvorba registru rizik.

Návrh byl společností přijat, s tím, že by mohlo jít o další způsob jak ve společnosti nahlížet na rizika a prioritizovat je.

5.2.2. Denní spolupráce zákazníků a vývojářů na projektu

Pro naplnění pilíře agilního manifestu, týkajícího se denní spolupráce zákazníků a vývojářů na projektu, bylo navrženo, aby Product Manager (zákazník) i zainteresované strany byly přizvány na denní standupy, které v praxi společnosti probíhají. Product Manager i zainteresované strany by tak mohly poskytovat zpětnou vazbu už v průběhu iterace.

Dle zpětné vazby Scrum Mastera je předložený návrh u agilního přístupu řízení projektů náročný v případech, kdy je software vyvíjen pro druhou stranu, tedy pro externího zákazníka. Ten se v praxi účastní jen na demech až v závěru iterace. Jelikož však vybraná společnost vyvíjí převážně projekty pro vlastní potřebu, jsou zainteresované strany i Product Manageři pravidelně zváni k účasti na denních standupech.

5.2.3. Řízení komunikace

Nyní ve společnosti funguje komunikační nástroj „Jabber“, který podle zkušeností zaměstnanců není vždy postačující. Dále je používáno mnoho dalších nástrojů, dle individuálních potřeb týmů i jednotlivců.

Ke zlepšení komunikace, jak v týmu, tak v celé společnosti, bylo navrženo implementovat jednotný online nástroj „Discord“. Nástroj je zcela zdarma a nezahrnuje jen textový, ale také hlasový komunikátor. Discord funguje na všech operačních systémech a není potřeba žádná instalace. Je velmi nenáročný na hardware i internetové připojení a zahrnuje rovněž IP a DDoS zabezpečení.

Navržená implementace nového komunikačního nástroje byla společností pozitivně přijata. Má mnoho funkcí a zároveň je jednoduchý a přívětivý, což by mohlo vést k podpoře a rozvoji komunikace uvnitř společnosti.

5.2.4. Stálá pozornost návrhu a technickému řešení

Pro tento pilíř agilního manifestu, který souvisí převážně s prototypováním projektů, bylo navrženo implementovat metodu „Lean Experience Mapping“. Metoda je vyvíjena tak, aby vyhovovala agilnímu přístupu řízení projektů. Zdůrazňuje, že je nutné nejprve testovat koncept a teprve poté použitelnost.

Metoda byla společností pozitivně přijata, je vhodná pro zkoumání toho, jak s produktem společnosti uživatelé zachází a v jakém kontextu jej používají.

5.2.5. Organizace týmů a definování rolí

Ve vybrané společnosti není úplně jasné, kdo je za co v projektu odpovědný, tedy, jaká je jeho role. K rozdělení základních odpovědností byla doporučena tvorba RACI matic pro řízení projektu (iterací) i jednotlivých User Story.

Navržené doporučení bylo částečně přijato. Rozdělení základních odpovědností chybí, agilní přístupy inklinují k týmové práci a odpovědnosti. Dle zpětné vazby Scrum Mastera, je přechod pro vybranou společnost poměrně komplikovaný. Proto by bylo vhodné tvořit alespoň RACI matici pro řízení projektu a tak zřetelně definovat základní odpovědnosti. RACI matice pro řízení jednotlivých User Story je podle zpětné vazby nadbytečná a neefektivní.

5.2.6. Rozšíření dokumentace

K rozšíření dokumentace při řízení projektu bylo doporučeno, aby tvorba dokumentace kódu, alespoň ve stručné formě, byla povinná. Pokud by byla povinná, noví vývojáři by při vstupu do běžícího projektu měli být schopni rychleji se zorientovat a začít pracovat. Také by tak docházelo k pravidelnému dokumentování průběhu vývoje projektu.

Dle zpětné vazby společnosti je dokumentace v rukou vývojářů. Úkolem Scrum Mastera je naučit tým této odpovědnosti za kvalitu, udržitelnost a schopnost převzít práci jiným týmem. Plošné nařízení dokumentace je v rozporu s agilním přístupem, který říká „dělej kde je potřeba“. To je nutno sledovat u každého projektu zvlášť. Ale určitě je nutné, aby se na dokumentaci nezapomínalo a to ani na rizika, analýzy atd. Jen striktně není určeno v jaké fázi a do jaké hloubky tyto kroky dělat. Dokumentace kódu by jistě pomohla začlenění nových vývojářů do projektu.

