

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

DIPLOMOVÁ PRÁCE



VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

NÁZEV DIPLOMOVÉ PRÁCE/TITLE OF THESIS

Agilní projektový management ve vybrané organizaci / Agile project management in the selected organization

TERMÍN UKONČENÍ STUDIA A OBHAJOBA (MĚSÍC/ROK)

červen/2024

JMÉNO A PŘÍJMENÍ STUDENTA / STUDIJNÍ SKUPINA

Bc. Tereza Hrubá / KEMMA03

JMÉNO VEDOUcíHO DIPLOMOVÉ PRÁCE

doc. RNDr. Mirko Křivánek, CSc.

PROHLÁŠENÍ STUDENTA

Odevzdáním této práce prohlašuji, že jsem zadanou diplomovou práci na uvedené téma vypracoval/a samostatně a že jsem ke zpracování této diplomové práce použil/a pouze literární prameny v práci uvedené.

Jsem si vědom/a skutečnosti, že tato práce bude v souladu s § 47b zák. o vysokých školách zveřejněna, a souhlasím s tím, aby k takovému zveřejnění bez ohledu na výsledek obhajoby práce došlo.

Prohlašuji, že informace, které jsem v práci užil/a, pocházejí z legálních zdrojů, tj. že zejména nejde o předmět státního, služebního či obchodního tajemství či o jiné důvěrné informace, k jejichž použití v práci, popř., k jejichž následné publikaci v souvislosti s předpokládanou veřejnou prezentací práce, nemám potřebné oprávnění.

Datum a místo: 1. 5. 2024, Tuchoměřice

PODĚKOVÁNÍ

Rád/a bych tímto poděkoval/-a vedoucímu diplomové práce za metodické vedení a odborné konzultace, které mi poskytl/a při zpracování mé diplomové práce.

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

SOUHRN

1. Cíl práce:

Hlavním cílem diplomové práce je sestavit implementační plán agilní transformace a navrhnout doporučení, jak může zavedení agilního projektového managementu vyřešit výzkumný problém. Vedlejším cílem je vyhodnotit současný stav projektového managementu ve firmě XY.

2. Výzkumné metody:

V diplomové práci byla dvakrát použita SWOT analýza – na zmapování současného stavu firmy a projektového managementu. Mezi první a druhou SWOT analýzou byl realizován kvalitativní výzkum v podobě polostrukturovaných rozhovorů, které trvaly od 60 do 90 minut. Rozhovory proběhly s 5 respondenty a byli vybráni k využití trojího pohledu (majitele, vedoucího vývoje, řadových zaměstnanců). Scénář k polostrukturovaným rozhovorům se týkal několika témat: role respondenta v IT projektech, plánování a průběh Sprintů, schůzky v rámci projektového managementu, pracovní prostředí, firemní kultura a možnosti agilní transformace. Získaná data byla vyhodnocena metodou otevřeného kódování. Výsledky ze SWOT analýz a kvalitativního výzkumu byly použity k zodpovězení výzkumných otázek, potvrzení/vyvrácení hypotéz a následně k návrhům implementačního plánu na uskutečnění agilní transformace.

3. Výsledky výzkumu/práce:

Obecná hypotéza 1 „*Časové odhady vývojových a testovacích požadavků ve Sprintech dělají nesprávní pracovníci.*“ byla potvrzena – časové odhady dělá pouze vedoucí vývoje.

Obecná hypotéza 2 „*Při časových odhadech požadavků ve Sprintech dochází k nedostatečné komunikaci mezi klíčovými lidmi a pracovníky.*“ byla také potvrzena, protože vedoucí vývoje nekomunikuje ohledně odhadů s vývojovým týmem.

Obecná hypotéza 3 „*V oddělení vývoje se nekonají agilní ceremonie.*“ byla potvrzena – žádné agilní ceremonie se ve firmě XY nekonají, pouze pravidelné schůzky.

4. Závěry a doporučení:

Autorka vybrala zavedení agilních metod Scrum a Kanban. Vzhledem k problémům s nestálou délkou Sprintů, neúplnými zadáními požadavků, nedostatečnou dokumentací, špatným časovým odhadům, určování priorit požadavků a zapojením klientů do procesu bylo navrženo: zkrácení Sprintů; přesnější podoba zadání a nový systém v jeho tvoření, určování časových odhadů a priorit; určení přesnější podoby dokumentace; metody Planning Poker, T-shirt sizing a Větší – menší pro časové odhady a efektivnějšího zapojení klientů.

KLÍČOVÁ SLOVA

Agilní projektový management, agilní transformace, Kanban, Scrum

VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Národní 2600/9a, 158 00 Praha 5

SUMMARY

1. Main objective:

The main aim of this thesis is to draw up the implementation plan for agile transformation and propose recommendations how agile implementation can solve the research problem. The secondary aim is to evaluate the current situation of project management in company XY.

2. Research methods:

SWOT analysis was used twice in the thesis – to map the current situation of company and project management. Between the first and second SWOT analysis, qualitative research was carried out in the form of semi-structured interviews which lasted from 60 to 90 minutes. Interviews were conducted with 5 respondents and they were selected to use a triple point of view (owner of company, head of development, other employees). The script of semi-structured interviews covered several topics: their role in IT projects, the planning and progress of Sprints, meetings within project management, the work environment, company culture and the possibilities of agile transformation. The obtained data were evaluated using the open coding method. The results from SWOT analysis and qualitative research were used to answer research questions, confirm/disprove hypotheses and then make proposals and to propose an implementation plan for the realization of the agile transformation.

3. Result of research:

General hypothesis 1 "Time estimates of development and test requirements in Sprints are made by the wrong workers" was confirmed – time estimates are made only by the development manager.

General Hypothesis 2 "There is a lack of communication between key people and workers in the time estimates of requirements in Sprints" was also rejected because the development manager does not communicate about the estimates with the development team.

General hypothesis 3 "There are no agile ceremonies in the development department." was confirmed – no agile ceremonies are held in the XY company, only regular meetings.

4. Conclusions and recommendation:

The author of thesis chose to implement Scrum and Kanban. Due to problems with the unstable length of Sprints, incomplete specification of requirements, insufficient documentation, poor time estimates, prioritization of requirements and involvement of clients in the process, it was proposed: shorten Sprints; a more precise form of assignment and a new system in its creation, determining time estimates and priorities; determination of a more precise form of documentation; Planning Poker, T-shirt sizing and Bigger - Smaller methods for time estimates and more effective client engagement.

KEYWORDS

Agile project management, agile transformation, Kanban, Scrum

JEL CLASSIFICATION

M15 IT Management, O22 Project Analysis, O36 Open Innovation

Vysoká škola ekonomie a managementu
Nárožní 2600/9a, 158 00 Praha 5

Vysoká škola ekonomie a managementu
Nárožní 2600/9a, 158 00 Praha 5

ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení:	Tereza Hrubá
Studijní program:	Ekonomika a management (Ing.)
Studijní skupina:	KEMMA03
Téma DP:	Agilní projektový management ve vybrané organizaci
Zásady pro vypracování (stručná osnova práce):	<ol style="list-style-type: none">1 Úvod2 Teoreticko-metodologická část Projektový management, agilní projektový management, přehled metodik agilního projektového managementu, metodika práce3 Praktická část Představení organizace, charakteristika projektového managementu v organizaci, agilní transformace organizace, shrnutí a návrh doporučení4 Závěr
Seznam literatury: (alespoň 4 zdroje)	<ul style="list-style-type: none">• DOLEŽAL, J. <i>Agilní přístupy vývoje produktu a řízení projektu: komplexně, prakticky a dle světové praxe</i>. Praha: Grada, 2022. 192 s. ISBN 978-80-271-3705-3.• RIGBY, D., ELK, S., BEREZ, S. <i>Doing Agile Right: Transformation Without Chaos</i>. Boston: Harvard Business Review Press, 2020. 252 p. ISBN 978-1-63369-870-3.• ŠOCHOVÁ, Z., KUNCE, E. <i>Agilní metody řízení projektů</i>. 2. vydání. Brno: Computer Press, 2019. 224 s. ISBN 978-80-251-4961-4.• WONG, M. <i>Corporate Agility: insights on agile practices for adaptive, collaborative, rapid, and transparent enterprises</i>. Hoboken: John Wiley & Sons, 2020. 256 p. ISBN 978-1119652267.
Harmonogram:	<ul style="list-style-type: none">• Zpracování cílů a metodiky do 25. 2. 2024• Zpracování teoretické části do 11. 3. 2024• Zpracování výsledků do 15. 4. 2024• Finální verze do 1. 5. 2024
Vedoucí práce:	doc. RNDr. Mírko Křivánek, CSc.

prof. Ing. Milan Žák, CSc.
rektor

V Praze dne 30. 1. 2024

Prof. Ing.
Milan Žák
CSc.

Digitálně podepsal Prof. Ing. Milan Žák CSc.
DN: cn=Prof. Ing. Milan Žák CSc., o=CZ, o=Vysoká škola ekonomie a managementu, a.s., givenName=Milan, sn=Žák, serialNumber=ICA - 10393535
Datum: 2024.01.30 10:18:17 +01'00'

Obsah

1	Úvod	1
2	Teoreticko-metodologická část práce	2
2.1	Projektový management.....	2
2.2	Agilní projektový management.....	4
2.2.1	Fáze agilního projektového managementu	7
2.2.2	Týmy a pracovní role.....	8
2.2.3	Agilní mindset a kultura	10
2.2.4	Agilní transformace	11
2.2.5	Agilní přístup vs. tradiční přístup ve firmě.....	13
2.3	Přehled metodik agilního projektového managementu.....	14
2.3.1	Scrum.....	14
2.3.2	Kanban.....	18
2.3.3	SAFe.....	19
2.3.4	Další metodiky agilního projektového managementu	19
2.4	Metodika práce.....	21
2.4.1	SWOT analýzy	22
2.4.2	Kvalitativní výzkum	22
3	Praktická část práce	24
3.1	Představení organizace.....	24
3.1.1	Řízení vývoje produktu firmy XY.....	25
3.1.2	SWOT analýza firmy XY	26
3.2	Charakteristika projektového managementu v organizaci	29
3.2.1	Analýza získaných dat z kvalitativního výzkumu	29
3.2.2	SWOT analýza současného projektového managementu firmy XY	35
3.2.3	Vyhodnocení výzkumu.....	38
3.3	Agilní transformace organizace	40
3.3.1	Důvody pro agilní přechod, analýza současného stavu, firemní připravenost	41
3.3.2	Plán agilní transformace, vize transformace, firmy a produktu.....	42
3.3.3	Výběr agilní metody	42
3.3.4	Definování týmových rolí a sestavení vývojového týmu	43
3.3.5	Pravidelné scrumové ceremonie.....	43
3.3.6	Příprava Sprint Backlogu	45
3.3.7	Plán pilotního Sprintu.....	45
3.3.8	Seznámení s agilní metodikou, upraveným průběhem vývoje a proškolení.....	46

3.3.9 Průběh pilotu	47
3.3.10 Postupné a úplné zavedení agility	47
3.4 Shrnutí a návrh doporučení	48
3.4.1 Návrhy doporučení	48
4 Závěr	54
Literatura	56
Přílohy	I

Seznam zkratk

OaM = obchod a marketing

XP = Extreme Programming

Seznam tabulek

Tabulka 1 SWOT matice – silné stránky firmy XY	26
Tabulka 2 SWOT matice – slabé stránky firmy XY	27
Tabulka 3 SWOT matice – příležitosti firmy XY	28
Tabulka 4 SWOT matice – hrozby firmy XY	28
Tabulka 5 SWOT matice – silné stránky projektového managementu firmy XY	36
Tabulka X SWOT matice – slabé stránky projektového managementu firmy XY	37

1 Úvod

Agilní projektový management je v dnešní době stále aktuální téma vzhledem k měnícímu se prostředí, chování zákazníků, potřebě neustále zvyšovat jejich spokojenost a díky principům, procesům a nástrojům, které umožňují firmám se rychle přizpůsobit poptávce. Jeho principy, fungování a procesy mají základ v Agilním manifestu, který byl vydán v roce 2001 americkými softwarovými inženýry. Oproti agilnímu řízení projektů může tradiční projektové řízení působit zastarale a neaktuálně, protože není na současnou dobu vhodně flexibilní.

Agilní fungování firmy přináší spoustu výhod, například efektivní komunikace a spolupráce mezi pracovníky, dynamika, vyšší rychlost reagování na změny a dodání produktů/služeb na trh, větší angažovanost zaměstnanců a přináší větší konkurenceschopnost firem.

Největším rozdílem ve srovnání s projektovým managementem je délka jednoho cyklu. Projektový management se skládá z jednoho dlouhého cyklu, který obsahuje všechny potřebné fáze a na jeho konci je výsledek. Agilní projektový management se skládá z množství stejných cyklů, ve kterých probíhají stejné fáze. Díky tomu se neustále kontrolují a zlepšují části produktů nebo služeb a je možnost rychlých změn bez zbytečného vrácení se zpět.

Kvůli všem výhodám firmy přecházejí na agilní projektový management, prochází tzv. agilní transformací. Můžou se pro ni rozhodnout z důvodu vyskytnutí problémů například nízká spokojenost klientů, která má finanční dopad, neuspokojivé výsledky i přes používání správných nástrojů, nepřebírání zodpovědnosti pracovníky, nezájem zaměstnanců o výsledek, nedodržování termínů nebo špatných časových odhadů.

Poslední dvě překážky se týkají firmy XY, kterou se zabývá tato diplomová práce. Původní problém byl, že vývojový tým nedodrží předem dohodnuté termíny zákazníků, což má dopad na pozdější dodání a současně finanční dopad v podobě pokut za zpoždění. Prostřednictvím SWOT analýzy a kvalitativního výzkumu bylo zjištěno, že příčina problému je ve špatných časových odhadech požadavků. A díky řádnému zavedení agilních metod se může problém vyřešit a lepší i další procesy ve firmě XY.

Proto hlavním cílem diplomové práce je sestavit implementační plán agilní transformace a navrhnout doporučení, jak může zavedení agilního projektového managementu vyřešit problém s časovými odhady. Vedlejším cílem je vyhodnotit současný stav projektového managementu ve firmě XY.

Teoreticko-metodologická část se zaměřuje na základy projektového managementu, agilního projektového managementu, agilních metodik a metodiky diplomové práce. Agilní projektový management zahrnuje svou definici, principy Agilního manifestu, fáze agilního řízení projektů, týmové role, agilní mindset, kulturu, transformaci a vyjmenování podstatných rozdíl mezi tradičním a agilním projektovým managementem. Mezi agilními metodami je detailně popsán Scrum, Kanban, SAFe a dále jsou zmíněny i méně používané metody. V závěru teoreticko-metodologické části je představena metodika, která je složena ze dvou SWOT analýz současného stavu firmy XY a projektového managementu, které byly sestaveny s majitelem. Následně byl proveden kvalitativní výzkum ve formě polostrukturovaných rozhovorů, ve kterých se do hloubky zjišťoval pohled dalších zaměstnanců na fungování vývojového oddělení, současných procesů a jaké změny by respondenti navrhly, aby se zlepšil současný stav. Z rozhovorů byly sepsány poznámky, které sloužily k analýze dat prostřednictvím otevřeného kódování, potvrzení či vyvrácení hypotéz a zodpovězení výzkumných otázek a výzkumného problému. Na konec byl sestaven implementační plán pro firmu XY a navrhnutá doporučení.

2 Teoreticko-metodologická část práce

Teoreticko-metodologická část se zabývá dvěma způsoby řízení projektů – tradičním a agilním projektovým managementem. První kapitola se zabývá definicí projektu, projektového managementu, jeho fázemi, životními cykly projektů a chybami projektového řízení. Druhá kapitola diplomové práce se věnuje představení Agilu, agilních principů dle Agilního manifestu, definicí agilního projektového managementu, jeho fázemi, agilními týmovými rolami, agilním nastavením pracovníků. Dále agilní transformací a rozdíly mezi tradičním a agilním řízením projektů. Třetí kapitola detailně popisuje fungování Scrumu, jeho artefakty a agilní ceremonie. Dotýká se dalších agilních metod, například SAFe, Kanban nebo Extreme Programming. Poslední kapitola teoreticko-metodologické části určuje výzkumný problém, výzkumné otázky, hypotézy a zabývá se metodikou SWOT analýz, polostrukturovaných rozhovorů a jejich limity.

2.1 Projektový management

Projektový management zastupuje tradiční postup v organizacích, které plánují, řídí a dokončují projekty. Aby bylo možné se zabývat projektovým management, je potřeba se nejprve zaměřit na definici projektu a jeho vlastnosti.

Project Management Institute (2021) definuje projekt jako činnost, která má omezenou délku projektu – vzhledem k dočasnosti je stanoveno, že má jasný začátek a konec. Dle Doležala et al. (2023) je projekt určitá změna, která má výchozí stav a dostane se do cílového stavu. Doležal et al. (2023) rozlišuje na projektu několik kritérií: ohraničení, potřeba projektového týmu, jedinečnost, riziko a souhrnnost. Vzhledem k tomu, že projekt nese prvky jedinečnosti a další kritéria vyjmenovaná výše, je potřeba jej rozdělit do menších a kontrolovatelných částí, tedy fází. Aby se projekt dostal z počátečního stavu do konečného, je nutné jeho správné řízení kvalifikovanými pracovníky, které je určeno projektovým managementem.

Doležal (2016, str. 16) označuje projektový management neboli projektové řízení za skupinu předpisů, tipů a zkušeností na řízení projektů. Igberaese (2022) charakterizuje projektový management jako způsob, díky kterému je možné systematicky řídit projekty od plánování po jeho ukončení.

Řízení projektu má tradičně 4 fáze: zahájení, příprava a plánování projektu, realizace a ukončení projektu. Před začátkem projektů odděleně probíhá předprojektová fáze, která obsahuje vymezení zadání a určení, zda se projekt bude realizovat či nikoliv (Doležal, Krátký, 2017, str.20).

Doležal a Krátký (2017, str. 20) uvádí, že cílem první fáze, **zahájení**, je zformovat a schválit zadání, ve kterém bude zaznamenán cíl, smysl, termín a omezení projektu. Autoři popisují, že na zadání se podílí všechny zainteresované strany, má být srozumitelné a v písemné podobě ve formě zakládací listiny projektu. Ve chvíli, kdy je listina schválena vedením, tak oficiálně začíná projekt.

Následuje **plánování**, ve kterém se rozhoduje, co, jak a kdy bude dodáno, zda jsou pro schválený projekt zajištěny všechny zdroje a které je případně potřeba sehnat (Doležal, Krátký, 2017, str. 21). V plánování hraje velkou roli projektový manažer, který udává časový rámec, realizaci a kontrolu projektu. Doležal a Krátký (2017, str. 21) objasňuje, že jako první se určí rozsah a průběh realizace, který odpovídá na otázku, co bude v rámci projektu vytvořeno, kdy to bude hotové a za kolik.

Po plánování se pokračuje fází **realizace**. Předem určený tým pracuje na projektu, jejichž práci kontroluje a monitoruje projektový manažer. Doležal a Krátký (2017, str. 21) dodává, že

projektový manažer zároveň kontroluje také celý projekt, aby směřoval k určenému cíli. Autoři dále popisují, že realizační část končí ve chvíli, kdy jsou dokončeny a předány všechny projektové úkoly, které jsou poté ověřeny projektovým manažerem.

Poslední fáze, **ukončení**, znamená uzavření projektu. Doležal a Krátký (2017, str. 21) píše, že se porovnává projektový plán s vytvořeným projektem a celkově se vyhodnotí. Se schválením projektu se oficiálně ukončí projekt.

Naproti tomu Hughes et al. (2019) řízení projektu rozděluje, konkrétně IT, do následujících fází: nastavení projektu, analýza požadavků, design, budování, testování funkcionalit softwaru, implementace, instalace, přechod na nový software, uzavření projektu a možná kontrola a údržba. Dále autor upřesňuje, že ještě před nastavením projektu probíhá zahájení a identifikace business case, ale nebere tuto část jako součást projektových fází – pouze se udávají důvody projektu, nastavuje se rozpočet a specifikují se různá omezení.

Wysocki (2019) do životního cyklu projektového managementu řadí těchto 5 procesů: scoping / rozsah, plánování, spouštění (launching), provádění (executing) a uzavírání (closing).

Výše vyjmenované procesy, které jsou odlišně pojaté dle autorů, se určitým způsobem uskuteční – za optimálních podmínek projekt dojde ke zdárnému konci, i když projde různými procesy. Obecně má mnoho životních cyklů projektového managementu začátek, průběh a konec. Firmy si mohou vybrat, jakým způsobem a v jakém pořadí se procesy uskuteční. Podle řazení procesů a způsobů se rozdělují životní cykly do modulů/možností. Doležal et al. (2023) rozlišuje 5 modelů životního cyklu projektu: inkrementální, waterfall, iterativní, agilní a hybridní.

Inkrementální přístup je postup, při kterém se postupně tvoří výsledek (Doležal et al., 2023). V IT prostředí může postup vypadat tak, že software bude kódován po částech. Zároveň v rámci inkrementálního přístupu se také testuje a uvede se daná část do provozu (Hughes et al., 2019). To znamená, že se část softwaru vyvine, následně otestuje a tímto způsobem dokola, dokud projekt není hotov.

Waterfall neboli vodopádový způsob projektů probíhá na principu na sebe navazujících kroků. Považuje se za velmi předvídatelný a tím pádem se plánuje jen nejbližší fáze, ke které se po jejím dokončení není nutné navracet (Doležal et al., 2023). Hughes et al. (2019) zastává opačný názor, že je možné přepracovávat již dokončené fáze, ale vyžaduje to další čas a je lepší si tak práci překontrolovat než se znovu vracet k ukončené části. Waterfall znázorňuje svou typickou organizační strukturu následovně: nahoře se nachází řídicí výbo, pak manažer projektu, pod kterým jsou specialisté, poradci administrativní podpora a několik garantů výstupu a v rámci každý garant výstupu má své členy týmu (Doležal, 2016, str. 41).

V **iterativním** modulu se jako první nanečisto vytvoří výsledek projektu a poté tým podrobněji pracuje na prvním návrhu až do konce (Doležal et al., 2023). Heath (2021) definuje iterativní přístup jako opakovaný vývoj dodávaný po malých částech.

Agilní metoda funguje na iterativním principu. Tým pracuje po částech, iteracích, díky kterým vytváří hodnotu a zároveň na jednotlivé iterace dostává od klienta zpětnou vazbu (Doležal et al., 2023). Agilní přístup má také výhodu, že snižuje problémy s byrokracií a tím pádem je možné projekt dokončit rychleji (Hughes et al., 2019).

Poslední přístup, **hybridní**, je přímo v rukou firmy – může si zkombinovat moduly, které jí vyhovují a dospěje díky nim k potřebnému výsledku.

V projektovém managementu se mohou firmy dopustit mnoho chyb. Podle Caldwell (2019) se v projektovém managementu často vyskytují následující chyby:

- Podnik zachází se členy týmu jako se s pouhými zdroji – s pracovní silou potřebnou k dosažení výsledku. Konkrétněji k tomu může docházet v případě, kdy se během plánování chybně určí výpočet zdrojů a pak dochází k přerozdělování úkolů.
- Tým stráví mnoho času plánováním vzhledem k tomu, že se v rámci tradičního projektového managementu plánuje celý projekt dopředu. Efektivnější by bylo trochu času stráveného plánováním věnovat skutečné realizaci.
- Kvůli dlouhému plánování se plán stane fixním. Pochopitelně se může počítat s možnými změnami, ale naruší to celý plán a tím pádem je potřeba nový.

Podle Doležala a Krátkého (2017, str. 22) může úspěšnost projektů nejvíce ohrozit například špatně definované zadání, nedostatek pracovních kapacit, časté změny v zadání během projektu nebo malá podpora od managementu firmy.

Aby projektový management fungoval efektivně a reálně pomáhal týmům dosáhnout cílů, měl by splňovat několik kritérií (Project Management Institute, 2017):

- zjistit a splnit očekávání stakeholderů (všech zúčastněných stran);
- splnit cíle projektu;
- aktivně řešit problémy;
- dodávat projekt nebo produkt včas;
- počítat s omezeními projektu;
- správně rozvrhnout a využít zdroje;
- reagovat na změny pozitivně;
- rozpoznat včas neúspěšné/nezdařené projekty, ukončit je a vzít si z nich ponaučení.

2.2 Agilní projektový management

Kapitola Agilní projektový management pojednává o základních agilních principech, metodách, týmových rolích, které stanovil Agilní manifest a představil tak koncept Agilu. Doležal (2022, s. 19–20) definuje Agile jako: „*Je to filozofie přístupu k problémům, sada hodnot a principů, které vedou k určitému typu organizační kultury nebo z určitého typu organizační kultury vyplývají – podle toho, jak se na ně podíváte.*“ Z toho lze odvodit, že Agile není pouhá metodika nebo teorie. Jedná se o komplexní koncepci, se kterou se mění nejen chápání, ale také organizační kultura.

Rubio (2022) definuje Agile jako přístup, který je iterativní a konstantně se zaměřuje na rychlé dodání řešení zákazníkovi s přidanou hodnotou. Autor to bere jako jednoduchou rovnici: když se klient/zadavatel postaví do středu projektu, podnik je schopen dodávat včas a rychle. Autor dále uznává, že je spousta definic agility a agilního projektového managementu, ale nejdůležitější je brát Agile jako rozdílný přístup od tradičního projektového managementu.

Stellman a Greene (2015) charakterizují agilní přístup jako komplex postupů, které pomáhají pracovat efektivně a zároveň se jedná o způsob otevřeného myšlení, které pomáhá týmu sdílet informace.

Kromě způsobu myšlení Catlin a Watkins (2021) označují Agile za metodu softwarového vývoje, která by měla být účinnější než tradiční plánovací metody. Byli to softwaroví vývojáři, kteří zformovali agilitu pod Agilní manifest (2001), který říká, že:

„Jednotlivci a interakce před procesy a nástroji.

Fungující software před vyčerpávající dokumentací.

Spolupráce se zákazníkem před vyjednáváním o smlouvě.

Reagování na změny před dodržováním plánu.“

Několik zdrojů uvádí, že při rozhodování mezi dvěma stranami by se měla upřednostnit levá strana, tzn. jednotlivci a interakce, fungující software, spolupráce se zákazníkem a reagování na změny. Pokud je to možné, mohou být splněny obě strany zároveň.

Pod čtyřmi základními principy se skrývají další pravidla a zásady, které by se měly určitým způsobem dodržovat, pokud chce firma být jakkoliv agilní.

- Princip **jednotlivci a interakce před procesy a nástroji** vyjadřuje, že by měli být vyzdvihováni lidé, týmy a jejich spolupráce nad zavedenými procesy a nástroji. Jak popisují Šochová a Kunce (2019) skupina plná lidí, kteří silně spoluprací a vytvářejí si vlastní nástroje, budou mít lepší výsledek než skupina lidí, kteří se řídí přesně podle procesů bez interakce. Neznamená to ale, že by neměly existovat procesy nebo nástroje. Jen by se měly přizpůsobit pracovníkům, protože od nich firma chce úspěch a výsledky;
- Princip **fungující software před vyčerpávající dokumentace** podle Goodpasture (2015) znamená, že by se mělo spíše soustředit na fungující produkt než věnovat obrovské úsilí dokumentaci. Příliš mnoho práce s dokumentací může být bráno jako mrhání časem a lidskými zdroji. Stejně jako v předchozím bodu princip nevyjadřuje, že je dokumentace zbytečná. Tady je naopak důležitá například v případě zaučování nebo přebírání práce;
- Princip **spolupráce se zákazníkem před vyjednáváním o smlouvě** znázorňuje nenahrazování smluv za bližší komunikaci s klientem a vzájemnou spokojenost. Jak uvádí Šochová a Kunce (2019), smlouvy jsou důležité, ale je také podstatné, co zákazník chce a potřebuje;
- Princip **reagování na změny před dodržováním plánu** formuluje podle Goodpasture (2015) možnost rychle a flexibilně reagovat na změny, a ne se striktně řídit produktovým plánem. Trendy, konkurence, technologie se neustále vyvíjí a mění, takže někdy není možné mít hodně dopředu plán.

Agilní manifest formuloval dalších dvanáct principů, které rozvíjí čtyři základní principy vyjmenované výše a umožňují je implementovat do firem.

První z nich si zakládá na **uspokojení zákazníků** prostřednictvím průběžného dodávání softwaru (Agilní manifest, 2001). Díky kontinuitě mohou vidět, jakou má pro ně hodnotu a poskytnout užitečnou zpětnou vazbu, kterou si tým vezme s sebou do další iterace na vylepšení produktu. Důraz je kladen na přinášející hodnotu klientům, protože by tým neměl dodávat něco, co jim nepřináší žádnou hodnotu. Princip také zahrnuje reakci na feedback, možného zapracování a adaptování se na úpravy.

Druhý princip uvádí, že **změny jsou nevyhnutelné** a měly by být přijímány i přesto, že mohou přicházet později než je v plánu (Agilní manifest, 2001). Měly by být brány za něco přirozeného, neomezujícího a tým by jim měl být otevřen a umět s nimi pracovat. Také by měl ale brát v úvahu, které změny jsou opravdu nutné, žádoucí a praktické a neignorovat harmonogram a plán projektu. Díky zavedeným změnám se může firma stát více konkurenceschopnou a předhonit své soupeře.

Třetí princip představuje **pravidelné dodávání fungujícího softwaru** v krátkých iteracích (Agilní manifest, 2001). Cílem v rámci každé iterace je část otestovat a tím se vyhnout riziku dodání zcela špatného softwaru. Zároveň díky krátkým intervalům je možné rychle reagovat na změny.

Čtvrtý princip říká, že je podstatné, aby mezi sebou **spolupracovníci komunikovali** denně a kontinuálně (Agilní manifest, 2001). Členové by se měli pravidelně setkávat, aby diskutovali o projektu, řešili problémy a plánovali další kroky. Jak popisují Anantatmula a Kloppenborg (2021), když spolupracovníci nejsou v kontaktu, mohou se zaměřovat na jinou práci a tudíž se zpomalí dodávka.

Pátým principem se udává důležitost **motivace a podpory členů** (Agilní manifest, 2001). Firma by měla vytvářet příjemné prostředí, které naplňuje jejich potřeby. Když budou členové dostatečně motivováni a podporováni, budou více kreativní, inovativní, odpovědnější a efektivně odvádět práci. Zaměstnavatel může nabídnout například zasloužené finanční ohodnocení, flexibilní pracovní dobu, práci ze zahraničí nebo možnost osobního rozvoje. Princip zahrnuje také důvěru. Členové by si měli mezi sebou věřit a vyšší management by měl věřit týmům, že svou práci dobře odvedou a budou plnit své závazky.

Šestý princip vyzdvihuje **efektivní sdělování informací** a volnou komunikaci, díky kterým se předchází nejasnostem a snižuje se riziko vzniku komunikačních šumů (Agilní manifest, 2001). Sdělování informací souvisí se čtvrtým principem, který zahrnuje brainstorming, debaty, agilní ceremonie a podobně.

Sedmý princip určuje **měření pokroku pomocí fungujícího softwaru** (Agilní manifest, 2001). Na rozdíl od tradičního přístupu, který klade důraz na plánování a dokumentaci, tady je v hlavní roli funkční ukázka softwaru. Pokrok souvisí se třetím principem, kdy sleduje pokrok v krátkých iteracích.

Osmý princip představuje schopnost dlouhodobě držet **stejné tempo** a uvažování o **udržitelnosti** (Agilní manifest, 2001). Agilita podporuje udržitelnost ve smyslu neplýtvání zdroji, časem a snaží se minimalizovat pracovní stres a přetížení. Přiměřené tempo je důležité i z důvodu kvality práce.

Devátý princip zdůrazňuje významnost **technické výjimečnosti a designu** (Agilní manifest, 2001). Cílem agility je dodávat kvalitní, funkční a efektivní software, který má dobrý design. Pokud například není kvalitní a má v kódu chyby, může vznikat technický dluh a tím pádem nesplňuje prvky technické výjimečnosti. Proto je potřeba softwaru věnovat neustálou pozornost.

Desátým principem se za agilní cíl bere **jednoduchost** (Agilní manifest, 2001). Je myšlena tak, že by vývoj softwaru neměl být zbytečně složitý, protože může způsobit zpoždění dodávky, vzniknutí úzkých hrdel a snížení efektivity a produktivity týmu.

Jedenáctý princip zdůrazňuje **samo-organizující týmy** (Agilní manifest, 2001). Základy těchto týmů jsou samostatnost a autonomie, které přispívají k lepší spolupráci a komunikaci mezi členy a dalšími týmy. Znamená to, že samo-organizující tým ví nejlépe, jaké má schopnosti, znalosti, potřeby, která se mu hodí k řízení projektů a případná omezení. I tento princip je propojen s ostatními – díky samo-organizaci se zvyšuje efektivita, rychlost, kvalita, motivace členů a je nutná komunikace a spolupráce.

V posledním, dvanáctém, agilním principu jde o **neustálé se zlepšování** a přemýšlení týmu nad zvyšováním jejich efektivity (Agilní manifest, 2001). Ne vždy všechno funguje, a proto je důležité, aby se členové ohlédlí zpět, i v rámci agilních ceremonií, vyhodnotili své výsledky a případně navrhli řešení, jak předejít již vniklým problémům. V praxi se zpravidla hodnotí vždy na konci iterace. Neustálé zlepšování je spojeno s přijímáním změn a zpětné vazby. Zároveň by měl mít tým čas a prostor na zkoušení nových postupů nebo procesů, aby zjistil, zda by mu fungovaly lépe než současné. Účelem neustálého zlepšování je také přinést zákazníkovi vyšší hodnotu softwaru nebo produktu.

Caldwell (2019) nazývá agilní projektový management metodikou, ve které se s projektem pracuje v krátkých intervalech a zároveň se může během průběhu upravovat. Podle autora se postupuje tímto způsobem proto, aby hodnota výsledného produktu byla díky lepší výkonnosti vysoká. Rovněž tak podnik dosáhne nižších nákladů a týmu bude umožněno mít proaktivnější přístup než v tradičním projektovém managementu. Rovněž lze agilní projektový management charakterizovat jako moderní přístup, který se zaměřuje na adaptabilitu, flexibilitu, kooperaci za používání agilních hodnot, týmových rolí a principů.

Agile udává určité hodnoty, principy, týmové role, ceremonie a artefakty vyplývající z Agilního manifestu, které jsou navzájem propojené. Díky artefaktům, ceremoniím a týmovým rolím je možné dodržovat a plnit hodnoty a principy – se zapojením všech částí agilního přístupu dohromady. Proto nemá smysl agilní metody implementovat postupně (Šochová, Kunce, 2019, str. 167.)

2.2.1 Fáze agilního projektového managementu

Caldwell (2019) rozděluje agilní proces do následujících fází:

- plánování projektu;
- vytvoření časové osy projektu, tzv. roadmapy;
- plánování releasu / vydání / vypuštění;
- plánování cyklů / Sprintů;
- pravidelné agilní ceremonie a pravidelná komunikace;
- pravidelné retrospektivy a zpětná vazba.

Při **plánování projektu** je důležité, aby tým před zahájením projektu plně pochopil konečný cíl. Vysvětlí se jim potenciální hodnotu, kterou úspěch v projektu přinese týmu, společnosti a jak by to mělo být dosaženo. Nastavuje se rozsah projektu ale tak, aby působil změnitelně.

