

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra systémového inženýrství



Diplomová práce

**Agilní transformace projektového řízení v
korporátní organizaci**

Jan Papoušek

Zadání práce

Autor práce: Bc. Jan Papoušek

Studijní program: Systémové inženýrství a informatika

Obor: Informatika

Vedoucí práce: doc. Ing. Jan Bartoška, Ph.D.

Garantující pracoviště: Katedra systémového inženýrství

Jazyk práce: čeština

Název práce: **Agilní transformace projektového řízení v korporátní organizaci**

Název anglicky: Agile Transformation in a Corporate Organization

Cíle práce: Cílem práce je návrh zlepšení stavu a řízení projektů při agilní transformaci na základě vlastního šetření (rozboru) ve vybrané korporátní organizaci.

Metodika: Prvním krokem bude výběr a studium vybrané literatury (odborných knih a článků), interních a veřejných materiálů (metodiky, interní dokumenty, aj.) vybraného partnera z praxe. Poté bude sepsána literární rešerše a první část praktické práce (úvodní popis vybraného partnera z praxe). V popisu budou figurovat všechny fáze životního procesu projektu od úvodní iniciace, přes definici požadavků pomocí user stories, vývoj samotného softwaru, testování a závěrečné nasazení do produkce, a to včetně následné zpětné vazby jeho uživatelů.

Druhým krokem (souběžným) bude provedení vlastního šetření (online nebo osobně) u partnera z praxe, v podobě rozhovorů a vlastního zapojení do projektových schůzek nebo schůzek projektového oddělení (PMO). Z vlastního šetření vzejde vlastní výzkum, který bude mít část kvalitativní a případně i kvantitativní – Na základě získaného přehledu bude zavedený systém fungující (přístupy řízení projektů) u partnera z praxe kriticky zhodnocen a porovnán s teorií agilních přístupů řízení. Na základě zhodnocení bude možné vytvořit doporučení pro konkrétní prostředí či případně doporučení do obecné praxe. V případě potřeby je též možné práci obohatit o cílené rozhovory s konkrétními pracovníky na projektu či o dotazníková šetření pro získání přehledu o názorech pracovníků na zavedené postupy a pořádky při řízení projektů jako takové. Vypracované vlastní návrhy a doporučení budou konzultovány a diskutovány průběžně s oslovenými odborníky z praxe.

Teoretická i vlastní výzkumná část práce se bude tematicky a teoreticky opírat o mezinárodní standardy a metodiky projektového řízení, přičemž práce bude kombinovat jak klasický vodopádový přístup pro řízení projektů, tak i agilní přístupy metody řízení projektů a týmů.

Vybraným partnerem z praxe bude společnost DHL IT Services.

Doporučený rozsah práce: 60–90 stran

Klíčová slova: AGILE, agilní transformace, projektové řízení, vodopádový model, SCRUM, Kanban, sprint, user stories, definice projektových požadavků, softwarový vývoj, zpětná vazba uživatelů

Doporučené zdroje informací:

1. AXELOS GLOBAL BEST PRACTICE. PRINCE2 Agile®. Norwich: Axelos, 2015. ISBN 978-0-11-331467-6.
2. Diaz J. a kol. (2011): Agile Product Line Engineering - A Systematic Literature Review
3. Hoda R. a kol. (2017): Systematic literature reviews in agile software development: a tertiary study
4. KERZNER, H. Project management: a systems approach to planning, scheduling, and controlling. Hoboken: John Wiley & Sons, 2013. ISBN 978-1-118-02227-6.
5. KERZNER, H. Project management 2.0: leveraging tools, distributed collaboration, and metrics for project success. Hoboken: John Wiley & Sons, 2015. ISBN 978-1-118-99125-1.
6. POPPENDIECK, M. -- POPPENDIECK, T D. -- SCHWABER, K. Lean software development: an agile toolkit. Boston: Addison-Wesley, 2003. ISBN 0-321-15078-3.
7. PROJECT MANAGEMENT INSTITUTE. A guide to the project management body of knowledge (PMBOK® guide). Newtown Square, Pennsylvania: Project Management Institute, 2013. ISBN 978-1-935589-67-9.
8. ŠOCHOVÁ, Z. -- KUNCE, E. Agilní metody řízení projektů. Brno: Computer Press, 2019. ISBN 978-80-251-4961-4.
9. ŠOCHOVÁ, Z. The great ScrumMaster: #ScrumMasterWay. Boston: Addison-Wesley, 2017. ISBN 978-0-13-465711-0.
10. Usman, M., Mendes, E., Weidt, F., Britto, R. (2014): Effort estimation in agile software development: a systematic literature review

Předběžný termín obhajoby: 2022/23 ZS – PEF

Čestné prohlášení

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

V Praze dne 13. 11. 2022

Poděkování

Rád bych touto cestou poděkoval vedoucímu práce docentu Janu Bartoškovi za jeho velmi cenné rady a milý a ochotný přístup. Dále bych rád poděkoval své vševědoucí partnerce Elišce za pevné nervy, neutuchající podporu a obrovské množství rad, především v oblasti slovosledu a gramatiky. Díky patří též mému otci, který svou promyšlenou a přesnou kritikou velkou měrou přispěl k závěrečné podobě práce. Nakonec bych rád poděkoval všem členům týmu u partnera z praxe za proaktivitu a nadšení, bez nichž by tato práce zřejmě nevznikla.

Agilní transformace projektového řízení v korporátní organizaci

Abstrakt

Tato práce se zabývá porovnáním teorie agilního projektového řízení s jejím praktickým použitím v konkrétní korporátní společnosti. Cílem této práce je na základě analýzy současného stavu implementace agilního projektového řízení u partnera z praxe navrhnout konkrétní změny, které povedou ke zlepšení tohoto stavu. Analýza je provedena na základě začlenění do zkoumaného týmu u partnera z praxe, porovnání zjištěného stavu s výsledky rešerše dostupných informačních zdrojů a dotazníkového šetření mezi členy týmu. Analýzou zjištěné nedostatky byly rozděleny na manažerské nedostatky vzešlé z nerespektování teoretických postupů a na technické nedostatky, které blíže reflektují místní specifika. V reakci na to byl tak vytvořen prvotní návrh změn, který byl následně upraven na základě zpětné vazby vzešlé z druhého dotazníkového šetření. Výsledkem tak jsou změny, které komplexně řeší situaci u partnera z praxe, a to jak z manažerského i technického pohledu. Výsledky této práce jsou tak vodítkem pro zlepšení situace u partnera z praxe. Zároveň je použita metodika uplatnitelná v libovolné firmě, která potřebuje analyzovat stav implementace agilního projektového řízení.

Klíčová slova

AGILE, agilní transformace, Projektové řízení, vodopádový model, SCRUM, Kanban, Sprint, user stories, definice projektových požadavků, softwarový vývoj, zpětná vazba uživatelů

Agile transformation in corporate organization

Abstract

The main topic of this thesis is a comparison of the practical implementation of Agile project management in a corporate organization with a theoretical approach. The goal of this thesis is to bring a proposal of changes which will improve the status of implementation of Agile project management in selected team at a partner company based on the detailed analysis of ASIS situation. The first step of the analysis is to integrate to the development team and to describe the situation in the team. Then the situation is compared to the review of theoretical research and based on the previous steps there is created a survey among the team members. Then there are identified management shortcomings, which come from not following the theoretical concepts and technical shortcomings which describe the strange specifics of the development team. Those management and technical shortcoming are then transferred to the first draft of changes to improve the situation. For the draft there is gathered feedback with the second survey. In the end there is created detailed overview of necessary changes, which tends to improve the situation from management and technical perspective. Results of the thesis can serve as foundation for the future changes at a partner company. Used methods can serve at any similar use-case because it provides universal tools to analyze the status of Agile project management implementation.

Key words

Agile, Agile Transformation, Project Management, Waterfall, SCRUM, Kanban, Sprint, user stories, requirements management, software development, user's feedback

Obsah práce

Zadání práce	3
Obsah práce	9
1 Úvod	13
2 Cíl práce a metodika	14
2.1 Cíl práce	14
2.2 Metodika práce.....	14
3 Teoretická část práce	15
3.1 Projektové řízení ve vztahu k vývoji softwaru.....	15
3.2 IT Projekt a jeho řízení.....	15
3.3 Fáze IT projektu	16
3.3.1 Zahájení projektu	16
3.3.2 Projektové plánování	17
3.3.3 Vypracování projektu	18
3.3.4 Monitoring a kontrola projektu.....	19
3.3.5 Uzavření projektu	19
3.4 Metoda Waterfall	20
3.4.1 Popis fází Waterfall metody	21
3.4.2 Typické role v metodě Waterfall	22
3.4.3 Výhody metody Waterfall	24
3.4.4 Nevýhody metody Waterfall.....	24
3.5 Agilní metodika.....	25
3.5.1 Agilní projektové řízení	26
3.5.2 Scrum	27
3.5.3 Kanban	34
4 Vlastní práce	36

4.1	Partner z praxe	36
4.2	Oddělení, ve kterém bude práce zpracována	37
4.3	Hlavní produkt týmu – ITS One	38
4.3.1	Project Hub	38
4.3.2	Planning Tool	39
4.3.3	Product Repository	40
4.3.4	Reporting Services	41
4.3.5	Service Quality reporting	41
4.4	Produkční tým	42
4.4.1	Manažer vývojového týmu	42
4.4.2	Konzultanti	42
4.4.3	Databázový specialista	43
4.4.4	Tester	43
4.4.5	Aplikační support	43
4.4.6	Developeři	43
4.4.7	Subproduct owněři	43
4.5	Proces vývoje produktu	43
4.5.1	Plánování procesu	44
4.5.2	Product backlog	46
4.5.3	Průběh schůzky Backlog Pre-Review	46
4.5.4	Průběh schůzky Backlog Review	46
4.5.5	Sprint	47
4.5.6	Sprint backlog	47
4.5.7	Určení náročnosti user story	48
4.5.8	Životní cyklus user story	48
4.5.9	Backlog Grooming	49
4.5.10	Daily Standups	50

4.5.11	Sprint Retrospective.....	50
4.6	Obecná charakteristika situace u partnera z praxe	53
4.7	Porovnání současného stavu s teorií implementace Agile	54
4.7.1	Porovnání týmových rolí s teorií Agile.....	54
4.7.2	Porovnání týmových schůzek s teorií Agile	56
4.7.3	Porovnání týmových technik s teorií Agile	57
4.8	Návrh změn na základě porovnání s teorií Agile	58
4.9	Identifikace technických nedostatků v procesu implementace Agile – dotazníkové šetření	59
4.9.1	Identifikace respondentů.....	59
4.9.2	Existuje prostor pro zlepšení implementace Agile?	60
4.9.3	Doplňující dotazníkové otázky	61
4.9.4	Shrnutí dotazníkového šetření	69
4.10	Prvotní návrh zlepšení implementace Agile.....	70
4.10.1	Kategorie Organizace týmu a práce.....	70
4.10.2	Kategorie Komunikace	72
4.10.3	Kategorie Řízení požadavků.....	73
4.10.4	Kategorie Procesní nedostatky.....	74
4.11	Vhodnost navržených řešení pro implementaci – dotazníkové šetření. 74	
4.11.1	Identifikace respondentů.....	75
4.11.2	Dotazníkové otázky	77
4.11.3	Kategorie organizační zlepšení.....	77
4.11.4	Kategorie komunikace	79
4.11.5	Kategorie Řízení požadavků.....	81
4.11.6	Kategorie Procesy	83
4.11.7	Shrnutí.....	84

4.12	Závěrečný návrh technických zlepšení po vyhodnocení dotazníkového šetření	85
5	Výsledky a diskuze	87
6	Závěr	89
7	Seznam obrázků, tabulek, grafů a zkratk	90
8	Seznam použitých online zdrojů.....	93
9	Seznam použitých knižních zdrojů	95
10	Seznam použitých odborných článků	95

1 Úvod

Vývoj softwaru se postupně od svých počátků vypracoval z pozice experimentálního vědního oboru do jednoho z hlavních hybatelů světové ekonomiky. Jako příklad je možné uvést, že jedny z nejhodnotnějších světových firem jako Microsoft, Google nebo Meta Platforms (Facebook), mají svůj business model postavený na vývoji softwaru, který různými způsoby ulehčuje život milionům lidí po celém světě.

Se vzrůstajícím objemem kapitálu a množstvím lidí, kteří byli do oboru zapojeni, vzrůstala i potřeba efektivně organizovat pracovní činnosti pro maximalizaci jejich pozitivních dopadů. Vyvinula se tak řada způsobů a metod projektového řízení, mezi nimiž vyčnívají metody Waterfall, jakožto ukázka klasického přístupu k projektovému řízení, a metoda Agile přímo reagující na nedostatky metody Waterfall. Přechod z metody Waterfall na metodu Agile se nazývá agilní transformace a jejímu správnému provedení se věnuje velké množství naučných publikací.

V posledních letech zažívá agilní transformace projektového řízení obrovský rozmach a je na ni často nekriticky nahlíženo jako na všelék organizačních nedostatků či problémů týmů zabývajících se vývojem softwaru. Můžeme se tak setkat s implementací Agile v rozličných prostředích, od malých vývojových týmů v malých firmách či organizacích, po masivní korporátní příklady, do kterých jsou zapojeny desítky týmů či stovky zaměstnanců. Existuje tak mnoho různých způsobů implementace Agile, což poskytuje jedinečnou příležitost porovnat vybrané příklady užití s dostupnou teorií.

Tato práce má za ambici komplexně popsat současný stav implementace agilního projektového řízení ve vybraném produkčním týmu v rámci korporátní organizace. V rámci popisu bude rozveden do detailu výsledný produkt týmu, produkční tým včetně jednotlivých zúčastněných rolí, způsob implementace agilního projektového řízení včetně tradičních ceremonií, postupů a metodik nebo i každodenní způsob organizace práce. Díky tomuto popisu bude následně možné využít získané poznatky pro definici prostorů pro zlepšení a následně tvorbu komplexní metodiky, která na příkladu zkoumaného týmu přesně popíše kroky potenciálně vedoucí ke zlepšení pracovní situace. Tato metodika má ambici stanovit postupy a témata takovým způsobem, aby mohla být v případě potřeby použita i v jiném než v této práci posuzovaném prostředí.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je metodický návrh na zlepšení stavu a pro řízení projektů při agilní transformaci na základě vlastního šetření (rozboru) ve vybrané korporátní organizaci. Hlavní výzkumnou otázkou bude: Existují nedostatky v řízení projektů při agilní transformaci ve vybrané organizaci?