6. Závěr

Cílem této diplomové práce bylo vypracování vlastních návrhů pro další rozvoj agilního řízení projektů ve vybrané společnosti na základě popisu a rozboru stávajících agilních postupů. V práci byla zmapována problematika týkající se agilního projektového řízení ve vybrané společnosti a byly vytvořeny vlastní návrhy pro další rozvoj řízení projektů.

Práce je rozdělena do dvou hlavních částí, teoretické a praktické. V teoretické části byly uvedeny základní pojmy projektového řízení, mezinárodní standardy projektového řízení a agilní přístupy projektového řízení, včetně definování agilního manifestu a vybraných metod. V úvodu praktické části je vybraná společnost popsána obecně, tedy její charakteristika, organizační struktura a průběh komunikace. Postupy agilního řízení projektů ve vybrané společnosti, byly nejprve popsány prostřednictvím pěti fází životního cyklu projektu dle mezinárodních standardů. Následně byl vybrán a popsán jeden konkrétní IT projekt, který byl prostřednictvím těchto fází analyzován. Na závěr analýzy případové studie, byly postupy společnosti porovnány s pilíři agilního manifestu a principy metody Scrum. Na základě tohoto porovnání byly identifikovány nedostatky, pro jejichž potlačení, či eliminaci jsou prezentovány vlastní návrhy a doporučení pro další rozvoj projektového řízení.

V závěru práce byly vlastní návrhy a doporučení diskutovány v praxi řízení projektů vybrané společnosti a byly jejím Scrum Masterem přijaty převážně kladně. Cíl diplomové práce byl tak naplněn a její přínos byl potvrzen vybranou společností, v jejíž praxi byla psána. Práce tak přinesla pohled na aktuální stav agilního řízení projektů ve vybrané společnosti, ale také vlastní návrhy a doporučení pro jeho další rozvoj.

7. Seznam použitých zdrojů

[1] SVOZILOVÁ, Alena. Projektový management. 2., aktualiz. a dopl. vyd. Praha: Grada, 2011. Expert (Grada). ISBN 978-80-247-3611-2.

[2] *Národní standard kompetencí projektového řízení: Příloha č. 3: Výkladový slovník pojmů v 3.2* [online]. 2012 [cit. 2016-03-09]. Dostupné z:

<http://old.ipma.cz/web/files/IPMA-CzNCB-slovník-pojmu-v3.2.pdf>

[3] BENTLEY, Colin. *The Prince 2 Practitioner: From practitioner to professional. Third edition.* New York: Routledge, Taylor & Francis Group, 2015. ISBN 9781138824119.

[4] DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO. *Projektový management podle IPMA.* 1. vyd. Praha: Grada, 2009. Expert (Grada). ISBN 978-80-247-2848-3.

[5] *A guide to the project management body of knowledge (PMBOK guide).* Fifth edition. Newtown Square, Pennsylvania: Project Management Institute, Inc., 2013. ISBN 9781935589679.

[6] *Manifest Agilního vývoje software* [online]. 2001 [cit. 2016-03-09]. Dostupné z: <http://agilemanifesto.org/iso/cs/>

[7] SUTHERLAND, Jeffrey Victor. *Scrum: The art of doing twice the work in half the time.* First Edition. New York: Crown Business, 2014. ISBN 9780385346450.

[8] SCHWABER, Ken a Jeff SUTHERLAND. *The Scrum Guide™: The Definitive Guide to Scrum: The Rules of the Game* [online]. 2013. [cit. 2016-03-09]. Dostupné z: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>

[9] RUBIN, Kenneth S. *Essential Scrum: A practical guide to the most popular agile process.* Upper Saddle River, NJ: Addison-Wesley, 2012. ISBN 0137043295.

[10] What is Kanban? In: *Leankit* [online]. 2016 [cit. 2016-03-09]. Dostupné z: <http://leankit.com/learn/kanban/what-is-kanban/>

[11] ŌNO, Taiichi. *Toyota production system: Beyond large-scale production.* Cambridge, Mass.: Productivity Press, 1988. ISBN 0915299143.

[12] ANDERSON, David J. *The principles of the Kanban method* [online]. 2010. [cit. 2016-03-09]. Dostupné z:

<http://www.djaa.com/principles-kanban-method-0>

[13] SCOTLAND, Karl. *Aspects of Kanban. Methods & Tools* [online], 2010, (Volume: 19, Issue: 1). [cit. 2016-03-09]. Dostupné z:

<http://www.methodsandtools.com/PDF/mt201002.pdf>

[14] POPPENDIECK, Mary a Tom POPPENDIECK. *Lean software development: An agile toolkit*. 10. print. Boston: Addison-Wesley, 2006. ISBN 9780321150783.