Vytvoření roadmapy je klíčové pro vývoj softwaru. Jde o časové rozdělení, které udává, jaké funkce budou tvořit finální produkt a předává informaci týmu, na co se má v každé fázi zaměřit.

V rámci časové osy je nastaveno jedno pevné datum releasu (**nasazení**), kterým končí celý projekt. Před zahájením projektu se vytvoří plán nanečisto pro vydávání funkcí. A při zahájení nového Sprintu se znovu přezkoumá a přehodnotí.

Před zahájením každého iterace se všechny zainteresované strany společně **naplánují**, čeho má být dosaženo. Součástí plánu je důležité se ujistit, že zátěž je rovnoměrně rozdělena mezi členy týmu.

Aby tým efektivně plnil své úkoly v každém Sprintu, musí se **pravidelně potkávat** a sdílet informace o aktuálních stavech. Na konci každé iterace se pak potká na **retrospektivě**.

Šochová a Kunce (2019, str. 215) zmiňují také tzv. Spotify model, které se firmy snaží často neúspěšně napodobit i přes poznámku Henrika, že se nejedná o model vhodný k "následování".)

Stejně jako tradiční projektový management ani agilní projektový management není dokonalý a během procesů může nastat řada chyb. Mezi 2 základní chyby Caldwell (2019) řadí velkou závislost na rychlém rozhodování a možné ztracení stopy. Pod velkou závislostí na rychlém rozhodování je myšleno, že agilní přístup není pro lidi, kteří mají tendenci zdržovat průběh projektu. Autor je názoru, že je pro lidi, kteří se dokážou rychle rozhodnout a jsou schopni zapojit do rozhodování i další členy týmu, protože mohou přijít s lepším řešením problému. Ztrátou stopy Caldwell (2019) myslí, že tým zapomene cíl kvůli rozdělení projektu na menší části. Obecně se proto doporučuje po každé hotové části projektu připomenout celému týmu, jaký je jeho hlavní cíl.

2.2.2 Týmy a pracovní role

Agilní týmy jsou nosným prvkem agilního řízení projektů, podniků a vývoje, jelikož v nich pracují lidé, kteří mají agilní mindset, jsou do jisté míry samostatní, kreativní a umí přijímat zpětnou vazbu. Zároveň v nich jsou zastoupeny nezbytné profese a kompetence, které jsou potřebné pro úspěšné dokončení projektu. Vyznačují se cross-funkčností, spoluprací, flexibilitou, adaptabilitou, autonomií a schopností rychle reagovat na změny.

Charakteristické agilní pracovní prostředí se podle Spotify frameworku dělí na tzv. Squad, Tribe, Chapter a Guilda. Lundberg (2020) popisuje, že při transformaci tradičního modelu by měl být rámec pouze inspirací a neměl by se přesně kopírovat, protože každá firma je odlišná. Dále autor zmiňuje, že rámec je založen na spolupráci, vstřícnosti a převzetí odpovědnosti, což jsou ideály každého týmu.

Po srovnání několika zdrojů by se každý Squad měl skládat z tzv. Product Ownera neboli vlastníka produktu, Scrum Mastera a vývojářů.

Product Owner je odpovědný za celý Squad a obecně pomáhá týmu dosáhnout výsledků. Březina (2020) ho popisuje jako ústředního člena týmu, který chápe cílového zákazníka a dokáže najít nejlepší řešení k uspokojení jeho potřeb. Je zodpovědný za prioritizaci a funkcionalitu produktu, který Squad společně vytváří. Od začátku určuje propozici produktu a dále pak určuje jeho směr do vývoje. Caldwell (2019) ho také považuje za klíčového člena, který zná produkt skrz naskrz a doplňuje jeho roli o výměnu informací a vzájemnou komunikaci mezi týmem, klientem a dalšími zainteresovanými osobami. Podle Kely (2019) by měl vlastník produktu využívat znalosti a zkušenosti z oblasti projektového řízení, podnikání a konzultování k vytvoření a vyvinutí správného produktu. Šochová a Kunce (2019, str. 47) jsou názoru, že Product Owner by měl 80 % svého času věnovat zákazníkovi a 20 % týmu.

Další zásadní pozicí je **Scrum Master**. Doležal (2022) ho popisuje jako člena, který má na starosti implementaci agilní metodiky Scrum a snaží se vytvářet samoorganizující (self-organized) a multifunkční (cross-functional) tým. Blair (2020, str. 29) dodává, že ale nemá řídicí funkci. Kromě implementace dohlíží na správné a efektivní používání Scrumu a pomáhá týmu odstraňovat vzniklé překážky. Caldwell (2019) nazývá Scrum Mastera jako projektového facilitátora, který také musí chápat vedení projektu. Protože dle autora je to osoba, která má za úkol reportovat a informovat další zainteresované osoby o uskutečněném pokroku. Šochová a Kunce (2019, str. 43) ho zároveň nazývají koučem, protože má motivovat členy týmu a starat se o jejich rozvoj. Autoři upozorňují, že kombinování dalších týmových rolí se Scrum Masterem špatně funguje. Docházelo by tak ke křížení pravomocí a mohly by nastávat konflikty rolí. Zároveň to ale neznamená nahrazení rolí, obě jsou potřebné.

Tým vývojářů vyvíjí kvalitní produkt, za který zodpovídají. Hunt (2018) uvádí, že ideální počet vývojářů v týmu je sedm až devět – mělo by jich být takové množství, aby se dokázali rychle přizpůsobovat změnám a dokončit svou práci. V agilním pracovním prostředí mohou být více zainteresováni do udávání priorit a plánování, protože k tomu mají odpovídající odborné znalosti. Blair (2020, str. 30) se domnívá, že plánování patří mezi hlavní úkoly vývojářů. Zároveň by jim měla být poskytnuta co největší autonomie a dostatek zdrojů, aby byli schopni vytvořit kvalitní produkt (Caldwell, 2019). Do dalších úkolů vývojového týmu patří spolupráce, časový odhad (estimace), vytváření reportů o aktuálních stavech, podávání hlášení o stavech, výrobení a doručení produktu (Blair, 2020, str. 30).

Meyer (2014) zmiňuje, že vzhledem k tomu, že **zákazník** by měl být dle agilních principů středem pozornosti, je možné ho přímo přijmout jako člena týmu. Dále autor uvádí, že míra zapojení není pevně daná. Agilita dává prostor firmám, aby ji přizpůsobily vlastnímu byznysu. Firmy ho mohou zapojit od začátku až do konce projektu nebo naopak jen na některé

workshopy nebo ceremonie týmu. Šochová a Kunce (2019, str. 55) uvádí, že je možné dát klientovi přístup přímo do Backlogu, kde může přímo pozorovat, co je hotové. Další možností je prezentovat hotovou práci po každé iteraci. Klient pravidelně vidí pokrok a je pak více chápavý k vyskytnutým překážkám (Šochová, Kunce, 2019, str. 55). Na druhou stranu je jen na firmě, jak si nastaví vztah s klientem, tak aby to fungovalo.

V některých situacích může do Squadu patřit **agilní projektový manažer**. Podle Hunt (2018) je pomocníkem Product Ownera a pomáhá týmu, kde je potřeba a tím zvyšuje jejich efektivitu. Na druhou stranu dle některých zdrojů není považován za člena týmu nebo jako povinný a jeho role může být tak rozdělena mezi další lidi v týmu. Šochová a Kunce (2019, str. 57) také dávají význam roli manažera, i přesto že patří do klasického projektového managementu. Manažer není Scrum Master ani neřídí tým, který by měl být multifunkční a samoorganizující. Úlohou manažera v agilním týmu je "vytvářet a zlepšovat fungující firemní prostředí". Dá se nazvat i tzv. agilním leaderem, který je po ruce Scrum Masterovi (Šochová, Kunce, 2019, str. 57).

Do agilního týmu je možné zařadit **mentora** (Caldwell, 2019), jehož úlohou je předávat své zkušenosti a rady vzhledem k jeho dlouhoháhlým zkušenostem s agilní metodikou. Dle Caldwell (2019) mentor může stát mimo agilní tým nebo i podnik.

I přesto že se podle Caldwell (2019) **vyšší management** nepodílí přímo na vývoji, tak hraje důležitou roli v agilním projektovém managementu – jeho povinností je udržovat upřímnou, otevřenou komunikaci, díky které ukazuje ostatním členům týmu, že se nemusí bát jeho autority.

Co se týká dalších členů ve Squadu podle Project Management Institute (2017) ho můžou doplňovat, pokud jsou jejich znalosti a dovednosti nezbytné k vytvoření produktu. Může se jednat o designéry, analytiku, testery a podobně.

Aby mohl tým dobře fungovat a být agilní, měl by přebírat plnou odpovědnost a zodpovědnost za svá jednání, důvěřovat, dostávat závazkům, nebát se konfliktů a projevovat týmový zájem o výsledky (Šochová, Kunce, 2019, str. 158–160). Autoři jsou názoru, že jakmile "selže jeden člen týmu", tak "selže celý tým".

Šochová a Kunce (2019, str. 160) záměrně odlišují **zodpovědnost** a odpovědnost. Podle autorů zodpovědnost vzniká ve chvíli, kdy tým přijme opatření, aby se situace v budoucnu neopakovala. Autoři uvádí za příklad stěžování si na délku Standupu – místo toho, aby si tým pracovníci stěžovali, vnesou to na Retrospektivě a domluví se na řešení, které bude vyhovovat všem.

Převzetím odpovědnosti Šochová a Kunce (2019, str. 160) myslí vymlouvání se, proč daný úkol nešel splnit – typickou "výmluvou" je, že to nějak půjde příště. Autoři popisují, že tým tímto způsobem nemá motivaci něco změnit a může to být způsobené nedůvěrou v sami sebe, management nebo podnikání.

S odpovědností je spojená **důvěra**, jak bylo popsáno v předchozí větě. Šochová a Kunce (2019, str. 158) se domnívají, že když neexistuje v týmu důvěra, tak členové schovávají své chyby, nepomáhají ostatním, nechtějí s ostatními spolupracovat nebo pomlouvají.

Dostání závazků představuje dodržení slibu, tedy řádné dokončení produktu/projektu. Když tým nedodrží závazky, může to být znovu spojené s nedůvěrou.

Nebát se konfliktů znamená nebát se si vyslechnout odlišné názory a místo vyvolání konfliktu o nich konstruktivně debatovat.

Šochová a Kunce (2019, str. 160) si pod **týmovým zájmem na výsledku** představují nastavení společného cíle, ve kterém tým vidí smysl.

2.2.3 Agilní mindset a kultura

Agilní mindset a kultura podporují celkové fungování agilního projektového managementu, proto je potřeba se kromě agilních metod a týmových rolí se zaměřit na agilní nastavení zaměstnanců.

Moreira (2017) prohlašuje, že agilní mindset má být založen na agilních principech a hodnotách, které rámuje Agilní manifest. Na druhou stranu Mundra (2018) uvádí, že se v agilním manifestu nikde nehovoří o mindsetu. Zároveň je ale všeobecně známo, že agilita požaduje odlišný způsob myšlení. Dle Mundry (2018) se mindset člověka skládá z jeho přesvědčení, hodnot, vnímání, postojů a propojuje ho s kulturou, se kterou dohromady tvoří ekosystém. Autor ho zjednodušeně popisuje tak, že kultura a mindset se navzájem ovlivňují a společně na ně mají dopad další okolnosti.

Podle (Waldock, 2015) se agilní mindset skládá ze stavu, obsahu a struktury. Autor popisuje stav jako přijetí agilní filozofie, obsah jako činnosti přispívající k úspěchu a strukturu jako metodologii rozpadající se na různé techniky a nástroje.

Další definici agilního mindsetu představuje Cole (2020) – podle něho se jedná o způsob myšlení, který je potřeba k dosahování nejlepší práce. Je to set hodnot a postojů, mezi které řadí například flexibilitu, podporu, závazek a neustálou chuť se zlepšovat.

Agilní mindset je rámec, který formuje způsob, jakým jednotlivci a týmy přistupují k řešení problémů, agilnímu plánování projektů a interakci s ostatními členy organizace. Jednou z klíčových charakteristik agilního mindsetu je orientace na zákazníka. Zaměstnanci pracují na tom, aby co nejlépe porozuměli potřebám zákazníka a zaměřují se na dodání maximální hodnoty prostřednictvím své práce. Důraz je kladen na rychlé reakce na změny a schopnost přizpůsobit se novým podmínkám a požadavkům.

Agilní mindset podporuje spolupráci a týmovou práci. Týmy pracují jako samostatné jednotky, které sdílejí společný cíl a odpovědnost za jeho dosažení. Důraz je kladen na otevřenou komunikaci, důvěru a respekt mezi členy týmu.

Transparentnost a otevřenost jsou dalšími prvky agilního mindsetu. Informace jsou sdíleny volně a otevřeně, aby všichni členové týmu měli přístup k potřebným informacím pro efektivní práci. Týmy také pravidelně reflektují nad svou prací a hledají způsoby, jak ji dále zlepšovat a zefektivnit.

Agilní mindset není pouze o používání konkrétních metodik, ale o změně kultury a hodnot v organizaci.

Pokud má firma zaměstnance s agilním mindsetem, není to všechno. Aby se jejich potenciál mohl vhodně využívat a podporovat, musí se mu být přizpůsobena firemní kultura. Podle Cole (2020) by taková kultura měla mít následující prvky:

- chování, v rámci kterého je třeba empatie a podpora;
- charakteristické schůzky, které se konají v AP (více jsou popsány v kapitole 2.3.1);
- hodnoty, kterých se budou pracovníci držet (například soustředěnost, respekt, odhodlání nebo upřímnost);
- styl komunikace a etiketa;
- odpovědnost ve smyslu, že každý je odpovědný za určitou část, ale na druhou stranu zahrnuje kolektivní odpovědnost.

Výše bylo popsáno, jak má vypadat ideální agilní kultura. Několik autorů stojí za názorem, že člověk nemusí přijít do firmy a být okamžitě nastaven na tento mindset. Moreira (2017) podotýká, že někteří zaměstnanci potřebují čas na přijetí nového druhu myšlení, přístupu a překlenout tzv. agilní propast. Autorka vysvětluje přesáhnutí agilní propasti tak, že kromě

přijmutí agilních hodnot by člověk měl být schopen vidět změnu v chování a kultuře. Konkrétně se na lidi zaměřují principy zahrnující spolupráci, důvěru, podporu a motivaci, které Moreira (2017) považuje za klíčové hodnoty agilní firemní kultury.

V každé firmě mohou nastat situace, kdy pracovníci nemají agilní mindset. Jak uvádí Denning (2018) ve svém výzkumu, takoví lidé mohou využívat jakýkoliv nástroj či proces a pracovat podle knihy a i přesto nepřinášejí žádné benefity. Naopak pokud zaměstnanci s agilním mindsetem odvádí svou práci, nezáleží na tom, s jakým nástrojem či procesem pracují, věci fungují správně.

Celkově lze konstatovat, že se do agilního mindsetu zařazuje přesvědčení, flexibilita, závazek, podpora, respekt, otevřenost, odvaha, soustředěnost, empatie, styl komunikace, kolektivní odpovědnost, upřímnost, etiketa, odhodlání a schopnost se neustále vyvíjet a zlepšovat.

2.2.4 Agilní transformace

Kandha (2018) definuje agilní transformaci jako cestu k uspokojení rychle se měnících potřeb zákazníků na trhu. Autorka jako výhody agility uvádí například zvýšení kvality produktu, rychlejší dodávky, zkrácení doby uvedení produktu na trh nebo vybudování motivovaného a výkonného týmu. I přes všechny výhody agilních metod se stává, že agilní transformace selhávají. Podle Kandha (2018) tato situace nastává až v 70 % procentech, zbylé firmy transformaci projdou díky svému prostředí a vedoucím. Důvody mohou být následující: nedostatečná podpora od vedení firmy, centralizované řízení, pokračování v tradiční organizační struktuře, nepřítomnost odborníka pro agilní transformaci nebo vnímání transformace jako pouhá změna procesů (Kandha, 2018).

Dle Martin (2019) je agilní transformace přechod v hodnotách. Mezi agilní hodnoty patří například efektivní a rychlá komunikace nebo zpětná vazba.

Březina (2020, str. 50) rozlišuje transformaci a adopci. Autor vnímá rozdíl ve vnímání principů, týmů, výkonnosti nebo změn. Například adopce je vnímání jako krátkodobá, kdež to transformace je dlouhodobá. Je důležité poznamenat, že dle autora je agilní transformace brána jako **změna celé firmy**. Dále autor uvádí, že agilní transformace musí projít celou firmou ve smyslu týmových rolí, organizační struktury, odpovědnosti a firemní politiky. Jako příklad změny pracovní role je možné uvést programátora, který byl nastavený na dostávání zadaných úkolů.

Mezi nejčastější důvody, proč firmy přechází na agilitu, Šochová a Kunc (2019, str. 31, 32) uvádějí:

- Klienti v dnešní době chtějí výsledky dostávat po malých kouscích velmi rychle, a proto je lepší jít iterativní cestou.
- Agilní metody pomáhají snadněji odhadovat časový plán projektu a zapojuje do něho celý tým.
- Dle zkušeností autorů je agilita účinnější z hlediska spolupráce více lidí. Například v rámci vývoje softwaru je efektivnější využít párové programování nebo Scrum než aby na vývoji pracovali jednotlivci oddělen.
- Agilní metody zapojují do procesu zákazníka, aby tým dostal informace o jeho očekávání, představách a prezentoval mu průběžně výsledky. Díky tomu se na celý tým přenesla odpovědnost za kvalitu.

Ještě než firma začne se změnami Šochová a Kunc (2019, str. 173) doporučují uspořádat týmové cvičení: nakreslí se kolo skládající se z 8 kousků, které znázorňují komunikaci, efektivitu, spokojenost zákazníka, spolupráci, reagování na změnu, kvalitu, spokojenost týmu

a předvídatelnost. Následně si tým projde části a ohodnotí je od 1 do 10. 1 znamená nejnižší spokojenost a 10 naopak největší spokojenost.

Pokud firma přemýšlí, jak udělat první krok, nemělo by to fungovat způsobem svolání celofiremní schůzky a oznámení majitele o přechodu na agilní projektový management nebo mít pouze k dispozici report od externí firmy (Šochová, Kunce, 2019, str. 173).

Březina (2020, str. 52) doporučuje agilní transformaci začít následujícími kroky:

- vysvětlit pracovníkům, jaká se chystá změna;
- definovat, co bude firma nebo vývojový tým dodávat;
- určit definici výsledku – hotového požadavku;
- složit tým/y podle strukturu produktu;
- pouze vlastní produktu udává cíl;
- vedení nebo manažeři udržet od vývojového týmu.

U zavádění agilních metodik Březina (2020, str. 54) radí zavádět změny postupně. Postupné zavádění je zamýšleno z důvodu adaptace na změnu a zjištění, zda funguje či ne.

Aby byla možná plná implementace agilních metod, je důležitý nastavit agilní mindset pracovníků a přijmout jisté agilní hodnoty. Caldwell (2019) mezi ně řadí: rychlý vývojový proces, klient na prvním místě, improvizace a adaptace.

V průběhu agilní transformace Šochová a Kunce (2019, str. 197) doporučuje začít IT oddělením, které v malých firmách nemají být zbytečně rozdělená na malé části, například testing a development je efektivnější sloučit do jednoho oddělení. Po úpravě IT oddělení autoři uvádí, že změna začne pronikat do dalšího oddělení – businessu. Přenastavením mindsetu se IT zaměstnanci budou více ptát na zadání od klientů a nebude jim stačit pouze předané zadání. Co se týče postupování napříč oddělení Šochová a Kunce (2019, str. 197) jsou názoru, že není nutné hned měnit organizační strukturu za předpokladu, že firmě vyhovuje.

U malých firem Šochová a Kunce (2019, str. 177) navrhují jednoduše začít, zkusit implementaci a nezapomenout na agilní kulturu a mindset. Březina (2020, str. 50) souhlasí s tvrzením o začínání, konkrétně něco zavést a něco se stane.

Spayd a More (2020) uvádí, že pokud chce firma přejít na agilní fungování a nemá s tím zkušenosti, měla by najít/najmout lidi, které ji s tím pomohou.

Lindsay (2024) u agilní transformace uvádí jako příklad Spotify, které nejprve přijalo agilní principy a experimentovali s nimi. Autorka dále uvádí, že poté začínali u Scrumu a posunuli se nad jeho implementaci, která následně vedla ke kulturnímu přizpůsobení.

Mezi časté omyly o agilní transformaci Březina (2020, str. 54) řadí:

- Přebírání modelů známých společností, které jim fungují (například Google, Spotify).
- Vedení se domnívá, že po agilní transformaci budou plány naplněny na 110 % a vše bude perfektně fungovat.

Pokud se firma nebo Scrum Master setkává s odporem přijmout, Březina (2020, str. 51) doporučuje se zamyslet, proč se vyskytuje tento typ překážek a porozumět pracovníkům, proč cítí odpor. Dále doporučuje řešit problémy na společných ceremoniích, ne na soukromých schůzkách. Problémy, které jsou krátkodobého charakteru, by se měly řešit na denních stand-upech, a dlouhodobější problémy by se měly na plánování či retrospektivě.

Agilní transformaci může brzdit management/vedení. Dle Březiny (2020, str. 183) agilní transformaci zpomalují kontrolování využití kapacit pracovníků, úpravy naplánovaného závazku, despekt k rozhodnutím týmu, neustálé změny ve členech týmu nebo nastavení grafu „velocity“ jako cíl.

Při transformaci tradičního pracovního prostředí je důležité, aby si firmy uvědomily, že když přechází na agilní prostředí, neznamená to nutně, že musí plnit všechny pravidla a hodnoty agility. Jak sama agilita hlásí, je na každé organizaci, jak si ji přizpůsobí.

2.2.5 Agilní přístup vs. tradiční přístup ve firmě

Agilní a tradiční přístup jsou dvě rozdílné filozofie a považují se za protikladné. Zřetelné rozdíly mezi přístupy vyjadřuje Agilní manifest porovnáním dvou stran a další z toho vyplývají. Agile se liší například v přístupu k vývoji, struktuře organizace, vedení týmů, pořadí procesů, dodávce řešení, formování postupu k cíli, práci se zpětnou vazbou, prostředí nebo nastavení mindsetu pracovníků.

Wright (2022) uvádí, že **tradiční přístup** je centrální a není moc prostoru pro změny na rozdíl od agilního, ve kterém se přímo podporují inovace díky experimentování a neustálou možností se učit.

Wright (2022) také popisuje rozdíly ve **struktuře** a **odpovědnosti**: tradiční uspořádání se skládá z odborníků, kteří spolu pracují, ale každý má svou roli. Když jeden pracovník dokončí svou práci, předá ji dalšímu odborníku. Tím, že se čeká na jednotlivá dokončení, může dojít ke zpoždění a protažení projektu. Je zde tedy jasně vytyčená odpovědnost za jednotlivé úkoly. Zatímco u agilního přístupu celý tým plný potřebných kompetencí dodává řešení a nedělí tolik odpovědnost na členy (Wright, 2022). Pokud se stane chyba, může za ní celý tým.

Dále se postupy liší v **cestě k cíli**. Jak Březina (2020) zmiňuje, cílem tradičního postupu je předvést celé řešení v termínu s domluvenými náklady. Agilní postup funguje na principu krátkých iterací, které zákazník může pravidelně sledovat a rychleji vidí výsledky.

Co se týče **pořadí procesů**, jsou dané kroky, kterými je nutné projít. V rámci tradičního přístupu by každá fáze projektu měla být nejdříve dokončena, aby se mohlo pokračovat dál (Wright, 2022). Agilní zastává možnou změnu pořadí nebo současné plnění.

Na rozdíl od tradičního řízení projektů, agilní přístup aktivně pracuje se **zpětnou vazbou**, která se sbírá a patří do vývojových činností (Březina, 2020). Dále pracuje se **zákazníkem** v průběhu procesů, kdežto tradiční až na konci.

Velkým rozdílem je také **plánování projektu**. V klasickém projektovém managementu se vše nejdříve naplánuje (i jednotlivé kroky), sepíše se detailní dokumentace, metodika, odhadne se, jak dlouho co bude trvat a vznikne kompletní plán, který je daný a nemusí se počítat se změnami (Wright, 2022). Na druhou stranu agilně se nebude detailně plánovat celý projekt, protože se předem počítá se změnami a dynamikou. Práce se rozdělí na části a každá část se bere jako samotný projekt, který se konzultuje, kontroluje a předává se zpětná vazba (Wright, 2022). Se zpětnou vazbou se dále pracuje a bere se z ní ponaučení pro další iterace.

Podle Blair (2020, str. 14) je tradiční a agilní projektový management postaven na podobných cyklech. Autor je názoru, že v obou managementech se iniciuje, plánuje, realizuje a následně uzavírá. Rozdíl je v tom, že v tradičním projektovém managementu výše vyjmenované kroky znamenají jeden cyklus, který se neopakuje, tedy jeden ukončený projekt. V agilním projektovém managementu kroky také znamenají jeden cyklus, který se ale provádí opakovaně kvůli rozdělení projektu do menších částí a neustálému zlepšování.

Wright (2022) ale zmiňuje, že oba přístupy mají **stejný cíl**: někdo vždy nese odpovědnost, řídí projekt a firma chce vytvořit přínosný produkt.

2.3 Přehled metodik agilního projektového managementu

Agilita si přizpůsobila a případně přetvořila principy fungování firmy, pracovní prostředí, management, kulturu, pracovní role a procesy díky **agilním metodám**. Vznikly proto, že k flexibilnímu a dynamickému způsobu práce jsou potřeba praktiky, které budou umožňovat rychle reagovat na příchozí změny, požadavky a bude se moci jejich fungování v průběhu projektů měnit. Obecně slouží metody týmům, které efektivně komunikují, spolupracují, jsou otevřeni změnám a novým myšlenkám. Zároveň agilní postupy zvyšují schopnost inovovat. Agilní metodiky se řídí podle tzv. iterativního a inkrementálního vývoje (Heath, 2021).

Mezi nejčastěji používané metody podle Ashmore a Runyan (2014) patří Scrum, Extreme Programming (XP), Kanban, Lean Software Development, Feature-Driven Development a Crystal.

2.3.1 Scrum

Power a Heavin (2021) definují Scrum jako jednoduchý rámec, díky němuž tým dokáže efektivně vyřešit složité problémy a zároveň dodá produkty v nejvyšší možné hodnotě. Scrum se na rozdíl od Kanbanu provádí v krátkých intervalech, které se nazývají **Sprinty**. Jejich délky závisí na druhu produktu, ale neměly by být delší než měsíc a zároveň musí být vždy stejně dlouhé (Březina, 2020, str. 61). Dle autora se také odvíjí délka o Sprintu od schopnosti vlastníka produktu určit vizi. I tak je stále na vlastníkovu produktu vytvořit zadání požadavků tak, aby byl tým schopen je stihnout v krátkém cyklu.

Podle Heath (2021) se Scrum rozděluje na tři části: tým, artefakty a ceremonie. Tyto části jsou navzájem propojené – tým pracuje pomocí artefaktů/informačních zdrojů a setkává se prostřednictvím ceremonií. Scrumový tým se skládá ze Scrum Mastera, Product Ownera a vývojového týmu. Více jsou týmové role popsány v kapitole 2.2.2. Stern (2017) zahrnuje do Scrumového týmu kohokoliv, kdo se podílí na projektu.

Podle Blair (2020, str. 29) do Scrum týmu patří 3 klíčové role: Product Owner, Scrum Master a vývojový tým. S přihlédnutím na artefakty úkolem Product Ownera (vlastníka produktu) definovat tzv. User Stories a určovat priority v Backlogu (Produktovém Backlogu).

Moran (2015, str. 19) charakterizuje Scrum jako agilní metodu specializující se na vývoj produktu s menším zatížením/snahou na řízení.

Artefakty Scrumu jsou Product Backlog, Product Increment a Sprint Backlog, které pomáhají týmu, aby bylo vše srozumitelné a daly se činnosti kontrolovat (Heath, 2021).

Dle Anantatmule a Kloppenborg (2021) patří mezi hlavní ceremonie: denní Standup (Daily Scrum), plánování Sprintu (Sprint Planning), Demo, Retrospektiva a tzv. Product Backlog Refinement. Heath (2021) doplňuje ceremonie ještě o Sprint Review.

Artefakty Scrumu

Mezi artefakty Scrumu Doležal (2022, str. 115–120) se řadí Product Backlog a jeho časové odhady, Sprint Backlog a Increment (přirůstek). Jak autor uvádí, každý artefakt má je vázán závazkem, například cílem produktu, Sprintu a tzv. Definiton of Done.

Březina (2020, str. 15) charakterizuje **Product Backlog** jako zásobník. Pod pojmem si lze představit neustále rostoucí seznam požadavků a úkolů spojených s produktem pro jeho zlepšování. Seznam může být tvořen vývojovým týmem, vedením nebo dalšími odděleními ve firmě. Doležal (2022, str. 115) popisuje, že úkoly, které se zde nachází, jsou tzv. Ready for Sprint a jsou již rozdělené na menší, realizovatelné a dobře popsané úkoly. Na přípravě a dělení úkolů slouží schůzka Product Backlog Refinement, která je popsána níže. Dle doporučení

autora úkoly měly mít takovou velikost, aby jejich délka splnění trvala maximálně polovinu Sprintu. Product Backlog se skládá z několika úrovní, priorit: nejvyšší, střední a nejnižší. V nejvyšší úrovni se nachází požadavky do aktuálního Sprintu. Střední úroveň je po částečném Product Backlog Refinementu a nejnižší tvoří nepřipravené úkoly.

Šochová a Kunce (2019, str. 73) doplňují, že Product Backlog má v rukou Product Owner, který za něj nese zodpovědnost. Autoři přidávají mezi vlastnosti požadavků také hodnotu pro zákazníka, tzn. Backlog by neměl obsahovat úkoly technického charakteru. Součástí obsahu požadavků jsou také priority a časové odhady.

Za časové odhady jsou dle Doležala (2022, str. 116) odpovědní vývojáři, ale může se stát, že nebudou zcela chápat zadání požadavků. Od toho je mají Product Ownera, který jim to vysvětlí a pomůže jim. Autor časové odhady na dny hodnotí jako ne zcela funkční, protože se pracovníci pak příliš upnou na číslo a i tak v rámci odhadnutého času není vždy požadavek ukončen (může pokračovat). Dle autora lépe funguje odhadování velikosti dle metod větší/menší, Planning Poker nebo T-Shirt. Kromě obecnějšího odhadování také lépe funguje skupinové odhadování z důvodu přesnosti. V praxi to znamená, že by odhad měli činit všichni vývojáři nebo testéři, ne pouze jeden člověk. Vývojář nemusí vzít v úvahu čas na testování a obráceně.

Druhým artefaktem ve Scrumu je **Sprint Backlog**, který je součástí Product Backlogu s ohraničenými požadavky pro daný Sprint. Doležal (2022, str. 118) uvádí, že představuje obraz odrážející práci v reálném čase. Na rozdíl od Product Backlogu, Sprint Backlog mají v rukou vývojáři, který během Sprintu aktualizují. Heath (2021) doplňuje, že Product Owner může doporučit požadavky k zařazení, ale rozhodnutí je na straně vývojářů z důvodu, že udávají časový odhad požadavků a mají přehled o svých kapacitách. Aby byl kompletní, tak kromě požadavků a úkolů obsahuje i cíl Sprintu. Šochová a Kunce (2019, str. 63) popisuje, že se cíl během Sprintu nemá měnit, protože je směr, kterým se má Sprint ubírat. Dle Heath (2021) cíl Sprintu udává „proč“ – proč proběhne daný Sprint.

Třetí artefakt **Increment** (přírůstek) udává hodnotu, kterou tým na konci Sprintu vypracoval. Doležal (2022, str. 119) specifikuje, aby bylo možné dodat hodnotu, musí být přírůstek použitelný, zákazník ho může otestovat a podat zpětnou vazbu. Autor dále uvádí, že během jednoho Sprintu lze dodat více inkrementů, které jsou předvedeny na Demu. Heath (2021) naproti tomu tvrdí, že ve Sprintech nemusí být vždy vytvořen přírůstek. V rámci přírůstků je velmi důležitá tzv. Definition of Done. V překladu jde o konkrétní určení, jak má přírůstek vypadat, aby byl považován za hotový. Autor doporučuje definice typu: otestováno, schváleno Product Ownerem, zapsáno do dokumentace nebo shoda s UX designem. Heath (2021) dodává, že vytvořené přírůstky musí dohromady spolupracovat a tak dohromady vytvoří celý software.

Scrumové ceremonie

Scrumové ceremonie umožňují týmům efektivně spolupracovat, dosahovat cílů a reagovat v požadované rychlosti na změny. Tyto schůzky mají za cíl podporovat hlavní a vedlejší principy Agilního manifestu, například komunikaci, transparentnost nebo spolupráci napříč týmy.

Stern (2017) radí, že při používání Scrumu by měl každý Sprint začít jeho plánováním a skončit hodnocením.

Scrumový inkrementální a iterativní vývoj funguje na principu dodání části funkčního softwaru na konci Sprintů (Heath, 2021). Vývoj začíná Sprintem 1, ve kterém tým vytvoří první část softwaru. Ke konci Sprintu 1 se první část předvede zbytku týmu a sesbírá se zpětná vazba. Zároveň celý tým zhodnotí uplynulý Sprint 1 a řekne si navzájem, co se povedlo, nepovedlo a co se dá do příště zlepšit. Poté začíná Sprint 2, ve kterém se tým řídí poznatky z předešlého Sprintu a vyrábí se další dodávka softwaru. Zbytek Sprintu 2 probíhá stejně jako sprint 1. Takto

iterace pokračují, dokud není hotový celý produkt/projekt a plně funkční. Během všech Sprintů se nemusí pracovat pouze na nových částech softwaru, ale opravují se chyby a zlepšují se základní funkce (Heath, 2021).

Sprint Planning neboli plánování Sprintu se koná na začátku každého Sprintu, kdy se plánuje, co se za danou časovou dobu musí stihnout (MacNeil, Cader, 2020). Zároveň se schůzkou ukončuje předchozí Sprint a pokud se některé úkoly nestihly v proběhlém Sprintu a mají prioritu, mohou být přesunuty do nadcházejícího běhu. Anantatmule a Kloppenborg (2021) uvádí, že při plánování množství práce a případné prioritizaci se pracovníci (vývojový tým, Product Owner, Scrum Master) spoléhají jeden na druhého. U plánování Sprintu Březina (2020, str. 63) radí, aby se počítalo s rezervou na možné vyskytnutí kritických problémů na produkci. Autor také uvádí, že plánování měsíčního Sprintu může trvat až 8 hodin. V případě, že takové plánování tým stihne velmi rychle, nemusí to nutně ukazovat na efektivnost – dle autorovy zkušenosti se stává, že se ve skutečnosti plánuje na jiných schůzkách.

Každý Sprint se plní požadavky nebo úkoly. Při fixní délce Sprintů se může stát, že daný požadavek bude moc velký na to, aby se stihl během jednoho Sprintu. Proto Březina (2020, str. 62) radí, aby se v těchto případech správně určila definice „done“. To znamená, že je tak určená i definice nehotového požadavku. Za příklad nehotového požadavku bere Březina (2020, str. 62) nestihnutí testů.

Dle Březiny (2020, str. 129) je význam Sprintu se zavázat k dokončení určených úkolů ve stanoveném termínu.