2.2 Metodika práce

První krokem při zpracování diplomové práce bude rešerše dostupné literatury (odborných knih, článků a online zdrojů) pro pochopení a popis současného stavu řešené problematiky. Dalším krokem je začlenění do vývojového týmu u partnera z praxe pro popis každodenních činností a produktových cílů týmu se zaměřením na analýzu současného stavu a funkčnosti agilního projektového řízení. Na základě dlouhodobé přítomnosti u týmu bude možné porovnat získané informace o způsobu implementace agilního projektového řízení s dostupnou teorií a v případě nesrovnalostí navrhnout jasné kroky pro zlepšení současného stavu.

V případě nejasností plynoucích z tohoto porovnání bude zahájeno dotazníkové šetření mezi členy produkčního týmu, jehož cílem bude identifikovat konkrétní nedostatky v procesu v předem určených potenciálně problematických oblastech. Tyto získané nedostatky je možné následně analyzovat a použít k vytvoření prvotního návrhu změn, které by měly vést ke zlepšení způsobu, jakým je vedena agilní transformace. Prvotní návrh změn bude použit jako podklad pro druhé dotazníkové šetření v rámci produkčního týmu, jehož cílem je poskytnout členům týmu prostor pro anonymní zhodnocení a případně doplnění navržených změn. Výstupy z druhého dotazníkového šetření poté budou využity k finálnímu návrhu změn, v rámci kterého budou jednotlivé kroky detailně metodicky popsány, vysvětleny a spojeny též s konkrétním plánem nasazení změny či analýzou rizik plynoucích z implementace. Tento vytvořený materiál bude možné, v případě zájmu, předat do užití partnerovi z praxe.

3 Teoretická část práce

V teoretické části práce bude provedena rešerše dostupných zdrojů na témata vývoj softwaru, IT projekt, fáze IT projektů, projektové řízení a typy IT projektového řízení. Jednou z ambicí této části je poskytnout čtenáři teoretický popis v této oblasti, aby následně v praktické části dokázal také sám posuzovat situaci u zkoumaného partnera z praxe. Především však bude teoretická část sloužit jako referenční bod pro porovnání s praktickou situací u partnera z praxe.

3.1 Projektové řízení ve vztahu k vývoji softwaru

Vývoj softwaru popisuje soubor činností v oblasti informatiky věnovaných procesu vytváření, navrhování, nasazení a podpory softwaru. Tento soubor činností se skládá z výběru metodiky, shromažďování požadavků, tvorby architektury, vývoje designu, tvorby procesních modelů, psaní kódu, testování, nasazení softwaru a řízení IT projektu jako takového (IBM, 2022). Vývoj softwaru je tedy velice obecný pojem zahrnující širokou škálu činností a dalších pojmů. Jednou ze zmíněných oblastí, která je pro úspěch vývoje softwaru zásadní, je řízení projektu. Tato oblast bude rozebírána dále jako téma diplomové práce.

3.2 IT Projekt a jeho řízení

Projekt je kus naplánované práce či aktivity s předem daným účelem, která má být dokončena v předem daném časovém období. (Cambridge Dictionary, 2022). Jedná se tedy o velmi široký pojem, pod nímž se dá představit řada časově ohraničených činností s určitým účelem, například i plánování dovolené. Tato diplomová práce se bude zabývat řízením IT projektů, což specifikuje rozsah aktivit čistě na oblast informačních technologií. IT projekt zahrnuje všechny aspekty plánování, návrhu, implementace, projektového řízení a školení v oblasti vývoje softwaru. (LawInsider, 2022). V teoretické části práce budou nejdříve popsány jednotlivé fáze IT projektu pro uvedení všech následujících textů do kontextu a poté bude následovat detailní teoretický popis Řízení IT projektu, který je hlavním tématem této diplomové práce.

3.3 Fáze IT projektu

Fází IT projektu je myšlen soubor aktivit v rámci řízení projektů, které spolu tematicky souvisí. Tyto fáze na sebe často navazují a každá z nich je zakončená nějakým standardním výstupem či výstupy. Dohromady tvoří takzvaný životní cyklus projektu. Jednotlivé aktivity jsou obvykle děleny do 5 fází:

- Zahájení projektu
- Projektové plánování
- Vypracování projektu
- Monitoring a kontrola projektu
- Uzavření projektu

Každá z těchto fází se liší zaměřením a náplní, řízení projektů je tedy velmi variabilní a různorodá činnost. (Project Management institute, 2000). V následujících kapitolách budou tyto fáze postupně popsány.

3.3.1 Zahájení projektu

Jedná se o první, a zřejmě i nejdůležitější fázi životního cyklu IT projektu. V rámci této fáze mají být definovány důvody, motivace a příležitosti plynoucí z projektu jako takového. Motivace a důvody pro zahájení projektu se nejčastěji popisují pomocí businessového záměru.

Cílem businessového záměru je definovat businessové zadání (popis následné funkcionality z pohledu přínosu pro zadavatele) a nastítnit využití finančních a dalších podnikových zdrojů pro potřeby splnění projektu. Businessový záměr by se měl řídit metodikou SMART (Specific – specifický, Measurable – měřitelný, Achievable – dosažitelný, Relevant – relevantní a Time-Bounded – časově vymezený) V rámci definice businessového záměru je vhodné identifikovat všechny projektové sponzory a zúčastněné strany, včetně jejich zájmů, preferencí atd. (There's a S.M.A.R.T. way to write management's goals and objectives, 1981).

Po definici businessového záměru se většinou provádí studie proveditelnosti, jejíž cílem je prozkoumat možná řešení businessového záměru, identifikovat nezbytnosti pro úspěch projektu a případně navrhnout záložní řešení. Studie proveditelnosti nemá řešit rizika plynoucí z projektu.

Po zpracování studie proveditelnosti by se obvykle měla vytvořit či zorganizovat tzv. projektová kancelář, jejíž úkolem je zajistit komunikaci v rámci projektu a poskytovat projektovému managerovi podporu v řízení projektu, například pomocí reportingu či skrze zajišťování nezbytných technikálií. (projectmanager.com, 2018)

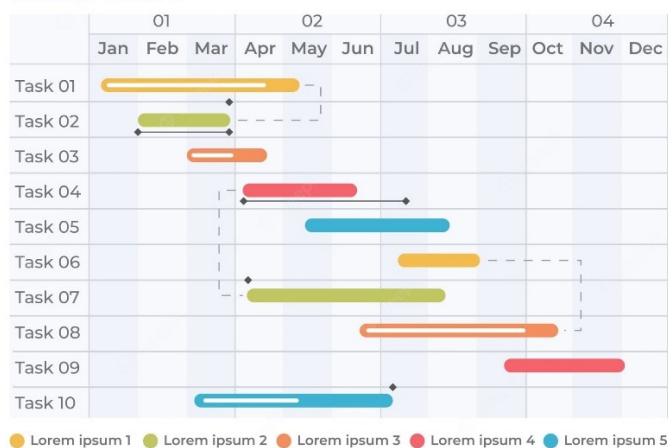
3.3.2 Projektové plánování

Cílem projektového plánu je poskytnout všem zúčastněným jasný směr a vysvětlit jim jejich roli a přínos potřebný pro úspěch projektu. Úspěšně provedené projektové plánování končí oficiálním zahájením projektu. Prakticky se tedy jedná o vytvoření časově organizovaného detailního výčtu na sebe navazujících aktivit vedoucích k dokončení projektu. Před vytvořením projektového plánu je však potřeba nejdříve provést řadu kroků, které umožní tento plán splnit lépe a přesněji. (Kerzner, 2009)

1. Definice funkčních požadavků pro přesnější naplánování rozsahu práce
2. Analýza rizik a z ní plynoucí návrhy pro jejich zmírnění
3. Analýza zdrojů (materiálních, technologických, lidských) a případné vznesení požadavků na jejich doplnění
4. Analýza a případná příprava komunikačního plánu
5. Analýza a případné vytvoření testovacího plánu
6. Definice projektových milníků a jejich dosazení do časového plánu
 - Nebo definice projektových KPIs pro kvantifikovatelný monitoring postupu

Po splnění těchto kroků je možné přistoupit k detailnímu rozplánování jednotlivých aktivit do podoby projektového plánu. Pro tyto účely se nejčastěji používá takzvaný Ganttův diagram (obrázek 1). V Ganttově diagramu je na ose Y k dispozici seznam činností a na ose X čas od zahájení projektu do jeho předpokládaného konce. Následně se k jednotlivým řádkům (činnostem) vybarvují patřičné buňky symbolizující zasazení dané aktivity do časového plánu. Typicky jsou tak vybarveny políčka od levého horního rohu směrem k pravému dolnímu rohu Ganttova diagramu. Též je možné seskupovat vybarvená políčka dle barev pro zvýraznění konkrétních tematicky společných činností.

Gantt Chart



Obrázek 1: Ganttův diagram, zdroj: [Freepick.com](https://www.freepick.com)

Takto zobrazený projektový plán by měl být doplněn o předem definované milníky, které budou kontrolovány se zúčastněnými stranami, k jednotlivým aktivitám by měli být dosazeni konkrétní za ně zodpovědní pracovníci a případně by měly být zaneseny závislosti (aktivita B nemůže začít, pokud není dokončena aktivita A)

Projekt je zpravidla formálně zahájen úvodní schůzkou všech zúčastněných stran, na kterém jsou definovány cíle projektu, rozsah projektu, časový rámec projektu, představeny jednotlivé zúčastněné strany, výzvy, rizika a nastíněny další nutné kroky. (University of Massachusetts Boston, 2022)

3.3.3 Vypracování projektu

Jedná se o zpravidla nejdelsí a nejnáročnější fázi životního cyklu IT projektu. Zároveň je potřeba zmínit, že tato fáze probíhá společně se čtvrtou fází – Monitoring a kontrola projektu a navzájem se doplňují.

Během této fáze se postupuje dle stanoveného projektového plánu a plní se jednotlivé aktivity v něm zahrnuté. Náplní práce projektového manažera je dohlížet, mapovat postup a vést všechny zodpovědné za jednotlivé aktivity k jejich úspěšnému a včasnému dokončení. Zároveň je jeho stěžejním úkolem zajistit a vést transparentní komunikaci se všemi zúčastněnými stranami. Pokud je v průběhu projektu zjištěno, že je potřeba provést změny v projektovém plánu, je úkolem projektového manažera, aby tyto změny patřičně zdokumentoval a všechny členy týmu o nich náležitě informoval. Změnové řízení je klasicky

nejproblematictější činností v této fázi projektu, protože vyžaduje vysokou míru flexibility a akceschopnosti od všech zúčastněných. (Kissflow Inc., 2021)

3.3.4 Monitoring a kontrola projektu

Cílem a smyslem této fáze je monitoring a faktická kontrola průběhu vypracování projektu. Jak již bylo zmíněno, běží současně s předchozí fází. Během této fáze je klíčové kontrolovat, zda project běží v souladu s časovým plánem, s odsouhlaseným rozpočtem, s plánovaným rozsahem a se správným řízením rizik (Adobe Workfront, 2022). K faktické kontrole jsou nejčastěji využívány následující nástroje a postupy:

- Status report, který sdružuje všechny informace o postupu v projektu, hodnotí případná rizika, a to jak z pohledu zúčastněných stran, tak z pohledu vývoje
- KPIs, což je sada v plánovací fázi definovaných metrik pro objektivní zhodnocení postupu (množství doručených funkcionalit za týden a podobně) či pro zhodnocení spokojenosti zákazníka s dodanými funkcionalitami (NPS)
- Projektové milníky a monitorování času a okolností jejich splnění
- Udržování komunikace a získávání zpětné vazby se zúčastněnými stranami (sponzory projektu)
- Kontrola nákladů – rostoucí náklady mohou plynout ze změnových požadavků či změněného rozsahu projektu
- Kontrola kvality – tlak na včasné doručení může znamenat sníženou kvalitu. Tu je potřeba hlídat především v součinnosti se sponzory projektu (zákazníky)
- Kontrola a předcházení rizik – kromě analýzy rizik v plánovací části je potřeba neustále monitorovat, aktualizovat a předcházet možným rizikům plynoucím ze změnových požadavků v projektu (Lewis, 1999)

3.3.5 Uzavření projektu

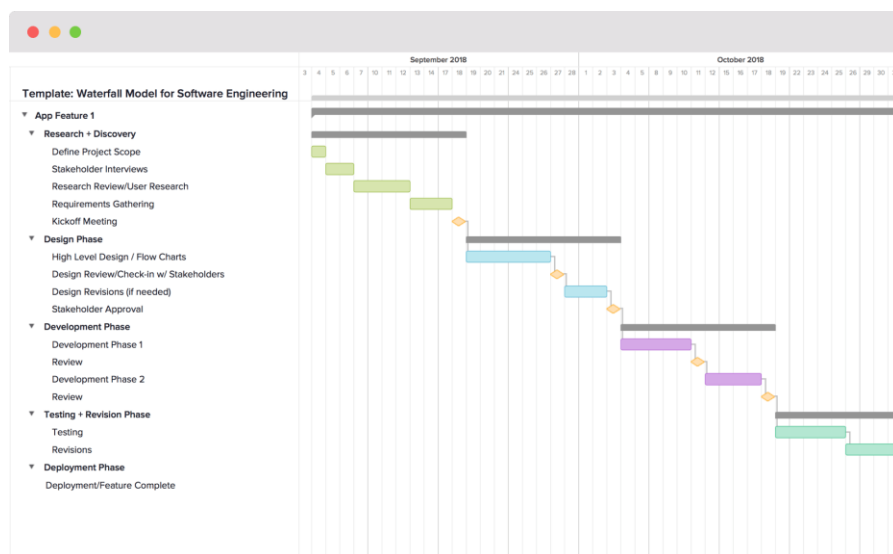
Tato fáze se věnuje formálnímu uzavření projektu, které nastává zpravidla po úspěšném dokončení všech naplánovaných aktivit. Jejím cílem je potvrdit a odkomunikovat dokončení všech aktivit, předání výsledků jeho sponzorům či zadavatelům, zhodnocení celkového průběhu nebo individuálních přínosů k úspěchu a tvorba doporučení plynoucích z průběhu tohoto projektu.

Za potvrzení o dokončení projektu je nejčastěji považována administrativní akceptace dokončení všech projektových aktivit zákazníkem či zadavatelem, například písemným potvrzením. Cílem komunikace v této fázi je zajistit, že všechny zúčastněné strany si jsou vědomy ukončení projektu a že změní své zaběhnuté pořádky a procesy tak, aby v nich správně zohlednily přinesené změny. (Project Management institute, 2000)

Zhodnocení celkového průběhu projektu a tvorba doporučení se nejčastěji provádí pomocí takzvané Zprávy o ukončení projektu. Zpráva o ukončení projektu je standardizovaný vzor používaný celosvětově pro objektivní zhodnocení úspěšnosti projektu. Její součástí je posouzení míry, s jakou projekt splnil představy zákazníka, identifikace dobrých zvyklostí plynoucích z práce, posouzení odchylek od původního projektového plánu a definice konkrétních zlepšení a přínosů do budoucna. (University of Massachusetts Boston, 2022)

3.4 Metoda Waterfall

Klasickým typem přístupu k řízení IT projektu, který se drží výše popsaných standardů a postupů, je takzvaná metoda Waterfall (vodopád). Ta je používána nejčastěji při vývoji nové softwarové funkcionality a skládá se z jasně oddělených fází, které na sebe navazují, ale téměř nikdy se nepřekrývají. Při zakreslení do Ganttova diagramu tak typicky vzniká kaskádovitý tvar (vodopád – Waterfall, obrázek 2).