[15] BECK, Kent a Cynthia ANDRES. *Extreme programming explained: Embrace change*. 2nd ed. Boston, MA: Addison-Wesley, 2005. ISBN 0321278658.

[16] BECK, Kent. *Test-driven development: By example*. Boston: Addison-Wesley, 2003. ISBN 0321146530.

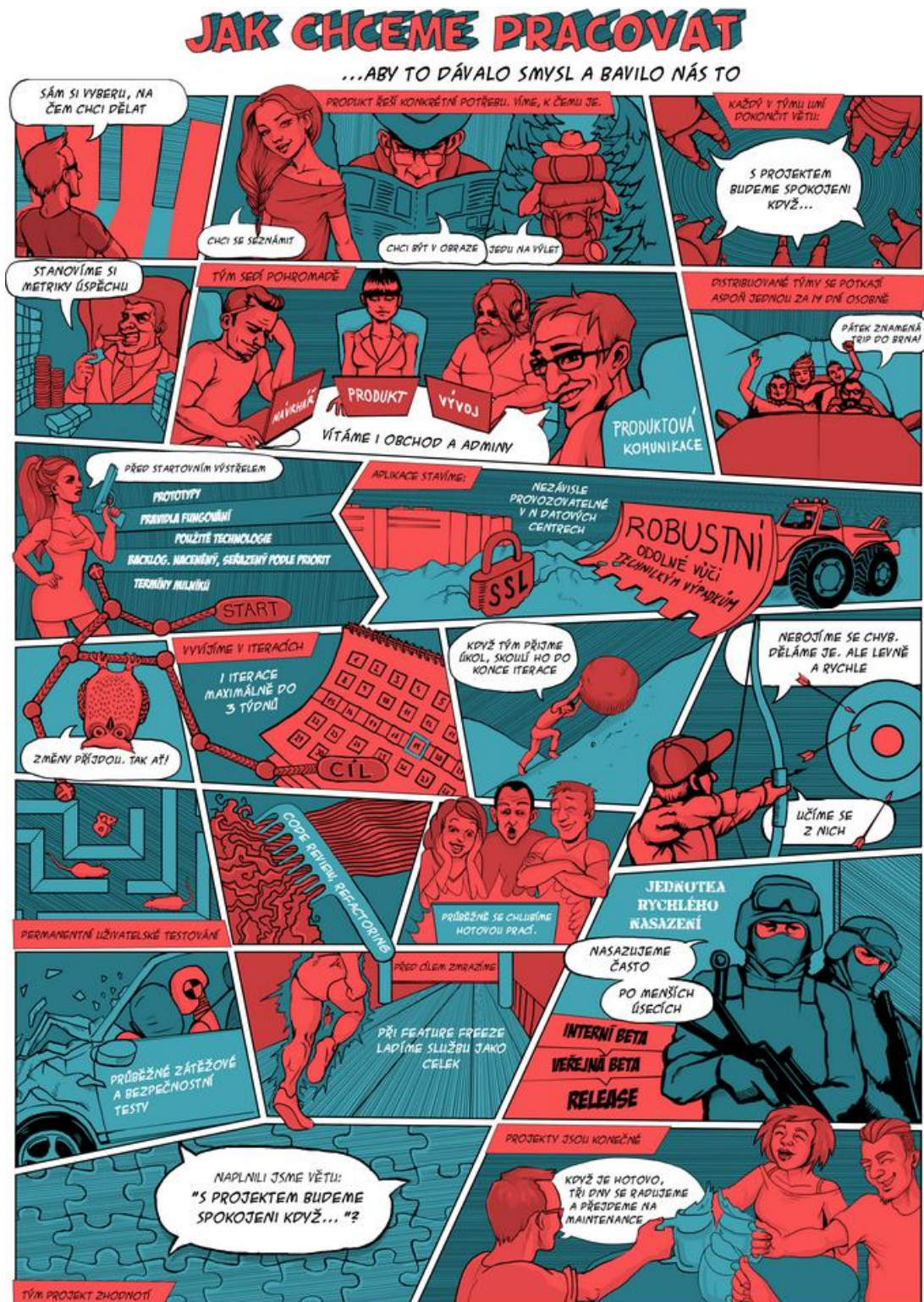
[17] Interní dokumenty anonymizované společnosti

8. Přílohy

Příloha 1: Komiks 81

Příloha 2: Řízené rozhovory..... 82

Příloha č. 1 - Komiks



Příloha č. 2 – Řízené rozhovory

1) Jaká je podoba a pozice projektové kanceláře v organizační struktuře společnosti?