Denní Standup je většinou 15minutová schůzka, kdy každý člen týmu informuje, na čem včera pracoval, co má v plánu dělat dnes a jaké jsou případné překážky (Anantatmula, Kloppenborg, 2021). Jak název napovídá, Standup se koná každý pracovní den ráno a podporuje efektivitu a produktivitu dne. Standup neboli Daily Scrum plní sdílecí funkci. Moderuje ho Scrum Master a doporučuje se, aby se odehrával mimo pracovní místa a aby během schůzky členové stáli (Šochová, Kunce, 2019, str. 107). Podle Šochové a Kunce (2019, str. 108) je ideální časová délka Standupu 15 minut. Autorka má zkušenost z praxe, ve které 7 lidí zvládlo Standup za 5 minut.

Během denního Scrumu určený člen týmu pokládá 3 otázky: Co jsi dělal/a včera, co budeš dělat dnes a existují nějaké překážky, které ti brání v práci (Stern, 2017). Šochová a Kunce (2019, str. 107) doplňuje otázky o dosažení cílů Sprintů.

Dle Březiny (2020, str. 127) není Standup o kontrole aktuálního stavu, je o odstraňování překážek, které mohly nastat a je zapotřebí je vyřešit v týmu například pomocí sdílení zkušeností.

I takto jednoduchý formát schůzky má svá úskalí. Standup by měl být pro všechny členy zajímavý, tzn. neměli by se nudit (Šochová, Kunce, 2019, str. 108). Autorka doporučuje, aby pracovníci hovořili o tom, co doopravdy dokončili a přispělo to to k cílům Sprintu. Místo toho, aby mluvili o tom, na čem včera pracovali. Podle autorky je psychologicky dané, když jeden člen týmu slíbí, že něco dnes zpracuje, tak nechce zítra zbytek týmu zklamat.

Product Backlog Refinement probíhá během Sprintu a má za cíl dospecifikovat úkoly, které jsou v Backlogu uloženy. Na této schůzce má znovu hlavní slovo Product Owner a jeho rolí je upřesnit úkoly, které nejsou dostatečně jasné (Anantatmula, Kloppenborg, 2021). Zároveň je zde kooperace s týmem, který dokáže odhadnout, zda je schopen daný úkol splnit či ne a zda se v případě nesplnění může rozdělit na menší části.

Backlog Refinement není pro změnu schůzka, ale činnost na které se podílí celý tým. Podle Šochové a Kunce (2019, str. 119) se jedná o komplexní týmové pochopení Backlogu, který obvykle trvá do 30 minut.

Demo, Sprint Review, se koná na konci Sprintu a jeho cílem je předvést celému týmu nové funkce a řešení, na kterých se pracovalo (Anantatmula, Kloppenborg, 2021). Autoři dále popisují, že rozhodovací slovo má Product Owner, který určuje, zda jsou nově vytvořené funkce dokončené nebo zda se vrátí zpět do Backlogu a bude se na nich znovu pracovat.

Retrospektiva je typ schůzky, na které se sdílí zpětná vazba a probíhá na konci Sprintu po Demu, aby si členové shrnuli, co během ohraničené doby fungovalo, nefungovalo a co je potřeba zlepšit. Dle Březiny (2020, str. 138) se jedná také o schůzku na odhalování překážek a najítí jejich řešení. Výsledkem by pak měl být konkrétní následující krok, co udělat pro vyřešení a aby se překážka znovu neobjevila.

Jak uvádí Dalton (2018), tento druh schůzky zastupuje princip z Agilního manifestu, který zastává stanovisko pravidelných intervalů, ve kterých tým přemýšlí jak pracovat efektivněji. Aby se z retrospektivy vyžilo co nejvíce a mohla efektivně fungovat, měla by mít pravidelný výskyt a organizovaná, například Scrum Masterem. Schůzka by se neměla probíhat po dlouhé době, proto se doporučuje po každé iteraci, Sprintu (Šochová, Kunce, 2019, str. 112). Šochová a Kunce (2019, str. 111) doporučuje retrospektivou, pokud firma neví kde začít s agilitou.

Pro zavedení retrospektivních schůzek Dalton (2018) doporučuje: dohodnout se na frekvenci schůzek, probírat problémy, zaznamenávat případná řešení a nápady na zlepšení, shrnout opatření a zapracovat je do Backlogu.

Šochová a Kunce (2019, str. 111) doporučuje retrospektivou, pokud firma neví kde začít s agilitou.

Retrospektiva by měla mít následující průběh (Šochová, Kunce, 2019, str. 111):

- představení a potvrzení agendy;
- sběr informací o aktuálním stavu – co je dobře a špatně;
- pochopení informací a identifikace problémů;
- hledání řešení, dohoda na konkrétních dalších krocích ke zlepšení situace a jejich shrnutí.

Z retrospektivy může vzejít, že tým postrádá danou specializaci nebo znalosti (Březina, 2020, str. 139).

Po každém ukončeném úseku se celý tým má sejít s vedením, klienty a prezentovat stav projektu/produktu a předvést dosavadní výsledek (Caldwell, 2019). Během takové schůzky by měl být prostor k diskusi, vzájemné výměně nápadů a poskytnutí pozitivní či negativní zpětné vazby.

Jedna strana Sprintu je splnění požadavků, za druhou stranu je možné považovat nasazení hotových požadavků na produkci. Dle Březiny (2020, str. 135) firmy, které používají Scrum, ukončují Sprints nasazením hotových úkolů na produkční prostředí. Autor popisuje, že s nasazením jsou spojené další aktivity typu integračních, zátěžových nebo regresních testů. Vzhledem k blížícímu se konci Sprintu a nasazení autor doporučuje pár dnů před koncem kontrolu, zda jsou kódy v pořádku, je vše otestované, dohodnout se na harmonogramu nasazení nebo si připravit poznámky k nové verzi.

Shrnutě Sprint Planning řeší, co je potřeba udělat, co se musí udělat a jaká je definice hotového úkolu. Denní Standup kontroluje plnění cíle Sprintu. Sprint Review monitoruje, co se opravdu udělalo a zda nastaly nějaké změny. A retrospektiva sleduje spolupráci týmu, co ne/funguje a zda se dodržela definice hotového úkolu.

Zavedení Scrumu

Březina (2020, str. 182) sestavil seznam předpokladů pro zavedení Scrumu, které by firmy měly splnit:

- Firma a Product Owner mají jasnou vizi produktu.
- Aktivní klient, který je zapojen do vývojového procesu.
- Zadání jsou ve formě, kterou je možné rozdělit na menší úkoly nesoucí hodnotu.
- Product Owner umí rozčlenit zadání na menší úkoly.
- Menší úkoly mají přesná zadání.
- V týmu jsou všechny potřebné znalosti a dovednosti ke splnění požadavků.
- Pracovníci věří v agilní transformaci a přistupují k ní pozitivně.
- Jen Product Owner stanovuje, na co se bude tým soustředit.
- Všechna zadání se zadávají stávajícím týmům (ne novým).
- Časovou náročnost úkolů odhaduje celý tým a odhady jsou respektovány firmou.
- Tým má čas a prostor experimentovat.
- Ve firmě je motivovaný, stabilní, zkušený tým, který umí být samoorganizující.
- Existuje pouze jeden informativní kanál, ze kterého tým čerpá zadání. Nejsou žádná zadání pouze po emailu.

2.3.2 Kanban

Kanban se používá ve firmách, ve kterých se XP nebo Scrum nehodí, což jsou nejčastěji údržbové nebo operativní činnosti (Šochová, Kunce, 2019, str. 151).

Podle Šochové a Kunce (2019, str. 151) má metoda Kanban jednu velkou překážku, a to že nic nenařizuje. To znamená, že podnik si všechno nastavuje podle sebe až na 3 základní Kanbanové principy: "omezit rozpracovanou práci, vizualizovat progres, minimalizovat čas průchodu". Březina (2020, str. 56) dodává princip neposouvání vadných nebo nedodělaných úkolů do dalších etap/stavů.

Březina (2020, str. 55) představuje Kanban jako vizualizaci procesů, minimalizaci rozpracovaných úkolů a používá se při zlepšování aktuálních procesů malými změnami. Autor uvádí, že je metoda Kanbanu založena na proaktivním přístupu pracovníků, kteří sami od sebe berou úkoly odshora seznamu podle svých časových kapacit.

Charakteristické pro Kanban je tabule, která je rozčleněná do sloupců, které popisují stavy zadaných úkolů například „to do“, „in progress“ nebo „done“ (Březina, 2020, str. 56).

V případě vyskytnutí kritických problémů se ostatní úkoly pozastaví a dokud se nevyřeší, nepokračuje ve standardní práci (Březina, 2020, str. 56).

Nevýhodami této metody jsou nemožnost včas odhalit překážky a závislosti, absence ceremonií, týmových rolí a plánování. Proto je Kanban u vývoje vhodný v případech, kdy je potřeba splnit požadavky zákazníků (Březina, 2020, str. 60).

Pokud se firma rozhoduje mezi Scrumem a Kanbanem, měla by uvážit, jaký je firemní cíl a jak ho chce dosáhnout.

Stejně jako Scrum i Kanban nastavuje určité nastavení mysli, a to štíhlé – lean (Stellman, Greene, 2014).

Hooda et al. (2023) popisuje, že Kanban nemá pro své využití určenou velikost týmů. Autoři dále shrnutě uvádí, že obsahuje fáze vizualizace, snižování rozpracované práce, správa a měření toku a zakončuje se opakovaným zlepšováním založené na datech.

2.3.3 SAFe

Knaster a Leffingwell (2019, str. 32) definují SAFe (Scaled Agile Framework) jako metodu, která má za cíl dodávat v nejkratší lhůtě a v nejvyšší možné kvalitě a hodnotě. Dle autorů se jedná o kombinaci agilních a lean principů. Agilní principy jsou převzaty z Agilního manifestu a lean principy z tzv. House of Lean, kterými jsou inovace, zlepšování, flow a respekt ke kultuře a lidem. Je to přístup pro řešení, která mají na starosti 50 až 125 pracovníků, u složitějších řešení i tisíce osob. Kane (2023) popisuje, že tato agilní metoda dává možnost firmám realizovat agilitu na podnikové úrovni. V praxi to znamená, že se SAFe používá ve velkých firmách ke škálování agilitu na různé podnikové úrovne – malé týmy, rozsáhlá portfolia. Dle Kane (2023) lze aplikovat na různé velikosti firem.

Principy SAFe se skládají například z variability, vnitřní motivace zaměstnanců, ekonomického pohledu, inkrementálních cyklů, rozdělení rozhodování, systémového myšlení a milníků k hodnocení (Knaster, Leffingwell, 2019, str. 55).

SAFe má 3 úrovně škálování:

- Team Level – soustředí se na práci jednotlivých agilních týmů, které pracují na splnění úkolů z Product Backlogu a dodávají inkrementální hodnotu produktu na konci každého Sprintu. Každý tým má svou vlastní iteraci, která zahrnuje plánování, vývoj, testování a předvedení výsledků. Využívají se Scrum ceremonie.
- Portfolio Level – portfoliová úroveň se zabývá strategickými rozhodnutími a plánováním na úrovni organizace jako celku. Jsou definovány strategické cíle a plány organizace, které se promítají do pracovních plánů na nižších úrovních. Zároveň se stará o alokaci zdrojů, rozhodování o prioritách mezi různými projekty a sledování celkového pokroku organizace směrem k strategickým cílům. Klíčovým artefaktem na této úrovni je tzv. Portfolio Backlog, který obsahuje všechny strategické iniciativy organizace a zajišťuje jejich transparentní sledování a řízení.
- Program Level – na této úrovni je koordinováno více pracovních týmů. Je odpovědná za integraci práce týmů a zajištění dodání komplexních funkcí a služeb. Pracuje se s většími plánovacími jednotkami, tzv. programovými inkrementy, které kombinují práci více týmů do jednoho společného dodání. Klíčovým prvkem na této úrovni je tzv. Program Increment (PI) Planning, který koordinuje práci týmů na jednotlivých inkrementech a zajišťuje, že všechny komponenty jsou dobře integrovány.

2.3.4 Další metodiky agilního projektového managementu

Kromě Scrumu, Kanbanu a SAFe existují i další metody, které jsou jejich alternativami. Všechny agilní metody mají ale stejný základ – stejné hodnoty a principy. Podle Moran (2015, str.14) každá agilní metodika jde do různých stupňů detailů.

Rubio (2022) mezi další agilní metodiky řadí:

- Adaptivní vývoj softwaru (ASD – adaptive software development);
- Agile modeling;
- Agile unified process (AUP);
- Disciplined agile delivery
- Dynamic systems development method (DSDM);
- Extreme Programming (XP);
- Feature-driven development (FDD);
- Lean Software development;
- Rapid application development (RAD);
- Scrumban.

Vzhledem k velkému množství jsou popsány pouze některé.

Extreme Programming (XP)

Moran (2015, str.16) pokládá Extreme Programming za vysoce disciplinovanou metodu požadující dodržování pravidel a velký závazek k testování softwaru.

Základní zásady XP jsou komunikace, jednoduchost, respekt, odvaha a zpětná vazba. Mezi hlavní týmové role Moran (2015, str.17) řadí analytika, testera, manažera, programátora, coache a zákazníka. XP používá typické artefakty jako user stories, spikes, akceptační testy nebo plán release/vydání. Dle autora začíná typický cyklus XP definováním user stories a jejich prioritizováním, následně probíhá modelování nebo prototypování, akceptační testování v rámci kterého se zapojuje zákazník/klient a končí ukázkou fungující části softwaru. Jako při klasické agilní metodě se popsaný cyklus provádí opakovaně až do úplného dokončení projektu.

Šochová a Kunce (2019, str.147) charakterizuje XP jako metodu, která se drží hesla, že když něco funguje, tak není důvod to dělat jinak. XP jako ostatní agilní metodiky dává vývojářům prostor k pozdějším úpravám, protože je to na jejich zodpovědnost (Šochová, Kunce, 2019, str. 147). 1-3 týdny jedna iterace

Jak uvádí Šochová a Kunce (2019, str. 147) v metodě Extreme Programming se nejvíc používá Pair Programming a Test Driven Development.

Stellman a Greene (2014) uvádí, že v rámci metody Extreme Programming je mnoho postupů, které se zaměřují na nejběžnější kritické situace a jak je vyřešit. Mezi dva hlavní postupy XP patří tzv. Pair Programming (párové programování) a Test-First Programming (programování na zkoušku). Oba přístupy se zakládají vysoké kvalitě. Například párové programování funguje stylem, kdy u jednoho počítače sedí dva vývojáři a jeden z nich píše kód. Smysl je dvojitá kontrola a díky tomu menší výskyt chyb (bugů).

Scrumban

Scrumban je hybridní kombinace Scrumu a Kanbanu, ve které se spojují dodávky požadavků a vizualizace postupů. Stellman (2019) představuje Scrumban jako metodu, díky které se vyvíjí, dodávají a udržují komplexní produkty. V rámci této metody je tým složen ze stejných osob jako Scrum: Product Owner, Scrum Master a další potřební členové. Tým také pracuje ve Sprintech a pořádá scrumové pravidelné ceremonie. Autor dále uvádí, že u jakékoliv kombinace agilní metody se Scrumem je nutná kontrola procesů na základě aktuálních dat a poznatků z praxe.

Jednou z klíčových vlastností Scrumbanu je možnost pružně reagovat na změny a prioritizovat práci na základě aktuálních potřeb a okolností. To znamená, že týmy mohou být více orientovány na zákazníka a lépe se přizpůsobovat měnícím se požadavkům a prioritám. Scrumban také klade důraz na kontinuální zlepšování procesů a maximalizaci toku práce, což je umožněno pomocí průběžného monitorování výkonnosti týmu a identifikace příležitostí k optimalizaci.

Další klíčovou vlastností Scrumbanu je limitace pracovního procesu (Work in Progress Limit – WIP Limit), která pomáhá zabránit přetížení týmu a udržovat stabilní tok práce. Týmy nastavují limity na počet úkolů, které mohou být současně ve fázi práce, což vede k lepšímu řízení pracovního procesu a zvyšuje pravděpodobnost dokončení práce včas.

2.4 Metodika práce

Poslední kapitola teoreticko-metodologické části diplomové práce se věnuje metodice výzkumů. Představí cíl diplomové práce, výběr respondentů, výzkumný problém, výzkumné otázky, hypotézy, od kterých se odvíjí otázky ve kvalitativním výzkumu a metody analýzy dat.

Hlavním cílem diplomové práce je sestavit implementační plán agilní transformace a navrhnout doporučení, jak může zavedení agilního projektového managementu vyřešit výzkumný problém. Vedlejším cílem je vyhodnotit současný stav projektového managementu ve firmě XY.

Pro sběr potřebných dat byly využity SWOT analýzy pro zmapování současného stavu firmy a projektového managementu. Následně byly provedeny kvalitativní polostrukturované rozhovory, aby autorka získala hloubku výzkumného problému, individuální pohledy zaměstnanců a vedení napříč organizací na problematiku (Hendl, 2016, s. 59). Získaná data byla použita k zodpovězení výzkumného problému, výzkumných otázek, potvrzení či vyvrácení hypotéz.

Výzkumný problém: Ve firmě XY byly v průběhu IT projektů zjištěny opakované špatné časové odhady, které mají za následek prodlužování Sprintů, finanční dopad na firmu a na možné zpoždění dodání zákazníkovi.

Výzkumná otázka 1 (VO1): Jak aktuálně funguje projektový management?

Výzkumná otázka 2 (VO2): Jak aktuálně probíhá plánování Sprintů?

Výzkumná otázka 3 (VO3): Jakým způsobem probíhají časové odhady IT projektů?

Výzkumná otázka 4 (VO4): Z jakých důvodů jsou špatně časově odhadovány požadavky v IT projektech?

Výzkumná otázka 5 (VO5): Proč dochází k prodlužování Sprintů?

Výzkumná otázka 6 (VO6): Jaká opatření přijmout pro přesnější časové odhady?

Na základě výzkumných otázek byly zformulovány následující hypotézy:

Obecná hypotéza 1: Časové odhady IT projektů dělají nesprávní pracovníci.

H0: Časové odhady dělají správní pracovníci.

H1: Časové odhady dělají nesprávní pracovníci.

Obecná hypotéza 2: Při časových odhadech IT projektů dochází k nedostatečné komunikaci mezi klíčovými pracovníky.

H0: Při časových odhadech IT projektů dochází k dostatečné komunikaci mezi klíčovými pracovníky.

H1: Při časových odhadech IT projektů dochází k nedostatečné komunikaci mezi klíčovými pracovníky.

Obecná hypotéza 3: Ve firmě XY se nekonají agilní ceremonie.

H0: Ve firmě XY se konají agilní ceremonie.

H1: Ve firmě XY se nekonají agilní ceremonie.

Pro výzkum diplomové práce byla oslovena firma XY z důvodů používání agilního projektového managementu a dostupnosti firemních materiálů a dat.

Vzhledem k tomu, že se firma nachází v IT sektoru, bylo nutné ji anonymizovat, informace o ní a poskytnutá data. Při zpracování diplomové práce autorka nejprve oslovila majitele firmy,

se kterým proběhla konzultace o náplni diplomové práce a poté se COO za účelem získání potřebných informací a dat. Obě strany si nastavily vzájemnou a průběžnou komunikaci. Po prvních konzultacích byl majiteli a COO představen cíl diplomové práce, navržená metodika a postup, jakým dojít ke splnění cíle. Následně se na začátku března uskutečnila schůzka s majitelem za účelem sestavení firemní SWOT analýzy. Po SWOT analýze byly domluveny termíny polostrukturovaných rozhovorů s předem probranou cílovou skupinou. Rozhovory proběhly s 5 respondenty, na základě kterých bylo možné získat podrobné informace o agilním projektovém managementu, sestavit jeho SWOT analýza, sesbírat individuální pohledy na problematiku, nalézt další překážky a navrhnout vhodná doporučení ke zlepšení agilního projektového managementu.

2.4.1 SWOT analýzy

Zkratka SWOT analýza pochází ze složení slov strengths, weaknesses, opportunities a threats. Dle Tylla (2014, str. 39) je základem SWOT analýzy určení silných stránek, slabých stránek z vnitřního prostředí firmy a příležitostí, hrozeb z okolí. Veber a kol. (2021, str. 243) popisují, že **silné a slabé stránky** se definují na základě vnitřních zdrojů firmy, které se týkají například vývoje, úrovně managementu, organizační struktury, značky firmy nebo její finanční pozice. Obecně lze do silných stránek zařadit jevy, které přinášejí výhody a do slabých stránek se přiřadí jevy, které ve firmě nefungují. Dále **příležitosti a hrozby** mohou být brány z vnějšího i vnitřního okolí firmy. Mezi ně patří třeba státní legislativa, růst odvětví, oborové know-how či velká konkurence. Obecně mohou příležitosti zahrnovat jevy, díky kterým by mohla firma uspět a hrozby jsou opakem příležitostí.

V diplomové práci byly použity SWOT analýzy z důvodu zhodnocení současného stavu projektového managementu ve firmě XY a následně sloužily jako podklad pro kvalitativní výzkum. Díky SWOT analýzám byl získán unikátní pohled majitele i přes jeho časové vytížení. Nejprve se autorka s majitelem zaměřila na identifikování všech čtyř částí, poté je majitel ohodnotil a případně se upravila formulace.

Hodnocení majitele se skládalo z čísel od 1 do 5, kde 1 znamenala málo důležité pro majitele a 5 nejvíce důležité pro majitele. U silných stránek a příležitostí znamenalo, čím vyšší hodnocení, tím lépe. A u slabých stránek a hrozeb znamenalo, čím vyšší hodnocení, tím hůře a měla by se těmto složkám věnovat velká pozornost.

Aby obě SWOT analýzy byly kompletní a obsahovaly více úhlů zaměstnanců, byly doplněny o výsledky z kvalitativního výzkumu.

Limity SWOT analýz

Pohled jedné osoby z firmy XY je zároveň i limitem. Z časových důvodů nebylo možné uspořádat celofiremní a celovývojovou schůzku pro zmapování aktuální situace ve firmě.

2.4.2 Kvalitativní výzkum

Kvalitativní výzkum byl veden formou polostrukturovaných rozhovorů s 5 zaměstnanci firmy XY. Do výzkumného vzorku byli vybráni majitel, COO, vedoucí vývoje, vývojář a tester z důvodu různorodosti pohledu na firmu, procesy a problematiku. Konkrétní vývojář a tester byli vybráni firmou. Vzhledem k anonymitě jsou místo jmen používány jejich pozice. Profily respondentů jsou popsány v kapitole 3.1.

Rozhovory probíhaly v následujícím pořadí:

- COO;
- tester;

- vývojář;
- vedoucí vývoje;
- majitel.

Pořadí bylo určeno na základě časových možností obou stran – respondenta a autorky. Rozhovory probíhaly v dubnu 2024 online přes Microsoft Teams s každým respondentem zvlášť. Důvod pro využití online nástroje byl, že vývojový tým pracuje mimo sídlo firmy. Se COO, testerem trvaly rozhovory 1,5 hodiny a s majitelem, vedoucím vývoje, vývojářem trvaly 1 hodinu. Respondenti byli informováni, za jakým účelem se rozhovory vedou, že se rozhovory nenahrávají, autorka si dělá poznámky z rozhovorů a vše je zcela anonymní. Respondenti byli také vyzváni k upřímnosti a otevřenosti.

K rozhovorům byl vytvořen scénář, který měl za cíl zmapovat stav agilního projektového managementu, sesbírat různé pohledy na problematiku, zodpovědět výzkumné otázky, potvrdit či vyvrátit hypotézy. Scenáře pro všechny respondenty měly téměř stejnou podobu, lišily se v závislosti na dané pracovní pozici. Okruhy a otázky byly zvoleny na základě firemní SWOT analýzy, zdrojů v teoreticko-metodologické části a prvků, které spadají do agilního projektového managementu. Poznámky z rozhovorů jsou vloženy do Příloh 2–6.

Kvalitativní výzkum byl vyhodnocen pomocí otevřeného kódování (Hendl, 2016, str. 211) a na základě dat byla sestaven implementační plán a navrhuta doporučení. Jak uvádí autor poznámky byly rozebrány, definovány jednotky, které byly pojmenovány a na základě nich vytvořeny kódy. Kódy posloužily k analýze dat a následně k vyhodnocení hypotéz, výzkumných otázek a výzkumného problému.

Limity kvalitativního výzkumu

Vzhledem k menšímu počtu zaměstnanců nebylo možné nasbírat více pohledů například vývojářů nebo testerů.

Druhým limitem byla citlivost informací, které se přímo týkají firmy XY. Jakmile respondent zmínil jméno firmy nebo tajné informace ohledně produktu, byly tyto informace vynechány nebo přeformulovány. Na vyhodnocení kvalitativního výzkumu to však nemělo vliv.

3 Praktická část práce

Praktická část diplomové práce je členěna na čtyři části. V první části je představena firma XY, její organizační struktura a její současný stav znázorněn pomocí SWOT analýzy. Druhá část obsahuje SWOT analýzu současného projektového managementu, vyhodnocení kvalitativních rozhovorů, potvrzení či vyvrácení hypotéz, zodpovězení výzkumných otázek a výzkumného problému. Třetí část je zaměřena na implementační plán agilních metod ve firmě XY a ve čtvrté části jsou navržena doporučení a opatření, dle kterých by se firma měla řídit.

Veškeré informace, které jsou zmíněné v praktické části, byly poskytnuty se souhlasem vedení na základě konzultací se CEO, COO a z interních norem firmy XY. Před odevzdáním diplomové práce proběhla společná kontrola, aby neobsahovala informace, které nesmí být zveřejněny.

3.1 Představení organizace

Předmětem zkoumání je vybraná firma, která se pro zachování anonymity v celé diplomové práci označuje jako XY. Firma XY je akciová společnost, která se zabývá vývojem IT softwaru ve zdravotnictví. Jako mnoho jiných firem je orientovaná prioritně na zákazníka – spokojeného zákazníka, který je brán jako základ obchodních úspěchů. Proto kromě softwaru nabízí také poprodejní služby typu správa, helpdesk, instalace nebo aktualizace. Pro více prozákaznický přístup poskytuje správu již vyvinutého softwaru u externího dodavatele.

Firma XY má své směrnice a normy, například bezpečnostní, procesní nebo další povinné, které vyžaduje její specializace. Pravidelně je kontroluje na základě svých vnitřních aspektů a podle potřebných změn je upravuje. Mezi vnitřní aspekty patří například organizační struktura, interní informační systémy, hodnoty společnosti, výkonnost firmy, firemní směrnice nebo vztahy se zaměstnanci a dodavateli. Kromě vnitřních složek také pravidelně monitoruje vnější aspekty, které vychází z interní PESTLE analýzy a podle aktuální situace je mění a přizpůsobuje se jim. Kromě vnitřních a vnějších vlivů se firma snaží neustále porozumět potřebám a očekáváním zainteresovaných osob, které řadí do svého integrovaného systémového managementu.

Firma XY má organizační strukturu do 20 zaměstnanců, kterou je možné vnímat jako hybridní – liniová/funkční s kříženými pravomocemi. Skládá se z několika oddělení: administrativa s certifikacemi, vývoj, podpora neboli support a obchod spojený s marketingem. Oddělení vývoje obsahuje jednotky zaměstnanců.

Na vrcholu firmy XY je CEO (majitel). Pod ním se nachází vedoucí softwarového vývoje, COO, vedoucí obchodu a vedoucí právních dodržování předpisů. Mimo výše vyjmenované funkce stojí externí konzultanti. Co se týče křížených pravomocí, tak majitel může nahrazovat roli COO, vedoucího softwarového vývoje a vedoucího obchodu. Naopak majitelovu roli může zastoupit pouze COO.

Majitel svou roli ve firmě popsal jako formální. V průběhu let se jeho role zmenšila a v současné době činí strategická a business rozhodnutí. Jak sám uvedl, udává směr firmy a pod ním se jeho rozhodnutí a směr formují dál, konkrétně do všech oddělení firmy. Kromě důležitých rozhodnutí se stará o firmu po finanční, mzdové, administrativní stránce a prochází si měsíční reporty. Reporty obsahují, co se za uplynulý měsíc splnilo, co se plánuje na nadcházející měsíc a v případě výskytu problémů, které je nutné z jeho úrovně řešit, převezme iniciativu.

COO má na starosti všechny provozní činnosti, kromě strategických rozhodnutí vývoje a finančního řízení.

Vedoucí vývoje vyvíjí backend online aplikace a vede celý vývoj o 3 zaměstnancích. U jeho náplně práce je důležité rozlišit desktop a online aplikaci, základ je stejný, ale má jiné funkcionality. I přesto, že má na starosti backend, je v kontaktu také s frontendem prostřednictvím zodpovídání dotazů od dalších pracovníků.

Vývojář sám vyvíjí frontend softwaru. V jeho podání se jedná o implementaci požadavků, které dostává prostřednictvím programu Jira.

Testerovou úlohou je kontrola kódu. Z jeho pohledu se jedná o postupné testování, tedy nečeká na balík hotových požadavků, testuje je postupně. Co je zásadní, tak dokud tester neřekne, že je požadavek není v pořádku, tak není brán jako hotový. Tester sám rozdělil svou náplň na 2 části: práce na nových požadavcích a přetestování starších požadavků. Na jednu stranu díky jeho práci se udržuje kvalita softwaru a klient je spokojený s produktem. Naproti tomu, ale označil obecně testování za prodělečnou činnost, protože necítí, že by vytvářel businessovou hodnotu.

Vzhledem k zaměření diplomové práce na agilní projektový management, bylo zkoumáno primárně oddělení vývoje. Aby bylo možné hlouběji zkoumat agilní projektový management ve firmě XY, bylo nutné nejdříve pochopit proces vývoje, který je popsán v následující kapitole 3.1.1.

3.1.1 Řízení vývoje produktu firmy XY

Vývoj produktu se ve firmě XY řídí aktualizovanou metodikou ke květnu 2024. Vývoj má několik fází: koncepce, příprava a plánování, realizace, testování, zpětná kontrola a nasazení.

V rámci první fáze **Koncepce** je hlavním cílem převzít od zákazníka požadavek, přeformulovat ho do zadání a ověřit s ním jeho popis. Požadavek do podoby businessového zadání dává dohromady vrcholový management (majitel, COO a vedoucí OaM) spolu s technickou podporou a finančním ředitelem. Obchodní ředitel pak ověřením finálního popisu zadání se zákazníkem.

V této fázi je potřeba vzít v úvahu srozumitelnost a proveditelnost požadavku a soulad s platnou politikou vývoje. Projektový manažer má možnost při nedostačujícím zadání požadavek vrátit, aby splňoval všechny podmínky. Jakmile je požadavek předán k realizaci a zařazen do Backlogu v programu Jira, je k němu přiřazen projektový manažer (vedoucí vývoje). Kromě vrácení požadavku k zákazníkovi, může nastat také úplné zamítnutí.

Záměrem druhé fáze **Příprava a plánování** je zjištění detailních informací o potřebných zdrojích, rizicích projektu a zároveň znovu předložit zákazníkovi nabídku, tentokrát nabídku realizace. Součástí detailních informací je také příprava harmonogramu a rozpočtu. Je možné, že při detailním rozboru budou identifikována nepřiměřená rizika, která mohou ještě změnit rozhodnutí projektového manažera.

V Plánování hraje velkou roli projektový manažer (vedoucí vývoje). Jeho úkolem je ověřit kapacity pracovníků, identifikovat rizika, možná propojení s jinými požadavky, připravit plán komunikace a rámcový plán.

Třetí fáze **Realizace** je výroba produktu, projektu nebo požadavku. Nejprve je nutné, aby zákazník potvrdil nabídku realizace. Poté projektový manažer rezervuje zdroje, dokončí harmonogram, přiřadí úkoly k odpovědným zaměstnancům. V průběhu fáze průběžně dohlíží na stavy úkolů dle časového plánu, vytváří reporty a případně průběžně ověřuje se zákazníkem.

Během Realizace firma XY využívá program Jira, konkrétně tzv. **Jira kanban board**, ve kterém se nacházejí sloupečky s názvy stavů požadavků: To do, Development in progress, Code

Review, Development done, Waiting, čtyři druhy Testing, Stage released, Prod released a Closed.

Ve chvíli, kdy je produkt nebo požadavek hotový, převezme ho tester, který ho otestuje a zkontroluje, zda se shoduje s jeho zadáním. Následně se předá zadavateli na schválení a pokud je v pořádku, tak se nasadí do aplikace a proběhne oficiální akceptace produktu.

Poté proběhne poslední fáze **Nasazení**, ve které proběhne vyúčtování, kompletace veškeré dokumentace a záznamů, vyhodnocení a ukončení projektu.

3.1.2 SWOT analýza firmy XY

Na základě výše uvedených informací o firmě, organizační struktuře a řízení vývoje byla provedena SWOT analýza ve spolupráci s majitelem s ohledem na vnitřní a vnější prostředí. Byly definovány silné, slabé stránky, příležitosti a hrozby. Smyslem SWOT analýzy bylo zanalyzovat současný stav firmy XY a vytvořit komplexní obraz firmy. Po jejím sestavení byla rozdělena na analýzu firmy a agilního projektového managementu. Dalším krokem bylo ohodnocení jednotlivých položek majitelem dle jeho důležitosti.

Po kvalitativním výzkumu byla SWOT analýza firmy doplněna o položky z pohledu dalších respondentů, u kterých nebylo možné provést ohodnocení z důvodu různorodého pocitu důležitosti. Buď byly přímo vysloveny nebo je autorka vybrala z poznámek rozhovorů dle definice SWOT analýzy, která je popsána v kapitole 2.4.1. Proto se jejich tvrzení nachází pod tabulkami silných, slabých stránek, příležitostí a hrozeb.

Z pohledu majitele má firma XY následující silné stránky viz Tabulka 1.

Tabulka 1 SWOT matice – silné stránky firmy XY

Silné stránky firmy XY	Hodnocení
Ve svém oboru je firma XY hodně pokročilá ve specializaci.	5
Firma XY má pouze 8 konkurentů na světě.	5
Firma XY má klienty ve 24 zemích.	5
Ve firmě XY dobře funguje support.	4

Zdroj: vlastní zpracování (2024)

Po kvalitativních rozhovorech je možné silné stránky doplnit o další položky:

- Firma se drží mezi TOP 4 na světě v oboru na světě podle nezávislých průzkumů.
- Skvělý certifikovaný produkt.
- Firma má individuální přístup k zákazníkům.
- Zákaznický servis ve firmě funguje dobře a efektivně.
- Zákazníci velmi potřebují produkt firmy XY k tomu, aby mohli vykonávat svou práci.
- Zaměstnanci si uvědomují důležitost produktu pro klienty.
- Dobrá, profesionální, efektivní komunikace mezi pracovníky.
- Zaměstnanci si navzájem vychází vstříc.
- Když se vyskytne problém, pracovníci jsou společně schopni ho vyřešit.
- Stabilní týmy pracovníků ve firmě.
- Loajalita zaměstnanců.
- Přátelské, profesionální, pozitivní pracovní prostředí.

- Majitel důvěřuje všem vedoucím (COO, vývoj, OaM, support).
- Vedoucí vývoje a vývojář důvěřují vedení firmy.
- Vedoucí vývoje důvěřuje svým podřízeným.
- Externě najatému procesnímu manažerovi se podařilo stabilizovat oddělení obchodu.