Obrázek 2: Metoda Waterfall po zakreslení do Ganttova diagramu, zdroj: teamgantt.com

V rámci Waterfall metody nejčastěji rozlišujeme 5 komplexních fází, které od sebe tematicky oddělují jednotlivé pracovní aktivity, navzájem na sebe chronologicky navazují a nepřekrývají se. Jedná se o Sběr požadavků, Design, Implementaci, Testing a Nasazení. (Adobe Workfront, 2022).

3.4.1 Popis fází Waterfall metody

Ve fázi Sběru požadavků je nezbytné zjistit, co všechno má být obsaženo v softwarovém přídatku a mít přehledné informace o místních specifikách. Typickými technikami pro tuto fázi jsou business analýza, dotazník, workshop, brainstorming. V této fázi jsou nejaktivnější IT a business analytici. Ukončit fázi sběru požadavků je možné ve chvíli, kdy jsou známy všechny detaily a podklady pro tvorbu systémového designu/architektury.

Ve fázi Designu systému dochází k využití sesbíraných požadavků k návrhu architektury, návrhem datových modelů, rozhodnutím o použitých technologiích, programovacích jazycích, hardwarových specifikách ve vývoji apod. Typicky jsou v této části nejaktivnější IT architekti, kteří jsou zodpovědní za veškeré modelování a návrhy.

Ve fázi Implementace systému dochází k vystavění (naprogramování) systému podle modelu vytvořeného v předchozí fázi. Přístupů k implementaci softwaru je několik, například monolitický přístup, spočívající v tvorbě jedné složité aplikace plnící veškeré požadované funkce, který je ale náchylný na selhání z důvodu své složitosti a komplexnosti nebo naopak dnes velmi populární přístup využívající tzv. Microservices tj. vytvoření malých subaplikací, které jsou navzájem integrované, ale v nichž výpadek jedné z nich neovlivní činnost těch ostatních. (Atlassian, 2022). V této fázi jsou nejaktivnější softwaroví inženýři. Jakmile jsou všechny požadované funkcionality naimplementované, je možné přistoupit k další fázi.

Další fází je Testování softwaru, jehož cílem je ověřit, zda vyvinutá funkcionalita odpovídá zadání od zákazníka. Pokud testování odhalí nedostatky, je možné danou část softwaru opravit v rámci implementační fáze. Je-li však nalezena zásadní systémová chyba, je třeba, aby se projekt vrátil na začátek a došlo k přepracování nejen implementační fáze, ale i obou částí předcházejících fází tak, aby byly nedostatky systematicky odstraněny. V této fázi pracují testeři.

Po úspěšném ověření, že nový softwarový přídavek splňuje všechny požadavky zákazníka, následuje fáze Nasazení (předání). V této fázi je výsledný produkt předán zákazníkovi do užívání, je nasazen k použití cílové audienci a je průběžně udržován a kontrolován. Jedná se o poslední fázi v rámci Waterfall metody, protože ta již teoreticky ani prakticky neřeší takzvanou run fázi (údržbu v provozu) životního cyklu softwarových produktů. (LucidChart, 2022) (Kerzner, 2009)

3.4.2 Typické role v metodě Waterfall

Jak již bylo nastíněno v předchozí kapitole, v každé z fází Waterfall projektu se zapojují typičtí odborníci (obrázek 3). Mají úzce specializované role, úzké spektrum odbornosti a v rámci jednoho projektu tak vystupují pouze v jeho částech. Typicky se tak tito odborníci zapojují do vícero projektů, protože na jednom projektu pro ně většinou není kontinuální využití po celou jeho dobu (s výjimkou projektového manažera).

Projektový manažer

- Zajišťuje celý projekt, zajišťuje komunikaci a schopnost spolupráce všech zúčastněných stran a zúčastněných odborníků. Konstruuje a udržuje projektový plán (Ganttův diagram) a srozumitelně jej vysvětluje všem zúčastněným stranám. Typickými dovednostmi projektového manažera jsou schopnost jasné a srozumitelné komunikace, vedení schůzek a schopnost organizování a plánování práce.

Business analytik

- Vytváří studii proveditelnosti, na jejímž základě se rozhoduje o projektovém záměru. Sbírá a detailně analyzuje požadavky zákazníka na budoucí funkčnost systému a předává je jako podklady pro následující technickou práci. Typickými dovednostmi je schopnost získat relevantní informace a ty následně srozumitelně předat techničtěji založeným kolegům.

IT architekt

- Je zodpovědný za komplexní návrh řešení, které má být zákazníkovi dodáno, za výběr správného architektonického modelu, výběr vhodných technologií a následně za vedení a organizaci týmu vývojářů především po technické stránce. Obvykle se jedná o nejzkušenějšího vývojáře, který ovládá široké spektrum technologií a je schopný na základě svého rozhledu určit správný směr.

SW vývojář (developer, inženýr)

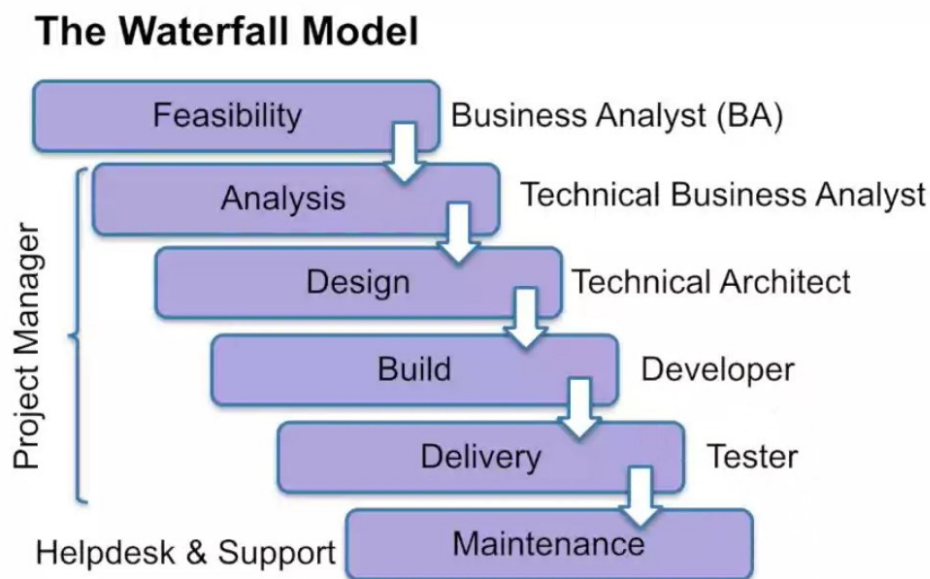
- Jedná se o odborníka, který umí používat konkrétní architektem zvolenou technologii a je tak přímo zodpovědný za vývoj dané funkcionality. Přebírá požadavky od business analytika a od architekta, a podle nich vytváří výsledné řešení. Klíčovou dovedností je schopnost ovládat požadované technologie (programovací jazyk apod.)

Tester

- Tester zodpovídá za ověření, že nové řešení v plném rozsahu odpovídá požadavkům zákazníka. Kontroluje funkčnost, implementaci všech požadovaných funkcí či schopnost přestát zatížení (performance testing). Klíčovými dovednostmi je znalost požadavků na řešení a znalost testovacích nástrojů a metodik.

Aplikační support

- Support zodpovídá za údržbu řešení poté, co bylo předáno zákazníkovi do užívání. Support může vést školení koncovým uživatelům, opravovat chyby u uživatelů či poskytovat zpětnou vazbu o využívání řešení. Klíčovými dovednostmi je znalost řešení a schopnost jej analyzovat a vysvětlovat jeho uživatelům. (Ayub, 2012)



Obrázek 3 Příklad rolí ve Waterfall metodice, zdroj: IT Recruitment Survival Guide (2017)

3.4.3 Výhody metody Waterfall

Za největší výhodu metody Waterfall je považována skutečnost, že používá jasnou a pevnou strukturu projektového řízení, což snižuje faktické nároky na schopnosti zaměstnanců. Pokud jsou dopředu jasně stanoveny a komunikovány úkoly, aktivity a cíle, tak každý člen zodpovědný za danou aktivitu ví naprosto přesně, na čem a proč má pracovat, a potřebuje tedy být obeznámen pouze s úzkým okruhem činností. Nejsou tak potřeba žádné speciální certifikace nebo tréninky pro úspěšné zapojení i méně zkušených členů produkčního týmu do procesu.

Další obecně uznávanou výhodou metody Waterfall je včasná definice projektového cíle, což všem členům produkčního týmu usnadňuje orientaci v projektu a snižuje šanci, že pozornost jednotlivých členů týmu bude odvedena mimo relevantní úkoly.

Neméně důležitou výhodou Waterfall metody je díky její jasně definované formě skutečnost, že usnadňuje výměnu informací mezi jednotlivými členy produkčního týmu, protože díky pečlivému projektovému plánování od začátku prací definuje kdo s kým a o čem má v dané projektové fázi komunikovat.

Obecně lze tyto výhody shrnout konstatováním, že metoda Waterfall díky striktnímu plánování umožňuje více specifikovat probíhající projektové činnosti, díky čemuž snižuje nároky na schopnosti jednotlivých zapojených stran. (LucidChart, 2022)

3.4.4 Nevýhody metody Waterfall

Za největší nevýhodu metody Waterfall je považována skutečnost, že díky svému striktně danému a dodržovanému projektovému plánu velmi limituje a ztěžuje možnosti, jak provádět změny v průběhu projektu. Chyby v postupu v metodě Waterfall jsou často odhalovány až téměř na konci celého projektu, ve fázi testování, což v krajních případech vede i k situacím, že je třeba začít kompletně od začátku, což může vést v rámci týmu i ke značným rozporům. Pokud je třeba provést změnu v průběhu projektu, nestačí pouze něco změnit v současné projektové fázi, ale je potřeba vyhodnotit dopady této změny na všechny následující činnosti a ty pak adekvátně poupravit.

Další zmiňovanou nevýhodou metody Waterfall je fakt, že snižuje vliv klienta (zákazníka či koncového uživatele produktu) na konečnou podobu produktu. Klient se totiž aktivně podílí na projektových přípravách především v úvodní fázi, kde odsouhlasí

projektový plán a dále je již na produktovém týmu, jakým způsobem tuto představu reálně implementuje.

Obecně jde tyto nevýhody shrnout konstatováním, že Waterfall metodika špatně reaguje na situace, kdy „práce nejde podle plánu“. (LucidChart, 2022)

V návaznosti na tyto skutečnosti proto byla vyvinuta nová metodika, která bude detailně rozebírána napříč celou prací.

3.5 Agilní metodika

V reakci na nevýhody metodiky Waterfall byla v roce 2001 definována takzvaná agilní metodika. Ta v ucelené podobě vznikla na shromáždění elitních amerických programátorů, kteří v té době již propagovali různé alternativní (odlehčené) metodiky vývoje softwaru jako například Extrémní programování (extreme programming) či Vývoj řízený vlastnostmi (feature driven development). Novou metodiku, vznikající jako propojení všech do té doby existujících odlehčených metodik, pojmenovali Agile a pro její potřeby sepsali tzv. Manifest agilního programování. V Manifestu je obsaženo následující:

Objevujeme lepší způsoby vývoje softwaru tím, že jej tvoříme a pomáháme při jeho tvorbě ostatním. Při této práci jsme dospěli k těmto hodnotám:

- *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*

Jakkoliv jsou body napravo hodnotné, bodů nalevo si ceníme více.

Řídíme se těmito principy:

- *Naší nejvyšší prioritou je vyhovět zákazníkovi časným a průběžným dodáváním hodnotného softwaru.*
- *Vítáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.*
- *Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.*
- *Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.*
- *Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.*

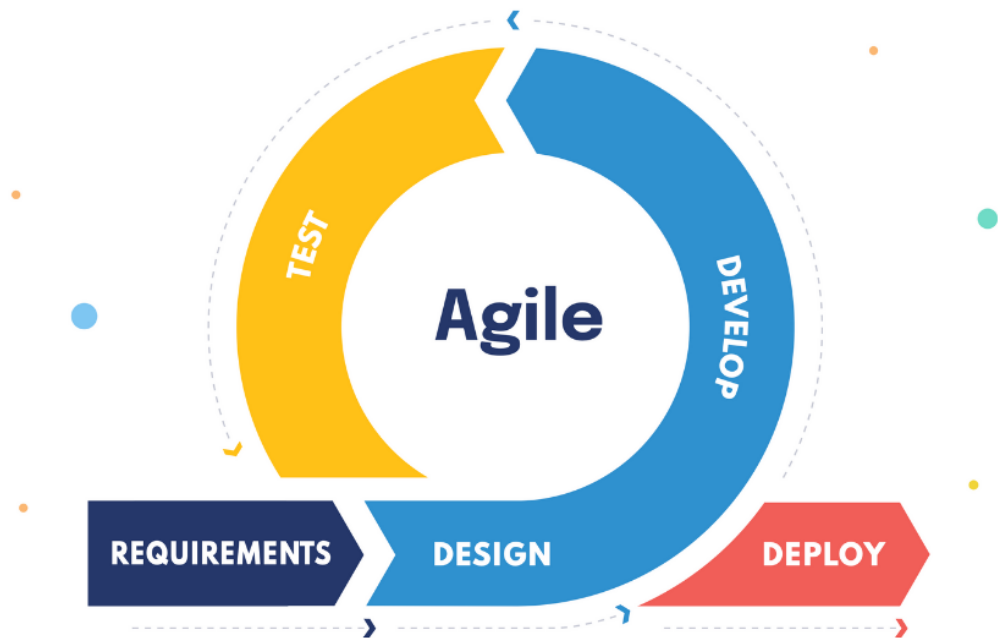
- *Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.*
- *Hlavním měřítkem pokroku je fungující software.*
- *Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.*
- *Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.*
- *Jednoduchost – schopnost maximalizovat množství nevykonané práce – je klíčová.*
- *Nejlepší architektury, požadavky a návrhy vzejdou ze samo-organizujících se týmů.*
- *Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování a zvyklosti.*

(Kent Beck a kol., 2001)

Jednotlivé tematické části, které si agilní metodika převzala a případně upravila do prostředí vývoje softwaru z metodik, ze kterých vyhází, budou popsány v následujících kapitolách.