„Žádná projektová kancelář ve společnosti není, o projekty se stará product manager. Procesy řídí scrum masteri, kterých máme celý tým. Jejich role se protínají, nemají stanovené žádné hranice, vše závisí na týmu a lidech co tu pozici vykonávají.“

2) Používá společnost vlastní směrnici nebo mezinárodní standard projektového řízení?

„Žádnou směrnici ani mezinárodní standard projektového řízení nepoužíváme. Vylučuje se to s principy agile metodik.“

3) Jakým způsobem je kontrolován a ohodnocován průběh práce?

„Každý den je stand up meeting, každý zde odprezentuje na čem právě pracuje, na čem bude pracovat a jestli nemá nějaké problémy. Dále jsou takzvané „plánovačky“, které začínají plánováním iterace. Iterace většinou končí demem. Demo je představení hotové práce zákazníkovi, kterým u nás je product manager. Na konci iterace také probíhá hodnocení práce, které je ve formátu zpětné vazby.“

4) Jaké metody agilního řízení jsou užívány? Pokud ano, o kterou se jedná, jaké jsou s ní zkušenosti a jak dlouho je využívána?“

„Projekty jsou nejčastěji řízeny kombinací různých agilních metod. Nejvíce je využíván Scrum, odbavování práce je založeno na principu Kanbanu. Dále využíváme Lean, Lean Startup, Design sprint, TDD a XP. Uvedené metody jsou využívány přibližně tři roky.“

5) Jak na agilní řízení pohlíží vrcholový management? Jak se tento pohled liší od linie?

„Management se stále obává plně pustit řízení projektů z rukou. Zaměstnancům se agile líbí, ale stále potřebují nějaké potvrzení, nebo schválení autority. Práce s týmem je snadná, ale je těžší přesvědčit top management.“

6) Dochází ke konfliktům mezi rolemi: projektový manažer, scrum master a product owner? Pokud ano, proč a kdy?

„Projektovými manažery jsou v našem případě scrum masteři. Scrum master má rozhodovací pravomoc, komunikuje s týmem a buduje v něm zodpovědnost. Product owner přináší vizi projektu, vymýšlí a představuje projekty. Také jedná se externími partnery a určuje prioritu projektů.“

7) Jaká dokumentace je vedena u projektů, které jsou řízeny agilní metodou? Jak a v čem se dokumentace liší od klasické projektové dokumentace?

„Projektová dokumentace jako taková není. Existuje backlog, který určuje vizi produktu, co má umět a jak se má chovat, neurčuje však, jak to má dělat. Dále vytváříme mindmapy, které zahrnují všechny milníky a vše co projekt ovlivňuje, v jaké fázi a proč. Každý tým pak zaznamenává průběh práce a je vytvářena dokumentace kódu. Na konci iterace je demo, kde je prezentováno to, co tým během iterace udělal. Způsob prezentace si řeší každý tým individuálně.“

8) Jaké přínosy společnost vidí v agilním řízení projektů?

Transparentnost, zpětná vazba od Product Ownera. Díky tomu, že skoro každý týden dodáváme funkční části kódu, tak nezhazujeme velké množství práce, když dochází ke změnám v projektu.

9) Jaké projekty jsou ve společnosti řízeny?

Převážně „Inhouse“ projekty IT charakteru. Webové služby a stránky, různé aplikace a podobně.

10) Jak se společnost stará o rozvoj projektového řízení a projektovou kulturu?

O rozvoj projektového řízení a projektové kultury ve společnosti se starají hlavně Scrum Masteři. Manažerskou praxi společnost rozvíjí pomocí interních lektorů, kteří vše šíjí na míru naší podnikové kultuře.

11) Jak probíhá alokace kapitálu a zdrojů v projektech?

Společnost netrápí finanční stránka, na experimenty je dostatek kapitálu. Případně je vše přerozdělováno na základě priority stanovené Top Managementem. Ve společnosti není na zaměstnance pohlíženo jako na zdroje, ale jako na lidi.

12) Jak jsou tvořeny projektové týmy?

Ve společnosti funguje otevřená kultura, projektové týmy se víceméně tvoří samy. Scrum Master s Product Managerem iniciují projekty a zaměstnanci se hlásí, na kterých by chtěli pracovat.