Z pohledu majitele má firma XY v oboru dobré postavení a je velmi pokročilá ve své specializaci, což potvrzují i nezávislé průzkumy, které ji dlouhodobě uvádějí v TOP 4 na světě. V průzkumech se hodnotí například produkt, jeho funkce, cena, kvalita nebo podpora. S čímž souvisí tvrzení skvělého certifikovaného produktu, individuálního přístupu k zákazníkům, uvědomění si důležitosti produktu zaměstnanci nebo dobrého, efektivního fungování zákaznického servisu a podpory. Majitel pokládá za silnou stránku i přesah do zahraničí, díky kterému má firma XY klienty přibližně ve 24 zemích.

Do silných stránek je možné zařadit i potřebnost softwaru – jedná se o produkt, který zákazníci, rozšířeně cílová skupina, potřebuje k výkonu své práce.

Za silnou stránku se také považuje přátelské, profesionální a pozitivní pracovní prostředí, ve kterém si zaměstnanci vycházejí vsříc, jsou schopni vyřešit vyskytnuté problémy a vedou profesionální komunikaci. Dle COO jsou výše vyjmenované stránky také dané stabilitou týmů, které spolu dlouho pracují a od toho je odvozená i loajalita a důvěra mezi vedením a zaměstnanci.

Poslední předností je externě najatý procesní manažer, který měl za úkol stabilizovat a zefektivnit fungování oddělení obchodu, což se mu povedlo.

Další částí SWOT analýzy jsou slabé stránky, které jsou vypsány v Tabulce 2.

Tabulka 2 SWOT matice – slabé stránky firmy XY

Slabé stránky firmy XY	Hodnocení
Těžký onboarding kvůli specializaci v oboru.	5
Firma nenabízí moc zaměstnaneckých benefitů.	3

Zdroj: vlastní zpracování (2024)

Dle poznámek z polostrukturovaných rozhovorů lze zařadit do slabých stránek také:

- Firma nemá podporu pro klienty 24/7.
- Jeden z respondentů by uvítal více komunikace o dobrých a špatných jevech ve firmě ze strany vedení.

Největší slabá stránka je těžký onboarding. Vzhledem k velké oborové specializaci je potřeba speciální know-how. To znamená, když firma najde vhodného kandidáta na pracovní pozici, přijme ho, tak dlouho trvá než získá potřebné znalosti k samostatnému plnění práce. Odhadem může onboarding trvat i šest měsíců.

Kvůli malé velikosti firmy si majitel uvědomuje malou nabídku zaměstnaneckých benefitů, které stojí nemalé peníze. Zároveň vnímání benefitů může být spojené i se zaměstnáváním externistů, protože obecně externisti nemají stejné benefity jako interní zaměstnanci. Mezi navrženými výhodami byly navrženy: přidání mzdy, více dovolené, penzijní připojištění, stravenky nebo 13. plat.

Mezi slabé stránky respondent zařadil nefungování podpory 24/7. Vzhledem k zákazníkům z 24 zemí, kteří žijí v různých časových pásmech, zatím není možné fungování podpory 24 hodin 7 dní v týdnu. Může to ale být bráno jako příležitost ke zlepšení služeb spojených s produktem. Další respondent z polostrukturovaných rozhovorů navrhl častější komunikaci ze strany vedení o dobrých a špatných jevech, které se ve firmě XY dějí. Z jeho pohledu se snáze mluví o dobrých jevech a hůře o špatných.

Z vnějšího prostředí mezi příležitosti patří (Tabulka 3):

Tabulka 3 SWOT matice – příležitosti firmy XY

Příležitosti firmy XY	Hodnocení
Uzavření kontraktů s dalšími zákazníky.	4
Uzavření kontraktů v zahraničí.	3
Nová legislativa	5

Zdroj: vlastní zpracování (2024)

Majitel vidí obchodní příležitost v uzavření nových kontraktů s dalšími zákazníky, ať v České republice nebo v zahraničí. A v oboru se chystá nová legislativa, která by pomohla businessu.

Z rozhovorů vyplynula jedna příležitost, a to rozšíření týmu obchodu z důvodu vymáhání faktur.

Poslední částí SWOT analýzy jsou hrozby, které jsou vyjmenovány v Tabulce 4.

Tabulka 4 SWOT matice – hrozby firmy XY

Hrozby firmy XY	Hodnocení
Klienti platí pozdě (faktury po splatnosti – nezávisle na opožděném doručení produktu).	5
Drahá certifikace k výrobě a správě produktu.	5
Někteří noví klienti po několika měsících dále nechtějí užívat produkt.	3

Zdroj: vlastní zpracování (2024)

Majitel nejvíce vnímá faktury po splatnosti, které jsou nezávislé na případném opoždění doručení produktu nebo splnění požadavku. Z jeho pohledu jsou bohužel opožděné platby v oboru podnikání běžné, protože schválení a poslání dané částky trvá delší dobu. S tím souvisí i vnímání ceny certifikace k výrobě a správě produktu. Pokud by klienti neplatili pozdě, tak by vnímaná cena certifikace byla přiměřená.

Menší hrozbou jsou noví klienti, kteří po několika měsících nechtějí dále používat produkt. Klienty lze přesvědčit k dalšímu používání například pomocí slev a podobně.

Kvalitativní výzkum ukázal navíc pouze jednu další hrozbu: rychlé finanční projevení na firmě v případě velkého množství faktur po splatnosti. I přesto, že se jedná o vnější prostředí, které z části nelze ovlivnit z důvodu fungování cílové skupiny, tak se ve firmě XY zařídilo efektivnější vymáhání faktur. Zavedla se například častější komunikace, posílání faktur na více

lidí najednou, zapisují se do CRM poznámky o klientech (zda je potřeba častá komunikace, v jakém termínu se ozvat) nebo komunikace v jejich rodném jazyce. Ze zkušeností firma XY zjistila, že se po nezaplacení faktur/y nemůže chovat ke klientům stylem odstríhnutí programu, protože je pro ně používání programu velmi důležité i nezbytné k práci.

3.2 Charakteristika projektového managementu v organizaci

Tato kapitola se zabývá charakteristikou projektového managementu, který zásadní fungování firmy XY. Jak již bylo zmíněno v předešlých kapitolách, diplomová práce se zaměřila na projektové řízení používané ve vývojovém oddělení. Aby bylo možné popsat komplexní fungování projektového managementu ve firmě XY, byly provedeny kvalitativní rozhovory napříč organizační strukturou, jak bylo představeno v kapitole XX. Proto jsou v podkapitole 3.2.1 nejprve zanalyzovány polostrukturované rozhovory pomocí otevřeného kódování a následně vyhodnoceny.

Ještě před kvalitativním výzkumem v rámci majitelova představení problému s časovými odhady a vytvoření SWOT analýzy s majitelem lze říct, že vývojový tým funguje v tzv. Sprintech, které trvají přibližně 3 měsíce. Zaměstnanci vývoje pracují s programem Jira, kde jsou sesbírány požadavky k realizaci s přiřazenými prioritami a časovými odhady. Ve firmě XY probíhají pravidelné schůzky, na kterých se řeší, co se za uplynutý týden událo a co se následující týden bude řešit.

3.2.1 Analýza získaných dat z kvalitativního výzkumu

V této podkapitole jsou zanalyzovány a popsány výsledky kvalitativního výzkumu – polostrukturovaných rozhovorů. Metoda polostrukturovaných rozhovorů byla navržena na základě šesti výzkumných otázek a z důvodu získání více pohledů na problematiku a hloubkového poznání. V Přílohách 2–6 jsou přiloženy poznámky z rozhovorů s respondenty.

Rozhovory byly rozděleny do tří částí.

První část stanovila, na jaké pozici zaměstnanec pracuje a co je jeho náplní práce. Cílem bylo vytvořit základní profil respondenta a zjistit jeho zapojení do vývoje.

Druhá část se zabývala projektovým managementem a fungováním vývoje – pravidelnými schůzkami, Sprinty, zapojením klientů, pracovním prostředím a plánování Sprintů s napojením na časové odhady.

Třetí část měla za cíl zjistit, co by respondenti změnili ve firmě, procesech a projektovém managementu.

Za použití otevřeného kódování byla identifikována témata, na základě kterých byly stanoveny následující kódy:

- profily respondentů;
- plánování Sprintu;
- průběh Sprintu;
- odhady práce;
- pravidelné firemní schůzky;
- kontakt s klienty;
- firemní kultura;
- navrhované změny.

První část obsahuje kód **Profily respondentů**, který je rozpracován v kapitole 3.1.

Druhá část rozhovorů je zahrnuta v kódech Plánování Sprintu, Průběh Sprintu, Odhady práce, Pravidelné firemní schůzky, Kontakt s klienty a Firemní kultura.

Plánování Sprintu

Pracovník, který plánuje obsah Sprintů, je **vedoucí vývoje**. Má sesbírané požadavky od oddělení supportu, vedoucího supportu, od svých podřízených a vstupy z desktopové aplikace. Desktopová aplikace vstupuje do Sprintů z důvodu přebírání komponent a funkcí do online aplikace. Všechny požadavky jsou sepsané v Backlogu, který je součástí programu Jira, ze kterého se vybírá práce na release. Vedoucí vývoje zařazuje požadavky po domluvě s vedoucím supportu, podle svého pocitu a odhadu. Na zařazování má vliv také vyvážená práce pro frontend a backend a přiřazené priority k požadavkům.

Firma pracuje s jedním Backlogem, ve kterém se nachází všechny požadavky (od nových i stávajících klientů). Dle testera se neodlišují požadavky od nových nebo stávajících klientů, jsou obecně brány jako požadavky.

Vývojář a tester nijak nezasahují do plánování Sprintu. Vývojář je názoru, že se nijak Sprintu neúčastní a tester ví, co bude obsahovat a kolik jeho časové kapacity bude potřeba. Oba se shodují, že o zařazování požadavků by měli rozhodovat vedoucí, oddělení supportu a na konec je úkolem vývoje odhadnout časové možnosti. Vývojář dodává, že pokud během zpracování požadavku zjistí o vydání nové verze kódu, tak vedoucímu řekne, že se vyplatí chvíli počkat, aby později nemusel požadavek předělávat.

Tester doporučuje do plánování Sprintu zahrnout 20% navýšení požadavků, možné nemoci a dovolené, které mohou trvat přibližně 1 týden.

Role majitele v plánování Sprintu je okrajovější, ale má možnost zasahovat dle své potřeby, například v případě businessových požadavků, chyb s velkým dopadem, většího využití funkce nebo požadavků od strategicky důležitých klientů.

Co se týče **priorit** spojených s plánováním tak vedoucí vývoje k nim přihlíží, ale mají menší vliv než další rozhodovací prvky. Důvodem je, že každý, kdo zadá požadavek, tak má jinou míru priority. Tak se podle zadavatele může jednat o velmi důležitý požadavek, ale ve skutečnosti může být jeho důležitost menší. Přiřazení priority pak také řeší vedoucí vývoje s vedoucím supportu, aby si vyjasnili, co spěchá a co ne. Ve vývojovém týmu vývojář pracuje s prioritami pouze tak, že se zeptá vedoucího vývoje, co má vysokou prioritu, protože se stává, že několik požadavků má zadanou stejnou prioritu, nebo v případě, kdy jsou zaneseny nové požadavky v průběhu Sprintu. Tester vnímá prioritu z pohledu hotfixů, které jsou velmi kritické. Domnívá se, že obecně budou mít větší váhu změny s dopadem na velké množství klientů.

O prioritách nerozhoduje tester nebo vývojář – tester je názoru, že by o nich mělo rozhodovat vedení (vedoucí vývoje, vedoucí OaM a další vedoucí).

Odhady práce

Vzhledem k tomu, že **vedoucí vývoje** převážně plánuje obsah Sprintu, tak dělá odhady požadavky za frontend a backend. Odhaduje na základě pocitů, priorit a urgency. Požadavky se snaží navrhnout tak, aby motivoval podřízené. Z rozhovoru lze vyčíst, že problém s časovými odhady souvisí s neúplným zadáním požadavku a jeho postupným doplňováním. Vedoucí vývoje přiznává, že kdyby měl přesnější zadání, byl by schopen přesněji odhadovat. Několikrát tento problém probíral s vedoucím OaM, ale bez řešení. Parafrázováním jeho slov je těžké hledat kompromis mezi splněním termínů a kvalitou, protože vedoucí OaM není tolik

v kontaktu s vývojem. Časové odhady se tedy ve firmě XY diskutují, ale kvůli nedokončeným diskusím a rozhodnutím vzniká pro vedoucího vývoje nejasná situace.

Nejasná situace a nedotažení změny může souviset s vnímáním a důvěrou ve vedoucího OaM, viz kód Firemní kultura. V současné době jsou oba vedoucí v rozporu, protože pro vedoucího vývoje je důležitější kvalita nad časem a pro vedoucí OaM je důležitý čas v důsledku unikání peněz.

Na vnímání odhadů a nestíhání požadavků mají členové vývojové týmu jiný pohled. Vývojář nestíhání plnění požadavků nijak neřeší. Podle něho se doteď vůbec nedělaly časové odhady, nezjišťovalo se, o kolik dnů nebo týdnů se požadavky nestihly a proč se Sprint prodloužil. Ze svých špatných zkušeností doufá, že se dělat nebudou. Vzhledem k tomu, že se svou práci snaží dělat maximálně dobře v co největší kvalitě, tak by chtěl mít i co nejpřesnější odhady. Aby dosáhl nejpřesnějších časových odhadů, tak by se musel hodně zamýšlet a i tak by si nezapomněl na všechny možné scénáře, které by měly dopad na čas.

Tester přiznává, že časové odhadování požadavků na testování je „časová alchymie“, nicméně umí vyhodnotit, kolik mu práce zabere času a snaží se do odhadu vejít. Pokud se jedná o chybu, může mu zabrat přibližně 1 hodinu. Když se jedná o novou funkci, která je většího rozsahu, může mu trvat i 1 týden testování.

Podle majitele je několik důvodů, proč dochází ke špatným časovým odhadům: zaměstnanci nepoznají komplexnost požadavků, bojí se odhadovat nebo jsou „líní“.

Průběh Sprintu

Všichni respondenti se shodli v délce sprintů, a to že **jeden Sprint trvá tři měsíce**. V přepočtu na rok se uskuteční čtyři Sprints. Od majitele, vedoucího vývoje a testera délka Sprintů vyzněla jako plán, protože není pevně nastavená jeho délka a stává se, že se prodlouží. Naproti tomu tester uvedl, že reálně je za rok více Sprintů než čtyři. Délka Sprintu je přímo závislá na počtu požadavků od klientů, u kterých se rozhodne o jejich zařazení. V reálném počtu Sprintů za rok lze konstatovat, že mezi majitelem a testerem dochází k rozporu.

Dle testera by mohly být Sprints kratší, ale stihly by se jen menší požadavky a nebyl by v nich čas na analýzu, vývoj a testing. Domnívá se, že Sprint o délce tří měsíců je rozumný.

Odhadem vedoucího vývoje se **do jednoho Sprintu zařadí sto požadavků**. V průběhu se doplňuje o další požadavky, okolo kterých probíhá nahodilá komunikace s vedoucím supportu (nejedná se o pravidelné schůzky). Společně se domlouvají, o jaký požadavek se jedná a jak je vývojově náročný. Podle toho se zařadí do běžícího nebo dalšího Sprintu.

Ohledně doplňování požadavků tester uvedl příklad: Do Sprintu se zařadí šedesát požadavků. Na jeho konci místo jich šedesáti požadavků je 120. Požadavky, které se přidávají v průběhu, jsou chyby nebo nové požadavky, u nichž je usouzeno, že nejsou tak časově náročné. Dle slov testera se sice upravuje Sprint, ale kostra zůstává stejná.

Po zeptání se vývojáře na průběh Sprintu, odpověděl, že dle nich nepracuje. Přesto že pro něj vyplývají požadavky, na kterých pracuje dle dohody, je mu jedno, zda pracuje na požadavku ze současného nebo následujícího Sprintu. Ví, že když dojde k jeho zvýšení priority, patří do aktuálního Sprintu.

Z pohledu vývojáře probíhá **vývoj požadavku** následovně:

- Vývojář vyvine požadavek a přesune ho v programu Jira do sloupečku „Development done“.
- Dá vědět vedoucímu vývoje, že je za něj hotový frontend a ve chvíli, kdy je hotový i backend, na kterém pracuje vedoucí vývoje, tak se požadavek předá na testera.

- Pokud tester zjistí, že se v požadavku vyskytuje chyba, vrátí jim ho na opravu. Tímto způsobem se proces opakuje, dokud požadavek není v pořádku.

Majitel v průběhu Sprintu kontroluje pokrok požadavků, analýzy z produkčního prostředí a požadavky klientů v programu Jira. Pro majitele vypadá typický průběh tak, že se mu oznámí přibližně 2 dny před koncem Sprintu, že se bude prodlužovat.

Během Sprintu se také rozhoduje o jeho **prodlužování**.

Z rozhovoru s vedoucím vývoje vyplynulo, že on reálně prodlužuje Sprints. Činí tak z důvodu kvality a upřednostnění splnění požadavků nad dodržením termínů. Pokud se vyvine tlak na dodání, tak dá přednost důležitějším požadavkům a další přesune do následujícího Sprintu.

Vývojář na prodlužování Sprintu neparticipuje. Pouze pokud mu požadavek trvá déle než bylo odhadnuto, tak dá vědět vedoucímu vývoje a ten rozhodne, co s požadavkem dál.

Tester ze své pozice chápe prodlužování Sprintu jako přidání dalších požadavků od klientů, které se musely rychle zanést a mělo smysl je přidat do aktuálního Sprintu. Tím pádem se prodlužování týká primárně vývoje. Je toho názoru, že když se začne vývoj protahovat, tak se bere z jeho testovacího času. V praxi to znamená, že do termínu nestihne otestovat vše a prioritizuje otestování požadavků, bez kterých produkt nemůže fungovat. Menší nebo méně důležité požadavky si nechá na otestování po jejich nasazení, které stihne otestovat cca do 2 dnů a domnívá se, že si toho klienti nestihnou všimnout, protože ne vždy hned začnou produkt používat. Méně důležité požadavky k otestování pozná díky tomu, že zná dobře produkt a ví, jaké části aplikace bude klient používat nejvíce.

Z rozhovorů lze usoudit, že tento problém nejvíce vnímá majitel, protože špatné časové odhady vedou k prodlužování Sprintu, pozdějšímu splnění požadavků klientů a tím pádem musí firma platit finanční penále. Ostatní respondenti (vedoucí vývoje, vývojář a tester) problém nevnímají tak kriticky.

Co se týče **vyhodnocování** Sprintů majitel hodnotí konec Sprintu sám. Sleduje reporty pracovníků, reporty pro klienty a hotfixy.

Dle vedoucího vývoje se uplynulý Sprint **detailně nevyhodnocuje**. Víc se řeší překážky pracovníků, které se vynesou a snaží se je vyřešit (děje to i během Sprintu).

Pravidelné firemní schůzky

Majitel, vedoucí vývoje, vývojář a tester chodí na pravidelné **schůzky vývojového týmu**, které se konají každý pátek. Ve firmě se ji hovorově říká „dev call“. Náplní schůzek je řešení aktuálních záležitostí, konkrétně co se událo za týden, co se chystá další týden a zda jsou k řešení možné vyskytnuté problémy. Vedoucí vývoje shrne výše vyjmenované záležitosti, požadavky, které řešil s podřízenými a vedoucím supportu a předloží otázky, které je potřeba projít ve větším plénu. Tester doplňuje, že se také řeší časové plánování, nové funkce, požadavky a změnové požadavky produktu. Majitel se účastní spíše pasivně, ale má připravené body, které případně potřebuje probrat. Vývojář se vyjadřuje pouze ke své práci. Schůzky trvají od 30 minut do 2 hodin a mají předem daný program.

Další schůzka, které se všichni respondenti účastní, je celodenní, která se koná **jednou za čtvrt roku** a je na ni povinná osobní účast. Na této schůzce se probírají podobné záležitosti jen za delší období.

Majitel má své pravidelné schůzky:

- jednou za týden se schází se COO a vedoucím obchodu a marketingu;
- jednou za měsíc, kde prezentuje, v jaké je firma stavu a výsledky za každé oddělení.

Vedoucí vývoje má k výše vyjmenovaným pravidelným schůzkám ještě jednou za rok setkání s majitelem, na které spolu diskutují cíle vývoje za rok, požadavky s důležitým dopadem na produkt. Tato schůzka nemá pevné datum. A jednou za týden si volá s pracovníkem v Holandsku ohledně desktopové aplikace.

COO má také osobní pravidelné schůzky jednou týdně s pracovníky z oddělení sales support, na kterých probírají aktuální úkoly a stavy vymáhání faktur po splatnosti a podobně. Aktuálně si oddělení prošlo změnami, které se usazují.

Obecně má každé oddělení pravidelné schůzky, na kterých se řeší podobné *věci*.

Kontakt s klienty

Všichni respondenti nepřichází do přímého kontaktu s klientem, pouze **formou požadavků**. Tester dostává požadavky prostřednictvím vedoucího vývoje.

Vývojář je názoru, že čím víc zpětné vazby, tím lépe, ale ve formě rozumných požadavků. Pokud je některý požadavek technicky neproveditelný, mělo by se zákazníkovi říct, proč byl jeho požadavek zamítnut.

Co se týče komunikace s klienty v průběhu vývoje / dodání požadavků vedoucí vývoje neví, zda taková komunikace probíhá – pokud ano, jde přes vedoucího supportu

Majitel by klienty rád zapojil, ale vzhledem k cílové skupině je zapojení těžké.

Firemní kultura

Do kódu „Firemní kultura“ bylo zařazeno pracovní prostředí a důvěra v pracovníky a vedení.

COO a tester popsali **pracovní prostředí** jako přátelské, pozitivní a profesionální. Pracovníci mezi sebou slušně komunikují, vychází si vstříc, mohou vyjádřit svůj názor a dokážou společně vyřešit problém. Napříč firmou si pracovníci tykají.

Co se týče důvěry, lze ji rozdělit dle respondentů na **důvěru ve vedení** (majitele, vedoucího obchodu a marketingu) a **důvěru v pracovníky**.

Důvěra v majitele je značná. Vývojář mu velmi důvěřuje a domnívá se, že produktu komplexně rozumí. Dokonce více než sám vývojář.

Důvěra ve vedoucího obchodu a marketingu je nižší než v majitele. Na vývojáře působí tak, že má know-how, ale jeho slovy „když do toho pořádně šlápne, tak by přivedl výsledky“. Vedoucí vývoje je s vedoucím obchodu a marketingu více v kontaktu a je názoru, že je velmi schopný ve svých oborech a vnímá, že by rád něco změnil. Bohužel změny nejsou dotaženy do koncenejší. Neví, co od vedoucího OaM má očekávat.

Naopak majitel plně důvěřuje všem vedoucím – vývoje, supportu, obchodu a COO.

Třetí část polostrukturovaných rozhovorů je obsažena v kódu Navrhované změny.

Navrhované změny

Posledním kódem jsou přímo navrhované změny respondenty nebo narážky. Změny jsou řazeny od nejčastějšího výskytu po individuální návrhy. Shrnutě se navrhnuté úpravy týkají délky Sprintů, zadání požadavků a dokumentace, časových odhadů požadavků, prodlužování

Sprintů a priorit požadavků. Další změny jsou rozděleny podle respondentů z důvodu individuálních návrhů.

U všech respondentů vyvstal stejný podnět/chyba/podnět a to délka Sprintů. Po opomenutí 3 měsíců, nemají Sprints **fixní délku** a prodlužují se.

4 respondenti (majitel, vedoucí vývoje, vývojář, tester) by uvítali úpravu **zadání požadavků a dokumentace**. Zadání a požadavku autorka spojila, protože jsou na sebe navázané. Ve chvíli, kdy by bylo výborně napsané zadání, bylo by psaní dokumentace snazší.

Vedoucímu vývoji by zjednodušilo práci, kdyby získal jasný obraz o tom, co by firma ráda zařadila do produktu, jaké požadavky by měl Sprint obsahovat a pomohl mu někdo s obsahem zadání požadavků. V současné době se zadání tvoří v průběhu práce na požadavcích. Zadání by ideálně mělo obsahovat: představa o vizualizaci požadavku, textace s překladem, fungování požadavku z pohledu uživatele aplikace, jeho umístění, všechny dopady požadavku do aplikace, scénáře včetně okrajových a errorů.

I přesto že vývojář řeší dokumentaci okrajově, narazil na situace, kdy je potřeba vyvolat hlášku v daných případech, tak neví například za jakých podmínek by se měla vyvolat. Aby dohledal informace, musí projít okolo nižších desítek požadavků v programu Jira, aby je dohledal. To znamená, není jedno místo, kde by byly napsané informace tohoto typu. V programu Jira mají napsaný pouze základ, který není dostačující. Vývojář je názoru, že bude nejspíš těžké udržovat dokumentaci aktuální, protože se informace převezme ze zadání požadavku, promítne se do vývoje a následně by bylo nutné ji zapsat do dokumentace.

Podle testera by mohla být dokumentace podrobnější a souhlasí s vedoucím vývoje ohledně vytváření v průběhu práce. Existují požadavky, u kterých se dokumentace doplňuje až u testování, protože tester narazí na chybějící scénáře nebo drobnosti. Zároveň by testerovi pomohla ucelenější dokumentace, což by mu pomohlo lépe si plánovat svou práci. Ve chvíli, kdy se dostane požadavek na testování, rychle si ho přečte a poté vymýšlí scénáře, za kterých musí produkt fungovat a otestuje je. Tester se domnívá, že jeho nepřipravenost může mít finanční dopad. Stejně jako vývojář hledá informace po několika požadavků v programu Jira. Aktuálně se na dokumentaci podílí vedoucí OaM, vývojáři a tester – navzájem se domlouvají, jak se dá požadavek vyřešit. Od vývoje je dokumentace příliš technická a neřeší všechny dopady a scénáře.

K otázce, kdo by měl dokumentace sestavovat a aktualizovat, se tester a vedoucí vývoje shodují na analytikovi.

Druhým nejčastějším problémem je **časové odhadování požadavků** související s prodlužováním Sprintů a neplněním termínů. Časové odhady jsou vnímány jako aktuální problém, o kterém se ve firmě začíná více mluvit.

Majitel by rád docílil toho, aby pracovníci relevantně odhadovali požadavky sami za sebe. Výhledově chce majitel s vedoucím OaM dojednat zodpovědnost za časové odhady.

Vedoucímu vývoji by s odhady pomohlo, kdyby s obsahem požadavků, dal mu přesnější zadání. Lze konstatovat, že se jedná o rovnici: přesnější zadání = přesnější časový odhad. Ohledně přesnějšího zadání se vrací k problému dokumentaci/zadání, který byl popsán výše.

S časovými odhady souvisí chtěná změna od vedoucího OaM, který má za cíl přesného dodržování termínů Sprintů.

Poslední navrhovanou změnou, na které se shodlo více respondentů, je **úprava priorit požadavků**. Vedoucí vývoje a vývojář naráží při práci na různorodost označování priorit požadavků. Vzniká to z toho důvodu, že každý zadavatel požadavku měří důležitost odlišně.

Tím pádem například zmenšení nebo změna fontu písma nemá pro vývojový tým tak velkou prioritu jako například bugy nebo hotfixy. I přes určení priorit požadavků si tak vedoucí vývoje při plánování Sprintu musí konzultovat znovu s vedoucím supportu/podpory, aby určili, co je doopravdy urgentní.

Dále už se jedná o individuální úpravy dle jednotlivých respondentů. Proto jsou níže úpravy seskupeny podle pracovních rolí.

Majitel by rád do budoucna zavedl několik změn:

- S vedoucím vývoje a vedoucím obchodu a marketingu by rád zavedl schůzku jednou za 14 dní, na které by společně probírali **průběh Sprintu**.
- Do **vyhodnocování Sprintu** by rád zapojil i vedoucího OaM.
- Majitel by rád **více zapojil klienta** a komunikaci s nimi, ale je to dle něj komplikované.
- A majitel by chtěl **nahradit sám sebe**, ze 100 %. Rád by docílit stavu, kdy by dostával reporty a udával by strategický směr a činil strategická rozhodnutí. Zároveň mu vadí, že týmy nejsou schopné rozhodnout bez něho.
- Podnět, který je možné brát jako příležitost na zlepšení, je dřívější oznámení majiteli o prodloužení Sprintu než pár dnů před jeho koncem.

Vedoucímu vývoje by se hodilo **více času na vývoj backendu**. Tím, že má ještě na starosti tým vývojový tým, plánování, administrativu, tak má pocit, že by potřeboval více času na získání hloubějšího know-how. Díky většímu know-how by byl schopen navrhnout lepší řešení a nevznikaly by některé chyby. Zároveň se mu vedení týmu a vývoj kříží ve smyslu **přerušování práce na backendu** – ve chvíli, kdy mu někdo zavolá, tak se vedoucímu přerušuje tok myšlenek a složitě se pak vrací k předešlé práci.

Kdyby byl vedoucí vývoje CEO, tak by více určoval **vizi směru vývoje** – čeho by se mělo ve vývoji dosáhnout v delším horizontu. Vize by měla směřovat od vrcholového managementu. Vize by měla vypadat například jako sblížení desktopové a online aplikace.

Vývojář by rád změnil **design aplikace, webové stránky** firmy a obecně více **zviditelnit produkt** (například na LinkedInu). Úpravy, které navrhl, se týkají více produktu a projektového managementu – sbírání zpětné vazby prostřednictvím dotazníků. Vývojář by se zákazníků ptal, co se jim líbí/nelíbí na produktu, zda jsou spokojeni s UI, jaké chyby se jim vyskytují v aplikaci nebo co by chtěli zavést. Následně by dotazníky někdo z firmy procházel a rozhodl, zda by se nápady klientů zavedly nebo zamítli. Důležité pokládá i to, aby se klientům sdělilo, proč se jejich nápad zamítl.

U testera se objevilo mnoho návrhů na změny, které již byly popsány v rámci kódů a SWOT matice. Rád by upravil komunikace ze strany vedení ohledně dobrých a špatných událostí, které se ve firmě XY dějí. Bylo to zařazeno do slabých stránek firmy. Tester by zavedl tento druh komunikace častěji, proaktivně a aby se prezentovaly i reálné špatné události. Také by uvítal, kdyby se špatné záležitosti prezentovaly tak, jak ve skutečnosti jsou. Vedení by mělo být v informování více proaktivní.

3.2.2 SWOT analýza současného projektového managementu firmy XY

SWOT analýza současného projektového managementu byla vytvořena na základě rozhovoru s majitelem, který se konal před kvalitativním výzkumem. Cílem SWOT analýzy bylo zmapovat aktuální stav projektového managementu se zaměřením na slabé stránky a najít první možné souvislosti s výzkumným problémem. Po jejím vytvoření se autorka spojila s majitelem, který ohodnotil od 1 do 5 silné a slabé stránky dle jeho důležitosti.

Po polostrukturovaných rozhovorech byla tato SWOT analýza doplněna o další položky od respondentů, u kterých nebylo možné provést ohodnocení z důvodu různorodého pocitu důležitosti. Stejně jako u SWOT matice firmy buď byly položky přímo řečeny nebo je autorka vybrala z poznámek rozhovorů dle definice SWOT analýzy. Proto jsou jejich tvrzení vysvětlena pod tabulkami silných a slabých stránek.

Dle majitele současné nastavení projektového managementu má následující silné stránky, viz Tabulka (5):

Tabulka 5 SWOT matice – silné stránky projektového managementu firmy XY

Silné stránky projektového managementu firmy XY	Hodnocení
Zaměstnanci z oddělení vývoje dělají dobře svou práci.	4
Zaměstnanci vývoje se snaží dodržovat termíny požadavků při dohlížení vedoucího OaM.	3
Zaměstnanci vývoje vědí, kolik firma vydělává a vidí spojení s klientem.	3

Zdroj: vlastní zpracování (2024)

Z poznámek polostrukturovaných rozhovorů lze do silných stránek zařadit:

- Velmi nízká chybovost, malé množství chyb a hotfixů.
- Do Sprintů se dostávají požadavky od více zúčastněných stran.
- Pracovníci z oddělení Vývoje mají možnost přidat své nápady do Sprintu.
- Vývojář má větší přehled o frontendu díky samostatné práci.
- Tester dokáže odhadnout, kolik času mu zaberou testovací požadavky.
- Vývojový tým má pravidelné páteční schůzky.

Pro majitele je nejvíc důležité, že zaměstnanci z oddělení vývoje odvádí dobrou práci. S propojením silných stránek firmy a postřehů z rozhovorů může dobrá práce znamenat malé množství chyb, bugů a hotfixů a zároveň kdyby pracovníci neodváděli dobrou práci, firma XY by pravděpodobně nebyla v TOP 4 firmách na světě ve svém oboru.

O zapojení vedoucí obchodu a marketingu se majitel zmínil ve SWOT analýze a také v rozhovoru, spolu s ostatními respondenty. Vedoucí OaM pracuje ve firmě od září předešlého roku a jak bylo uvedeno v analýze rozhovorů a SWOT analýze firmy, stabilizoval obchod a má pomoci také s nastavením vývoje a mít nad ním pevnou ruku. Což odráží silnou stránku, že se zaměstnanci vývoje snaží dodržování termíny. Více jsou termíny a časové odhady rozepsány v kapitole 3.2.1.

Majitelova poslední silná stránka, že zaměstnanci jsou již informováni o finanční stránce firmy XY a vidí spojení s klientem.

Za silnou stránku je možné považovat, že vedoucí vývoje do plánování Sprintu zapojuje více stran – zohlední nápady podřízených a další důležité požadavky, které pochází i z jiných oddělení kromě vývoje nebo vedení.

Individuální stránky pracovníků, které fungují, jsou testerova schopnost odhadnout časovou náročnost testovacích požadavků a velká přehlednost o frontendu, což může být zároveň také bráno jako slabá stránka.

Poslední silná stránka současného projektového managementu je pravidelnost schůzky vývojového týmu. Každý pátek se tým online sejde a probírají, co se za týden stalo, proběhlo, kde se vyskytly problémy a na čem se bude pracovat další týden. Je možné tuto schůzku pokládat za náznak agilního projektového managementu.

Všechny slabé stránky současného fungování projektového managementu, které majitel vyjmenoval, se týkají oddělení vývoje (Tabulka 6).

Tabulka 6 SWOT matice – slabé stránky projektového managementu firmy XY

Slabé stránky projektového managementu firmy XY	Hodnocení
Zaměstnanci vývoje špatně reportují a plánují Sprints.	5
Zaměstnanci vývoje nerespektují slovní spojení „Klient = náš pán“.	5
Při zjištění špatných časových odhadů zaměstnanci vývoje nechtějí převzít zodpovědnost.	5
Když mají zaměstnanci vývoje vyřešit určitou situaci/problém, trvá jim to zbytečně dlouho.	4
Když nastane problém, zaměstnanci vývoje netáhnou za jeden provaz.	4
Když nastane problém, loajalita zaměstnanců vývoje se sníží.	3
Zaměstnanci vývoje nerespektují termíny.	3

Zdroj: vlastní zpracování (2024)

Po kvalitativních rozhovorech je možné slabé stránky doplnit o další položky:

- Sprints nemají fixní délku.
- Sprints se prodlužují i přes nedodržení termínů.
- Testerovi se při časovém tlaku ukrajuje čas na testování ze Sprintu.
- Vedoucí vývoje dělá časové odhady požadavků na základě pocitu.
- Odhady, které činí vedoucí vývoje se nikdy nestihnou.
- Zaměstnanci vývoje nepoznají komplexnost požadavků.
- Vývojář nebere ohledy na časové odhady požadavků.
- O změně časových odhadů se z pohledu vedoucího vývoje jen mluví bez akce.
- Pracovníci, kteří zadávají prioritu k požadavkům, mají rozdílná měřítka.
- V programu Jira je zapsaný pouze základ dokumentace.
- Zadání a dokumentace se vytváří a doplňuje po celý vývojový proces.
- Majitel vyhodnocuje Sprints sám.