3.5.1 Agilní projektové řízení

Zatímco v klasickém projektovém řízení je projekt pojat jako ucelený plán s jasně definovanými fázemi, které v určitý čas začínají i končí a které lze v grafu zobrazit jako kaskádovitý vodopád, agilní projekt se nejlépe ukazuje jako kruh (obrázek 4). Smyslem agilního přístupu k projektovému řízení je totiž proces tvorby závěrečného produktu po malých částech, z nichž každá tato malá část sama o sobě přináší zákazníkovi určitý užitek. (Poppendick, et al., 2003) Tyto části jsou uzavřené funkcionality, které jsou samy o sobě navrhnuté, implementovány a otestovány s nepřetržitým získáváním zpětné vazby od zákazníka. Zákazník má tedy daleko vyšší vliv na závěrečnou podobu produktu i na prioritizaci prací k závěrečnému produktu vedoucí. Vzhledem ke značným specifikům tohoto postupu (větší nároky na přehledné vedení komunikace či schopnost improvizovat) jsou často od zodpovědných osob za projektové řízení vyžadovány speciální certifikace na agilní metodiku. (Atlassian, 2022)



Obrázek 4: Agile project management, zdroj blog.feedbackloop.com

3.5.2 Scrum

Scrum je metoda, která pomáhá lidem, týmům a organizacím vytvářet hodnotu prostřednictvím adaptivních řešení pro složité problémy. Tuto metodu vytvořili Ken Schwarber a Jeff Sutherland na začátku devadesátých let a definovali ji v článku s názvem The Scrum guide, viz. následující citace.

Scrum se ve zkratce dá popsat následovně.

Scrum master má za úkol vytvořit a udržovat funkční prostředí, kde:

- *Product owner zadává práci na komplexním problému do product backlogu.*
- *Produkční tým pracuje během sprintu na komplexním problému. Výsledek práce se nazývá přídavek hodnoty.*
- *Produkční tým a jeho zákazník zkontrolují výsledky a přizpůsobí podmínky na další sprint.*
- *Opakovat od začátku*

(The Scrum Guide, 2020)

Klíčovým prvkem Scrumu je tzv. sprint. Sprint je pravidelné opakování takových pracovních činností, během kterých je definován a implementován přídavek. Sprinty na sebe přímo navazují a v rámci každého sprintu je splněn alespoň jeden přídavek. Délka sprintů je

záměrně volena co nejkratší, aby v rámci sprintu nedocházelo ke změnám jeho priorit, typická délka je 1–4 týdny.

V rámci Scrum můžeme najít následující tematické celky: pojící základní hodnoty, typické role, typické schůzky, a typické techniky. V následujícím textu budou stručně teoreticky představeny.

Scrum staví na následujících základních hodnotách:

- Sebeorganizace produkčních týmů
 - Týmy jsou zodpovědné za všechna rozhodnutí související s potřebou doručit přídavek. V rámci produkčního týmu se nerozvíjí žádná sub-hierarchie, jedná se o soudržnou malou jednotku profesionálů, která se soustředí v daný čas na jeden cíl.
- Rychlá a pravidelná dodávka přídavků
 - Každý z členů produkčního týmu se soustředí na pravidelné a postupné dodávání přídavků, každý člen týmu má vždy jasně stanovený cíl, na kterém pracuje.
- Kontrola a adaptace
 - Týmy neustále kontrolují a hledají způsoby, jak zefektivnit svou činnost skrze retrospektivní evaluaci a monitorování vývoje ve sprintu. Na základě evaluace se snaží zlepšovat pro příští sprinty.
- Transparentnost procesu
 - Týmy sdílí veškeré informace tak, aby se každý člen týmu mohl nejlepším možným způsobem zapojit do činnosti tak, aby byl splněn cíl. Žádný člen týmu neskrývá informace před ostatními kvůli vlastnímu prospěchu.
- Časová ohraničenost všech činností
 - Vše, s čím se ve Scrumu pracuje, musí mít definovaný začátek a konec pro zajištění disciplíny a důrazu na včasnou dodávku. Nejčastější časovou hranicí je nejbližší konec sprintu.

V rámci Scrumu registrujeme tři typické role, které může jedinec nabýt.

- Product owner
 - Jedná se v podstatě o vedoucího prací na produktu. Jeho primárním cílem je maximalizace přidané hodnoty plynoucí z práce produkčního týmu. Toho product owner dosahuje následujícími činnostmi.
 - Rozvíjí a jasně komunikuje produktové cíle

- Vytváří, udržuje a jasně vysvětluje jednotlivé položky product backlogu
- Je zodpovědný za prioritizaci prací na položkách product backlogu
- Zajišťuje, že product backlog je všem přístupný a srozumitelný.
- Z tohoto vyplývá, že má pravomoc delegovat práci a zodpovídá za další členy produkčního týmu. Pozice product ownera musí být obsazena jedním člověkem (rada product ownerů je nepřijatelná), ale zároveň product owner může být zodpovědný za více produktů. Jeho pozice musí být plně respektována a ve společnosti podporována. (The Scrum Guide, 2020)

Scrum master

- Jedná se o osobu, která usnadňuje průběh prací v rámci Scrumu. Je zodpovědný za to, že každý rozumí specifikám Scrumu a odstraňuje překážky bránící členům produkčního týmu v efektivní práci. (Šochová, 2018)
- Zároveň je tato role zodpovědná za celé nastavení systému Scrum. Vede školení o chování a práci v rámci Scrum v celé organizaci, a je k dispozici všem členům týmu v případě jakýchkoliv problémů, které je potřeba řešit, a které přímo nesouvisí s technickou náplní práce.
- scrum master též slouží jako pomocník product ownerovi v nalézání technik a postupů pro efektivní definování produktových cílů, efektivní manipulaci s položkami v product backlogu a úspěšnou komunikaci se stranou zákazníka. (The Scrum Guide, 2020)
- Produkční tým (vývojáři, inženýři)
 - Je zodpovědný za dodání přídatku a všechny práce s tím související, od odhadování pracnosti (plánování sprintu), přes udržování sprint backlogu, po samotný technický návrh a implementaci řešení. Obvykle mívá 3–9 členů, a obsahuje zástupce všech technických rolí (analýza, vývoj, testing), které jsou potřeba pro doručení přídatku. Tyto technické role se velmi liší podle charakteru práce, která je na produktu potřeba. Jejich zodpovědností je kvalita přídatku, přizpůsobování denních činností doručení přídatku v rámci cílů sprintu. (The Scrum Guide, 2020)

V rámci každého sprintu se vyskytují pevně dané a rozvržené pravidelné schůzky, které obstarávají většinu komunikace potřebné pro doručení přídatku. Cílem těchto schůzek je získání priorit sprintu od zákazníka, následně definice cílů sprintu, sledování průběhu sprintu, odstraňování překážek v probíhajícím sprintu, předání přídatku zákazníkovi na schválení a závěrečná retrospektiva sloužící pro definici prostorů pro zlepšení v příštích sprintech. (Deemer, et al., 2010)

- Grooming
 - Během této schůzky se pravidelně upravují detaily položek v product backlogu tak, aby odpovídaly požadavkům a prioritám zákazníka. V rámci schůzky se schází product owner, zástupce/zástupci týmu developerů a zástupce/zástupci strany zákazníka a diskutují, které položky product backlogu již nejsou relevantní a je možné je vyřadit, které položky backlogu je třeba upravit či přeformulovat tak, aby odpovídaly novým skutečnostem, a kterým položkám je potřeba zvýšit jejich prioritu tak, aby se dříve dostali do rozsahu příštího sprintu
- Sprint Planning One
 - Této schůzky se účastní produkční tým a product owner. Product owner prezentuje jednotlivé položky v product backlogu a produkční tým na základě dostupných informací odhaduje náročnost jejich implementace. Na základě toho je stanoveno, kterými položkami se bude tým zabývat v tomto sprintu a budou tedy přesunuty do sprint backlogu a na které položky zřejmě nebude dostatečná kapacita a je třeba je ponechat v product backlogu do příštího sprintu. Též je možné nedostatečně vysvětlené či chybně zadané položky nechat přeformulovat či znovu zadat pro příští sprint.
- Sprint Planning Two
 - Této schůzky se účastní product owner a produkční tým. Společně prochází sprint backlog a detailně plánují aktivity, které je potřeba provést v daném sprintu tak, aby se během něj podařilo zdárně doručit všechny položky. Je třeba stanovit ideální pořadí, v jakém dojde k implementaci položek, aby byla dodržena případná logická návaznost. Po dohodě o plánu sprintu se přistupuje k implementaci jednotlivých položek podle plánu.
- Daily scrum/Standup
 - Každodenní schůzka produkčního týmu, product ownera a scrum mastera, kde je kladen důraz na to, aby byla co nejkratší a nejstručnější. Častou praktikou je, že

se schůzka jmenuje Standup, protože členové týmu u schůzky stojí, aby je nemožnost sedět motivovala mluvit stručně a jasně. Schůzka především slouží k operativní koordinaci v rámci sprintu. Každý člen týmu má sdělit ostatním, na čem pracoval od poslední schůzky, na čem bude pracovat do příští schůzky a co případně blokuje jeho hladký postup. Případně blokace řeší při schůzce se scrum masterem. (The Scrum Guide, 2020)

- Sprint Review

- Této schůzky se účastní jak členové produkčního týmu, tak zástupci zákazníka. Schůzka se odehrává na konci každého sprintu. Během schůzky jednotliví členové produkčního týmu prezentují přídavky doručené v rámci uplynulého sprintu a následně se zástupci strany zákazníka poskytují zpětnou vazbu. V rámci této schůzky jsou tedy přídavky ohodnoceny, zda je možné je považovat za doručené, či ne (definice dokončeného).

- Sprint Retrospective

- Této schůzky se účastní všichni členové produkčního týmu včetně scrum mastera a product ownera. Produkční tým hodnotí práci v uplynulém sprintu, diskutuje, co se podařilo doručit, které aspekty práce proběhly hladce a u kterých čelili překážkám. Tým se na základě této diskuze snaží identifikovat prostor pro zlepšení do dalšího sprintu či vytvořit novou praktiku, která umožní trvale zlepšit problematickou oblast. (Deemer, et al., 2010)

Ve scrumu se pracuje též s typickými technikami či objekty, které pomáhají standardizovat organizační procesy a maximalizovat dostupnost klíčových informací. Každý objekt se váže na zadaný cíl podle kterého je vyhodnocován a podle kterého se k danému objektu přistupuje.

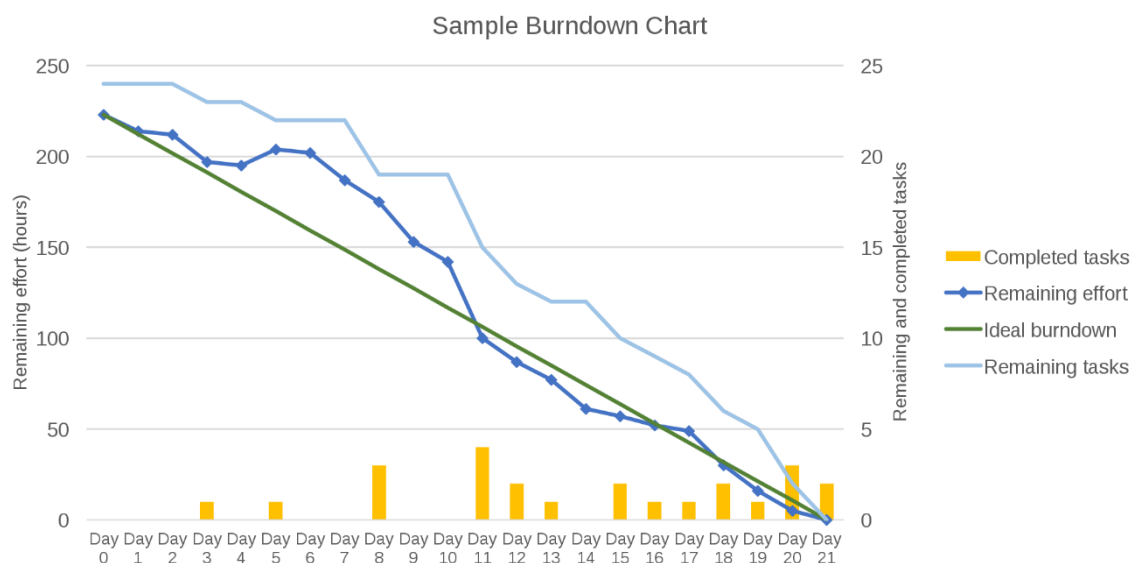
- Product backlog

- Jedná se o uspořádaný seznam všeho, co by mohlo být v budoucnu potřeba implementovat pro potřeby produktu a je tak jediným zdrojem požadavků (a práce) na produkční tým. O product backlog se stará product owner, který je jediný, kdo má možnost položky do něj přidávat, upravovat, mazat či měnit jejich prioritu. Pokud zákazník vyžaduje novou funkcionalitu, musí si ji nejdříve dojednat s product ownerem, který ji následně zařadí do product backlogu

- Produktový cíl
 - o Jedná se o dlouhodobý cílový stav produktu se všemi zvažovanými funkcionalitami. Všechny položky product backlogu vedou k dosažení produktového cíle a produkčnímu týmu tak slouží jako styčný bod.
- Sprint backlog
 - o Sprint backlog je podmnožinou product backlogu. Jedná se o souhrn všech položek, které mají být implementovány v rámci daného sprintu tak, aby mohl být splněn cíl sprintu. Sprint backlog je jednoduše přístupný všem členům produkčního týmu, pro které je hlavním zdrojem práce v daném sprintu podle toho, jak byl stanoven v rámci Sprint Planning Two.
- Cíl sprintu
 - o Tím je myšleno objektivní zlepšení plynoucí ze souhrnu přídaveků doručených během jednoho sprintu. Cíl sprintu není definován tak exaktně jako produktový cíl, protože v průběhu sprintu je možné narazit na řadu neočekávaných komplikací. Slouží především pro stanovení směru vývoje v daném sprintu a motivuje všechny členy produkčního týmu se společně podílet na jeho naplnění.
- Položka backlogu/user story/ Přídavek
 - o Jedná se o konkrétní objektivní požadavek, objem práce, který vývoj produktu posune blíže produktovému cíli. V rámci Sprint Planningu jsou jednotlivé položky backlogu ohodnoceny z hlediska pracnosti, diskutovány případné problémy, které by mohly nastat v průběhu jejich implementace a následně podle jejich popisu členové produkčního týmu pracují na konkrétní funkcionalitě. Termínem user story je myšlena technika, kdy je Položka backlogu popsána z pohledu budoucího uživatele dané funkcionality. Kvalita vypracování položky se sleduje pomocí definice dokončeného a v momentě, kdy položka splní definici dokončeného, je ji možné označit jako přídavek. (The Scrum Guide, 2020)
- Definice dokončeného (definition od done)
 - o Jedná se o kritéria kvality, po jejichž splnění je možné považovat položku backlogu za úspěšně doručenou a tedy za přídavek. Bez splnění definice dokončeného není možné prezentovat dokončenou položku zákazníkovi. Tuto definici stanovuje zvláště pro každou položku product owner ve spojení se zákazníkem. (Wibas GmbH Management consultancy, 2013)

Některé týmy rozšířily klasické pojetí Scrumu o další techniky, které jim pomáhají v orientaci v množství naplánované práce.

- Definice připraveného (definition of ready)
 - o Podobně jako definice dokončeného označuje stav, kdy je možné ukončit práce na položce backlogu, tak definice připraveného označuje stav, kdy jsou pro danou položku zpracovány veškeré relevantní podklady k tomu, aby na nich produkční tým mohl začít pracovat. De facto se tak jedná o kontrolu kvality zpracování user story, ze které čerpá výhody především produkční tým.
- Burndown/Burnup graf (Obrázek 5)
 - o Představují způsob, jak graficky vyjádřit množství zbývajících práce v současném sprintu. Na ose X je udáván čas ve sprintu a na ose Y množství zbývajících práce (v podobě položek backlogu nebo součtu hodin jejich odhadované pracnosti). Tento graf pomáhá produkčnímu týmu se jednoduše orientovat v tom, jak si v rámci daného sprintu vedou, zda jsou napřed proti plánu, nebo zaostávají a je třeba tuto situaci řešit. Liší se v tom, že v burndown grafu se množství zbývajících práce odčítá shora dolů a v burnup grafu zdola nahoru. Burndown graf je zřejmě rozšířenější. (Improving Software Development Using Scrum Model by Analyzing Up and Down Movements on The Sprint Burn Down Chart: Proposition for Better Alternatives, 2009)

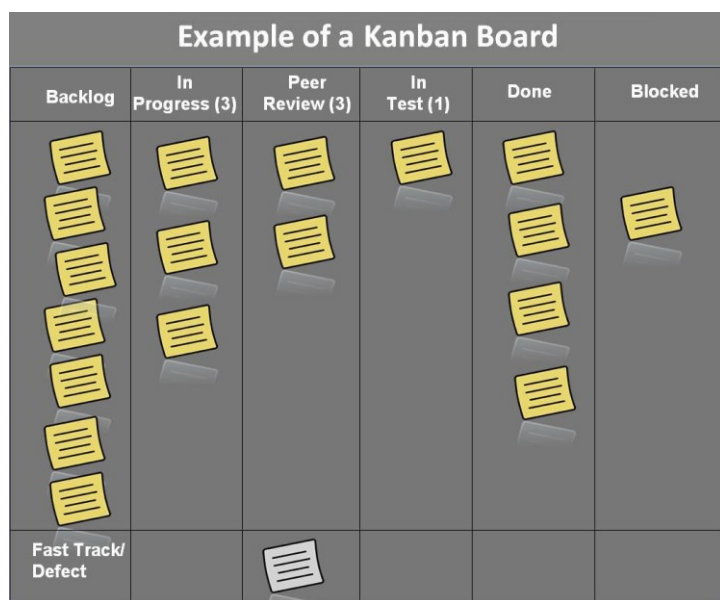


Obrázek 5: Burndown Chart, zdroj: wikipedia.org

3.5.3 Kanban

Kanban je technika, která se v rámci agilní metodiky používá pro vizualizace pokroku ve zpracování pracovních položek (aktivit). Nebyla však vyvinuta pro účely IT vývoje. Byla vyvinuta v Japonsku ve firmě Toyota pod vedením Taiichi Ona, na konci 40. let 20. století, kde se využívala pro optimalizaci logistických procesů po vzoru supermarketů tak, aby bylo možné držet pouze přesně tak velké zásoby surovin, jaké jsou nezbytně potřeba. Pro tento systém se tak vžilo označení JIT (Just in Time).

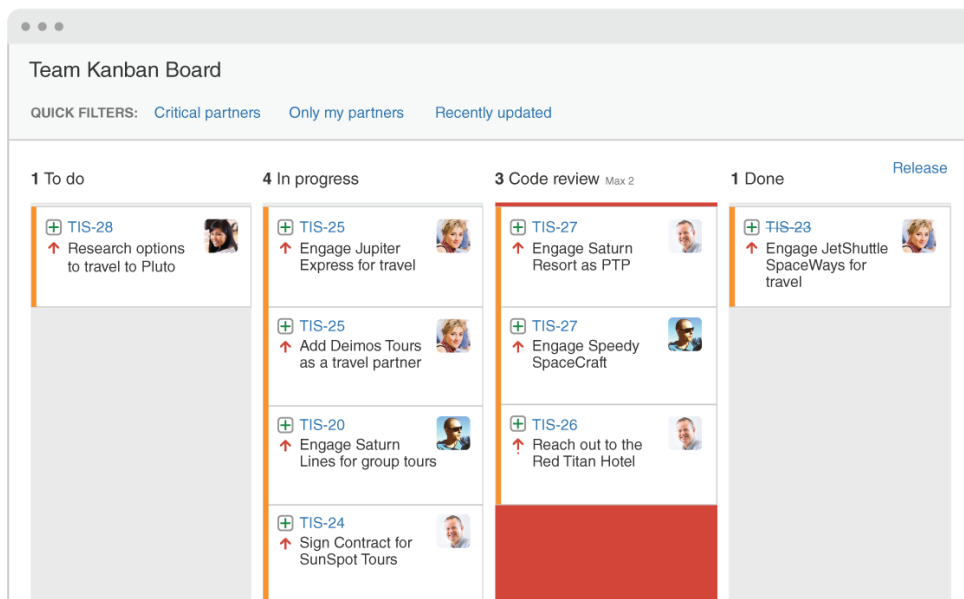
Středem Kanbanu v IT prostředí je tzv. Kanbanová tabule, která je využívána pro zobrazení jednotlivých položek sprint backlogu, na kterých produkční tým pracuje, toho, v jakém stavu práce na nich je a toho, jak důležitá daná položka v rámci sprintu je. Každá pracovní položka (nejčastěji ve formě jedné user story) bývá ukázána jako kartička, ke které se hlásí konkrétní člen produkčního týmu. Tím, že je někomu přiřazena je jednoznačně určeno, kdo je za její stav zodpovědný. Dále je tato kartička zařazena do jednoho z dostupných sloupců, což označuje stav jejího zpracování (Obrázek 6). Tyto stavy je možné definovat libovolně, ale nejčastěji se používají: V přípravě, Bude zpracováno, Ve zpracování a Dokončeno. V těchto stavových sloupcích se jednotlivé položky řadí odshora dolů podle své důležitosti.



Obrázek 6: Kanban tabule, zdroj: wikipedia.org

Díky tomuto systému je možné velmi rychle získat přehled o stavu prací v daném sprintu, o tom, v jaké oblasti jsou práce potenciálně pozadu či napřed, či zda je někdo v týmu nedostatečně využit nebo naopak přetížen. Pro zobrazování a práci s týmovým kanbanem je

možné buď používat klasickou tabuli s nálepkami nebo dedikovaný software, který umožňuje upravovat stav Kanbanu kdykoliv (Atlassian, 2022). Nejrozšířenějším programem pro organizaci Kanbanu je Jira od firmy Atlassian (Obrázek 7).



Obrázek 7: Kanban v SW Jira, zdroj: Atlassian

4 Vlastní práce

4.1 Partner z praxe

DHL IT Services je interní poskytovatel IT služeb skupině Deutsche Post DHL (DPDHL), která operuje ve 220 zemích světa. DPDHL je přepravní firma, sídlící v Bonnu v Německu. Deutsche Post zajišťuje poštovní služby v Německu a DHL zastává celosvětové kurýrní služby. Součástí korporátu je několik business divizí:

- Deutsche Post, eCommerce and Parcel – zajišťuje poštovní služby v Německu
- DHL Express – je největší divizí DPDHL a zajišťuje expresní přepravu zásilek po světě a je například majoritním vlastníkem několika aerolinií
- DHL Global Forwarding – Freight – zajišťuje přepravu zboží po zemi, tedy po silnicích a železnicích
- DHL Delivery Supply Chain – zajišťuje opakovanou strategickou logistiku pro zákazníky, provozuje sklady apod.

DHL IT Services pracuje na principu 24 hodin denně, 7 dní v týdnu, 365 dní v roce v datových centrech a kancelářích na třech kontinentech – Amerika (Mechanicsburg, Westerville a Tempe v USA; Mexiko; Kostarika; Brazílie), Evropa (Praha v ČR; Bonn a Darmstadt v Německu) a Asie (Cyberjaya v Malajsii; Chennai v Indii). DHL IT Services v Praze spravuje největší datové centrum v České republice. (DHL IT Services, 2022)

DHL IT Services se skládá ze dvou hlavních oddělení – Solution Delivery a Production Services. Solution Delivery je oddělení zodpovědné za dodávání řešení, tj. softwaru, know-how a konzultací pro potřeby jednotlivých divizí businessu. Oddělení Production Services je zodpovědné za poskytování podpory a udržování provozu interních i externích aplikací, udržování hardwaru i softwaru napříč celou DPDHL.

Solution Delivery se dělí na dvě hlavní části – „Solution Delivery Express, DGFF and DSC“ a „Solution Delivery P&P a ERP“. První zmíněná poskytuje přednostně systémy pro tři v názvu zmíněné DHL divize, druhá poskytuje software pro potřeby Deutsche Post. Tyto dvě části se dělí na další podorganizace, zde nazývané domény, kde již dochází ke skutečné organizaci práce a projektů. Tato diplomová práce bude zpracovávána v rámci domény SD PMO, Analytics and Tools, která se ve firemní hierarchii řadí pod Solution Delivery Express, DGFF and DSC.

4.2 Oddělení, ve kterém bude práce zpracována

Doména SD PMO, Analytics and Tools slouží především jako podpůrné oddělení pro všechny ostatní domény. V tomto oddělení se nevytváří žádný software na objednávku jednotlivých business divizí, jako v ostatních doménách, ale vytváří interní software (například ITSSOne) sloužící jako interní aplikace ostatním doménám pro správu IT Projektů a pro dohled nad kapacitním plánováním, sledováním KPIs a jiných analytických metrik. SD PMO, Analytics and Tools se skládá ze tří týmů – SD Analytics and Planning, SD PMO a SD Tools.

SD Analytics and Planning je tým skládající se z 6 členů. Tým zajišťuje pro celé oddělení Solution Delivery analytické a plánovací práce. Hlavní náplní práce týmu je analytický reporting, například pomocí platformy Microsoft PowerBI. Předmětem reportingu je pracovní nasazení zaměstnanců (míra přesčasů a efektivita práce), míra spokojenosti zákazníků s provedením projektů (metriky jako NPA), reportování dostupnosti lidí s určitými technickými dovednostmi a případné shromažďování podkladů pro proces vedoucí k otevření nových pracovních pozic v rámci Solution Delivery. Též je zde monitorována a plánována účast lidí na projektech, díky čemuž dochází k lepší optimalizaci pracovního výkonu lidí (pokud pro softwarového inženýra kódujícího v jazyku JAVA není uplatnění v jeho domovském týmu, tento tým mu hledá uplatnění na jiných projektech v rámci celého Solution Delivery).

SD PMO je projektová kancelář celého oddělení Solution Delivery, kterou tvoří 5 členů. SD PMO monitoruje a hlídá kvalitu projektového řízení. Lidé v tomto týmu evidují práci odvedenou na projektech, náklady na projekty, relevanci projektů a zpětnou vazbu zákazníků na práci odvedenou na projektech. Též zodpovídají za úvodní tréninky nastupujících projektových manažerů a vedou rozličná školení a konference projektových manažerů napříč celým Solution Delivery. Zároveň shromažďují požadavky a návrhy na zlepšení pracovních podmínek projektových manažerů.

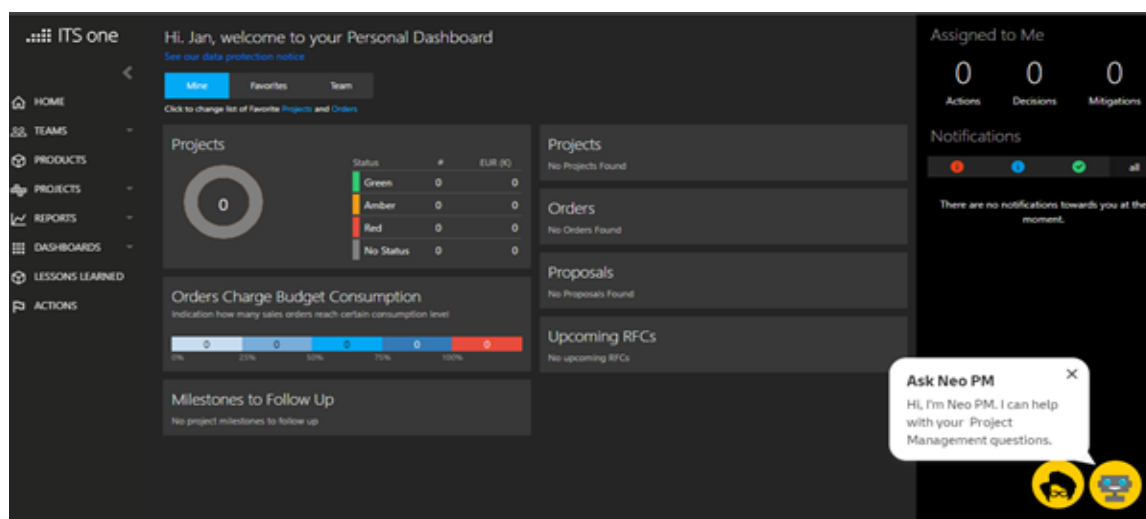
Třetím týmem v rámci domény je SD Tools. Tento tým je v rámci domény největší, zahrnuje 11 zaměstnanců. Členy týmu jsou vývojáři, konzultanti a testeři. Hlavní náplní práce týmu je vývoj a následná údržba klíčové interní aplikace ITSSOne, která pro potřeby Solution Delivery funguje jako souhrnná platforma, v níž je zastřešena administrace práce projektových manažerů, kapacitní plánování napříč odděleními, jsou v ní nalezeny výstupy analytického oddělení a další podpůrné a informativní nástroje využívané především

managementem v rámci Solution Delivery. Vývojářské práce jsou vedeny způsobem Agile a na příkladu tohoto týmu bude zpracována tato diplomová práce.

Pro zpracování práce jsou tedy velmi příhodné podmínky – tým, který zpracovává v agilním způsobu řízení interní aplikaci, úzce spolupracuje s projektovou kanceláří.