Z pohledu majitele je nejvíce závažný špatný reporting, plánování Sprintů, nerespektování klientů a nepřebírání zodpovědnosti. Naopak je pro něj nejméně důležité nerespektování termínů.

Jak je popsáno v kapitole 3.2.1, ve které se analyzují získaná data z rozhovorů, je to společně propojené. Když zaměstnanci špatně naplánují Sprint, nedostatečně časově odhadnou požadavky ke splnění a ve chvíli opoždění nechtějí převzít zodpovědnost, a tak nerespektují zadané termíny.

Špatné časové odhady je možné spojit s nepoznáním komplexnosti požadavků, odhadováním na základě pocitů, tvoření zadání během vývojového procesu a základní dokumentací. Vzhledem k nepřesným časovým odhadům se Sprints prodlužují a může tak vzniknout

nerespektování časových odhadů. Prodlužování Sprintů je tak závislé na nestálých délkách a tím pádem se ubírá testerovi čas na testování, protože se upřednostní vývoj.

Za slabou stránku je možné považovat osobní vyhodnocování Sprintů majitelem, protože si vyhodnocuje Sprint pro sebe a neúčastní se ho více lidí, více pohledů.

Z kvalitativního výzkumu nevyplývaly žádné vnější příležitosti. Z **vnitřních příležitostí** je možné vyjmenovat:

- Větší sblížení desktop a online aplikace produktu.
- Probíhají diskuse mezi vedoucím vývoje a vedoucím OaM ohledně lepších časových odhadů požadavků.
- Vytvořit strukturovanější schvalovací proces požadavků do vývoje.
- Externě najatý procesní manažer na zavedení pevné ruky ve vývoji a dohlížení na dodržování termínů.
- Určit prvky zadání, která musí obsahovat.
- Více se věnovat vyhodnocení uplynulých Sprintů.
- Nastavit pravidla komunikace mezi vedoucím vývoje a jeho podřízenými.

Do **vnějších hrozeb** je možné z rozhovorů zařadit:

- Pomalá nebo žádná reakce od knihoven třetích stran při vznesení požadavku na opravení chyby.
- Restartování serverů při nasazování požadavků.

Jsou to vnější vlivy, které nelze ovlivnit a zároveň všechny vzešly od vývojáře a byly prezentovány jako méně kritické.

3.2.3 Vyhodnocení výzkumu

Cílem této kapitoly je vyhodnocení dat z kvalitativního výzkumu a tím připomenutí informací, které byly zjištěny v rámci analýzy.

Pro diplomovou práci byla získána data prostřednictvím firemních metodik, rozhovorů s majitelem a COO, sestavováním SWOT analýz a kvalitativního výzkumu. Cílem polostrukturovaných rozhovorů bylo zodpovězení výzkumného problému, výzkumných otázek a ověření či vyvrácení hypotéz, které jsou zformulovány v kapitole XX.

Po analýze dat z kvalitativního výzkumu lze vyvodit různorodost pohledů majitele a níže postavených pracovníků na výzkumný problém a fungování současného projektového managementu.

Vyhodnocení kvalitativního výzkumu velmi opíralo o kódy z otevřeného kódování, které jsou popsány v kapitole 3.2.1. Pro úplnost jsou připomenuty:

- profily respondentů;
- plánování Sprintu;
- průběh Sprintu;
- odhady práce;
- pravidelné firemní schůzky;
- kontakt s klienty;
- firemní kultura;
- navrhované změny.

Kódy, které se prolínají výzkumnými otázkami a hypotézami jsou:

- plánování Sprintu;

- průběh Sprintu;
- odhady práce;
- pravidelné firemní schůzky;
- navrhované změny.

Nejprve jsou vyhodnoceny hypotézy a výzkumné otázky.

Pro obecnou hypotézu 1 „*Časové odhady vývojových a testovacích požadavků ve Sprintech dělají nesprávné pracovníci.*“ byly stanoveny následující hypotézy:

H0: Časové odhady vývojových a testovacích požadavků ve Sprintech dělají správné pracovníci.

H1: Časové odhady vývojových a testovacích požadavků ve Sprintech dělají nesprávné pracovníci.

Jak již bylo několikrát zmíněno, časové odhady požadavků ve Sprintech dělá vedoucí vývoje. Vzhledem k tomu, že to ale dle rozhovorů nefunguje, lze konstatovat, že se jedná o nesprávnou osobu. Dle agilní metodiky by měl časové odhady dělat celý vývojový tým. **Obecnou hypotézu 1 je proto možné potvrdit.**

Obecná hypotéza 2 byla stanovena: „*Při časových odhadech požadavků ve Sprintech dochází k nedostatečné komunikaci mezi klíčovými lidmi a pracovníky.*“

H0: Při časových odhadech požadavků ve Sprintech dochází k dostatečné komunikaci mezi klíčovými lidmi a pracovníky.

H1: Při časových odhadech požadavků ve Sprintech dochází k nedostatečné komunikaci mezi klíčovými lidmi a pracovníky.

Jak bylo popsáno výše, časové odhady přiřazuje vedoucí vývoje a dále je s nikým nediskutuje. Tím pádem je **obecná hypotéza 2 potvrzena.**

Pro obecnou hypotézu 3 „*V oddělení vývoje se nekonají agilní ceremonie.*“ byly formulovány hypotézy:

H0: V oddělení vývoje se konají agilní ceremonie.

H1: V oddělení vývoje se nekonají agilní ceremonie.

Podle popisů respondentů lze soudit, že se v oddělení vývoje se nekonají agilní ceremonie. Konají se pravidelné páteční schůzky, kde se řeší několik věcí najednou, které by se dle agilní metodiky měly rozdělit do několika schůzek. Plánování a vyhodnocování Sprintu se také neuskutečňuje v rámci agilní ceremonie. Proto lze **obecnou hypotézu 3 potvrdit.**

VO1: Jak aktuálně funguje projektový management?

Odpověď na první výzkumnou otázku lze nalézt pod kódem Průběh Sprintu a kapitole 3.1.1.

VO2: Jak aktuálně probíhá plánování Sprintu?

K zodpovězení druhé výzkumné otázky slouží kód **plánování Sprintu**, který je detailně popsán v kapitole XX. Shrnutě plánování probíhá tak, že vedoucí vývoje pracuje s Backlogem, kde má nasbírané požadavky od několika vstupních stran – oddělení supportu, vedoucího supportu, vývojového týmu a z desktopové aplikace. Do daného Sprintu zařazuje požadavky dle urgencí, vyváženosti mezi frontendem a backendem, dle svého pocitu, odhadu a po domluvě s vedoucím supportu. Může se stát, že do plánování Sprintu vstoupí majitel v případě, kdy potřebuje vyvinout businessový požadavek od strategicky důležitých klientů. Ostatní členové vývojového týmu nezasahují do jeho plánování.

Do Sprintu vstupují požadavky od nových i stávajících klientů, podle testera se to nerozlišuje.

VO3: Jakým způsobem probíhají časové odhady IT projektů?

Na třetí výzkumnou otázku lze odpovědět pomocí kódu **odhady práce**. Vedoucí vývoje dělá časové odhady požadavků za frontend a backend, které posuzuje dle svého pocitu, priorit a urgency. Snaží se je odhadovat optimisticky tak, aby motivoval tým ke splnění.

Tester časově odhaduje své požadavky na testing.

VO4: Z jakých důvodů se špatně odhadují požadavky v IT projektech?

Z rozhovoru s vedoucím vývoje lze vyčíst, že jsou požadavky špatně odhadovány z důvodu jejich neúplného zadání. Tím pádem je těžké udělat časový odhad požadavku, u kterého nejde při plánování poznat jeho komplexita. Zároveň se zadání požadavku a dokumentace doplňuje postupně a tak se časem pozná, co se za daným požadavkem doopravdy skrývá.

Dle majitele ke špatnému odhadování dochází z důvodu nepoznání komplexnosti požadavků, strachu zaměstnanců učinit nebo z jejich lenosti.

VO5: Proč dochází k prodlužování Sprintů?

Vedoucí vývoje je osoba, která reálně prodlužuje Sprint. Prodlužuje je proto, aby se v daném cyklu splnily všechny požadavky a kvalitně. Upřednostňuje splnění před dodržením termínu. V případě, že je tlak na dodání některých požadavků, tak se upřednostní a ostatní posune do dalšího Sprintu.

Tester vnímá prodlužování Sprintu přidáním dalších požadavků v jeho průběhu z důvodu urgency. Z jeho pohledu se primárně prodlužuje kvůli vývoji, ne testingu – to znamená, když se prodlužuje Sprint, ubírá se zároveň z jeho prostoru na testování.

VO6: Jaká opatření přijmout pro přesnější časové odhady požadavků?

Vzhledem k tomu, že požadavky časově odhaduje vedoucí vývoje, tak navrhoval přesnější zadání požadavku. Dle vedoucího vývoje by mělo zadání obsahovat: představu vizualizaci požadavku, textaci s překladem, fungování požadavku z pohledu uživatele aplikace, jeho umístění, všechny dopady požadavku do aplikace, scénáře včetně okrajových a errorů.

Lze tedy konstatovat, že se jedná o rovnici: přesnější zadání = přesnější časový odhad.

3.3 Agilní transformace organizace

V předposlední kapitole je popsán konkrétní implementační plán agilní transformace, který popisuje přechod firmy XY na agilní řízení projektů a zároveň pomůže vyřešit její problémy. Hlavním cílem diplomové práce je sestavit implementační plán agilní transformace a navrhnout doporučení, jak může zavedení agilního projektového managementu vyřešit výzkumný problém. Vedlejším cílem je vyhodnotit současný stav projektového managementu ve firmě XY.

Vzhledem k tomu, že firma staví požadavky zákazníků na jeden produkt (desktop a online aplikace), který se kontinuálně upravuje a vylepšuje, bude implementační plán představovat ideální návod, jak zavést agilitu pro celý produkt. Implementační plán se opírá o doporučení z odborných knih, teorie agilních metod a dat z polostrukturovaných rozhovorů.

Prostřednictvím kvalitativního výzkumu vyplynulo, že by přechod na agilní řízení projektů vyřešil následující překážky: nestabilní délka Sprintů, neúplný obsah zadání, nedostatečná dokumentace, špatné časové odhady požadavků, úprava určování priorit požadavků, vyhodnocování Sprintu, zapojení klienta do procesu a vize směru vývoje.

Agilní projektový management určí Sprintů fixní délku Sprintů, systematictější plánování a průběh, díky kterému by se odstranil nedostatečné zadání požadavků, neúplná dokumentace, chybné časové odhady a zaneslo by se efektivnější určování priorit požadavků.

Je důležité zmínit, že firma XY již dodržuje několik pravidel agilních metod, které je vhodné zachovat a pracovat s nimi dále:

- pravidelnost schůzek (Scrum);
- používání programu Jira (Scrum, Kanban);
- rychlá komunikace;
- důvěra v management a podřízené (Scrum);
- průběžné upravování a zlepšování produktu (Kanban);
- sloučení vývoje a testingu do jednoho oddělení.

Implementační plán se skládá z následujících částí (všech částí se účastní agilní expert):

- 1) Firma XY si ujasní, **proč** chce přejít na agilní metody a co se jí díky nimlepší. Provede **analýzu současného stavu** a zkontroluje tak **přípravenost** na agilní transformaci.
- 2) Určí se **plán** transformace, její krátkodobé a dlouhodobé cíle, jaké obsahuje kroky. Zároveň s plánem definuje **vize** agilní transformace, firmy a produktu.
- 3) Firma XY si vybere **agilní metodu**, která je pro ni vhodná.
- 4) Díky dané agilní metodě se definují **nové týmové role** a nově se postaví **vývojový tým**.
- 5) Na základě agilních metody se zavedou pravidelné schůzky dle Scrumu.
- 6) Vedení s agilním expertem předpřipraví Sprint Backlog na Sprint Planning.
- 7) Naplánuje se **pilotní Sprint** s možností ho upravovat.
- 8) **Tým se seznámí** s agilní změnou, pilotním Sprintem a je proškolen.
- 9) Proběhne pilotní Sprint.
- 10) V návaznosti na vyhodnocení Sprintu proběhne postupné a úplné zavedení agility.

Poté, co si vedení rozmyslí agilní transformaci, se všechny kroky plánují s agilním expertem. Kroky č. 3, 4, 5 probíhají souběžně vzhledem k jejich provázanosti.

3.3.1 Důvody pro agilní přechod, analýza současného stavu, firemní připravenost

Vedení firmy XY si může sama ujasnit důvody pro přechod na agilní řízení projektů například prostřednictvím analýzy současného stavu, která byla již v rámci diplomové práce provedena. Nebo vyhledá agilního experta z konzultační firmy, který se zabývá agilními transformacemi a zhodnotí připravenost firmy na přechod – jak je vyspělá, zda má potřebný potenciál a je připravená na změny. Kromě výše uvedeného expert také zjišťuje časové a finanční zdroje, dosavadní efektivitu komunikace, spolupráci, vizi firmy a produktu a možná rizika. Agilní expert potřebuje navnímat firemní prostředí, chování zaměstnanců, jejich znalosti, dovednosti, jaká je podpora vedení, vývojové prostředí a identifikovat místa ke zlepšení.

Ve chvíli, kdy agilní expert usoudí, že má firma XY potřebný potenciál, mohou se společně domlouvat dál: jaké výhody pro ni má agilita, co je nutné změnit, průběh agilní transformace a implementace, jak pro změnu nakoupit zaměstnance, měření úspěšnosti a kam se chce firma prostřednictvím agility dostat. Díky tomu si firma plně uvědomí, co všechno znamená přechod na agilní projektový management.

Vzhledem k tomu, že přechod není pouze o vedení, firma XY zjistí názor zaměstnanců na agilní fungování a jejich předešlé zkušenosti, které se vezmou na vědomí.

V této fázi hrají hlavní roli vedení firmy XY, najatý agilní expert a dle autorčiných doporučení z kapitoly XX i Scrum Master.

3.3.2 Plán agilní transformace, vize transformace, firmy a produktu

Firma si již v předešlých krocích uvědomila, proč chce učinit přechod na agilní transformaci a co se změní. Ještě než se firma XY pustí do agilní transformace, tak je nutné, aby si vydefinovala krátkodobé a dlouhodobé cíle transformace. Mezi krátkodobé cíle transformace může patřit například seznámení a proškolení pracovníků, zavedení agilních nástrojů nebo vytvoření pracovního prostředí pro pilotní Sprint. Do dlouhodobých cílů lze zařadit transformaci kultury, kontinuální zlepšování, agilní vedení nebo jiný způsob řízení.

Společně s agilním expertem se určí plán agilní transformace: její příprava, začátek a průběh. Autorka úmyslně nezmiňuje konec, protože v agilním fungování se počítá s kontinuálním zlepšováním. Plán má konkrétně definovat výběr agilní metody, pilotní Sprint a jeho průběh, sestavení pilotního týmu dle principů zvolené agilní metody, seznámení týmu/firmy se změnou, sestavení Sprint Backlogu, změny ve vývojovém procesu, vyhodnocení pilotu, měření jeho úspěšnosti a jak v případě úspěšnosti a ochoty vedení pokračovat v dalším zavádění. Konkrétní popis jednotlivých fází je popsán v dalších kapitolách.

Výhodou je si také připravit nejčastější negativní scénáře. Již v této fázi je potřeba počítat s možným odporem zaměstnanců, být na něj připraven a diskutovat s nimi možná řešení.

Vize agilní transformace udává, jak má firma XY po agilní transformaci, v tomto případě i po pilotním Sprintu. Firma si určí jasné cíle dle SMART metody, například dodání požadavků klientů do určených termínů.

Ve fázi plánování a vize hrají hlavní roli také vedení firmy XY, najatý agilní expert a Scrum Master.

3.3.3 Výběr agilní metody

Autorčin výběr agilní metody pro firmu XY se zakládal na velikosti vývojové týmu a firmy XY, zaměření firmy, zákaznických požadavků, zavedených postupů, fungování a jaké problémy je třeba vyřešit. Vzhledem k tomu, že každá metoda pokrývá určitou část a nebyla by jako samostatná metoda dostačující, byly na základě literatury z teoreticko-metodologické části diplomové práce vybrány dvě agilní metody splňující účel: kombinace Kanbanu a Scrumu (Scrumban). Tyto dvě metody byly zvoleny z důvodů většího důrazu na vhodnou délku Sprintů, systematického plánování, scrumových agilních ceremonií, řádného používání scrumových artefaktů, pravidelné kontroly průběhu Sprintů, omezení rozpracovaných požadavků, efektivnější využití programu Jira/Confluence a z již používaných stavů požadavků v Kanbanu. Scrum vyvažuje negativní stránky Kanbanu, například nepřítomnost pravidelných ceremonií, plánování a nemožnost včasného odhalení problémů.

Obě metody mají ve firmě XY základ, který již funguje/existuje: princip pravidelných schůzek, používání programu Jira, rychlá komunikace, důvěra v management a podřízené, průběžné upravování a zlepšování produktu a sloučení vývoje a testingu do jednoho oddělení.

Vzhledem k tomu, že se program Jira již používá a firma XY má licence, nebude pro ni vznikat náklad. Je ale možné, že se bude muset zvážit počet zaměstnanců, kteří program používají – zda mají do Jiry přístup ti, co s ní reálně potřebují pracovat i po zavedení agility.

Je potřeba si dát pozor na to, aby zavedení agilních metod neovlivnilo míru chybovosti.

O výběru se domlouvá vedení firmy XY, agilní expert a Scrum Master.

3.3.4 Definování týmových rolí a sestavení vývojového týmu

Společně s výběrem agilní metody a plánováním pilotního Sprintu je potřeba myslet na přerozdělení rolí ve stávajícím týmu a sestavení nového dle agilních principů. Díky zmapování aktuálně zastoupených znalostí a dovedností, které provedl agilní expert, se sestaví nový vývojový tým a jejich profily. Při sestavování agilního týmu by mělo vedení s agilním expertem myslet na to, že mohou nastat problémy s adaptací na nové role.

V současné době se vývojový tým skládá z vedoucího vývoje, vývojáře, dvou testerů a jednoho vývojáře v Holandsku. Vedoucí vývoje pracuje na backendu online aplikace, vývojář na frontendu, oba testéři testují a vývojář v Holandsku pracuje na backendu desktopové aplikace. Všichni členové vývojového týmu pracují vzdáleně, vidí se osobně jednou za kvartál na kvartální schůzi. Vzhledem k tomu, že vývojový tým je dlouhodobě stabilní, tak má rozsáhlé know-how okolo produktu, u kterého je stále potřeba myslet na jeho specifickou. Z kvalitativního výzkumu vyšlo, že mají mezi sebou vzájemnou důvěru, která poskytuje náležitý základ.

Dle agilní metodiky se základ týmu má skládat z Product Ownera, Scrum Mastera, vývojářů a testerů. Z čehož vyplývá, že chybí zastoupení Product Ownera a Scrum Mastera. Z polostrukturovaných rozhovorů vyplynulo, že **Scrum Master** nebo agilní coach nemá ve firmě XY žádné zastoupení.

Co se týče **Product Ownera**, tak ho ve firmě XY nelze učebnicově určit vzhledem k tomu, že se jedná o přebírání požadavků z oddělení supportu/podpory – zadávají se mimo vývojový tým. Zároveň se v některých případech jedná o požadavky, které mají dopad na více klientů najednou anebo naopak na jednoho. Měl by to být člověk s rozsáhlým know-how produktu, vynikající znalostí cílových skupin, schopností sdílet informace, určovat, co a jak se bude realizovat, zastupovat komunikace mezi zainteresovanými osobami, upřesňovat zadání požadavků a nést celou zodpovědnost za funkcionality produktu. Z pohledu zadávání požadavků by se mohlo jednat o vedoucí oddělení podpory, který ale na druhou stranu nemá rozsáhlé technické know-how.

Kromě týmových rolí je také potřeba zajistit komunikaci a spolupráci prostřednictvím pravidelných schůzek (kapitola 3.3.5), vytvoření otevřeného pracovního prostředí.

Fáze Pilotního Sprintu a Nastavení agilních týmových rolí jsou brány jako předběžný plán, se kterým se seznámí vývojový tým a bude prostor ho před ostrým spuštěním změnit s ohledem na zpětnou vazbu pracovníků.

Definování týmových rolí účastní vedení, agilní expert a Scrum Master, ale přímo v definování mají hlavní slovo Scrum Master a agilní expert. Při sestavování nového vývojového týmu jsou znovu rovnocenní.

3.3.5 Pravidelné scrumové ceremonie

Vzhledem k tomu, že vývojový tým má aktuálně pouze jednu pravidelnou schůzku, kde se řeší všechno a zbytek se komunikuje nárazově, je potřeba schůzkám a komunikaci dát řád. Ze zvolených agilních metodik vyplývá, že se budou implementovat agilní ceremonie dle Scrumu. Zavádění schůzek reaguje na zpětnou vazbu nárazové komunikace od vedoucího vývoje, větší dohled nad průběhem Sprintu, informovanost všech pracovníků a snadnější odhalení problémů se splněním termínu.

Pilotní Sprint začne **Sprint Planningem**. Sprint Planning probíhá pouze jednou, na začátku Sprintu a může trvat až 8 hodin. Během schůzky se stanoví cíl Sprintu, naplánuje se práce z požadavků, které jsou uloženy v Backlogu. Vedoucí vývoje bude přidávat požadavky do

Sprint Backlogu a vývojový tým bude k požadavkům přiřazovat časové odhady, dle nich určovat priority, řešitele a stanoví se tzv. definition of done.

Co se zachová, je určování velkých priorit vedením z důvodu strategicky důležitých zákazníků. Tento princip bude dodržován z důvodu businessového hlediska – cílem je zavést agilní fungování ve firmě, ale na druhou stranu je potřeba řídit finanční stránku firmy.

Do Sprintu se zahrne časová rezerva z důvodů dovolených, nemocí, hotfixů, bugů z předešlého Sprintu a možného přidání dalších požadavků od zákazníků, které budou považovány za kritické. Proto by se Sprint nebude plánovat tak, aby byl hned na svém začátku plný. Je jednodušší přidávat požadavky, než je pak odkládat a nakonec přesunout do dalšího Sprintu. V případě, že Backlog bude obsahovat velký požadavek, rozdělí se do menších částí, které budou zvladatelné v rámci jednotlivých Sprintů a zároveň bude možné je stihnout včas.

Po rozřazení požadavků v Backlogu bude pilotní Sprint spuštěn.

V průběhu pilotního Sprintu se budou pořádat Standupy a Product Backlog Refinement. Agilní metodika udává, že **Standupy** jsou denního charakteru. Na druhou stranu vývojář v Holandsku pracuje převážně samostatně a tester 2 dny v týdnu. To znamená, že by se na Standupech potkávali pouze vedoucí vývoje a frontend vývojář. Autorka proto doporučuje nastavit ze začátku Standupy každý druhý den a v případě, kdy vznikne častější potřeba, přenastaví se na denní schůzky. To samé u délky – Standupy začnou na 15 minutách a podle potřeby se prodlouží nebo zkrátí. Neměly by se z nich ale stát řešitelské schůzky. V rámci pilotního Sprintu bude na průběh dohlížet agilní expert a Scrum Master a budou pokládány 3 druhy otázek, jak je uvedeno v kapitole 2.3.1: „Na čem si pracoval včera? Na čem budeš pracovat dnes? Vyskytly se některé překážky, které ti brání v práci?“. Vývojový tým si určí čas, ve kterém mu to bude vyhovovat. Zavedení Standupu reaguje také na poznámku vedoucího vývoje, kdy ho občas náhodná komunikace s podřízenými vytrhuje z práce.

Product Backlog Refinement je může probíhat například jednou za 14 dní. Záleží na délce Sprintu, ale minimální četnost je jednou za Sprint, aby bylo provedeno přezkoumání, aktualizace Backlogu a zkontrolování priorit požadavků. Na schůzku se připravuje Product Owner a Scrum Master, aby se zorientovali v aktuálním stavu Sprintu, připravili si agendu a případné body k diskusi. S přizpůsobením pro firmu XY by se měly během této schůzky také probírat nově přidané požadavky.

Na konci Sprintu se odehrají dvě agilní ceremonie: demo a retrospektiva. Na **Demu** budou zaměstnanci vývoje prezentovat nové funkce a požadavky, na kterých se pracovalo během pilotního Sprintu. Délka schůzky je ovlivněna množstvím hotových požadavků.

Retrospektiva se také odehrává jednou za Sprint a účastní se jí všechny zainteresované strany. Tato schůzka uzavírá Sprint a jejím cílem je sdílení zpětné vazby, konstruktivního sebehodnocení pracovníků a shrnutí, co fungovalo/nefungovalo, jaké se objevily problémy a co do příštího Sprintu zlepšit. Na schůzce se určí konkrétní následující kroky.

Retrospektiva po pilotním Sprintu hraje velkou roli, protože se zároveň celkově vyhodnotí, zda byl úspěšný či ne. Víc se na ni bude rozebírat, jak fungoval vývojový tým a vedení i řadoví zaměstnanci budou prezentovat výsledky a vyjádřit své názory na pilot. Jak uvádí Březina (2020, str. 139), z retrospektivy může vyplynout, že ve firmě chybí daná specializace, dovednosti nebo znalosti.

Pravidelných ceremonií se bude účastnit celý vývojový tým, Scrum Master, analytik a agilní expert. Agilní expert a Scrum Master je povedou a rozdělí si mezi sebou role, aby se nekřížili. Vedení se jich může pasivně účastnit, ideálně aby se účastnili maximálně Sprintu Planningu a Retrospektivy.

Všechny scrumové ceremonie se budou konat v čase, aby byli pracovníci svěží, ale zároveň jsou nejméně pracovně vytiženi.

3.3.6 Příprava Sprint Backlogu

Před fází Seznámení vývojového týmu se změnou a spuštěním pilotu si vedení, vedoucí vývoje, Scrum Master, agilní expert předpřipraví a uklidí Backlog na začátek Sprintu. Dotažen do finální podoby bude až na schůzce Sprint Planning.

Aktuálně se v rámci programu Jira využívá **Kanban board**, který je propojen s Backlogem a používají se stavy: To do, Development in progress, Code Review, Development done, Waiting, Testing, Stage released, Prod released a Closed. Vzhledem k tomu, že takto nastavené stavy požadavků ve firmě XY fungují, zachovají se jejich tak.

Ve stavu **To do** by se měly nacházet všechny požadavky pro daný Sprint, pilotní Sprint, seřazené dle priorit, které se určí na Sprint Planningu. Z tohoto stavu si bude vývojový tým sám volit svou práci.

Do stavu **Closed** se budou dostávat pouze hotové požadavky, které zde budou uloženy do konce pilotu a případně budou čekat na možné chyby v provozu. Ve chvíli, kdy se ukončí Sprint, se všechny požadavky ve stavu Closed archivují, a tak v aktuálním Kanbanu zaniknou.

Názvosloví se ponechá stejné, tedy User Stories v programu Jira se budou nazývat požadavky.

3.3.7 Plán pilotního Sprintu

Autorka ještě před spuštěním celé agilní transformace doporučuje uspořádat pilotní Sprint z následujících důvodů:

- Tým získá první společné zkušenosti s agilitou, setká se s problémy, které bude muset řešit a naučí se s agilitou pracovat.
- Při pilotu je možné, že se narazí na možné překážky, které by mohly v budoucnu bránit úplnému agilnímu přechodu. Tím, že se identifikují v brzké fázi, bude snadnější je vyřešit.
- Pilot je první krok firmy, na kterém se budou budovat další kroky transformace – určuje si tak základ. Je to také první krok, který bude vnímat více lidí než jen vedení firmy XY a v případě, že bude pilotní Sprint úspěšný, bude větší šance pro nakoupení zaměstnanců, ostatních oddělení a motivace pracovat efektivněji než původně.
- Pro zaměstnance je to příležitost se ohlédnout na celý vývojový proces, identifikovat silné a slabé stránky v akci a najít způsoby ke zlepšení.
- Pilotní Sprint má menší finanční dopady i v případě neúspěchu než rovnou zavádění celé agility a pozdější zjištění překážek.
- I přesto, že agilní expert vyhodnotí v předchozích krocích identifikuje potenciál firmy XY, je pilot přímý důkaz vhodnosti daných agilních metod.

Cílem pilotního Sprintu bude především dodání hotových požadavků a jejich nasazení na produkční prostředí, dodržení časových termínů požadavků, větší dohled a přehled o průběhu Sprintu, efektivní komunikace mezi zaměstnanci, víc komplexní zadání požadavků, ověření nové formy dokumentace, vyhodnocení vhodnosti agilních metod a úspěšnosti oproti předešlému nastavení Sprintů. Cíle budou definovány dle SMART metody.

Plán pilotního projektu bude obsahovat:

- vybrané agilní metody;
- stanovení konkrétních cílů Sprintu dle požadavků klientů;

- vytvoření nového týmu a určení agilních týmových rolí;
- příprava Backlogu na začátek Sprintu;
- seznámení vývojového týmu s pilotním Sprintem;
- proškolení vývojového týmu;
- Sprint Planning a prioritizace požadavků;
- průběh pilotu a další pravidelné ceremonie;
- vyhodnocení a zpětná vazba na pilotní Sprint;
- rozhodnutí o dalších krocích.

Firma XY si v rámci plánu pilotu definuje konec Sprintu – zda to budou hotové požadavky nebo nasazení do aplikace.

Na pilotním Sprintu se vyzkouší i jeho kratší délka. Nebude trvat 3 měsíce, ale nastaví se měsíční délka. Měsíc autorka vybrala z důvodu, že se ve většině odborných knih uvádí jako maximum. Vzhledem k tomu, že se zkrátí a po jednom Sprintu není možné vyhodnotit, zda byl efektivnější než v původním nastavení, tak proběhnou 3 pilotní Sprints. Všechny budou mít ze začátku nastavený stejný průběh, jak je popsáno v diplomové práci a budou mít prostor se flexibilně měnit a reagovat na připomínky od zaměstnanců.

3.3.8 Seznámení s agilní metodikou, upraveným průběhem vývoje a proškolení

Až této fáze se zúčastní všechny zainteresované strany a představí se jim změna, plán a průběh pilotního Sprintu. Do zainteresovaných stran budou zařazeni: vedení, vedoucí vývoje, vedoucí supportu, vývojáři a testeři. S přihlédnutím na aktuální situaci firmy XY by mezi ně mohl patřit také vedoucí OaM, jehož úkolem je zlepšit fungování vývojového týmu.

Při vysvětlování agilního přechodu je nutné pracovníkům sdělit důvody, proč by firma XY ráda přešla na agilní projektový management, jaké překážky pomůže vyřešit, její přínosy, očekávání, cíle, porovnání se současným fungováním a představit složení agilního vývojového týmu během pilotu, základní agilní principy a nové pravidelné schůzky. Během seznamovací schůzky budou mít pracovníci prostor k otázkám, diskusi a možným zásahům do plánu. Zároveň vedení s agilním expertem bude mít připravené otevřené body ohledně přípravy a průběhu pilotního Sprintu, které by rádo probralo s ostatními zaměstnanci. Například se může otevřít diskuse na téma kompletního zadání požadavků, které zmínil vedoucí vývoje (kapitola 3.4.1).

Se základními principy se seznámí i ti, kteří je již v minulosti používali, aby informace, které zaměstnanci dostanou, byly shodné. A současně dát zkušenějším zaměstnancům možnost sdílet své zkušenosti a díky nim se lépe zformují změny.

Zainteresovaní zaměstnanci budou proškolení agilním expertem v základních agilních principech fungování a používání Scrumu, Kanbanu. Školení se zahájí vysvětlením důvodů a cílů agilní transformace. Agilní expert se Scrum Master se zaměří na výhody agilního přístupu jako zvýšená flexibilita, rychlejší dodávání hodnoty a lepší spokojenost zákazníka. Následně je seznámí s principy agilního přístupu – iterativní vývoj, samoorganizace týmu a pravidelná zpětná vazba. Detailně rozeberou Scrum, s týmovými rolmi (Scrum Master, Product Owner, tým), ceremoniemi (Sprint Planning, Standup, Product Backlog Refinement, Demo, Retrospektiva) a artefakty (Product Backlog, Sprint Backlog, Increment). Facilitátoři předvedou reálnou ukázkou simulace Scrumu a Kanbanu a znovu si projdou definice jednotlivých stavů v Kanbanu a případně je upraví. Na metodiku napojí role a odpovědnost jednotlivých členů. Na závěr agilní expert a Scrum Master dostanou od zaměstnanců zpětnou vazbu a budou odpovídat na případné otázky. Po celou dobu školení mají prostor k otázkám, obavám a diskusi.

Před spuštěním pilotního Sprintu vedení a agilní expert nechají dostatečný prostor k diskusi a akceptaci týmu. Pokud ne, může se stát, že pilot nebude fungovat z důvodu nesoučinnosti celého týmu.

3.3.9 Průběh pilotu

Během pilotní Sprintu vedení, agilní expert a Scrum Master sledují jeho průběh, mění se stav Sprint Backlogu, průběžné splňování požadavků, spokojenost zákazníků, efektivitu nově zavedených ceremonií a pozorují objevující se problémy, překážky, které analyzují, hledají řešení a při retrospektivě z nich zformulují tzv. lessons learned. V průběhu se také dohlíží na množství práce jednotlivých pracovníků, aby měli byli stále zaměstnání a nemuseli čekat. Pokud například tester nemá na sebe přiřazené požadavky k otestování, tak monitoruje aktuální verzi a opravuje možné nesrovnalosti. Určitým způsobem takto funguje teď, a proto se tento princip ponechá.

Pilotní Sprint bude také dostatečně flexibilní, aby bylo možné měnit jeho nastavení za pochodu a nefungovalo se dle původního plánu celý Sprint. Může to být nepříjemné pro zaměstnance a vytvořili si tak odpor k dalším změnám.

Na konci pilotního Sprintu se uspořádá retrospektivní schůzka, která je detailněji popsána v kapitole 3.3.5.

3.3.10 Postupné a úplné zavedení agility

Po vyhodnocení pilotního Sprintu, který bude prohlášen za úspěšný, by firma XY měla ještě pár Sprintů v aktuálním nastavení.

Postupně se bude objevovat méně překážek a ve chvíli, kdy vedení usoudí, že je proces dostatečně stabilizován, může se posunout v dalším zavádění agilního řízení požadavků. Společně s agilním expertem by se měla stanovit vize, strategie dalšího přechodu, cíle, časový harmonogram a zaměstnanci by měli stále cítit podpora od vedení, efektivní komunikaci a dostávat od nich pravidelnou zpětnou vazbu.

V průběhu stabilizace by mělo počítat v kontinuálním zlepšováním a je možné, že budou potřeba další školení ze strany agilního experta nebo minimálně jeho podpora.