4.3 Hlavní produkt týmu – ITS One

ITS One je webová softwarová platforma zastřešující několik interních aplikací pro potřeby DHL IT Services (Obrázek 8). Přístup do ní a do jejích hlavních aplikací má každý zaměstnanec. Frontend platformy je postavený v Typescriptu a JavaScriptovém frameworku React, backend v Node.js, databáze je na interních MSSQL serverech. Pro každou aplikaci je stanovený její product owner, který komunikuje s jejími uživateli, na základě této komunikace tvoří požadavky pro další vývoj ve formě user stories a předává je týmu vývojářů pro zpracování. Důležitým faktorem je též sledování uživanosti jednotlivých aplikací pomocí Google Analytics.

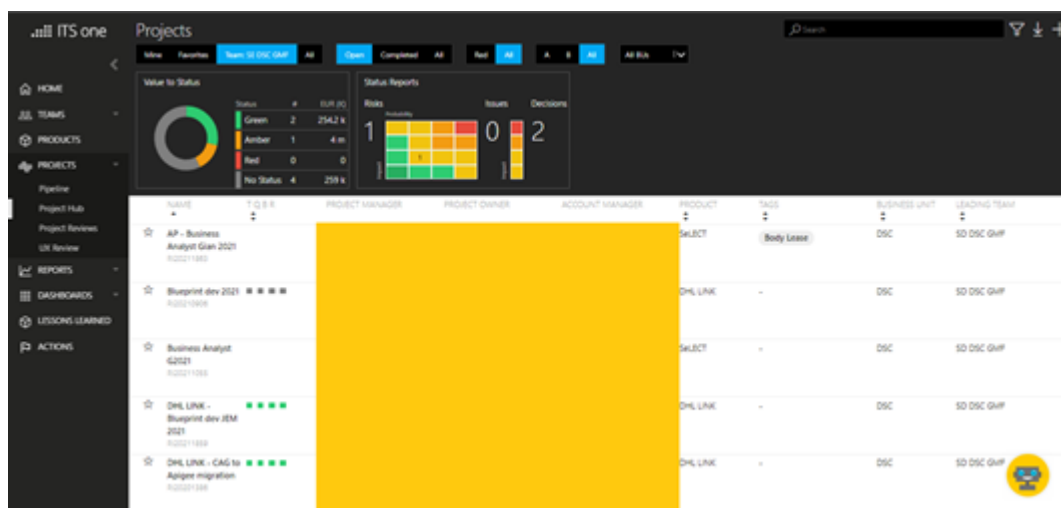


Obrázek 8: Úvodní stránka ITSONe, zdroj: Archiv autora

4.3.1 Project Hub

Nejdůležitější a nejvíce používanou aplikací je Project Hub, kterou vlastní lidé z PMO (projektové kanceláře) (Obrázek 9). Slouží pro administraci projektů, naceňování projektů, přehled pokroků v projektech. Též v něm mohou projektoví manažeři identifikovat a zapsat rizika, která vidí v průběhu životního cyklu projektu. Na základě vstupů z této aplikace probíhají jednání seniorního managementu, která slouží k revizi postupu v daných projektech. Nově se pro potřeby této aplikace vyvíjí zásadní komponenta, a to integrace do

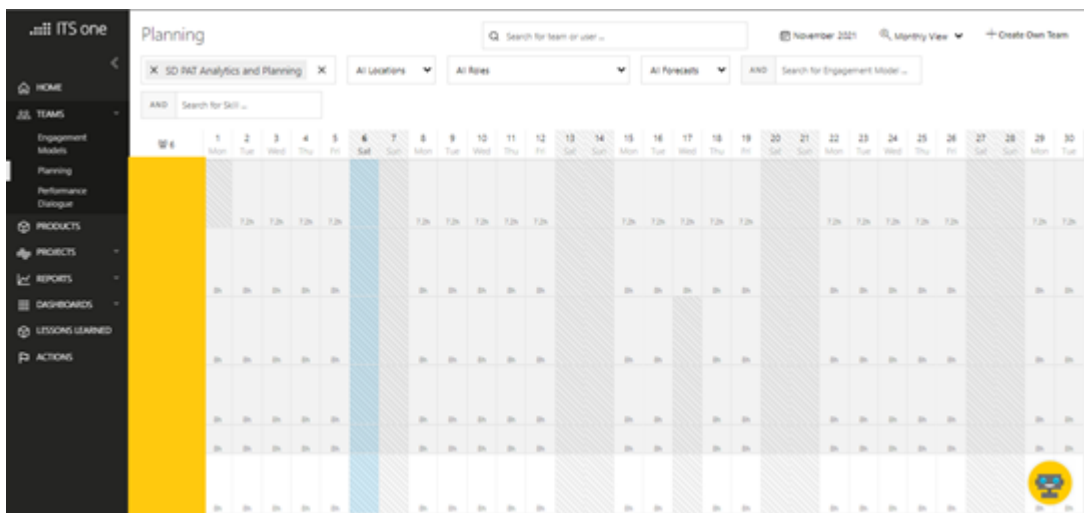
CRM systému Salesforce, což by umožnilo projektovým manažerům provádět a udržovat projektové účetnictví přímo v Project Hubu.



Obrázek 9: Project HUB, zdroj: Archiv autora

4.3.2 Planning Tool

Druhou nejvíce používanou aplikací je Planning Tool (Obrázek 10). Slouží k plánování a vizualizaci dlouhodobého zapojení zaměstnanců do projektů. Jejimi hlavními uživateli jsou projektoví manažeři, týmoví manažeři a lidé zodpovědní za interní kapacitní plánování. V Planning Tool lze zobrazit celou organizační strukturu, zadané schopnosti a zkušenosti jednotlivých zaměstnanců s prací s konkrétními technologiemi a časovou osu. Do časové osy si zaměstnanci zadávají své aktivity na projektech a lze tedy monitorovat, kolik procent času v následujících měsících stráví prací na kterém konkrétním projektu, nebo jestli mají volnou kapacitu. To je velmi důležité pro interní zaměstnance zodpovědné za kapacitní plánování, kteří v tomto toolu dohledávají a spojují poptávku po pomoci na projektech se zaměstnanci s dlouhodobou dostupnou kapacitou.



Obrázek 10: Planning Tool, zdroj: Archiv autora

4.3.3 Product Repository

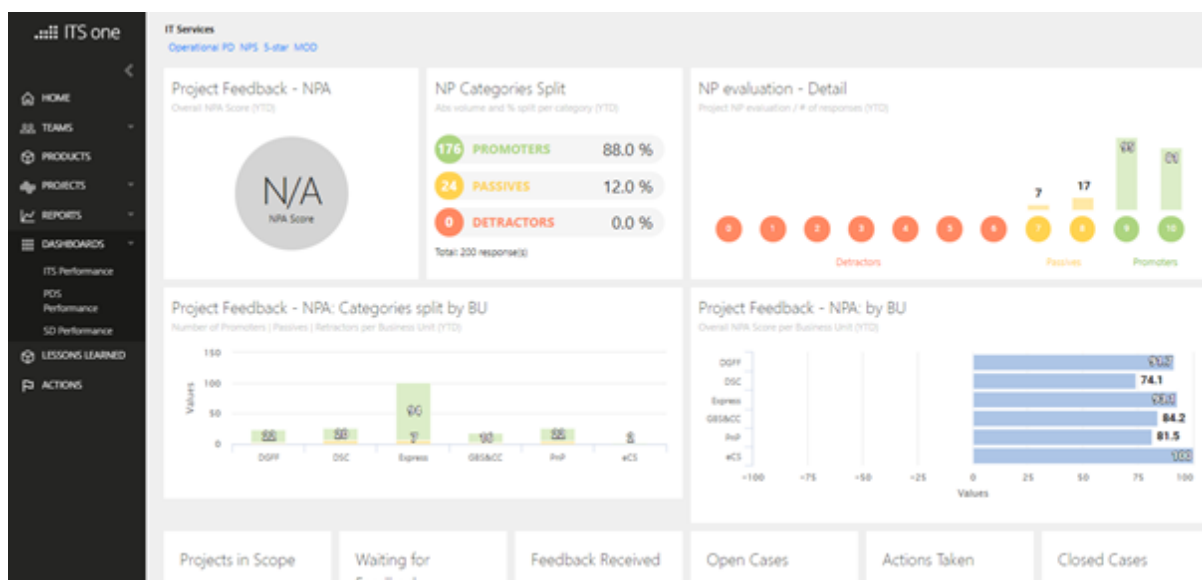
Třetí nejdůležitější aplikací je Product Repository, ve které jsou ukládány a administrovány všechny informace o produktech vyvinutých v IT Services (Obrázek 11). Zásadní je především pro následný reporting, protože interní produkty jsou klíčovou datovou položkou, která spojuje interní záležitosti, jako zapojení zaměstnanců do práce na projektech a najímání nových zaměstnanců, se záležitostmi externími, jako je poptávka po službách IT Services a podobně.

ID	NAME	SUBJECT MATTER EXPERT	BUSINESS UNIT	TAGS	ITS SD LEADING TEAM
2085	Abbott Hens		Supply Chain	SoS	
1236	ACA		Express	SWL/b1, ProductRepository, EGAP	SD EXP Clearance & Global Trading
10440	Accounting Monitor		Global Forwarding	ProductRepository	SD Data Services
2234	ACES		Express	ProductRepository, OCR	SD ERP Divisions CBI
2089	ACM		Express	ProductRepository, EGAP	
10532	ACS - 3P App		Express	ProductRepository	SD EXP Sent & Containerization
10430	AD Support DE		Corporate Center (CC), Express, Freight, Global Business Services, Global Forwarding, P&P, Supply Chain, eCommerce	Pos DE Infrastructure, Puretesting	Vir_CRM_PDS-other-RUN
10074	ADC		Express, Global Business Services	EGAP, ProductRepository	SD EXP CTC CBI - Global Billing
10396	Address API		Global Business Services	ProductRepository	SaaS & eCS ROW PRGeCS ROW PRG
1820	Address Management		Express	SWL/b1, ProductRepository, EGAP	SD EXP Address Data & Mobile Solutions
10341	Address Service		Express	SoS	eCom
10521	Administration ASIM UB P&P		P&P	ProductRepository	SD BER
2244	ADVANCE		Express	ProductRepository	SD EXP BI PRG
2207	AdvancedQCC		Express	ProductRepository	SD EXP Shipment Data & Quality
10361	AET_PUC_ETL		Express	SoS	Express Operations
10213	AFR Gateway Schedule		Global Forwarding	ProductRepository	SD DSC Solutions C
10252	AFR Visualization Tool		Global Forwarding	ProductRepository, SoS, Puretesting	DGFF
2189	AFSM tool		Supply Chain	ProductRepository	SD DSC Consulting PRG

Obrázek 11: Product Repository, zdroj: Archiv autora

4.3.4 Reporting Services

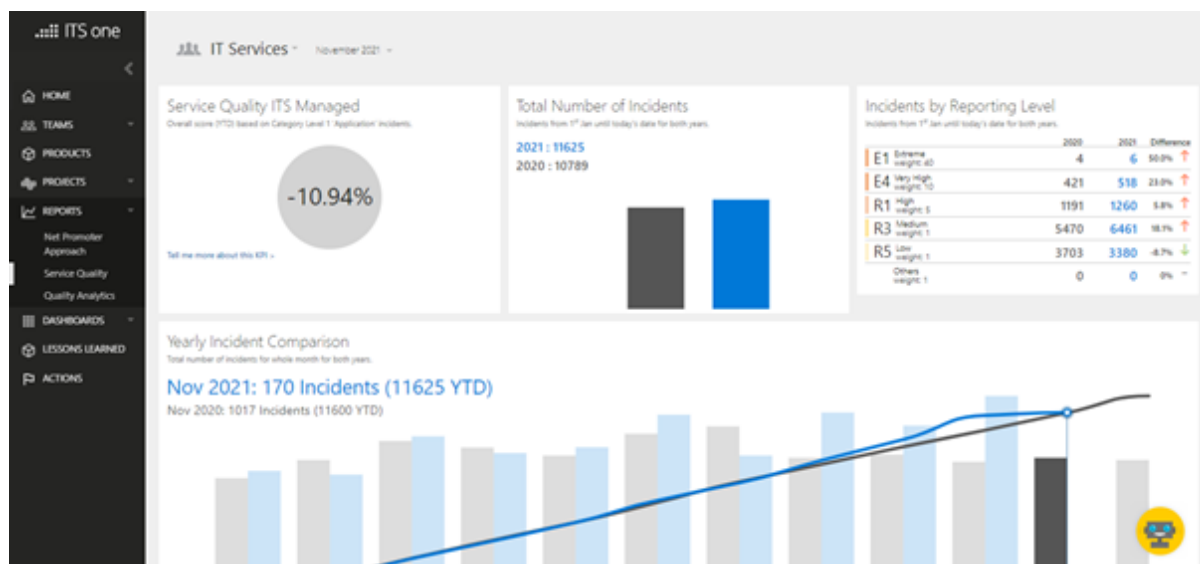
Další důležitou aplikací je Reporting Services, kterou využívá především seniorní management (Obrázek 12). Jednou složkou, která vystupuje nad ostatní a je zásadní pro monitorování fungování ITS, je NPS (Net Promoter Score). Net Promoter Score je KPI (Key Performance Index) vyvinuté pro měření loajality zákazníků firmy představené v roce 2003 článkem F. Reichhelda “The One Number You Need to Grow”.



Obrázek 12: NPS Reporting, zdroj: Archiv autora

4.3.5 Service Quality reporting

Kromě reportingu NPS má důležité místo i Service Quality reporting (Obrázek 13). Ten spočívá v analýze množství a závažnosti interních incidentů (či ticketů) vznesených kvůli nefunkčnosti nebo problémům spojeným s dodanými produkty. Jeden incident může vznést kdokoli v firmě a obsahuje informaci o tom, která služba a jakým způsobem nefunguje (či pokud funguje jinak, než je očekáváno). Též je v ticketu přiřazena informace o závažnosti problému stanovená na základě množství uživatelů, kterých se dotýká. Na základě všech těchto informací je vypočtena výsledná metrika Service Quality.



Obrázek 13: Service Quality Reporting, zdroj: Archiv autora

4.4 Produkční tým

Tým je složen z 11 členů, kteří sídlí celkem ve čtyřech lokacích – v Česku, v Indii, Malajsii a na Filipínách.

4.4.1 Manažer vývojového týmu

Manažer týmu je zároveň solution architektem, product managerem a hlavním styčným bodem, který stojí u produktu od počátku jeho vývoje. Vytváří architekturu, určuje vzhled, funkcionalitu a další směřování produktu. Dále je zodpovědný za výběr a nábor všech ostatních spolupracovníků pro tento produkt. Též píše user stories při iniciaci prací na nových strategických přírůstcích a určuje priority dalšího vývoje.

4.4.2 Konzultanti

Přímo do vývojářského týmu jsou zahrnuti dva konzultanti. Oba jsou zároveň subproduct ownery v rámci agilního procesu vývoje. Jejich náplní práce je monitorovat uživanost produktu, vytvářet user stories či pomáhat s jejich tvorbou subproduct ownerům. Pomáhají koordinovat práci a jsou zodpovědní za vedení některých meetingů v rámci agilního procesu vývoje. Též v případě potřeby vypomáhají s testováním přírůstku.

4.4.3 Databázový specialista

Zajišťuje a monitoruje, že operace s daty v rámci vývoje probíhají správně, a též dohlíží na dodržování GDPR regulí v rámci produktu.

4.4.4 Tester

Testuje všechny nově vyvinuté části softwaru. Všechny přídatky do kódu musí projít testovacím procesem, jehož smyslem je odhalit jakoukoliv nedokonalost před jejich nasazením do produkce. Tester je též zodpovědný za nasazování změněných částí softwaru do produkce.

4.4.5 Aplikační support

Řeší stížnosti uživatelů na nefunkčnost částí produktu. Pokud je schopen, sám tyto problémy řeší. Pokud není, předává požadavky na odstranění potíží formou user stories vývojovému týmu.

4.4.6 Developeři

Zbytek týmu je tvořen developery různých zkušeností bez rozdělení kompetencí a zaměření. Ovládají práci v JavaScriptu, TypeScriptu, SQL či Javě. Práci na user stories si rozdělují dle svých preferencí a přání.

4.4.7 Subproduct owněři

Vzhledem ke značné rozsáhlosti produktu není možné podchytit veškeré jeho potřeby uvnitř týmu. Proto se na vývojovém cyklu podílí i lidé, kteří nejsou přímo začleněni do vývojového týmu. Jednotlivé subaplikace ITSONe se vyvíjí jako produkt pro projektovou kancelář či pro analytický tým, a proto za každou z těchto jednotlivých aplikací (Project Hub, Planning Tool) vystupuje jeden člověk z těchto týmů jako subproduct owner, který zajišťuje komunikaci a předávání požadavků pro již fungující aplikace. Též identifikují oblasti a prostor pro nový vývoj pomocí user stories.

4.5 Proces vývoje produktu

Vstupy do procesu vývoje: Celkem jsou rozlišovány dva vstupy do procesu vývoje – strategické a operativní. Strategické vstupy určuje product manager. Product manager se

schází se seniorním managementem a zajišťuje určování strategických cílů produktu. Na těchto setkáních je určován směr a termíny pro nasazení požadovaných změn do tohoto produktu. Operativní vstupy určují především subproduct ownery na základě zpětné vazby od uživatelů jejich aplikací či od potřeby jejich týmů, které dané aplikace využívají.

Určení priorit: V aplikační rovině (Project Hub, Planning Tool) jsou za určování priorit vývoje odpovědní subproduct ownery daných aplikací. Ti hodnotí své user stories podle MoSCoW metodiky (Must, Should, Could, Won't). Následně jsou tyto požadavky vzešlé z aplikační roviny porovnány s požadavky ze strategické roviny (které mají přednost) a product manager rozhodne o zařazení konkrétních user stories do vývojového backlogu.

Iniciace vývojového procesu: Vývojový proces funguje na principu týdenních cyklů, takzvaných sprintů. Před začátkem sprintu product manager určí cíle následujícího sprintu. Pro klíčové části připraví user stories sám či přímo kontaktuje subproduct ownery, kterým vysvětlí, co má být obsahem dané user story, a ti ji následně vytvoří.

4.5.1 Plánování procesu

Proces vývoje se připravuje týden před jeho samotným zahájením. Pro tento účel jsou rozvrženy dvě schůzky – Backlog Pre-Review a Backlog Review. Cílem je týden před samotným zahájením vývoje rozhodnout, které user stories budou součástí sprintu začínajícího příští týden. Též se user stories ladí po formální stránce a rozhoduje se, zda jsou z hlediska produktu relevantní. Zásadním nástrojem, který je při plánování používán, je product backlog. Product backlog je udržován v nástroji GitHub for Enterprise.

user story označuje v kontextu DHL IT Services funkcionalitu Issue dostupnou v nástroji GitHub for Enterprise. user story je základní způsob komunikace product ownerů s vývojovým týmem. Pro potřebu vývoje a udržování systému rozeznáváme dva typy user stories – Feature a Bug.

Feature označuje popis nového požadavku na vývoj softwaru. Může popisovat novou funkci, úpravu stávající funkce na základě změněných požadavků nebo i odstranění funkce.

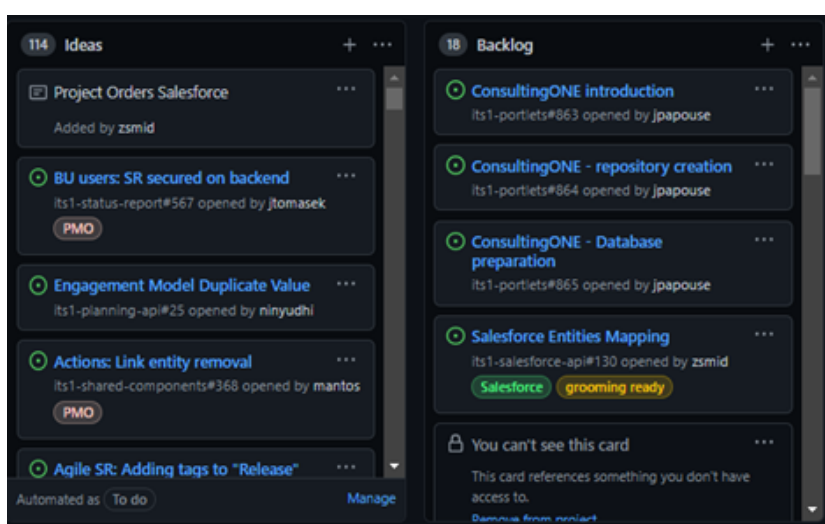
- Při vytváření feature se dodržují následující pravidla:
 - Každá user story začíná popisem funkce ve stylu:
 - “As a user/admin/project manager...” - vysvětlení kdo žádá
 - “...I need to add a field to the taskbar...” - vysvětlení co žádá
 - “...therefore, I am able to filter the document” - proč to žádá
 - Pokud je potřeba, následuje detailní techničtější popis funkcionality (jaká data, jaké agregace apod.).
 - Na závěr jsou přidána kritéria pro přijetí user story jako dokončené:
 - “This story is accepted when filter works in 100 % of use cases.

Bug označuje popis požadavku na odstranění chyby v softwaru. Tato chyba mohla být odhalena buď v průběhu UAT (User Acceptance Testing, uživatelské akceptační testování) nebo dále v průběhu používání softwaru.

- Každá user story označená jako Bug obsahuje tři odstavce:
 - Replication of the bug – návod pro developera, jak napodobit situaci, při které vznikla chyba, může se jednat o odkaz či sekvenci kroků.
 - Observed behavior – popis chování, které chyba vykazuje, ideálně doplněná přímo o obrázky chybného chování.
 - Expected behavior – popis chování, jaké by měla funkcionality správně vykazovat.
- Dále je možné v případě potřeby doplnit o detailnější technické specifikace.

4.5.2 Product backlog

Product backlog se sestává z user stories uspořádaných do dvou sloupců – Ideas a Backlog (Obrázek 14). Ideas je místo, kam se dávají veškeré user stories hned po jejich vytvoření. V tomto sloupci probíhá jejich ladění a zůstávají v něm, dokud jejich owner nerozhodne o přesunutí do sloupce Backlog. Ve sloupci Backlog jsou user stories ohodnoceny product managerem z hlediska jejich relevantnosti. Když projdou kritérii, jsou označeny tagem Grooming Ready, což znamená, že budou prezentovány vývojovému týmu a zařazeny do následujícího sprintu.



Obrázek 14: Product backlog, zdroj: Archiv autora

4.5.3 Průběh schůzky Backlog Pre-Review

Tato schůzka se koná vždy v úterý ráno a trvá maximálně 30 minut. Účastní se jí všichni subproduct ownéři a příležitostně product manager. Schůzka je vedena a organizována dle dohody jedním z interních subproduct ownerů (konzultantem). Jednotliví subproduct ownéři prezentují své user stories, které mají v product backlogu ve sloupci Ideas. user stories jsou buď označeny jako vyhovující a přesunuty do sloupce Backlog, nebo jsou jako nevyhovující ponechány ve sloupci Ideas, kde čekají na úpravy.

4.5.4 Průběh schůzky Backlog Review

Tato schůzka se koná vždy ve čtvrtek ráno a trvá maximálně 60 minut. Účastní se jí všichni subproduct ownéři a product manager. Schůzka je vedena a organizována dle dohody jedním z interních subproduct ownerů (konzultantem). Jednotliví subproduct ownéři

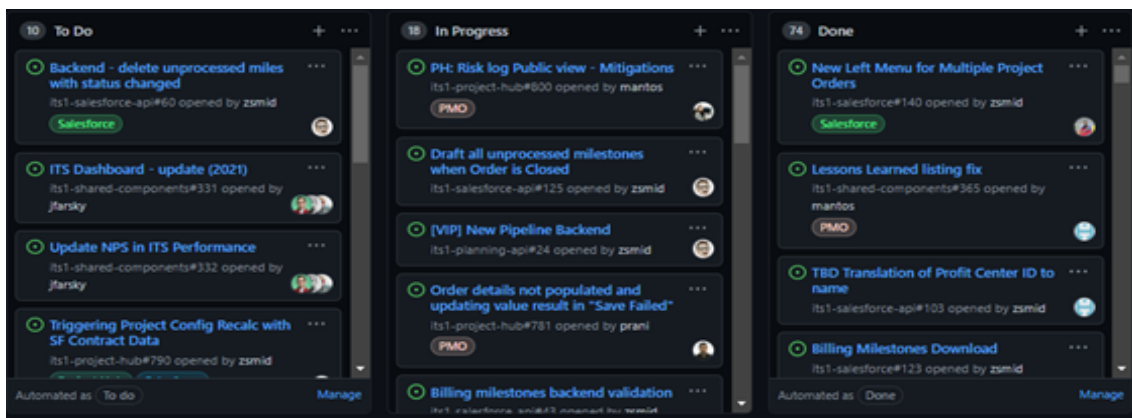
prezentují své user stories, které mají v product backlogu ve sloupci Backlog. Jejich cílem je stories obhájit před product managerem, který je následně, pokud je s jejich obsahem a formou spokojen, označí tagem Grooming ready. Nevyhovující stories jsou vráceny do sloupce Ideas k formálnímu přepracování. Touto schůzkou je ukončeno plánování vývojového procesu, který pokračuje další týden už samotným vývojem.

4.5.5 Sprint

Sprint trvá jeden pracovní týden. Začíná v pondělí definicí a vysvětlením obsahu, který má být daný týden vytvořen. Pro tento účel je plánována schůzka s názvem Backlog Grooming, během níž jsou user stories ohodnoceny z hlediska předpokládané náročnosti pomocí nástroje Plan IT Poker. Podle této informace si jednotlivé stories rozdělí příslušní developéři na základě svých znalostí, preferencí a přání. Každý z následujících dnů probíhá krátká schůzka zvaná Daily Standup, kde každý developer popisuje status své práce. Sprint končí v pátek shrnutím týdenního postupu na schůzce Sprint Retrospective, kde je popsán prostor pro zlepšení pro příští sprint. Sprint je plánován a udržován v nástroji sprint backlog.

4.5.6 Sprint backlog

Sprint backlog slouží k mapování postupu v rámci konkrétního sprintu. Sestává se ze třech sloupců – To Do, In Progress a Done (Obrázek 15). Do sloupce To Do přechází user stories z product backlogu poté, co mají příznak Grooming ready a jsou ohodnoceny podle své náročnosti v nástroji Plan IT Poker. Z tohoto sloupce na začátku každého týdne developéři vybírají, na kterých user stories budou daný týden pracovat. Výběr provedou přiřazením sebe do popisu user story přímo v GitHubu. Po tomto přiřazení přesunou user story do sloupce In Progress, v něm zůstává do té doby, než je na něm práce oficiálně dokončena, což znamená splnění vývojových kritérií, revize kódu dalšími dvěma developery a otestování testerem. Po ukončení tohoto procesu je story přesunuta do sloupce Done, kde je v případě potřeby rozebírána na schůzce Sprint Retrospective.



Obrázek 15: Sprint backlog, zdroj: Archiv autora

4.5.7 Určení náročnosti user story

user stories jsou ohodnoceny z hlediska náročnosti během schůzky Backlog Grooming. Jedná se o stěžejní součást vývojového procesu, protože na základě něj si developeři rozdělují user stories pro práci v současném sprintu. Pokud je story špatně naceněna, může se stát, že ji developer nestihne zpracovat, nebo naopak ji zpracuje příliš rychle a následně bude nějaký čas nevyužit. Proces nacenění začíná tím, že product manager nebo subproduct owner vybere user story s tagem Grooming ready a prezentuje ji všem přítomným developerům. Ti mají možnost vznášet dotazy na požadované výstupy a přemýšlejí nad způsoby implementace. Následně se product manager a všichni developeři přihlásí do online nástroje Plan It Poker, kde developeři hlasují anonymně v jeden čas o náročnosti implementace na stupnici: Coffee -> XS -> S -> M -> L -> XL. Na hlasování mají přibližně deset sekund a po uplynutí času product manager vyhlásí výsledky hlasování. Při jednohlasné shodě je story okamžitě ohodnocena výslednou náročností, při nejednoznačné shodě mají možnost ti, kteří hlasovali pro menšinou možnost, vysvětlit ostatním své důvody. Při nejednoznačném výsledku se obvykle pro jistotu volí varianta vyšší náročnosti, aby nedocházelo k přetěžování developerů.

4.5.8 Životní cyklus user story

User story je vytvořena jako objekt Issue v nástroji GitHub v repozitáři, v němž se nachází zdrojový kód části aplikace, které se user story věnuje. Poté, co si user story přiřadí k sobě daný developer, začne pracovat na úpravách kódu u sebe na počítači. Každý developer má povinnost alespoň jednou denně nahrát všechny změny, které za den provedl, ze svého počítače do GitHubu ke zbytku zdrojového kódu. Tento upload se označuje jako

commit. Jednotlivé změny kódu lze sdružit jako pull request a připojit k původní user story. V tom momentě je user story k dispozici dalším developerům pro revizi, většinou story revidují developeři, kteří mají již zkušenosti s danou částí aplikace. Cílem revize je odstranění chyb v kódu ještě před samotným začátkem testování. Jakmile pull request obdrží dvě kladná hodnocení, případně jsou-li odstraněny všechny námitky revizorů, story přechází pod testera. Tester nahraje verzi kódu (aplikace) s provedenými změnami na testovací server, kde testuje, zda byly splněny všechny požadavky. Projde-li story testováním, tak v případě, že se jedná o rozsáhlejší změny, přichází na řadu uživatelské testování. Pokud všechny testy proběhnou bez problémů, změny v kódu jsou sloučeny s hlavní, produkční verzí a nahrány do produkce (verze, kterou vidí uživatelé). Nahrání nové verze do produkce se děje většinou jednou týdně mimo hlavní období vytíženosti produktu a zahrnuje více schválených změn najednou a označuje se jako release. Po nahrání do produkce je user story přesunuta do sloupce Done a končí tak její životní cyklus.