Dostatečně stabilní vývojový tým se pozná podle několika faktorů:

- stal se z něho samoorganizující tým;
- ustálila se jeho výkonnost a již několik Sprintů odvádí pozitivní výsledky;
- dodržuje dohodnuté plány po časové a obsahové stránce;
- tým má schopnost se přizpůsobit změnám a neustále se zlepšovat;
- komunikace nikde nevázne;
- tým je motivován dosáhnout cílů;
- do retrospektiv se promítá čím dál více kladných výsledků a pozitivní zpětná vazba zaměstnanců.

Dalším směrem, kterým se může firma XY vydat v rámci vývojového týmu prostřednictvím agilní transformace, je například zapojení klienta do procesu, změna firemní kultury a formování agilní mindsetu pracovníků. Nebo může začít se zaváděním agility v dalším odděleních a postupně tak transformovat celou firmu.

3.4 Shrnutí a návrh doporučení

Implementační plán agilního projektového managementu představuje strukturovaný proces, jak má firma XY přejít na agilní fungování. Začíná **ujasněním** si firmy, proč se chce agilně transformovat a co jí agilita přinese. Ve chvíli, kdy má jasno, tak poptá **agilního experta** nebo coacha z externí agentury, který se zabývá agilními transformacemi a společně začnou projednávat její průběh, aby dokázala představit, co všechno transformace obnáší. Na začátku procesu si agilní expert zmapuje současný stav firmy XY, zjistí, jaké znalosti a dovednosti jsou zastoupeny zaměstnanci a zhodnotí připravenost firmy na změny a přechod. Pokud agilní expert usoudí, že firma má potřebný potenciál, tak s vedením zahájí vytváření **plánu transformace**, cíle, vize firmy a produktu, na které se napojí cíle a vize přechodu, výběr agilní metody, uspořádání pilotního sprintu, časový harmonogram pilotu a následné transformace, zformování nového vývojové týmu, definování agilních týmových rolí, zavedení agilních ceremonií, seznámení a proškolení zainteresovaných zaměstnanců a jaké změny budou provedeny jako první. Zároveň se také připraví Backlog na pilotní Sprint. Po vypracování výše uvedeného plánu se uskutečnění seznámení a představení změny zaměstnancům, na kterém jim budou také sděleny důvody k přechodu na agilní fungování a budou mít prostor klást otázky, sdílet své zkušenosti, obavy a mít čas na vstřebání. Následně pracovníky, kteří se budou účastnit pilotního Sprintu, zaškolí agilní expert, vysvětlí se jim jejich týmové role a znovu se jim nechá prostor pro adaptaci nebo otázky. Během představení a školení je potřeba počítat s připomínkami zaměstnanců a pokud budou oprávněné a konstruktivní, tak je zanést do plánů.

Pilotní Sprint se odstartuje schůzkou Sprint Planning, na které všichni společně sestaví Backlog, vyberou požadavky dle termínů a kapacit, určí priority, časové odhady a přiřadí řešitele k jednotlivým požadavkům. Následně se spustí Sprint a zaměstnanci začnou pracovat na požadavcích. V průběhu Sprintu bude agilní expert s vedením sledovat aktuální stavy, sbírat postřehy, řešit překážky a budou probíhat pravidelné ceremonie. Každý den nebo dle nastavení budou organizovány Standupy, v průběhu Sprintu Product Backlog Refinement a na konci Demo a retrospektiva.

Retrospektiva bude současně sloužit k vyhodnocení pilotního Sprintu a poté vedení s agilním expertem rozhodne o úspěšnosti či neúspěšnosti. V případě úspěšnosti bude dalším krokem kontinuální zlepšování Sprintů a stabilizace procesů. Ve chvíli, kdy bude tým zvyklý na nové způsoby a agilní fungování bude zintegrováno do běžného provozu vývojového týmu, bude se agilita rozšiřovat do dalších oddělení.

3.4.1 Návrhy doporučení

Jak bylo uvedeno v předchozí kapitole, agilní transformace může vyřešit překážky firmy XY. I přesto autorka doporučuje stanovit pravidla, která určí řád a společně s agilní metodikou budou tvořit celek:

- Nestabilní délka Sprintů se vyřeší díky fixně nastavené délce Sprintů, která se na rozdíl od skutečnosti nebude prodlužovat.
- Neúplný obsah zadání a nedostatečná dokumentace selepší na základě stanovení pravidel.
- Špatné časové odhady požadavků se usnadní díky krátkým iteracím, protože pak pracovníci budou schopni určit časovou náročnost úkolu a během Sprintu ji upřesňovat.
- K úpravě určování priorit požadavků dojde prostřednictvím Sprint Planningu, na kterém se kromě rozřazování požadavků budou společně určovat také priority;
- Vyhodnocování Sprintu bude rozšířeno do formy týmové retrospektivy a je popsáno v kapitole 3.3.5.
- Zapojení klienta do procesu.

- Víze směru vývoje bude více komunikována.

Je důležité zmínit, že zavedení doporučení nebo opatření není možné dělat po jednom – musí se jednat o společnou implementaci, protože jsou společně provázané. Stejně jako je v teorii uvedeno, že při zavádění agilní metodiky není praktické fungovat pouze podle částí a tím skončit.

Měnící se délka Sprintů

Jak budou Sprints fungovat při zavedení agilních metod, je popsáno v kapitole XX. Mezi doporučení autorka zařadila neprodłużování Sprintů tak, jak se děje v současné době. V praxi to znamená obrátit využívání Sprintů – místo prodłużování raději přesunout nestihnuté požadavky do dalšího Sprintu. V plánování se má počítat s časovou rezervou pro stihnutí všech požadavků.

Respondenti během rozhovorů používali pojmy Sprint a Release. Ve firemním názvosloví se používají oba pojmy a znamenají to samé – Sprint/Release trvá 3 měsíce. Agilní metodika uvádí, že Sprint nemusí vždy končit Releasem, nasazením požadavků do ostrého provozu. Pokud by firma XY trvala při agilní transformaci na Releasech jednou za 3 měsíce, tak je zachování možné a Sprints budou mít význam, který udává Agile. Tedy Sprints budou součástí Releaseů. Za předpokladu dlouhodobého fungování měsíčním Sprintů, bude 1 Release obsahovat 3 Sprints.

Podoba zadání

I přesto, že by přechod na agilní fungování pomohl firmě vyřešit výše zmíněné problémy, tak se dle autorky zavedou jednotná pravidla pro zadání, která se budou dodržovat a budou brána jako minimální podoba. Vzhledem k tomu, že zadavatelů požadavků je několik, určí se minimální podoba zadání, kterou budou všichni respektovat a pokud napíší k zadání více poznámek než bylo nutné, bude to bráno jako plus.

Vedoucí vývoje, který aktuálně dostává zadání požadavků od oddělení supportu a testera, v rozhovoru popsal, jak by mělo vypadat zadání požadavku:

- jak má požadavek vypadat;
- textace, která s ním souvisí;
- překlad textace;
- místo/a, na kterých se bude požadavek vyskytovat;
- UX forma.

Obecně by zadání dle něho mělo „splňovat všechna místa z hlediska uživatele, která to ovlivní“ a případně také vydefinované krajní situace, ve kterých může dojít k errorům.

Lze konstatovat, že pokud by zadání mělo fungovat tímto způsobem, tak bude docházet ke konfliktu a dá se říct, že by ani nefungovalo. Pracovníci ze supportu nemají takové know-how, aby dokázali popsat, jak má požadavek vypadat ze stránky UX a aby analyzovali, kam všude mít požadavek dopad, krajní situace a error. Tester má sice UX znalosti, ale není jeho úkolem vytvářet textace a překládat je. Nestíhal by pak svou testovací roli, obzvláště s omezenou kapacitou 2 dny za týden. Z tohoto poznatku vyvstává potřeba, kterou identifikovalo několik respondentů, a to najmutí analytika.

Za předpokladu, že by firma XY nabrala analytika, tak by oddělení supportu mělo svá pravidla na zadání, která dokážou splnit. Následně by se zadání předalo jemu, určí by krajní situace a dopad požadavku. Ideální by bylo najmutí seniorního analytika, který má zároveň přesah do UX, aby se určil design požadavku ještě před vývojem. Z pohledu komplexnějšího zadání požadavku nábor seniorního analytika s přesahem do UX tvoří výhodu. Na druhou stranu kvůli

specifičnosti produktu je možné, že jeho onboarding bude trvat odhadem pár měsíců než mu zcela porozumí. Proto by se zavedly pravidelné schůzky pravděpodobně s vedoucím vývoje nebo testerem na předávání know-how, aby byl analytik postupně schopen doplňovat zadání sám.

Shrnutě by tvorba zadání požadavku vypadala tak, že by základ napsali pracovníci z oddělení supportu, poté by se přesunulo k analytikovi, který by doplnil UX formu, dopady požadavku do aplikací, definoval by krajní situace a errorry. Takto by zadání bylo hotové a předalo se na vývoj. Pro vývojáře je důležitá UX forma, dopady a errorry, pro testera také dopady, krajní situace a errorry. S touto formou zadání se vyjde vstříc i testerovi, který byl měl dostatečně předpřipravené zadání a mohl se ještě před samotným testováním mapovat testovací scénáře.

Vzhledem k tomu, že se jedná o velký zásah do procesů požadavků a najmutí nového seniorního analytika, nebude výsledek vidět okamžitě. Vyhodnocení úspěšnosti proběhne až po několik Sprintech.

V případě, kdy analytik nebude nabrán, se náplň práce vedoucí vývoje, vývojáře a testera rozšíří a rozdělí se mezi ně zadání požadavků. Začátek procesu je stejný, to znamená minimální základ sestaví pracovníci z oddělení supportu. Poté tester definuje UX formu. Další parametry se určí společně v týmu, jelikož má každý člen vývojového týmu jiný pohled na požadavek. Tedy dopady požadavku, krajní situace a errorry určí hromadně vedoucí vývoje, vývojář a testeri. Pokud by zadání požadavku mělo do budoucna fungovat tímto způsobem, vedlo by to k navýšení kapacit testera, zvýšení časové náročnosti všech vývojových pracovníků a tím pádem by měli dostat vyšší mzdu.

Autorka proto doporučuje nábor analytika, protože se lépe bude udržovat systém práce a časová náročnost zaměstnanců vývojového oddělení se zvýší pouze dočasně, počas jeho onboardingu.

Podoba dokumentace

Vzhledem k anonymitě firmy XY není možné v diplomové práci detailně rozebírat náplň dokumentace, proto je doporučení obecnějšího a procesního charakteru. Během polostrukturovaných rozhovorů docházelo k zaměňování pojmů zadání požadavku a dokumentace. Zároveň je ve firmě několik druhů dokumentace – zaznamenávací / technická / programátorská, regulatorní a uživatelská. Regulatorní dokumentaci firma XY splňuje a je pravidelně kontrolována příslušnými orgány, protože je vázaná legislativou. Společně s ní není náplní diplomové práce ani uživatelská, protože je příliš specifická a tajná.

Technická neboli programátorská dokumentace není dle testera tak podrobná, jak by měla být a dle vývojáře měla obsahovat informace pro testera a popis implementace požadavku. V tomto bodě se kříží několik záležitostí. Co by mělo dle vedoucího vývoje patřit do dokumentace, co do zadání požadavku a vytváření dokumentace vývojářem, aby měl tester dostatek informací, nemusel ji vymýšlet „za pochodu“ a objeovat chybějící informace až při testování.

Zadání požadavku je obecně definováno jako souhrn základních informací, které je vstupem do Sprintu a obsahuje například očekávání od požadavku, vlastnosti nebo hlavní funkce. Zároveň slouží jako vstup pro dokumentaci, která daný požadavek rozvádí a ve které nastává prostor pro analýzu. Zadání je vytvořeno na začátku Sprintu a převážně se zachovává jeho podoba, pouze pokud se změní potřeba požadavku (vlastnost, funkce a podobně). Dokumentace se tvoří v průběhu Sprintu a postupně se doplňuje, dokud komplexně nepopisuje daný požadavek. Má obsahovat analýzu požadavku, dopady, jak se má požadavek chovat (i v krajních scénářích) a jejím účelem je popsat návrh, vývoj a testování požadavku.

Přímo tvoření dokumentace je v podobném stavu jako podoba zadání a vyvstává stejná potřeba náboru analytika s přesahem do UX designu.

Na základě rozhovorů je nutné nejprve definovat, co je zadání požadavku, a technická/zaznamenávací dokumentace a jaký by měl být jejich obsah tak, aby to vyhovovalo všem zainteresovaným stranám. Nově bude zadání tvořit oddělení supportu a ve chvíli, kdy se zadání dostane k analytikovi ho přenesou do dokumentace a začíná pracovat na potřených analýzách a dalších činnostech. Postup by byl stejný jako u tvorby zadání: určí UX formu, dopady, krajní situace a errorry. V případě, kdy by nebyl vybrán seniorní analytik a stala se z toho činnost pro stávající zaměstnance, vypadalo by to stejně jako u tvorby zadání. Navýší se náplň práce a časová náročnost zaměstnanců vývoje.

V rámci tvorby zadání a dokumentace vznikají firmě XY dvě možnosti: zadání bude velmi široké a do dokumentace se překloupí později nebo zadání bude základního charakteru a ihned po přijetí požadavku se z něho stane dokumentace. Z toho důvodu se autorka znovu přiklání k náboru analytika. V obou případech vzniká práce pro analytika stejného množství, ale rozdíl je časový.

Špatné časové odhady

Díky změnám podoby zadání a následné tvorby dokumentace budou časové odhady snadnější. Spolu se zadáním je nutné změnit také délku Sprintů a plánování Sprintů, které mají na odhady vliv.

Ve chvíli, kdy se zkrátí délka Sprintů na XX, bude obsahovat méně požadavků než dříve. Tím pádem bude pro vývojový tým i časově méně náročné odhadovat menší množství a zabere jim to kratší dobu. Místo určování odhadů v časových jednotkách (hodiny, dny) autorka doporučuje použít agilní metody typu „Planning Poker“, „T-shirt sizing“ nebo „Větší – menší“.

Doležal (2022, str. 117) označuje Planning Poker za přesné relativní odhadování. Metoda funguje na principu škály čísel například 0, 1, 3, 4, 7 až po maximální hodnotu třeba 200. K číslům se zároveň přiřadí symboly, která značí, že je zadání požadavku nesrozumitelné, velmi komplexní nebo potřebuje upřesnit. A ve chvíli, kdy se určují časové odhady, každý člen týmu vybere dle sebe odpovídající číslo (velikost požadavku) a případně i symbol. Autor uvádí, že se nejedná o kompromis nebo čisté rozhodnutí Product Ownera, jakou má požadavek velikost. V důsledku to může snížit zodpovědnost, protože to za daného člena týmu určil někdo jiný.

T-shirt sizing se odvíjí od velikostí oblečení: XS, S, M, L, XL, XXL. Doležal (2022, str. 117) vysvětluje, že by se přiřazená velikost požadavků měla zakládat na již provedených, historických požadavcích a podle nich si určit, o jakou velikost se jedná.

Metoda Větší – menší je obecného charakteru a je založena na principu srovnávání požadavků. Odhadování funguje tak, že se určuje, jaký požadavek je větší či menší a podle toho je uspořádán Product Backlog, ze kterého se pak berou požadavky do Sprint Backlogu.

Pro vývojový tým autorka navrhuje nejprve vyzkoušet metodu T-shirt sizing a v případě, že bude nevyhovující, tak přejít na Planning Poker.

Při plánování Sprintu je důležité brát v úvahu velikost požadavků (XS, S ...) a přiřazenou prioritu, která je popsána níže.

Priority

Počet zadavatelů souvisí i s určováním priorit. V programu Jira lze nastavit 3 úrovně důležitosti: vysoká, střední a malá. Vzhledem k různorodosti požadavků se může stát, že 3 úrovně budou působit nedostatečně. Poté se musí postupovat dle pravidel, která si firma XY interně nastaví. V určování priorit je možné jít několika cestami, přičemž se zachová nejvyšší priorita u požadavků, které jsou velmi kritické nebo se jedná o bugy/hotfixy (níže hromadně označováno jako chyby). V praxi bugy a hotfixy budou vždy na prvním místě, protože mají přímý dopad na

klienty a vzhledem k důležitosti produktu není možné, aby produkt nesprávně fungoval delší dobu než je to nutné. V Jire budou označeny jako bug (již zavedeno), což je odlišné označení od ostatních úkolů a bude jim přiřazena vysoká priorita. Pokud se ve Sprint Backlogu vyskytne více chyb, může se buď postupovat hierarchicky nebo o jejich pořadí rozhodne vývojový tým na denních standupech. Můžou nastat situace, kdy se chyby objeví mimo časový rámec pravidelných schůzek a bude potřeba je vyřešit. Pak je samozřejmě v rámci efektivní komunikace a nastavené spolupráce možné se domluvit na řešení mimo stanovené časy.

Při plánování každého Sprintu se vždy vytvoří časová rezerva na jejich možný výskyt a nebude se do rezervy zasahovat. Ideálně, aby se v případě jejich reálného menšího výskytu ani nezkracovala a místo toho se může ke konci Sprintu splnit požadavek o vhodné velikosti z Product Backlogu (za předpokladu, že jsou všechny požadavky ve Sprint Backlogu hotové).

Několik cest je zde pro požadavky s jinými prioritami než jsou kritické chyby.

První cesta je, že se zavede systém přiřazování priorit. Z rozhovorů vyplynulo, že občas vedení nebo businessové oddělení firmy XY určuje prioritu požadavku z důvodu strategicky důležitého klienta a vývojový tým tuto potřebu chápe, protože tak vysoko nevidí. Je možnost, aby tento druh požadavků měl také vysokou prioritu a vývojový tým to bude dále respektovat. Ve chvíli, kdy ve Sprint Backlogu bude více těchto požadavků, budou rozhodovat termíny nebo se detailněji prodiskutují na schůzkách Product Backlog Refinement. Z toho vyplývá povinná podmínka k zadání požadavků – reálný termín dodání.

Co se týče stanovování priorit dalších požadavků je možné, aby je určovali zadatelé stejně jako teď, ale dodržovali systém. Nebo zadavatelé nebudou zadávat úrovně priorit, doplní termín splnění a podle něj si vývojový tým určí prioritu na Sprint Planningu. Když požadavky přijdou v průběhu Sprintu, tak se určí priorita na Product Backlog Refinementu.

Další požadavky je možné brát velmi individuální, a proto aby tento systém priorit mohl fungovat, je potřeba ve firmě XY zmapovat všechny druhy požadavků a na základě zkušeností vývojového týmu rozhodnout, zda se jedná o vysokou, střední nebo nízkou prioritu. Zároveň tento systém splňuje výroky respondentů, kteří mají pocit, že nemohou určovat priority z důvodu neznání detailu. Mezi druhy můžou patřit například požadavky s dopadem na velké množství klientů, důležité funkcionality nebo požadavky pro jednoho klienta.

Zapojení klienta

Dle agilní metodiky se má klient zapojovat v průběhu procesu, v některých případech se z něj může stát oficiální člen týmu.

Vzhledem k cílové skupině nemá firma XY možnost přijmout klienty jako členy týmu (když pracují na jejich požadavku). Firma s klientem komunikuje na začátku procesu, kdy si ujasňují zadání a na konci při předání. V průběhu práce firma s klientem nekomunikuje, pouze když se ozve sám. U malých požadavků typu přidání tlačítka do aplikace nemá smysl zapojovat klienta v průběhu, protože všechny potřebné informace jsou v zadání a ukáže se až výsledek. Při předání výsledku se s ním diskutuje, zda je výsledek odpovídá jeho očekávání. U větších požadavků (například nové přihlášení do aplikace) firma může již využít zpětné vazby zákazníka. Větší, časově náročné požadavky budou nově rozděleny do menších částí, jejichž maximální velikost bude polovina Sprintu. V případě kdy velký požadavek bude rozdělen například do dvou Sprintů, firma na konci prvního Sprintu předá nebo prezentuje aktuální stav a podobu s klientem a získá tak zpětnou vazbu.

Implementace výše popsaného procesu může být pro firmu XY velmi komplikovaná. Autorka proto navrhuje, aby zpětná vazba byla například součástí písemné dohody se zákazníkem.

Další doporučení

Mezi další doporučení byly zařazeny menší návrhy nebo alternativy.

S ohledem na propojení firemních překážek a agilní transformace, autorka doporučuje nejprve najmout **seniorního analytika** s přesahem do UX, u kterého se kromě požadovaných znalostí a dovedností bude hledat i agilní mindset. V případě, že by byl problém najít takovou osobu, je možné ustoupit z UX a po přijetí ho nechat proškolit. Dále je potřeba před transformací upravit zapojení majitele do vývoje. V rozhovoru s ním bylo řečeno, že by se majitel rád částečně nahradil manažerem, který by byl hlavou finálních rozhodnutí za vývoj místo majitele. Autorka navrhuje spojit **roli „manažera“ se Scrum Masterem**, který ve firmě XY chybí a jeho role musí být zaplněna při agilní transformaci. Proto je potřeba ho najmout ještě před změnou a zároveň by se měl účastnit přípravy implementačního plánu agility. Za agilní transformaci bude zodpovědný Scrum Master. U pracovních pozic je interně nutné posoudit, zda mají být zaměstnání na plný nebo částečný úvazek.

Již v implementačním plánu agility (kapitola XX) je několikrát zmíněn tzv. **agilní expert** z externí společnosti. Autorka navrhuje externího agilního experta proto, že má mnoho zkušeností s agilními transformacemi, dokáže s přechodem rychle začít, zohlednit náklady, komunikaci, kulturu ve firmě a má vyzkoušené, co a jak funguje a současně dokáže své znalosti o agilitě, ověřené způsoby a postupy přizpůsobit na míru. Pro představu, jak externí společnosti provádějí klienty agilní transformaci, bylo vybráno několik konzultačních agentur.

Integrated Consulting Group (2024) pomáhá firmám zjistit, zda je agilita vhodná, jaké nástroje mají využívat a jak by ve firmě vypadal agilní přechod. Společnost uskuteční analýzu současného stavu a dle výsledků navrhne postup. Kromě svého coache, který povede transformaci, zavedou opatření efektivní implementace a sestaví časovou osu (roadmapu). Roadmapa začíná vizí, definicí, přechází na přípravu přechodu, adaptaci, učení a končí upevněním zavedených přístupů a nechání prostoru pro další firemní rozvoj. Organizace Integrated Consulting Group pracovala například s firmami jako Albert Česká republika, ČSOB, Dr. Max, Kooperativa, UPC, Škoda Auto nebo Vodafone.

Mezi agilní coache provádějící ve firmách agilní transformaci patří Zuzana Šochová. Šochová (2024) začíná změnu školení pilotního týmu, které trvá 2 dny. Před školením je možné mít workshop s vedením, aby se určila očekávání nebo důvody. Následuje domluva, co jak bude probíhat a při nastavování nových procesů klade důraz na změnu mindsetu. Poté se proces transformace nasměřuje na Product Ownera, Scrum Mastera a další zúčastněné osoby. Třetí fáze znázorňuje tzv. Recharge, který představuje samostatné pokračování firmy. Tím, že postupuje sama a zkouší, co ji funguje/nefunguje, tak dospěje do bodu, kdy potřebuje nový pohled. V tu chvíli se přechází na další fázi – posunutí se na další úroveň.

Poslední firmou, kterou autorka vybrala, je Deloitte. Deloitte (2024) poskytuje službu Agilní Transformation, kterou klienta provede krok po kroku. Deloitte klade důraz na nastavení správných očekávání, a proto začíná nejdříve vzájemnou dohodou a monitoruje organizační strukturu, týmové role nebo specifika klienta. Poté postupuje po malých krocích, aby šel klient správným směrem. K této službě nabízí navíc nastavení agilního mindsetu, seznámení s agilními metodami nebo vyhodnocení pracovního prostředí.

V rámci výše vyjmenovaných firem autorka navrhuje Integrated Consulting Group, protože nabízí komplexního průvodce agilní transformací a díky referencím a množství klientů na svých webových stránkách mají pozitivní výsledky.

Další návrh od autorky k agilní transformaci je **správné načasování** uspořádání pilotního Sprintu a spuštění změny. Vzhledem k tomu, že firma XY má po celý rok sezónu, tak se pilotní Sprint spustí v momentě, kdy začíná jakýkoliv Sprint za současného fungování (jednou za tři měsíce).

4 Závěr

Tématem diplomové práce byla agilní transformace a její implementační plán ve firmě XY. Proto hlavním cílem diplomové práce bylo sestavit implementační plán agilní transformace a navrhnout doporučení, jak může zavedení agilního projektového managementu vyřešit výzkumný problém. Vedlejším cílem bylo vyhodnotit současný stav projektového managementu ve firmě XY.

Diplomová práce byla rozdělena na 2 hlavní části: teoreticko-metodologickou a praktickou část.

První kapitola teoreticko-metodologické části se zaměřila na základy projektového managementu, jednotlivé fáze a přístupy projektů. Druhá kapitola se zabývala definicí Agilu a agilního projektového managementu, rozebráním Agilního manifestu a vysvětlením jednotlivých principů, fázemi agilního procesu, týmovými rolemi, agilním midsetem a kulturou, teorií agilní transformace a rozdíly mezi tradičním a agilním projektovým managementem. Třetí kapitola představila nejznámější metodiky agilního projektového managementu, konkrétně Scrum, Kanban a SAFe. V rámci Scrumu byly popsány jeho principy: základní týmové role, artefakty a ceremonie. Poslední kapitola se věnovala metodice diplomové práce. Byl definován výzkumný problém, výzkumné otázky a navrženy hypotézy. Po sestavení metodiky dle Hendla bylo představeno sestavení SWOT analýz na zhodnocení současného stavu firmy XY a aktuálního fungování projektového managementu a průběh kvalitativního výzkumu ve formě polostrukturovaných rozhovorů. U obou metod byly nastíněny vyskytnuté limity.

V praktické části byla anonymně představena firma XY, organizační struktura, náplně pracovních pozic respondentů a řízení vývoje dle firemní metodiky. Poté byla rozebrána SWOT analýza firmy XY a jejich silných, slabých stránek, vnějších příležitostí a hrozeb. Data do analýzy byly sesbírány formou rozhovoru s majitelem a následně doplněny z poznámek polostrukturovaných rozhovorů. V další podkapitole byl představeno současné fungování projektového managementu a zanalyzována data z kvalitativního výzkumu pomocí otevřeného kódování. Z analýzy rozhovorů vznikly kódy: profily respondentů, plánování Sprintu, průběh Sprintu, odhady práce, pravidelné firemní schůzky, kontakt s klienty, firemní kultura a navrhované změny. Následně byla sestavena druhá SWOT analýza, která identifikovala silné stránky, slabé stránky, příležitosti a hrozby projektového managementu a vytvořila tak kompletní obraz fungování projektů.

Cílem SWOT analýz a kvalitativního výzkumu bylo odpovědět na výzkumný problém, výzkumné otázky a potvrdit či vyvrátit hypotézy.

Byl stanoven výzkumný problém: Ve firmě XY byly v průběhu IT projektů zjištěny opakované špatné časové odhady, které mají za následek prodlužování Sprintů, finanční dopad na firmu a na možné zpoždění dodání zákazníkovi.

Na základě problému byly vytvořeny výzkumné otázky:

- VO1: Jak aktuálně funguje projektový management?
- VO2: Jak aktuálně probíhá plánování Sprintu?
- VO3: Jakým způsobem probíhají časové odhady IT projektů?
- VO4: Z jakých důvodů jsou špatně časově odhadovány požadavky v IT projektech?
- VO5: Proč dochází k prodlužování Sprintů?
- VO6: Jaká opatření přijmout pro přesnější časové odhady?

A poté zformulovány hypotézy:

- Obecná hypotéza 1, která byla potvrzena: Časové odhady IT projektů dělají nesprávní pracovníci.
- Obecná hypotéza 2, která byla potvrzena: Při časových odhadech IT projektů dochází k nedostatečné komunikaci mezi klíčovými pracovníky.
- Obecná hypotéza 3, která byla potvrzena: Ve firmě XY se nekonají agilní ceremonie.

V návaznosti na získaná data a jejich vyhodnocení, byl sestaven implementační plán agilní transformace:

- 1) Firma XY si ujasní, **proč** chce přejít na agilní metody a co se jí díky nimlepší. Provede **analýzu současného stavu** a zkontroluje tak **připravenost** na agilní transformaci.
- 2) Určí se **plán** transformace, její krátkodobé a dlouhodobé cíle, jaké obsahuje kroky. Zároveň s plánem definuje **vize** agilní transformace, firmy a produktu.
- 3) Firma XY si vybere **agilní metodu**, která je pro ni vhodná.
- 4) Díky dané agilní metodě se definují **nové týmové role** a nově se postaví **vývojový tým**.
- 5) Na základě agilních metody se zavedou pravidelné schůzky dle Scrumu.
- 6) Vedení s agilním expertem předpřipraví Sprint Backlog na Sprint Planning.
- 7) Naplánuje se **pilotní Sprint** s možností ho upravovat.
- 8) **Tým se seznámí** s agilní změnou, pilotním Sprintem a je proškolen.
- 9) Proběhne pilotní Sprint.
- 10) V návaznosti na vyhodnocení Sprintu proběhne postupné a úplné zavedení agility.

V rámci jednotlivých kroků bylo určeno, jaký bude jejich průběh a jakých pracovníků se bude pilotní Sprint týkat.

Autorka vybrala pro firmu XY zavedení Scrum a Kanban (Scrumban) a podle metod i zavedení týmových rolí, artefaktů a scrumových ceremonií. Je kladen důraz na kontrolování průběhu pilotního Sprintu a plnění agilních ceremonií.

Poslední kapitola praktické části diplomové práce tvoří shrnutí a návrhy autorčinných doporučení. Vzhledem k problémům s nestálou délkou Sprintů, neúplnými zadáními požadavků, nedostatečnou dokumentací, špatným časových odhadů, určování priorit požadavků a zapojením klientů do procesu bylo navrženo: zkrácení Sprintů; přesnější podoba zadání a nový systém v jeho tvoření, určování časových odhadů a priorit; určení přesnější podoby dokumentace; metody Planning Poker, T-shirt sizing a Větší – menší pro časové odhady a efektivnějšího zapojení klientů.

Na závěr autorka doporučila správného načasování pilotního Sprintu, nábor seniorního analytika a Scrum Mastera, najmutí agilního experta z externí společnosti a navrhla tři agentury, které splňují záměr agilní transformace.

Literatura

Monografie

BŘEZINA, A. *Agilní transformace: Proč bývá tak křehká?*. 1. vyd. České Budějovice: Kopp, 2020. 216 s. ISBN 978-80-7232-521-4.

DOLEŽAL, J. *Agilní přístupy vývoje produktu a řízení projektu: komplexně, prakticky a dle světové praxe*. 1. vyd. Praha: Grada, 2022. 192 s. ISBN 978-80-271-3705-3.

DOLEŽAL, J., KRÁTKÝ J. *Projektový management v praxi: Naučte se řídit projekty!*. 1. vyd. Praha: Grada Publishing, 2017. ISBN 978-80-271-9495-7.

DOLEŽAL, J., KRÁTKÝ, J., HÁJEK, M., LACKO, B., CINGL, O., Ježková Z. *Projektový management: Komplexně, prakticky a podle světových standardů*. 2. vyd. Praha: Grada Publishing, 2023. 432 s. ISBN 978-80-271-6724-1.

HENDL, J. *Kvalitativní výzkum: Základní teorie, metody a aplikace*. 5. vyd. Praha: Portál, 2016. 494 s. ISBN 978-80-262-1968-2.

ŠOCHOVÁ, Z., KUNCE, E. *Agilní metody řízení projektů*. 2. vyd. Brno: Computer Press, 2019. 224 s. ISBN 978-80-251-4961-4.

Odborné knihy a časopisy

ANANTATMULA, V., KLOPPENBORG, T. *Be Agile Do Agile*. 1. vyd. USA: Business Expert Press, 2021. 218 p. ISBN 978-1-95334-995-8.

ASHMORE, S., RUNYAN, K. *Introduction to Agile Methods*. 1. vyd. USA: Addison-Wesley Professional, 2014. 336 p. ISBN 978-0-321-92956-3.

BLAIR, A. *Agile Project Delivery: A Practical Approach for Corporate Environments Beyond Software Development*. 1. vyd. Kanada: Canadian Scholars, 2020. 290 p. ISBN 978-1-773-38215-9.

CALDWELL, C. *Agile Project Management: The Complete Guide for Beginners to Scrum, Agile Project Management, and Software Development*. 1. vyd. USA: Alakai Publishing LLC, 2019. 152 p. ISBN 979-8201889630.

CATLIN, R., WATKINS, C. *Agile Auditing: Fundamentals and Applications*. 1. vyd. Kanada: John Wiley & Sons, 2021. 336 p. ISBN 978-1-119-69348-2.

COLE, R. *Agile Now: Your quick start introduction to agile*. 1. vyd. Velká Británie: Pearson Education Limited, 2020. 160 p. ISBN 978-1-292-23200-0.

DALTON, J. *Great Big Agile: As OS for Agile Leaders*. 1. vyd. USA: Apress, 2018. 329 p. ISBN 978-1-4842-4206-3.

DENNING, S. *The Age of Agile: How Smart Companies Are Transforming the Way Work Gets Done*. 1. vyd. USA: AMACOM, 2018. 336 p. ISBN 978-0-81443-910-4.

GOODPASTURE, J. *Project Management the Agile Way: Making it Work in the Enterprise*, 2. vyd. USA: J. Ross Publishing, 2015. 392 p. ISBN 978-1-604-27115-7.

HEATH, F. *The Professional Scrum Master (PSM I) Guide: Successfully practice Scrum with real-world projects and achieve your PSM I certification with confidence*. 1. vyd. Velká Británie: Packt Publishing, 2021. 174 p. ISBN 978-1-80020-556-7.

HOODA, S., SOOD, V., SINGH, Y., DALAL, S., SOOD, M. *Agile Software Development: Trends, Challenges and Applications*. 1. vyd. USA: John Wiley & Sons, 2023. 365 p. ISBN 978-1119896838.

HUGHES, B., IRELAND, R., WEST, B., SMITH, N., SHEPHERD, D. *Project Management for IT-Related Projects*. 3. vyd. Velká Británie: BCS Learning & Development, 2019. 162 p. ISBN 9781780174853.

HUNT, J. *PMI-ACP Project Management Institute Agile Certified Practitioner Exam Study Guide*. 1. vyd. Kanada: Sybex, 2018. 408 p. ISBN 978-1-119-43463-4.

IGBERAESE, D. *Introduction to Project Management: A Source Book for Traditional PM Basics*. 1. vyd. USA: Routledge, 2022. 288 p. ISBN 978-1-032-13737-7.

KANDHA, U. *Essentials of Agile Transformation*. 1. vyd. Indie: Notion Press, 2018. 120 p. ISBN 978-1-64249-993-3.

KANE, R. *SAFe 6.0: Your Roadmap for Agile Scaling*. Německo: epubli, 2023. 81 p. ISBN 978-3757568726.

KELLY, A. *The Art of Agile Product Ownership: A Guide for Product Managers, Business Analysts, and Entrepreneurs*. 1. vyd. Velká Británie: Apress, 2019. 168 p. ISBN 978-1-4842-5168-3.

KNASTER, R., LEFFINGWELL, D. *SAFe Distilled – Applying the scaled agile framework for lean enterprises*. 2. vyd. USA: Addison Wesley, 2019. 315 p. ISBN 978-0-13-517049-6.

LINDSAY, G. *Clearly Agile – A Leadership Guide to Business Agility*. 1. vyd. Velká Británie: Practical Inspiration Publishing. 2024. 296 p. ISBN 978-1788605502.

LUNDBERG, A. *Successful with the Agile Spotify Framework: Squads, Tribes and Chapters - The Next Step After Scrum and Kanban?*. 1. vyd. Německo: Books on Demand, 2020. 58 p. ISBN 978-3-7519-8137-8.

MACNEIL, D., CADER, A. *Scaling Agile with Jira Align: A practical guide to strategically scaling agile across teams, programs, and portfolios in enterprises*. 1. vyd. Velká Británie: Packt Publishing, 2020. 326 p. ISBN 978-1-80020-321-1.

MARTIN, R. *Clean Agile: Back to Basics*. 1. vyd. USA: Pearson, 2019. 240 p. ISBN 978-0135782002.

MEYER, B. *Agile!: The Good, the Hype and the Ugly*. 1. vyd. Švýcarsko: Springer, 2014. 170 p. ISBN 978-3-319-05155-0.

MORAN, A. *Managing Agile – Strategy, Implementation, Organisation and People*. 1. vyd. Švýcarsko: Springer International Publishing, 2015. ISBN 978-3-319-16262-1.

MOREIRA, M. *The Agile Enterprise: Building and Running Agile Organizations*. 1. vyd. USA: Apress, 2017. 280 p. ISBN 978-1-4842-2391-8.

MUNDRA, S. *Enterprise Agility: Being Agile in a Changing World*. 1. vyd. Velká Británie: Packt Publishing, 2018. 490 p. ISBN 978-1-78899-064-6.

POWER, D., HEAVIN, C. *On Becoming Agile*. 1. vyd. USA: Business Expert Press, 2021. 110 p. ISBN 978-1-63742-009-6.

Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. 6. vyd. USA: Project Management Institute, 2017. 756 p. ISBN 978-1-62825-391-7.

Project Management Institute. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. 7. vyd. USA: Project Management Institute, 2021. 370 p. ISBN 978-1-62825-667-3.

RIGBY, D., ELK, S., BEREZ, S. *Doing Agile Right: Transformation Without Chaos*. Boston: Harvard Business Review Press, 2020. 252 p. ISBN 978-1-63369-870-3.

RUBIO, M. *The Mini Book of Agile: Everything you really need to know about Agile, Agile Project Management and Agile Delivery*. 1. vyd. Packt Publishing, 2022. 50 p. ISBN 978-1-8032-3871-5.

SPAYD, M., MADORE, M. *Agile Transformation: Using the Integral Agile Transformation Framework to Think and Lead Differently*. 1. vyd. USA: Addison-Wesley Professional, 2020. 256 p. ISBN 978-0133137392.

STELLMAN, A., GREENE, J. *Learning agile: Understanding Scrum, XP, Lean, and Kanban*. 1. vyd. USA: O'Reilly Media, 2015. 417 p. ISBN 978-1-449-33192-4.

STERN, T. *Lean and Agile Project Management: How to Make Any Project Better, Faster, and More Cost Effective*. 1. vyd. USA: Taylor & Francis, 2017. 354 p. ISBN 978-1-4987-3916-0.

TYLL, L. *Podniková strategie*. 1. vyd. Praha: C. H. Beck, 2014. 275 s. ISBN 978-80-7400-507-7.

VEBER, J. a kol. *MANAGEMENT: Základy, přístupy, soudobé trendy*. 1. vyd. Praha: Ekopress, s.r.o., 2021. 415 s. ISBN 978-80-87865-69-9.

WALDOCK, B. *Being Agile in Business: Discover Faster, Smarter, Leaner Ways to Work*. 1. vyd. Velká Británie: Pearson Education Limited, 2015. 200 p. ISBN 978-1-292-08372-8.

WONG, M. *Corporate Agility: insights on agile practices for adaptive, collaborative, rapid, and transparent enterprises*. Hoboken: John Wiley & Sons, 2020. 256 p. ISBN 978-1119652267.

WRIGHT, C. *Agile Project Management, Assurance and Auditing: A practical guide for auditors, reviewers and project teams*. 1. vyd. Velká Británie: It Governance Publishing, 2022. 129 p. ISBN 978-1-78778-357-7.

WYSOCKI, R. *Effective Project Management: Traditional, Agile, Extreme, Hybrid*. 8. vyd. USA: Wiley, 2019. 656 p. ISBN 978-1-119-56278-8.

Internetové zdroje

Agile Transformation. *Deloitte.com* [online]. 2024 [cit. 2024-04-30] Dostupný z WWW: <https://www2.deloitte.com/cz/cs/pages/risk/solutions/agile-transformation.html>

Agile Transformation. *Integratedconsulting.cz* [online]. 2024 [cit. 2024-04-30] Dostupný z WWW: https://www.integratedconsulting.cz/poradenstvi/rozvoj-lidi-a-organizace/agile-organization/?gad_source=1&gclid=Cj0KCQjw0MexBhD3ARIsAEI3WHKbkTHduP1RYAyU0Hv6XcePUagH-AonyabQoq-ULT2nPPFeULWZAMMaAp6AEALw_wcB

Manifesto for Agile Software Development [online]. 2001 [cit. 2024-03-11]. Dostupný z WWW: <https://agilemanifesto.org/principles.html>

ŠOCHOVÁ, Z. Agilní transformace a enterprise coaching. *Sochova.cz* [online]. 2024 [cit. 2024-04-30]. Dostupný z WWW: <https://sochova.cz/agilni-transformace-a-enterprise-coaching.htm>

Seznam příloh

Příloha 1 Scénář k rozhovorům	I
Příloha 2 Poznámky z rozhovoru s majitelem	IV
Příloha 3 Poznámky z rozhovoru se COO	VII
Příloha 4 Poznámky z rozhovoru s vedoucím vývoje	X
Příloha 5 Poznámky z rozhovoru s vývojářem	XV
Příloha 6 Poznámky z rozhovoru s testerem	XX

Přílohy

Příloha 1 Scénář k rozhovorům

Jaká je Vaše role v IT projektech? Zodpovědnost / úkoly / ideálně / co byste změnil

S jakými problémy/výzvami se v práci potýkáte? Co by je pomohlo vyřešit?

Jaké je podle Vás role majitele v IT projektech? Jaká by podle Vás měla ideálně být?

Jaké máte **pravidelné schůzky** Vývoje?

- jak často se provádí agilní ceremonie Vývoje? (standup, retrospektiva, review, refinement, sprint planning / proč se některé ceremonie neprovádí)
- co všechno se na nich děje? Detaily
- předem daná agenda
- kdo všechno se jich účastní
- Lessons learned v rámci retrospektiv nebo na konci každého projektu

Sprinty

- Co podle Vás znamená Sprint? Jak si ho představujete?
- (Jakým způsobem jsou definovány Sprinty?)
- Jak jsou dlouhé?
- Máte fixní délku Sprintů?
- Proč jsou Sprinty tak dlouhé?
- Jaké jsou výstupy v rámci každého Sprintu? Jsou definovány?
- Jakým způsobem jsou Sprinty plánovány?
- Mají všechny úkoly/storky stejnou prioritu? Podle čeho se řídí jejich priorita?
- Jaký je Váš pohled na množství práce, které vývojáři/testeři udělají za 1 sprint?
- Kolik toho stihnete za 1 sprint?
- Kontrolujete zaměstnance / Sprinty průběžně?
- Jak monitorujete/kontrolujete progres?
- Kdy zjistíte, že zaměstnanci nestíhají (v důsledku špatného odhadu plánování)?
- Jakým způsobem jsou hodnoceny výsledky za uběhlý Sprint?
- Popište typický průběh Sprint (detailně) - plánování, změny, jak moc je zapojen, je tam něco fixního/flexibilního
- Co byste na nich změnil?

Estimace

- Jaký je Váš pohled na současnou estimaci?
- Jak podle Vás dochází ke špatné estimaci od pracovníků?
- Proč k ní dochází? / Proč to podle Vás špatně odhadují / Kde je chyba?
- Proč zaměstnanci Vývoje nechtějí převzít odpovědnost za špatnou estimaci?
- Jak se u zaměstnanců Vývoje projevuje nerespekt termínů?
- Jaké jsou hlavní překážky, které zaměstnanci mají při odhadování?
- Odhadují to zaměstnanci Vývoje sami nebo mají pomoc? / Odhadují časovou náročnost projektu kvalifikovaní / správní lidé?
- Jak často zaměstnanci špatně odhadují? Při jakých projektech – společný rys?
- Jaká je strategie při zjištění špatného odhadu? / Jaká je obvyklá reakce na špatný odhad?

- Kdy se zjistí / v jaké fázi projektu se zjistí, že se byla uskutečněna špatná estimace?
- Má špatná estimace vliv na kvalitu?
- Využívají se data z minulých projektů ke zlepšení estimace?
- Zkoušeli jste estimaci již nějak zlepšit?
- Co by Vám pomohlo ke zlepšení estimace?
- Ovlivňuje / jak ovlivňuje volnost pracovníků dodržování termínů?

Projektový management

- (Jak moc zapojujete zákazníka do procesu?)
- (Jaká je současná úroveň povědomí a zkušenosti týmu vývoje ohledně agilních metodik?)
- Jaký je Váš pohled na současné řízení vývoje? / fungování Sprintů
- Jak by se podle Vás měl ideálně zapojovat klient do procesu?
- Jak by podle Vás měl ve firmě ideálně fungovat agilní projektový management?
- Jaké máte zkušenosti s agilní metodikou? Jaký na ni máte názor?

Změny (transformace)

- (Jaká máte očekávání ohledně změny na agilní přístup? / očekávání v dlouhodobém měřítku)
- Jaké změny byste zavedl, aby Vývoj / řízení projektů fungovalo správně? / Co všechno byste byl ochoten změnit, aby firma fungovala lépe?
- Jaké máte představy o tom, jak by měla probíhat implementace řešení / agilní transformace?
- Jaké by bylo Vaše zapojení po zavedení změn? / Jak si představujete Vaše ideální zapojení ve vývoji / agilním projektovém managementu?
- Jaká jsou rizika v rámci agilní transformace? Jak je eliminovat?
- Jak byste motivoval týmy, aby na změny přešly ...
- Jaká je Vaše vize fungování po změnách?
- Jaký vliv budou mít změny na agilní pracovní prostředí, podnikatelské cíle? + Jaký vliv měly změny ve fungování Sprintů na doručování / cíle atd.?

Pracovní prostředí / kultura

- Jak byste popsal pracovní prostředí ve firmě? / Jak vnímáte atmosféru a pracovní prostředí?
- Jak byste popsal současnou firemní kulturu? (co byste na ní změnil / proč / co postrádáte)
- Jaké jsou její základní hodnoty/principy? Jak se tyto hodnoty odrážejí v každodenní práci zaměstnanců?
- Jak funguje komunikace a spolupráce ve firmě (otevřené diskuse, podpora sdílení nápadů, respektování rozmanitosti nápadů)? Jaký je Váš názor na komunikaci a spolupráci ve firmě?
- Jaký je styl vedení (hierarchický, participativní)?
- Jak se management podílí na podporování firemní kultury?
- Jaká je péče o zaměstnance (benefity, rozvoj, work-life balance)?
- Co myslíte o současné motivaci?
- úroveň zájmu a podpory vedení firmy

- Jaká je Vaše zkušenost s managementem a vedoucími ve firmě?
- Jaká je Vaše důvěra v management a vedoucí?
- Jak vnímáte Vaši úroveň autonomie a svobody v práci?
- Jak byste si představoval firemní kulturu do budoucna?

Obecné

- Co podle Vás společnost potřebuje?
- Co by Vám pomohlo při Vaší práci?
- Když byste byl CEO/majitel, co byste změnil?

Příloha 2 Poznámky z rozhovoru s majitelem

*Jaká je Vaše **role** v IT projektech / ve firmě?*

- Role CEO je spíše formálního typu.
- Dělá strategická a business rozhodnutí. Pod majitelem se jeho rozhodnutí formují dál. Udává high level směr firmy a oddělení (obchod, podpora a vývoj).
- Nejčastěji deleguje činnosti na COO a vedoucího obchodu a marketingu.
- Když se všechny jeho činnosti shrnou, pracuje reálně 1 až 2 dny v týdnu (pondělí, pátek). Občas i více, pokud se něco mimořádného / kritického.
- V rámci firmy se stále stará o mzdy, finance a administrativu. Současně se nezapojuje do mikromanagementu.
- Prochází měsíční reporty od každého oddělení a zaměstnance. Reporty obsahují informace, co se za daný měsíc událo, co se plánuje a jaké nastaly problémy, které je potřeba vyřešit. Když zjistí, že je něco špatně, tak sám převezme iniciativu a řeší dané problémy.

*Co pro Vás znamená **Sprint**?*

- Znamená to Release.

Jak jsou dlouhé? Máte fixní délku Sprintů?

- Každý Sprint / Release trvá 3 měsíce. Tedy za rok máme 4. Dá se říct, že se snažíme, aby byly fixní, ale v rámci zpoždění nejsou pokaždé stejně dlouhé.

Jaká je Vaše role ve Sprintech?

- Majitel schvaluje, co se dá do Sprintů – má poslední slovo. Ale aktuálně obsah Sprintu schvaluje také vedoucí obchodu a marketingu.
- Majitel by rád zavedl každých 14 dnů schůzku s vedoucím vývoje a vedoucím obchodu a marketingu, aby s nimi probral průběh aktuálního Sprintu.

Jakým způsobem jsou plánovány Sprints? Jakou roli v plánování máte? Mají všechny požadavky stejnou prioritu? Podle čeho se řídí jejich priorita?

- Priority požadavků určuje, když cítí potřebu, například když jsou některé požadavky businessové. Tzn. když přišel požadavek strategicky důležitý klient nebo když se jedná o požadavek / funkci, která je na vázaná na více zákazníků nebo chyby v softwaru, které mohou způsobit velké škody.

Kontrolujete požadavky nebo zaměstnance v průběhu Sprintu?

- Majitel se často kouká, jaký je progres požadavků v programu JIRA. Kontroluje analýzy z produkčního prostředí a co tam klienti dělají.

Jakým způsobem je vyhodnocován uplynulý Sprint?

- Konec Sprintu majitel sám vyhodnocuje. Sleduje tzv. hotfixy a reporty pracovníků a reporty pro klienty. Do budoucna by do vyhodnocování rád zapojil vedoucího obchodu a marketingu.

Jak vypadá typický průběh Sprintu?

- Většinou cca 2 dny (jednotky dnů) před koncem Release dostane informaci, že se pravděpodobně prodlouží, aby se stihli všechny požadavky. Ať už se jedná o Release na testování nebo produkci.

Co byste rád změnil na Sprintech?

- Majitel by rád změnil komunikaci na klienty a dokumentaci.
- Dokumentace je potřeba ze 2 úhlů: zaznamenávacího a regulačního.
 - o Regulační dokumentace by měla obsahovat to, co je důležité. S tím je spojená i prioritizace požadavků – požadavek je důležitější, protože to požaduje regulace nebo audit.

Co naopak funguje na Sprintech?

- Release se dokončí.
- Požadavky, které se dokončí v rámci Release, mají velmi nízkou chybovost, malé množství chyb a hotfixů. Domnívá se, že je to i kvůli důležitosti správnosti / bezchybnosti softwaru.

Jaké máte pravidelné schůzky?

- Jednou za týden má majitel schůzku se COO a vedoucím OaM.
- Každý pátek se účastní vývojové schůzky a výzkumné schůzky.
 - o Na každou schůzku má připravené body k prodiskutování.
 - o Na vývojové schůzce má spíš pasivní / poslouchací účast.
 - o Na výzkumné schůzce je více aktivní.
- Jednou za kvartál má celá firma osobní celodenní schůzku.
- Jednou měsíčně je schůzka, kde prezentuje, v jakém je firma stavu a ukazují se souhrnné výsledky za každé oddělení.

*Proč podle Vás dochází ke špatné **estimaci**? Jak k ní dochází? Proč zaměstnanci nechtějí převzít odpovědnost za špatnou estimaci? Odhadují to sami zaměstnanci nebo to někdo dělá za ně?*

- Protože se pracovníci bojí odhadů nebo jsou líní je dělat.
- Párkrát jsme to už zkoušeli a nefungovalo to.
- Majitel to za ně dělal 3 roky nazpět a aktuálně by to měli dělat sami. Je možné, že odhady dělají, ale nepišou je k požadavkům.
- Majitel by chtěl docílit toho, aby si požadavky naceňovali sami a správně / relevantně.
- S vedoucím obchodu a marketingu se majitel snaží dojednat zodpovědnost za odhady (výhledově / v nejbližší době).
- Pracovníci dělají špatné odhady, protože nepoznají komplexnost požadavku.

*V jakém kontaktu je s **klientem**?*

- Majitel je v nepřímém kontaktu s klienty prostřednictvím požadavků.
- Rád by je více zapojil, ale je to komplikované vzhledem k jejich přístupu.

*Jaká je Vaše **důvěra** v podřízené?*

- Majitel má plnou důvěru v COO, vedoucího vývoje a vedoucího obchodu a marketingu.

*Když byste byl majitel / CEO, **co byste změnil**?*

- Majitel by rád nahradil sám sebe ze 100 % - něco ve stylu provozního ředitele.
- Chtěl by docílit toho, že bude udávat pouze strategický směr a dělat strategická rozhodnutí. Ze své „nové“ pozice by dostával pouze reporty, aby kontroloval realitu.
- Majiteli vadí, že týmy nejsou schopny finálního rozhodnutí bez něho.

Příloha 3 Poznámky z rozhovoru se COO

*Jaká je Vaše **role** ve firmě?*

- COO se aktuálně stará o všechny provozní věci, které se týkají firmy – kromě finančního řízení a strategických rozhodnutí, která se týkají vývoj.
- Další část práce je certifikace a chod firmy po administrativní stránce.
- Vede oddělení „sales support“ – včasné vystavování faktur, jejich vymáhání.
- Má na starosti holandský trh.

*Jak jste do **vývoje** zapojená?*

- Okrajově zapojená
- Certifikace – s kolegy z vývoje řeší, když je něco potřeba popsat do technické složky.
- Pokud má požadavek (normy), tak s vývojem komunikuje a ve chvíli, kdy se shodnou, tak se to zapracuje. Ale nezadáva vývojové požadavky.
- Má přístup do programu JIRA a do dalších vývojových nástrojů. Přístupy má proto, aby si mohla vytáhnout potřebné věci pro audit nebo aby zadala nějaký úkol na vývoj (např. změna všeobecných podmínek).

Jak se Vám pracuje s vývojem, když třeba potřebujete něco zadat v rámci dokumentace?

- Pracuje se s nimi dobře, tým je stabilní a všichni se znají dlouho. Díky tomu spolu navzájem umí komunikovat.
- Komunikují spolu na Slacku, emailu.
- Když se jedná o něco akutního nebo většího, tak si nejdřív zavolají a poté požadavek zadá do Jiry. Zjišťuje, zda se vejde do Sprintu nebo zda to půjde do dalšího Sprintu. Pokud je to něco hodně velkého, tak to řeší s majitelem a ten to na dev callech řeší dál a přidá do plánu.

Podle čeho hodnotíte, zda je požadavek velký? Nebo to hodnotí oni?

- Aby se určila velikost, tak si zavolají a rozhodnou společně.
- Může se stát, že u požadavku je termín, pak to řeší s vedoucím vývoje.
- Pokud je to něco hodně důležitého a akutního, tak zadává termín v Jire a ohlídá si to. Termíny zadává s týdenní rezervou, když by se tam vyskytl nějaký problém (ideálně).
- Občas se stane, že požadavky, které zadá, nemají takovou prioritu jako si myslí.
- Ale nikdy se nestalo, že by požadavek nesplnili.

*Jak vnímáte **role ostatních lidí ve vývoji**? Ať už vedoucího vývoje nebo dalších zaměstnanců.*

- Vnímá je pozitivně, přichází s nimi minimálně do styku. V profesionálním kontaktu z vývoje je pouze s vedoucím vývojem.

Majitel vnímá nerespekt termínů a absenci pevné ruky. Jak to vnímáte Vy?

- Vnímá to stejně, ale z doslechu.
- Bavili se o tom s externě najatým procesním manažerem a ten by s tím rád něco udělal, zavedl pevnější ruku – lepší dodržování termínů.
- Problém není, že by se požadavky neplnily, ale když si dají termín na Release, tak se posouvá. Ale tím jak se neúčastní vývoje, tak není možné říct proč.
- Z profesního hlediska sepsala náplně jednotlivých pracovníků a jaké jsou procesy – konkrétně sepsala, kdo co dělá, s kým kdo komunikuje. Od toho si slibují zjednodušení,

zrychlení procesů. Aby všichni nepracovali na jednom požadavku, což se dělo v obchodě. Aktuálně zde probíhá změna procesů.

Jakou úlohu zastává externí manažer?

- Byl najatý jako externista, aby pomohl stabilizovat obchod a svým způsobem ve firmě uklidit. Protože se ve firmě znají všichni dlouho, tak se to dá považovat i za stinnou stránku. Proto majitel přišel s tímhle nápadem a oddělení obchodu by se tak probralo.
- Obecně externí manažer měl nastavit konkrétní struktury, dodržování pravidel – aby se obchod mohl věnovat pouze novým obchodům a nezatěžovala ho administrativa.
- Pomohlo to a nakoplo to, že přišel někdo zvenku.
- Externista je vidět v organizační struktuře jako vedoucí OaM. Má pod sebou interní a externí obchod.

Jaký je rozdíl mezi technickým a prodejním supportem?

- Prodejní support se stará o stávající zákazníky.
- Technický support řeší technické dotazy, například nefunguje tlačítko.

Jaké máte pravidelné schůzky?

- Schůzky jednou týdně v obchodu. Obchodníci mají mít připravený report – co se změnilo, report forecastu, jak se změnil rozdělané úkoly, co se plánuje, kolik má domluvených školení atd.
- Zapojí se do nich i sales support (2 lidi), který pomocí umělé inteligence vyhledává studené kontakty. Ale ukazuje se, že je malá odezva. Probírají například jakým způsobem se posunuly fakturace (kolik je faktur posplatnosti, co se zaplatilo, co se nezaplatilo, proč se neposouvají) nebo zda se vyskytl problém, který je potřeba eskalovat směrem nahoru.
- Hlavním úkolem sales supportu je starání o interní obchody a administrativu spojenou s péčí o stávající zákazníky.
- Výhledově by měly tyto schůzky jednou za 14 dní. Je to výhledově, protože téměř každý měsíc se najde věc, která by se dala zlepšit (zjednodušení, zapojení dalších lidí).
 - o Proč 14 dní? Je to časově náročné – každý týden, 5 lidí, 2 hodiny. Ukazují se v podstatě ty samé tabulky, s menšími rozdíly. Takže dokud si procesy nesednou, tak to bude jednou za týden. Reporting (zpracovávání dat) bude probíhat každý týden. Během 2 týdnů se toho víc stane (klient zaplatil atd.).

Eskalace / vymáhání faktur

- Moc se neeskaluje, protože se snaží vymáhání vyřešit komunikací.
- V rámci prozákaznického přístupu a specifikům produktu se nelze chovat stejně jako u klasických licencí, tzn. když zákazník neplatí, tak jim zrušit licenci. Klienti doopravdy potřebují tento produkt k tomu, aby mohli vykonávat svou práci.
- U některých klientů vystavují faktury automaticky.
- Někdy se stává, že ji faktur klient nevšimne a ozve se po 4 měsících, že tady má nějakou fakturu a neví, co s ní.
- Pro firmu je důležité, aby za produkt zaplatili a pro klienty je důležité používání.
- Řešení je komunikace a rozšiřování posílání faktur/komunikace na více lidí (kopie v emailech) + komunikace v jejich rodném jazyce (například klienti ze Španělska reagují lépe na komunikace ve španělštině než v angličtině).

- „Vyhrožování“ nebo informace, že klientovi skončí licence moc nefungovalo.
- Nově se zavedlo, že se klienta zeptají, zda si přeje prodlužovat. Když se neozve, tak za 1-2 týdny, zda je všechno v pořádku, zda všechno funguje.
- Do CRM zapisují více poznámky ke klientům, jak platí faktury, kdy je dobré se ozývat.

Zákazník – kontakt

- Se stávajícími klienty ano.
- Ale s úplně novými klienty ne.

Jak byste popsala pracovní prostředí?

- Přátelské, pozitivní prostředí (v kanceláři).
- Profesionální komunikace, na úrovni. Vychází si navzájem vstříc. Když se vyskytne problém, tak jsou společně schopni ho hned vykomunikovat.
- Lidé z vývoje pracují vzdáleně.
- Každý může svobodně vyjádřit svůj názor, není to diktatura.

Jaké jsou silné stránky firmy?

- Firma je hodně prozákaznická a má individuální přístup k zákazníkům (opravy faktur, více školení pro jednoho zákazníka). (nejsilnější stránka)
- Kdykoliv, když zákazník zavolá, tak to obchodník zvedne.
- Skvělý certifikovaný produkt
- Produkt/firma se drží mezi TOP4 na světě podle nezávislých průzkumů – hodnotí support, cenu, co všechno produkt umí apod.
- Mezi top 4 je firma jediná, která má daný typ certifikace.
- Zákaznický servis taky funguje dobře, je stále na telefonu. A i když není, tak rychle reaguje ve dne, noci, protože firma má zákazníky z různých časových pásem.

Jaké jsou slabé stránky?

- Firma nemá 24/7 support.
- Malý tým. Dovede si představit, že by nabrali cca 2 lidi.
- Rychle se projeví, když více zákazníků neplatí včas. Je těžké to predikovat.
- U 20 % faktur po splatnosti je nutná častější komunikace.
- Rozšíření týmu pomohlo s vymáháním faktur.

Příloha 4 Poznámky z rozhovoru s vedoucím vývoje

pozn. Vedoucí vývoje nazývá Sprint Releasem.

Jaká je Vaše role v IT projektech?

- Role vedoucí vývoje je vývoj aplikace – je zodpovědný za vývoj backendu online aplikace.
- Tvoří zadání frontendu. Zodpovídá dotazy frontendu a řeší věci z hlediska uživatelského rozhraní nebo toho, jak to má vypadat.
- Na backendu online aplikace je sám a desktopovou aplikaci dělá člověk z Nizozemska.

Z knihoven se používají části na desktopovou a online aplikaci – je to společná část. Díky tomu se obě aplikace v některých částech shodují, ale stále jsou mezi nimi rozdíly.

Desktopová aplikace si natahuje funkcionality a pamatuje si jejich nastavení a nemusí to pamatovat někdo jiný a snadněji to pak vyhodnocuje.

Online aplikace není schopna si nic vypočítat z frontendu a musí si sáhnout do backendu. Má rozdílný přístup.

Z pohledu zákazníka mají různé verze vliv – v online aplikaci mohou být některé věci snadnější. V desktopové aplikaci se dá i víc vytvářet, ale využívají to zhruba 1 % klientů.)

*Jakým způsobem jsou **plánovány Sprints**? Máte fixní délku Sprintů? Jaké jsou vstupy v rámci každého Sprintu? Podle čeho plánujete Release? Kolik požadavků stihnete za 1 Sprint?*

- Vedoucí vývoje plánuje obsahy Releasů – co se v daném Releasu bude dělat.
- Firma nefunguje tak, že by měla 14denní Sprints.
- Cíl je mít 4 Releasy za rok, tzn. 1 Release / Sprint trvá 3 měsíce.
- Jsou zde vstupy od supportu, vedoucího supportu, testera (s přesahem do UX), každý z týmu Development a vstupy z desktopové aplikace. Je zde snaha přebírat věci z desktopové aplikace do online aplikace – co je hotové v desktopu, tak je vstupem pro online aplikaci. (Desktopovou aplikaci vyvíjí člověk v Holandska.)
- Tým Developmentu má zároveň prostor na nápady, mohou být jakékoliv.
- Do jednoho Realesu se vejde přibližně 100 požadavků. Během Releasu se doplňují další požadavky klientů. Jsou to požadavky, které by nepočkaly a na základě domluvy s vedoucím supportem se doplní.
- Ohledně doplňování nejsou žádné pravidelné schůzky, probíhá náhodná komunikace s vedoucím supportem. A domluví se společně, co to je, jak je náročný na vývoj a zda jsou schopni to přidat do Releasu.

*Jak vypadá **Backlog**? Podle čeho vybíráte požadavky do Releasů? Mají všechny úkoly/storky stejnou prioritu? Podle čeho se řídí jejich priorita?*

- Backlog jsou sepsané požadavky a z něho se vybírá práce do Releasů.
- Vedoucí vývoje je vybírá na základě domluvy s vedoucím supportem a poté podle svého odhadu / pocitu. Aby byla vyvážená práce pro backend i frontend. Podle priorit také vybírá, ale není to hlavní – nějakým způsobem to zreviduje. Priority u požadavků jsou řešené tak, že ten, kdo požadavek zadá, k němu napíše prioritu. Vedoucí vnímá, že když v Backlogu jsou požadavky od různých lidí, tak každý má priority jinde.
- Poté ještě požadavky konzultuje s vedoucím supportem, aby se vyjasnilo, co je urgentní a co ne.
- Priority jsou na základě důležitosti nebo urgencye.

- Požadavky, které jsou důležité a mají dopad na produkt, tak vedoucí vývoje řeší jednou za rok s majitelem – důležité cíle za rok, čeho chce dosáhnout. Schůzka nemá fixní datum.

Na základě čeho děláte odhady požadavků?

- Vedoucí vývoje dělá odhady podle pocitu nebo podle priorit, urgency. K požadavku přidá datum.
- Odhady požadavků jsou optimistické a prostřednictvím nich se snaží pracovníky motivovat.
- Aktuálně se vedou diskuse s vedoucím OaM, který zjišťoval, proč jsou špatné odhady, proč se to nestihá a proč se to posouvá. Vedoucí obchodu je v situaci, kdy neví, jak to má dělat. Asi je teď fáze, kdy vedoucí OaM vymýšlí lepší způsob, jak to dělat.
- Odhaduje požadavky za frontend a backend.

Když se to teď intenzivně řeší, znamená to, že kvůli špatným odhadům vznikly problémy?

- Většinou z toho problémy nebyly. Ale teď na to začal vedoucí OaM tlačit, aby se to stihalo, takže z toho asi nějaké problémy jsou.
- Podle vedoucí vývoje jsou problémy v komunikaci – vedoucí OaM má představu, že utíkají peníze, pokud se práce nestihne do odhadu. A vedoucí vývoje se mu snaží vysvětlit, že pro dev tým je důležitá kvalita a ne čas. Oba vedoucí jsou v rozporu.
- Vedoucí vývoje není schopen říct, zda by nepomohlo vědět, kolik peněz utíká kvůli zpoždění -> asi by to nic nezměnilo.
- Pomohlo by mu, kdyby mu někdo řekl, co do produktu chce firma dát. Někdo pomohl s obsahem požadavků, dát přesnější zadání. A poté by se odhady daly udělat přesněji. Zadání se tvoří za pochodu a snaží se motivovat tým, aby sami přicházeli s nápady, jak udělat práci lépe (v průběhu implementaci). Vedoucí vývoje nezapisuje nějak zvlášť, ale vylepšuje se zadání v rámci implementace a tester poté hodí další nápady, jak vylepšit daný požadavek.
- Pokud by byl ve firmě člověk, který by fungoval jako analytik nebo vlastník produktu, a byl by schopný říct, jak to má vypadat, tak by byl schopen udělat přesnější odhad.
- Zadání se aktuálně dostává od supportu, testera – vstupy do Releaseů.
- Vedoucí vývoje by ocenil, aby případný analytik rozšířil zadání požadavků – aby bylo více konkrétní.

Co by mělo zadání ideálně splňovat?

- Zadání by mělo splňovat všechna místa z hlediska uživatele, která to ovlivní + jak to bude vypadat + textace + překlad textací + jak to má fungovat z hlediska uživatele + na kterých místech aplikace to má být + UX forma (pop-up atd.).
- Pokud by zadání měl dělat analytik, tak by mělo obsahovat i „edge casů“, možné errorry.

Jak dlouho by podle Vás trvalo vytvořit zadání?

- Vedoucí vývoje si myslí, že to bude trvat dlouho. Analytik by to musel vymyslet, musel by nad tím přemýšlet. Musel by si nastudovat technologii, jak funguje.

Kolik požadavků z Release denně dokončíte?

- Vedoucí vývoj neví. Je důležitá velikost požadavku – každý je jinak veliký. Z programu JIRA bychom to asi dokázali získat.

- Ale průměrně by se to dalo vypočítat na jeden požadavek denně.

Proč se prodlužují Releasy?

- Pro vedoucího vývoje je důležitější mít všechny požadavky hotové než je včas dodat.

Jak vyhodnocujete Sprint?

- Detailně ne.
- Jsou sepsané požadavky, které se dokončily a nápady, které se zadávají do dalších požadavků.
- Spíš než vyhodnocení se řeší, zda se u někoho vyskytuje zásek, tak se to vznese a přemýšlí se, co se s tím dá dělat. Záseky se řeší i během Releasů a probírají se na pátečním dev callech.

Vyskytuje se přesouvání požadavků do dalších Releasů?

- Vedoucí vývoje se tomu snaží vyhnout. Nestíhání požadavků se řeší posunutím / prodloužením Releaseu, aby se všechny požadavky stihly.
- Občas se vyskytuje tlak na vydání, tak se snaží upřednostnit důležitější požadavky. A to méně důležité se posune do dalšího Releaseu.

Proč se nesplnění řeší posunutím?

- Pro vedoucího vývoje je důležitější, aby to, co tým chce vložit do produktu (funkčnosti), tak tam reálně bylo než dodržení termínu. Není tak důležité, zda to bude vloženo o týden dřív nebo později.

Když prodloužíte Release, má to nějaké dopady? (finance, zákazník)

- Vedoucí vývoje to není schopen říct. Obecně tam důsledek není, protože je to produkt firmy a zákazník ho využívá. Postupně se produkt vylepšuje.
- Zákazník za produkt zaplatí ve chvíli, kdy si ho koupí.
- Může tam být, že v rámci získávání „dealu“ někdo něco slíbí a čeká na to. Ale pokud na to zákazník čeká, tak vedoucí supportu urguje.

Jaké máte pravidelné schůzky?

- 1x za týden má tým Developmentu schůzku (dev call), kde se řeší aktuální věci – co se událo a co se chystá na příští týden. Schůzka probíhá tak, že vedoucí vývoje shrne, co řešil se svými podřízenými a vedoucím supportu a sepíše si dotazy, které se nashromáždily během týdnu a projde se účastníky.
 - o V poslední době zavedly strukturovanější meetingy „meeting minutes“, kde se zaznamenává, jaké věci se probraly a věci, které se do budoucna chtějí řešit a sleduje se to.
- 1x za týden call s člověkem z Holandska, který pracuje na desktopové aplikaci. Vedoucí supportu pošle dopředu agendu.
- 1 za rok má vedoucí vývoje schůzku s majitelem, kde si určují důležité cíle. Nemá fixní datum.
- Z vývojového týmu má přístup do Jiry každý.
- Pokud chce někdo ze supportu zadat požadavek na vývoj, tak ho zadává vedoucí supportu (má do JIRY přístup). A pokud je to z jiných oddělení, tak teď je nový proces

(kontinuální improvement proces), který funguje tak, že se to pošle na člena interního prodejního týmu. Vedoucí vývoje to bere jako schvalovací proces.

- Člen prodejního týmu to sbírá a vedoucí OaM to zaštiťuje. Vedoucí OaM to vyhodnocuje v týmu.

Zkoušeli jste někdy Standupy?

- Nezkoušeli.
- Tester pracuje 2 dny v týdnu (úterý, pátek), takže by se setkával jen s vývojářem. Pokud pracuje večer, tak si stejně ráno zavolají. Komunikace / volání vyplývá ze situací.

Co by Vám pomohlo při práci?

- Vedoucímu vývoje by pomohlo, kdyby mu někdo řekl, co se má dát do Releasu – myšleno obsah požadavků.
- Obsah požadavků znamená zadání požadavků a tím pádem by byly přesnější odhady.
- Odhaduje požadavky za frontend a backend.
- V rámci backendu a vedení lidí se to kříží (role). Lidi s ním komunikují a do toho musí vyvíjet. Ve chvíli, kdy mu třeba někdo zavolá, tak se mu přeruší tok myšlenek a složitě se k němu vrací. Neví, co s tím.
- Tím, že je zodpovědný za plánování a administrativu navíc, tak by potřeboval víc času na vývoj (backend). Cítí, že na některé věci nemá tolik času, kolik by chtěl a potřeboval. A cítí, že by potřeboval mít čas na projití se některých věcí, aby je chápal víc do hloubky – produkt je dost specifický a je potřeba velkého know-how. Díky tomu by nevznikaly některé chyby, protože měl větší know-how.
- Díky většímu know-how by byl schopný navrhnout lepší řešení. Nevymýšlel by je zbytečně složitě a neoptimálně.

*Přicházíte do kontaktu se **zákazníkem**?*

- Pouze ve formě požadavků.
- Přímější kontakt se zákazníky by mu nepomohl.

*Komunikuje se během Releasu se **zákazníkem**?*

- Vedoucí vývoje si myslí, že ne.
- Pokud ano, tak to jde od vedoucího supportu.

*Jak byste popsal **přístup vývojářů a testerů** k práci?*

- Lidé pod vedoucími vývoje dělají svou práci dobře a je s nimi maximálně spokojen. Snaží se jim nechat prostor pro nápady a jejich realizaci. Jsou to lidé, o které se může opřít a nechová se k nim jako k podřízeným. Jsou to individuality, které mají velký přínos, protože to jsou profesionálové.

*Když se k nim nechováte jako k podřízeným, v čem spočívá **Vaše vedení**?*

- Mají dotazy a s nimi jdou za vedoucími vývoje. Nenutí jim to. Když potřebují něco řešit, jdou za ním. Ale nefunguje to obráceně.
- Zadá jim práci, kterou mají dělat a pokud mají dotazy nebo nápady, tak to řeší s ním.
- Jediná věc je, že fungují vzdáleně – dá jim důvěru. Ale je tam omezený prostor je časově sledovat – nedá se přímo kontrolovat, zda pracují od X do X.

- Pro vedoucí vývoje je to komplikované ve smyslu, že vidí jen výsledek, ale není schopný to sledovat blíže. Kdyby byli všichni v jedné kanceláři, bylo by to pro něho snadnější a i snadnější komunikace. Je to pro něj trochu překážka.

Potřebujete vidět, v jakém stavu je daný požadavek? Je pevně nastavená pracovní doba?

- Asi ne. Ale není schopný říct, že lidé pracovali od X do X. Nemá úplně časovou kontrolu. Kdyby se po něm chtělo, aby kontroloval lidi, od kdy do kdy byli v práci, tak je to pro něho nereálné.
- Vývojář pracuje od 8 h do 15/16 h.
- Tester pracuje od 10 h do 16 h a poté si to posouvá podle sebe.

Jaká je Vaše důvěra v management / vedení?

- Management (majitel a vedoucí OaM)
- Majitel nemá tolik času na firmu. S ním se vidí jednou za čtvrt roku na čtvrtletním meetingu.
- Od majitele by očekával, že by dával vědět hlavní priority vývoje za půl nebo 1 rok. Očekával by větší vstupy do vývoje.
- S vedoucím OaM se poslední dobou potkává na dev callech. Je velmi schopný v obchodu a marketingu ale co se týče vývoje, tak má vedoucí vývoje pocit, že vedoucí OaM by rád něco změnil. Složitě hledají kompromis, protože vedoucí OaM není tolik v kontaktu s vývojem.
- Vedoucí vývoje má pocit, že vedoucí OaM má potřebu něco řešit, ale nemá na to čas. Neví, co má od něj očekávat. Teď je to neznámá situace, protože neví, jak má fungovat. Vedoucí OaM má nějaká očekávání, chce něco změnit, ale v malém týmu není možné něco měnit, protože jsou časově vytíženi svou prací.

Co by vedoucí OaM rád změnil?

- Pro něj jsou důležité odhady a rád by je měl přesné. Aby Releasey měly pevné termíny, aby se to neposouvalo.
- Požadavky, co jdou do vývoje, by měly procházet přes schvalovací proces a pokud jsou větší požadavky, tak aby to schvaloval vedoucí OaM nebo majitel. Aby tam byl strukturovanější schvalovací proces.
- Myšlenka dobrá, ale je otázka do jaké míry na to budou kapacity. Ale asi nemá čas nebo energii se tomu věnovat do hloubky.
- Nakousnutí probíhá na pravidelných pátečních schůzkách nebo přes cally s vedoucím OaM, na kterých se snaží dostat do obrazu.

Co byste jako CEO změnil?

- Vedoucího vývoje zajímá pouze dopad na vývoj.
- CEO a vývoj – byla by dobrá vize shora, kam má vývoj směřovat – high level vize, čeho chce dosáhnout. Aby to nebylo řešení věcí, které hoří, ale aby tam byl přesah do toho, kam se vývoj má dostat. Delší časový horizont.

Jaká by měla být vize vývoje?

- Co největší sblížení 2 aplikací v určitém časovém horizontu. Aby to nešlo naopak od sebe.
- Online aplikace – možná by byl zajímavý reporting nad daty (co nejvíc využívá).

Příloha 5 Poznámky z rozhovoru s vývojářem

*Jaká je Vaše **role** v IT projektech?*

- Vývoj softwaru na frontendu (interní zaměstnanec).

*Jaká je Vaše **náplň práce**? (úkoly)*

- Úkolem vývojáře je implementovat požadavky z programu JIRA. Po komunikaci s vedoucím vývoje, který požadavky zadává, tak je vývojář zpracuje. Požadavky jsou od klientů nebo přidání nových funkcí (features) = jednotlivé úkoly, na kterých se postupně pracuje. Maximálně se k tomu přidá něco nového.
- Kdo mu co řekne, to udělá.
- Vývojář je nejnižší v organizační struktuře.

*Jaké máte **překážky**, které Vám brání v práci? Například problémy nebo něco nefunguje.*

- Při používání knihoven třetích stran se někdy vyskytne problém, který mu brání v implementaci daného požadavku. Když daný problém najde, tak vznesе požadavek na opravení. Buď se to nezavede vůbec nebo to trvá dlouho, protože to nebudou měnit kvůli jednomu zákazníkovi.
- Když se jedná o větší požadavek, tak to prochází schvalovacím procesem, ve kterém v knihovnách musí zjistit, zda to nepotřebuje i někdo jiný a jaký je přínos. A pokud se to nerozhodnou implementovat, tak musí vývojář najít jiné řešení.
- Tato situace se stává občas. Brzdí ho to, ale nezastavuje mu to práci. Ve firmě to nemůže nikdo ovlivnit.
- Ale vývojář vnímá, že mají knihovny víc kladů než záporů.

Co jsou knihovny třetích stran?

- Knihovny s kódem (poskytovatelé), který následně používá. Obsahuje komponenty nebo funkce, které mu pak usnadňují práci. V podstatě jsou to hotové části.
- Vývojář použije části kódů a udělá z toho finální produkt.

Když trvá knihovnám dlouho zanést požadavky nebo je nezavádí vůbec, proč změny nezavedete sám? (zasáhnutím do kódu)

- Přímo do jejich kódu nemá vývojář přístup. Ale teoreticky by to šlo, jenomže ve chvíli, kdy knihovny vydají novou verzi, tak by vývojář přišel o implementovaný požadavek. Takže by to musel znovu implementovat a takhle neustále dokola.

*Jaký vypadá typický **průběh Sprintu/Releasu**?*

- Vývojář úplně na Sprints nejede. Organizuje je někdo jiný.
- Ze Sprints pro něj vyplývají úkoly/požadavky, na kterých pracuje a přistanou mu v Backlogu. Je mu jedno, zda je to současný nebo další Sprint.
- Jen když někdo zařadí požadavek (stane se z něj vysoká priorita), tak ví, že je to do současného Sprints. Ale jinak to neřeší.

To znamená, že pracujete čistě podle požadavků v Backlogu nebo máte možnost si určitým způsobem určovat jejich prioritu?

- Všechny požadavky mají nějakou prioritu. Kdyby nekomunikovat s vedoucím vývoje, tak by si „vylistoval“ seznam požadavků, seřadil by si to podle priorit. Ale díky

komunikaci zjistí, co má podle vedoucího vývoje prioritu. Protože několik požadavků má stejnou prioritu, tak je lepší se zeptat.

- Ale vývojář nepracuje tak, že by si sám seřadil požadavky podle priorit a vyvíjel je.
- A kdyby vedoucí vývoje byl na dovolené, tak by si to vylistoval sám a jel by odshora dolů.

Jak často s ním komunikujete?

- V podstatě denně. Každý den
- S vedoucím vývoje neřeší, když na jednom požadavku dělá více dnů, ale když pak přijde seznam dalších požadavků, tak s ním řeší, co má dělat dřív.
 - o Kdyby mu chodily nové požadavky každý den, tak se ho ptá každý den. Je to na pár minut.

Máte nastavené schůzky, kterých se účastníte?

- Vývojář se účastní se pravidelných pátečních schůzek, které trvají půl hodiny až 2 hodiny.
- Na pátečních schůzkách se řeší všechno možné – na čem se bude pracovat, na čem se pracovalo.
 - o Na schůzkách se vyjadřuje ke svojí práci, ale prezentuje tam spíš vedoucí vývoje.
- Také se účastní osobní schůzky jednou za čtvrt roku, kde řeší podobné věci.

Účastníte se plánování Sprintů?

- Vývojář se plánování Sprintů neúčastní, protože jdou mimo něj.
- Plánování je mezi vedoucím vývoje, vedoucím supportem a asi dalším vedením.
- Necítí potřebu se účastnit plánování Sprintů. Kdyby se tam objevilo něco, co by mu vadilo, tak se ozve. Akorát by mu to bralo čas na práci.
- *Ozvání:* Například kdyby se do Sprintu vložil požadavek a při jeho zpracovávání vývojář zjistí, že v současné verzi poskytnutého kódu je to řešení složitě, ale za 14 dní vyjde novější verze, ve které je to jednodušší, tak by řekl směrem nahoru, že se vyplatí počkat.
 - o Raději si počká na lepší způsob než aby to dělal horším způsobem a pak to musel přepracovávat.

Máte ještě nějaké další schůzky?

- Další schůzky ne, ale hodně komunikuje s vedoucím vývoje (i několikrát denně).
- Nejvíce komunikuje s vedoucím vývoje, navzájem se doptávají, protože backend má spolupracovat s frontendem a vyměňují si nápady na různé požadavky. Navzájem si usnadňují práci.

*Jak řešíte, když zjistíte, že **nestiháte vyřešit všechny požadavky Sprintu**?*

- Vývojář to neřeší. Snaží se maximálně dělat, co může.
- Doteď se nejel na časové odhady požadavků a doufá, že nepojede. Podle něho se doteď nezmýšlelo, o kolik se přetáhlo a proč.
 - o Ze zkušeností vývojáře: když dělal odhady na požadavky, tak to vždy nějak odhadl a ukázalo, že špatně. + Chtěl mít odhad co nejpřesnější, protože se snaží pracovat precizně a díky tomu se nad tím musel hodně zamýšlet (brát v úvahu

situace, které mohou nastat a těch bylo hodně). Existuje spousta situací, na které si člověk nevzpomene hned a zjistí je až při implementaci. To se zpětně promítne do časového odhadu a pak to pro něj byla ztráta času.

- Když by se ho někdo jen tak zeptal na odhad, tak mu řekne nějaké číslo.

Participujete pak na prodlužování Sprintu?

- Spíš ne.
- Když zjistí, že na nějakém požadavku pracuje déle než bylo odhadnuto, tak o tom informuje vedoucího vývoje a ten rozhodne co s ním.

Může se stát, že odvedete svou práci – vyvinete požadavek, pošlete to dál, tak se Vám to vrátí zpět?

- Když je požadavek dodán, tak by se to stávat nemělo.
- Obecně by měl požadavek projít procesem / sloupečky, které jsou v programu JIRA.
- Vývojář, když splní požadavek, tak ho v programu JIRA přesune do sloupce „done development“. A to je maximum, kam to přesune. Což ale neznamená, že je požadavek úplně uzavřen, pouze z vývojářova pohledu.
- Poté se požadavek přesune na vedoucího vývoje, protože on ví, zda je hotový frontend s backendem. Když ano, tak se to nasadí do testovací verze a tester to zkontroluje. A buď je úspěšné nebo ne.
 - o Když je neúspěšné, tak je to zpátky u vedoucího vývoje a ten napíše do požadavku, co to má dělat nebo je to již jasné z popisu testera.
 - o A takhle dokola, dokud to není v pořádku. Pak vedoucí vývoje může požadavek v programu JIRA uzavřít.
- Jsou tam ještě další stavy (sloupce) například „Waiting“, kdy se může čekat na knihovnu třetích stran než požadavek implementuje.

Řešíte i bugy nebo hotfixy?

- Řeší a je to stejný proces. Jsou ve formě požadavku s danou prioritou a ten se musí co nejdříve nasadit do aplikace.
- Když přijde požadavek na hotfix, tak že vrátí, zapracuje a rovnou se „hodí do aktuálního kódu“.

Co na procesech a Vaší práci (dobře) funguje?

- Vývojář jako silnou stránku bere, že na frontendu pracuje sám. Nikdo jiný mu nesahá do kódu a díky tomu má větší přehled.
- Na druhou stranu je to i slabá stránka, protože mu požadavky trvají déle.

*Napadají Vás **slabé stránky**? (Co nefunguje nebo je prostor ke zlepšení)*

- Cca jednou za 14 dní se stane, že když vývojář nasazuje na server, tak na disku dojde místo a tím pádem je server nedostupný a nemůže pracovat. Tím pádem musí čekat, až daný server někdo restartuje.
- Dále má vývojář nápady přímo k produktu:
 - o Nastavení správných priorit požadavků, například zmenšení písma nebo změna fontu by neměla mít vysokou prioritu.
 - o Kompaktní design/layout – nesnažit se všechno dát na jednu stránku. Na druhou stranu zákazníci neradi hodně scrollují. Zároveň zákazníci chtějí scrollovat

v jednotlivých sekcích. Používají hodně klávesové zkratky a minimálně myš, takže se mění klasické klávesové zkratky na ty, které jsou zvyklí.

Zkoumá se ve firmě zákaznická použitelnost?

- Asi firma sbírá zpětnou vazbu od zákazníků, co se jim líbí/nelíbí. Ale určitě se nezkoumá, zda jim vyhovuje rozložení stránky, jak je umístěné menu.

Probíhá komunikace mezi testery, Vámi a backendem?

- Když se tester dostane k něčemu, čemu nerozumí nebo potřebuje dovysvětlit. Ale je to v menší míře než s vedoucím vývoje. Asi se nejdřív zeptá jeho a pak mě.

Řešíte dokumentaci nebo reporty?

- Dokumentaci vývojář řeší okrajově.
- Minulý týden se to ve firmě řešilo, že když je potřeba vyvolat hláška v určitých případech, tak neví, jak se má vyvolat, za jakých podmínek apod. Nikde to není napsané. Je to třeba v 50 požadavcích v Jire a člověk to musí hledat.
- Asi bude i těžké udržovat dokumentaci aktuální, protože když se to napíše do požadavku, promítne se to do vývoje, tak se to musí dostat i do dokumentace.
- Je otázka, či by to měla být práce. Zda by to měla být práce vývojáře (byl by ochotný ji sestavovat) nebo zda by to měla být práce toho, kdo zadává požadavky.
- Na wiki je zapsaný základ, ale není dostatečný.
- Je rozdíl mezi uživatelskou dokumentací a programátorskou dokumentací.
 - o Programátorská dokumentace by měla popisovat implementaci, informace pro testera. Protože tester někdy neví, zda to, co otestoval, se chová správně nebo by tam měla vyskočit hláška.
 - o Uživatelská dokumentace se dá stáhnout v PDF v aplikaci. Je to návod pro ty, kteří s aplikací pracují.

Kdybyste byl CEO, co byste změnil?

- Vývojářům by přidal zaměstnancům peníze a benefity typu 25 dní dovolené, 13. plat, penzijní připojištění, stravenky (asi spíš pro interní zaměstnance).
- Obchod to už pravděpodobně dělá, ale více zviditelnit a prezentovat firmu na LinkedInu – obecně více zviditelňovat produkt.
- Zlepšit web firmy/produktu z pohledu obsahu. Asi to bylo bráno, že je to efektivní, ale vůbec mi není sympatický. Není přehledný.

Jaká je vaše důvěra ve vedení / managementu?

- Docela dost.
- Věří, že se vedení / management snaží něco dělat s firmou a produktem.
- Majiteli věří, protože si myslí, že produktu rozumí po technické stránce, které nerozumí ani vývojář (komplexně). Což byl i jeden z důvodů, proč šel pracovat do firmy.
- Co se týče vedoucího OaM“
- , tak to vypadá, že také ví, co dělá. „Když by do toho pořádně šlápnul, tak by přivedl výsledky.“
- Vývojář by neočekával, že se management nebude angažovat do jeho práce – říkat mu, co má jak dělat. Chtějí výsledky a ty podává.
- Nově se sepsaly popisy práce, ale moc to neřeší.

*Jak by měl být podle Vás zapojen **zákazník** do vývoje/procesů?*

- Ve formě rozumných požadavků.
- Někdy mají zákazníci scestnou představu, že by něco mělo fungovat, ale z technického hlediska je to nereálné.
- Požadavky od zákazník přicházejí aktivně. Ale neví přesně, jak se dostávají do firmy (prostřednictvím čeho).
- Čím víc feedbacku, tím je to lepší.
- Měli by se jim dávat dotazníky, kde by měli psát, co se jim na produktu líbí a co ne. A pak by je měl někdo procházet a rozhodnout, zda se na nápadu bude dělat teď / v budoucnu nebo vůbec.
 - o Vysvětlit zákazníkům, proč se jejich požadavek nezavede.
- Když vývojáři přijde požadavek divný/zvláštní, tak se raději zeptá vedoucího vývoje, aby případně byla odhalena chyba.

Co by měl podle Vás dotazník obsahovat?

- Zda jsou spokojení s UI, rychlostí, jestli se jim v produktu vyskytují chyby nebo co by chtěli zavést.

Příloha 6 Poznámky z rozhovoru s testerem

Jaká je Vaše role v IT projektech?

- Pozice testera – účastní se kontroly kódu. Ve chvíli, kdy se něco vyvine, tak se to přesouvá k testerovi.
- Je alokovan na 2 pracovní dny v týdnu (MDs). Když je akutní situace, tak i více.
- Ve firmě to není rozdělené tak, že nejdřív proběhne celý vývoj a pak se to celé testuje. Práci přebírá postupně a může ji vracet.
- Nestane se, že by celý den netestoval (neměl práci). Bere to tak, že je to agilní / průběžné. Průběžně řeší testování a přebírá požadavky.
- Dokončený požadavek není, dokud tester neřekne, že je to v pořádku.
- Ve firmě není analytik, takže tím, jak to testuje, tak má největší znalost softwaru, jak funguje jako celek. Hodně se nachází i na začátku vývoje nebo to vyplyne až u testování. Tzn. vývojáři to nějak vyvinuli, tester usoudí, že je to nepoužitelný, tak jim to vrátí.

Pracujete konstantně nebo čekáte, až bude mít vývojář požadavek hotový?

- Testerova práce se dá rozdělit na 2 části: práce na nových požadavcích a přetestování starších požadavků.
- Ve chvíli, kdy nemá co testovat, tak se věnuje přetestování starších požadavků. + Má požadavky, které nejsou v aktuálním vývoji, ale objeví se chyby v produktu.
- Dříve tady bylo méně lidí a nebyly nastavené procesy (zvykání). Ze začátku bylo víc Sprintů a dávaly se do nich méně časově náročné požadavky, aby se to stihlo. Ale stávalo se, že pak k testerovi přišla velká hromada na konci Sprintu (týden / 2 týdny), což bylo nezvladatelné. Vzhledem k časovému tlaku testoval nejdůležitější věci a ty méně důležité si nechal až když to bylo venku (věci, které nemají na zákazníka takový dopad).
 - o Pak se to nastavilo na postupné testování.

Jak jsou aktuálně nastavené Sprints?

- Cíle je mít 4 Sprints za rok. Ale záleží, jak chodí požadavky od klientů. Teoreticky 4, prakticky jich může být víc.
- Každý Sprint je jinak dlouhý – podle naplánovaných požadavků.
 - o Poslední Sprint se prodloužil o měsíc, protože přišly požadavky od klientů, které měly smysl a musely se opravit rychle a zaměstnanci nemohou pustit požadavky částečně. Tím pádem to rozšíří vývoj.
 - o Neznamena to úplně, že je to špatně naplánované, ale taky to má svou roli. Mělo by se počítat s 20 % navýšení kapacity (požadavků), nemocemi, dovolený (+ 1 týden navíc). Zástupy jsou.

Stává se často, že se musí Sprints doplňovat?

- Občas a záleží taky na klientech – větší, dávají zpětnou vazbu, dobré vztahy.
- Když přijdou požadavky, tak jsou zajímavý, ale jsou tam také chyby – drobné chyby od stávajících klientů. A poté by se dalo rozlišit, zda je kritické požadavky nebo další požadavky, které se zařadí do dalších Sprintů.
- Ale o zařazování požadavků do Sprintu a jejich prioritách nerozhoduje tester.
- Noví klienti nemají vliv na délku Sprintu.
- 1 produkt = 1 Backlog = všichni zákazníci

- Kritika od „nových“ nebo stávajících klientů je obecně brána jako kritika. Pak se rozlišují jen priority (velký klient). O prioritách rozhoduje business.
 - o Business to rozebírá s vývojem ohledně kapacit – debata na více úrovních.
 - o Ale jsou i požadavky, které se musí udělat prioritně – jsou kritické -> hotfixy mimo Sprints.
- Jsou požadavky, které ovlivní celý produkt nebo individuální, které uvidí daný zákazník.
 - o Taky to hraje roli v prioritách. Obecně by asi měla větší váhu změna pro všechny.

Proč jsou Sprints dlouhé zrovna 3 měsíce? (proč nejsou kratší)

- Můžou být kratší. Ale v delších Sprints je čas na vývoj, analýzu a testing.
- Kdyby se dělaly kratší, tak se stihnou jen drobnosti (notifikace, textace ...). Větší změny by se do kratších Sprints nevešly a třeba by je nestihl ani otestovat.
 - o Je nutné odchytat všechny varianty, které jsou časově náročnější.
 - o Hotfixy se řeší okamžitě, může to trvat cca 2 dny.
 - o Kritické věci se opravují okamžitě.
- Dokumentace není tak podrobná jako kdyby byl ve firmě analytik. Vymýšlí se hodně za pochodu a mohou se tak opomenout drobnosti, které se objeví až při testingu.
- Podle testera je kvartální Sprint nejrozumnější.

*Proč je špatně, že **dokumentace** není tak podrobná?*

- Tester by měl dopředu vědět, co a jak má testovat a pokud to není zcela zpracované, tak se z toho stává agilní testování.
- Když přijde nový požadavek, tak si ho rychle načte a pak vymýšlí scénáře a jak to otestovat. Kdyby byla dokumentace dobře zpracovaná, tak se na požadavky může připravit.

Když na ně nejste připraven, vytváří tak problémy? (časové, finanční atd.)

- Finanční asi ano.
- Z hlediska testování – pokud by se jednalo o méně seniorního testera, tak vznikne časový problém a tím pádem i finanční.
- Informace, které tester má, tak denně komunikuje s vývojáři a supportem / businesssem.
 - o Pokud něčemu nerozumí, tak se zeptá.

Kdo by měl vytvářet dokumentaci?

- Dokumentaci by měl vytvářet analytik, který chybí, takže jsou všichni analytici.
- Vývojáři vidí požadavek na začátku, tester ho vidí na konci.
- Dokumentaci nahrazuje vedoucí OaM, vývojáři a tester. Dohromady se o tom baví – jakým směrem vyřešit požadavek, použitelnost.
- Aktuálně dělá dokumentaci vývoj, ale je velmi technická a neřeší dopady (analytik).
- Dříve nebyl analytik potřeba, ale teď kvůli rozměrům produktu, by byl potřeba.

*Účastníte se **plánování Sprints**?*

- Tester se Sprints účastní, ale nezasahuje do nich (nemá možnost). Ví, co bude obsahovat a jaký časový objem bude potřeba.

- Ale neovlivňuje, jaké požadavky se do něj zařadí (není schopný posoudit). Od toho je ve firmě business = vedení / management (vedoucí), support a potom vývoj, aby řekl, jaké jsou časové možnosti.
 - o Majitel má ze své pozice poslední slovo.

Jaké máte pravidelné schůzky?

- Tester se účastní týdenních pátečních vývojářských schůzek.
 - o Řeší se nové funkce, testerovy požadavky, změnové požadavky, aktuální stav, na čem celý týden pracoval, co chybí udělat, časové plánování.
 - o Před Sprintem se řeší nové požadavky a čas.
 - Od testera tam chodí nejasnosti, upraví systému/produktu.
 - o Je předem daná agenda. Požadavky se sbírají celý týden.
- Pravidelné týdenní schůzky má každé oddělení.

Časové plánování znamená, že se upravuje Sprint?

- Do Sprintu se dostane například 60 požadavků (představa). Ale když se na konci hodnotí Sprint, tak ze 60 požadavků jich je 120.
 - o Navíc požadavky: chyby, nové požadavky od klientů ze supportu (nejsou časově náročné).
- Sprint se upravuje, ale kostra zůstává stejná.
- Zatím tester nezažil, že by se Sprint musel zkracovat.

Jak se odhaduje čas na 60 požadavků?

- Špatně, ale je to ze zkušeností.
- 1 chyba pro testera = 1 hodina práce navíc.
- Funkcionalita pro testera = cca týden testování (nové přihlašování).
- Časová alchymie – záleží na požadavcích
 - o U sebe to dokáže zhodnotit.
 - o Snaží se do odhadu vejít.

Dostáváte se nějak do kontaktu se zákazníkem?

- Ani ne. Pouze skrz požadavky zadávané od vedoucího vývoje, které tester má otestovat (kontakt se zákazníkem mu nechybí).

Jaké změny byste zavedl, aby fungovaly procesy?

- Nikdy to nebude ideální, jednorožci neexistují.
- Za testera by se mělo zapracovat na dokumentaci.
- Za něho jinak vše funguje (business, support, vývoj)

Jak byste popsal současnou firemní kulturu?

- Přátelské a profesionální prostředí, všichni spolu dlouho pracují. A když se vyskytne problém, tak si lidé neberou osobně.
- Tester přiznává, že když si 2 lidé týkají, tak občas člověk může ztratit zábrany.

Když byste byl CEO, co byste změnil?

- Komunikaci.

- Testerovi přijde, že by se ze strany vedení (majitel, vedoucí OaM) mělo více komunikovat – dobré věci, ale hlavně ty špatné. A říkat realitu.
- Komunikace vize je dobrá, ale komunikovat i horší věci.
- V současné době to nějakým způsobem je na kvartálních schůzkách, kde se řekne firemní vize, vize produktu (ne vize vývoje). Říká se tam, jakým směrem se firma ubírá, jaké jsou výsledky, co se plánuje.
 - Tester by se zaměřil na kratší dobu.
 - Když se zeptá, tak dostane odpovědi, ale musí se zeptat.
 - Přijde mu logické, že o dobrých věcech se mluví a o špatných ne.

Jak by mělo vedení / management zasahovat do práce?

- Dřív mělo CEO poslední slovo, ale když je to na sílu, tak to není úplně dobré – v zájmu firmy. Firma by měla fungovat jinak než rozhodnutí 1 člověka, i když ho přesvědčuje několik lidí.
 - Teď už se to neděje – poslední 1-2 roky.
- Asi vždy budou věci, do kterých majitel zasáhne – v rámci rozpočtu. Teď do toho mluví z hlediska omezeného kapitálu. Ale nemluví do obsahu produktu apod.

Jaké jsou silné stránky nastavených procesů?

- Lidé – přátelské prostředí, efektivní komunikace
- Loajalita zaměstnanců
- Teď se pracovníci naučili zapisovat, takže každý požadavek má historii a lze to tedy dohledat – co vývojáři opravili, jak to opravili, na co si má dát tester pozor (nemusí dohánět lidi). = vedení a evidování práce
- Funguje odezva businessu k vývoji – jsou schopni se o tom bavit na schůzkách. Někdy možná moc schůzky, opakují se pro něj informace (na dev callu ne).

Jaké jsou slabé stránky?

- Neucelená dokumentace, která chybí; je zdlouhavé dohledat informace v Jire.
- Starší procesy:
 - Jelo se hodně na sílu – zasahoval do majitele, klient („one man show“).
- Sprint je naplánován na nějakou dobu (3 měsíce) a z nich by měl mít tester měsíc testování.
 - Testing není pro firmu výdělečná činnost, ale dává produktu kvalitu.
 - Z businessového hlediska se upřednostňuje vývoj a ze 3 měsíců si většinu vývoj ukrajuje pro sebe. Čas se ukrajuje z jeho testování.
 - Vývoj vydělává peníze a zákazník se tak uspokojí.
 - Menší problém je, zda to vyhodí chybu – je důležitější, že to klient dostal.
 - Kvůli ukrajování nemá čas vše zkontrolovat, pak dělá taky důležitější věci a pak je to honem do produkce.
 - Čím méně dostane času, tím horší kvalita.
 - Co jde za něj do produkce, tak tam nejsou závažné problémy. Pak to samozřejmě kontroluje ještě dál (staré věci).

- Na začátku Sprintu se věnuje regresním testům a stačí mu to na 2 dny, protože klienti ještě nezačali program používat, tak si toho nevšimnou. Rychle to zadá na vývoj a pak se z toho udělá hotfix.
- Myslí si, že je to běžná situace.
- Ale testing je velmi důležitý z hlediska oboru a specifika produktu.

Proč si myslíte, že je testing prodělečná činnost?

- Testeři nic nevyvíjí, nedávají přidanou businessovou hodnotu.
- Drží kvalitu produktu, klient je spokojený a nebude chtít vrátet licenci z důvodu nefungování produktu. Díky tomu je to výdělečné.
- Bez testera to vývojáři vyvinou a prodají. Jen se to pak vrátí, protože to nebude fungovat.

- V rámci vývoje se dělají také errorové stavy.
 - Testerová pozice je, aby se jim předcházelo.
 - Celý Sprint by měl testovat produkt, zda všechno funguje správně.
 - Testuje jen nejčastější situace.
 - Je důležité si uvědomit, že zákazníci nevyužívají celý produkt, takže když pak někdo použije méně častou část, tak se může stát, že to vyhodí chybu. Neměl na to zařazený test.
 - Když se něco takového objeví, tak si to zařadí do testů.
 - Na otestování celého systému má na konci Sprintu 2 dny a nejde to celé otestovat. Takže otestuje to nejdůležitější.
 - Cílem vyhledávat takové stavy? Ano, ale je to navíc. Jeho úkolem je kontrolovat produkt v rámci zadání a zároveň podle testerova pocitu.
- Jsou chyby, které chybou nejsou, protože je to takhle v zadání. Například oprávněné vyskakování error hlášky.
- Ví, za jakým účelem je produkt vyráběn. Ví, jaké části bude nejčastěji klient používat.
 - Pak to může mít dopad na priority.



Agilní projektový management ve vybrané organizaci

Bc. Tereza Hrubá, KEMMA03

Řešená problematika

úvod

Firma XY podniká v oboru IT a zabývá se vývojem produktu ve zdravotnictví. Vzhledem k problému je otevřená agilní transformaci a změnám, které pomohou vyřešit problém.

problém

Ve firmě XY byly v průběhu IT projektů zjištěny opakované špatné časové odhady, které mají za následek prodlužování Sprintů, finanční dopad na firmu a na možné zpoždění dodání zákazníkovi.

přístup

Byl zmapován současný stav firmy XY a projektového managementu.
Byl proveden kvalitativní výzkum.

Postup řešení

zdroj

- Odborná literatura
- Schůzky s pracovníky
- Interní firemní metodiky
- Polostrukturované rozhovory

získávání

- Osobní konzultace
- Email
- Rozhovory s majitelem
- Online rozhovory s pracovníky

zpracování

- SWOT analýzy
- Otevřené kódování

Postup řešení

V01: Jak aktuálně funguje projektový management?

V02: Jak aktuálně probíhá plánování Sprintu?

V03: Jakým způsobem probíhají časové odhady IT projektů?

V04: Z jakých důvodů jsou špatně časově odhadovány požadavky v IT projektech?

V05: Proč dochází k prodlužování Sprintů?

V06: Jaká opatření přijmout pro přesnější časové odhady?

Obecná hypotéza 1: Časové odhady IT projektů dělají nesprávní pracovníci.

Obecná hypotéza 2: Při časových odhadech dochází k nedostatečné komunikaci mezi klíčovými pracovníky.

Obecná hypotéza 3: Ve firmě se nekonají agilní ceremonie.

Implementační plán

- 1) Firma XY si ujasní, **proč** chce přejít na agilní metody a co se jí díky nim zlepší. Provede **analýzu současného stavu** a zkontroluje tak **připravenost** na agilní transformaci.
- 2) Určí se **plán** transformace, její krátkodobé a dlouhodobé cíle, jaké obsahuje kroky. Zároveň s plánem definuje **vize** agilní transformace, firmy a produktu.
- 3) Firma XY si vybere **agilní metodu**, která je pro ni vhodná.
- 4) Díky dané agilní metodě se definují **nové týmové role** a nově se postaví **vývojový tým**.
- 5) Na základě agilních metody se zavedou pravidelné schůzky dle Scrumu.
- 6) Vedení s agilním expertem předpřipraví Sprint Backlog na Sprint Planning.
- 7) Naplánuje se **pilotní Sprint** s možností ho upravovat.
- 8) Tým se seznámí** s agilní změnou, pilotním Sprintem a je proškolen.
- 9) Proběhne pilotní Sprint.
- 10) V návaznosti na vyhodnocení Sprintu proběhne postupné a úplné zavedení agility.

Doporučení



Stálá délka Sprintů



Nábor analytika, Scrum Mastera






Najmutí externího agilního coache



Časové odhady dle metod Planning Poker, T-shirt sizing, Větší – menší

Závěr

-  **Práce přinesla návrh implementačního plánu přechodu na agilní fungování firmy XY.**
-  Novým řešením je přechod na agilní řízení projektů.
-  Problematika byla posunuta díky kvalitativnímu výzkumu, otevřenému a upřímnému přístupu pracovníky firmy XY.

**DĚKUJI ZA
POZORNOST**