4.5.9 Backlog Grooming

Jedná se o schůzku, kterou začíná týdenní sprint. Koná se vždy v pondělí v devět hodin ráno a účastní se jí product manager, subproduct ownéři a developeři. Je rozvržena na dvě hodiny. Během této schůzky jsou developerům prezentovány a vysvětlovány user stories s tagem Grooming ready. Cílem je popsat specifikace a motivace stojící za konkrétními user stories či nastínit možné problémy při jejich implementaci. Zároveň je potřeba zodpovědět developerům všechny otázky týkající se stories, aby byli schopni vyhodnotit náročnost implementace. Pokud již nejsou žádné dotazy ani doplňující odpovědi, přechází se k hlasování o náročnosti, jak je popsáno v kapitole Určení náročnosti user story. Po určení náročnosti je story přesunuta do sloupce backlogu To Do a schůzka pokračuje popisem další story. Schůzka nemusí nutně trvat dvě hodiny, v případě potřeby bývá ukončována dříve.

4.5.10 Daily Standups

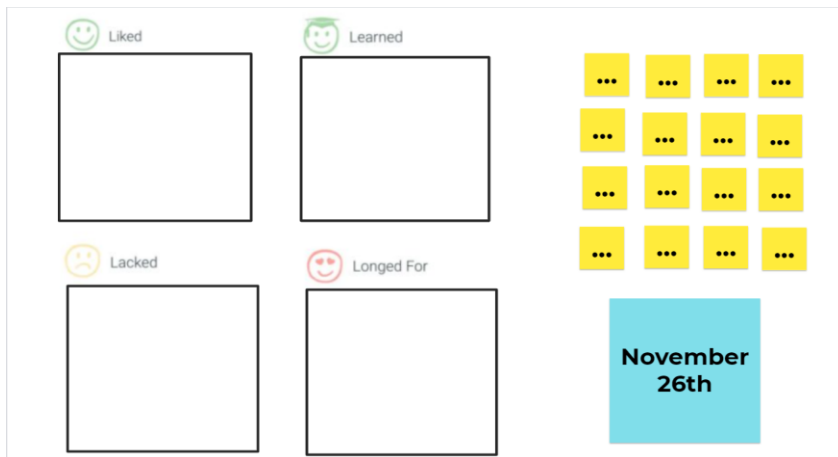
Daily Standup je schůzka, která se odehrává v úterý, ve středu a ve čtvrtek od deseti hodin. Jedná se o krátkou, maximálně třicetiminutovou schůzku, které se účastní product manager, subproduct owněři a developeři. Všichni účastníci by měli během schůzky stát a mít zapnutou kameru, aby se zvýšila její autenticita a jedinečnost. Odtud pochází název Daily Standup. Během schůzky každý ze členů týmu odpovídá na tři otázky.

- 1) Na čem jsem doteď pracoval?
- 2) Na čem odteď budu pracovat?
- 3) Narazil jsem na nějaké překážky?

Cílem je, aby všichni členové týmu měli přehled o tom, na čem ostatní pracují a případně si mohli navzájem pomoci. Též lze schůzku brát jako možnost pro komunikování změny priorit ve vývoji či jiných ad-hoc záležitostí.

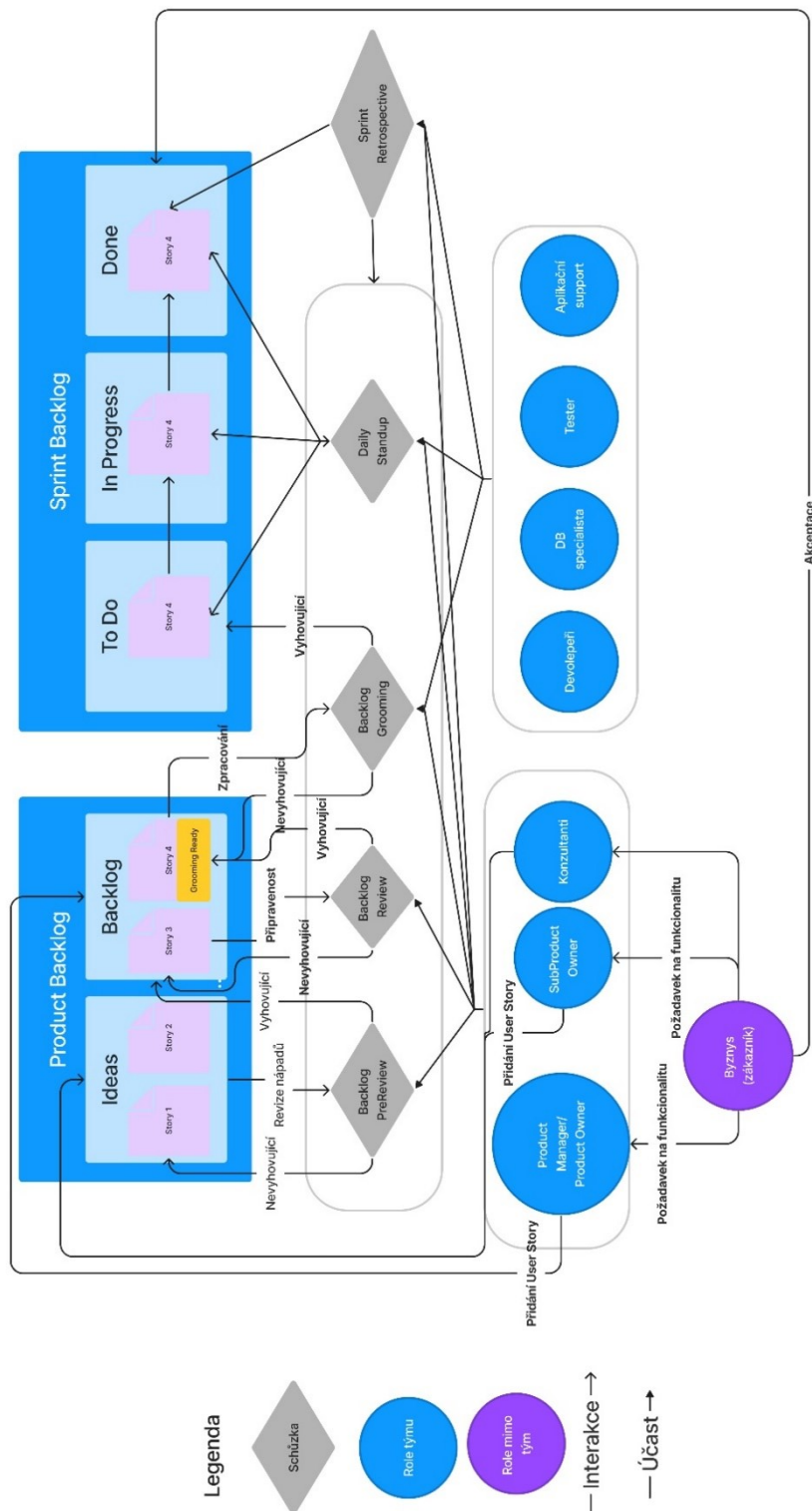
4.5.11 Sprint Retrospective

Sprint Retrospective je schůzka, která se odehrává v pátek od deseti hodin. Jedná se o půlhodinovou schůzku, během níž mají všichni členové, kteří se podíleli na vývoji v rámci tohoto sprintu, využít prostor pro poskytnutí zpětné vazby. Pro tento účel se používá nástroj JamBoard od Google, kde má každý možnost anonymně napsat svůj komentář do jednoho ze čtyř oddílů (Obrázek 16). Oddíly jsou: Liked, Learned, Lacked a Longed for. Do oddílu Liked má každý možnost vložit například pochvalu pro dalšího člena týmu nebo dát najevo, co se mu ve stávajícím sprintu líbilo. V oddílu Learned mají všichni prostor podělit se s ostatními o to, co se během tohoto sprintu naučili. Oddíl Lacked slouží pro vylíčení zklamání či prostoru pro zlepšení pro příště, například pokud nejsou developeři spokojeni s kvalitou připravovaných user stories, mohou tak nejučinněji udělat zde. Oddíl Longed for slouží jako místo pro návrhy na zlepšení, avšak bez negativního podtextu. Tato schůzka není většinou naplněna relevantním obsahem, protože jednotliví členové týmu se poměrně zdráhají dávat zpětnou vazbu.



Obrázek 16: JamBoard pro Sprint Retrospective, zdroj: Archiv autora

Na následujícím diagramu (Obrázek 17) je možné najít komplexní popis procesu vývoje softwaru u partnera z praxe.



Obrázek 17: Procesní diagram, zdroj: Archiv autora

4.6 Obecná charakteristika situace u partnera z praxe

Během pozorování pracovní atmosféry a zvyklostí u partnera z praxe se objevily určité náznaky, že některé zavedené pořádky nefungují podle očekávání. Tyto náznaky budou rozebrány v rámci této kapitoly a budou použity jako podklady pro vypracování dotazníkového šetření, které si bere za cíl zjistit názor členů produkčního týmu na tuto problematiku.

Jedním z klíčových varovných signálů byla skutečnost, že situace, kdy byla zpožděna dodávka softwarové funkcionality, nebyla ničím výjimečná, často se objevovalo zpoždění přibližně o jeden týden proti původnímu předpokladu u menších přídavků, v případech komplexnějších problémů se jednou vyskytlo i zpoždění v řádu měsíců. Je ale potřeba zmínit, že poměry v týmu nebyly jedinou příčinou tohoto velkého zpoždění (změny u zákazníka).

Za další varovný signál lze považovat situaci, která nastala hned dvakrát nezávisle na sobě, kdy bylo svoláno uživatelské akceptační testování (forma testování, kdy je funkcionality zpřístupněna uzavřenému okruhu jejích budoucích uživatelů pro odhalení skrytých nedostatků) kvůli novému softwarovému přídavku a testování se muselo hned v zárodku přerušit, protože služba byla v daný moment nepoužitelná. Znamená to, že zřejmě selhalo či vůbec neproběhlo testování základních funkcionalit.

Dále se objevily náznaky nedostatečné komunikace mezi všemi členy týmu, konkrétně vysoká zdrženlivost developerů v aktivním zapojování se do týmových tematických debat či v poskytování zpětné vazby na proces, což mnohdy vedlo ke značné neefektivitě využití času v naplánovaných schůzkách.

Toto jsou všechno náznaky, že v současném týmu by mohlo být vhodné zauvažovat nad změnou zavedených pořádků, mimo jiné pro zlepšení pracovních výkonů. Prvním krokem bude porovnání současného stavu s teorií agilního projektového řízení a následně vytvoření dotazníkového šetření, které si bude klást za úkol detailně rozkrýt problémové oblasti v současném stavu implementace Agile.

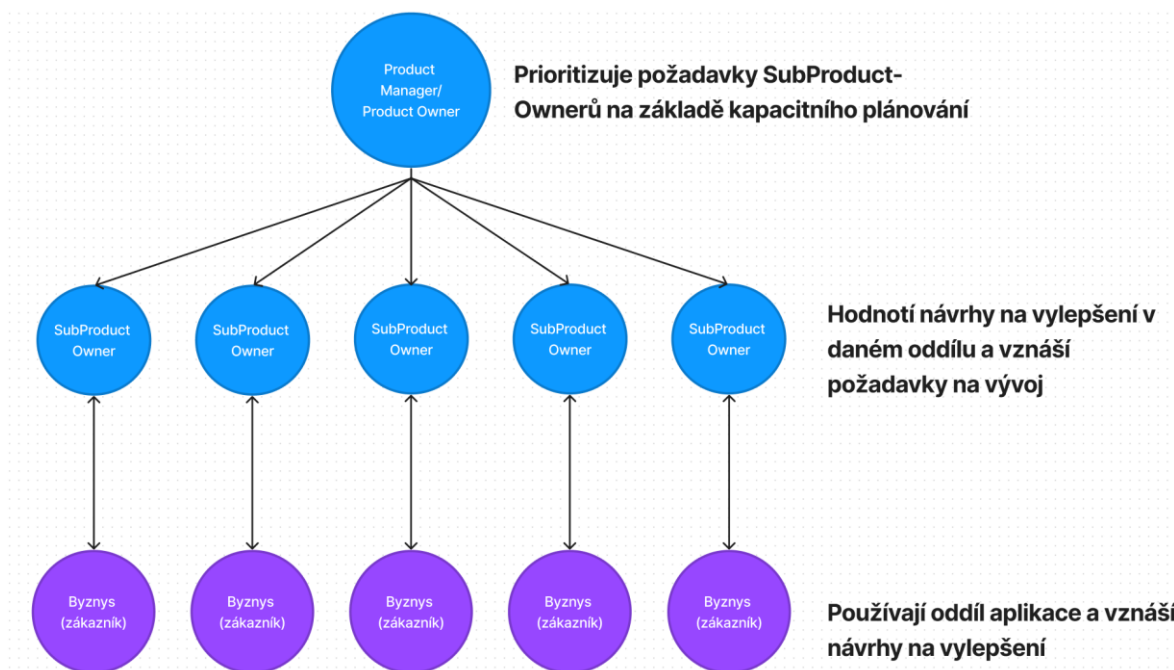
4.7 Porovnání současného stavu s teorií implementace Agile

Jak již bylo uvedeno v teoretické části práce, agilní projektové řízení se vyznačuje typickými rolami, schůzkami a technikami. Každý z těchto objektů má jasně stanovený účel, kterým přispívá ke správné a bezproblémové implementaci Agile. Při nedodržení těchto navržených postupů hrozí, že celý proces nebude fungovat způsobem, jakým má, a bude pro členy týmu přítěží. V této kapitole bude porovnána zde popsaná implementace Agile v týmu SD Tools s dostupnou teorií a na základě tohoto porovnání budou navrženy změny, které by měly zlepšit současný stav.

4.7.1 Porovnání týmových rolí s teorií Agile

V rámci teorie Agile se vyskytují tři hlavní role – product owner, scrum master a člen vývojového týmu. Podle teorie je každá z těchto rolí nezastupitelná a pro agilní vývoj softwaru nepostradatelná.

Pozice product ownera by podle teorie měla náležet striktně jednomu člověku, jehož hlavním úkolem je maximalizovat přinesené přidané hodnoty z práce týmu, viz kapitola 7.5.2. V rámci posuzovaného produkčního týmu je kvůli značné rozsáhlosti aplikace ITSSOne funkční systém, kdy za jednotlivé subaplikace, jako jsou Planning nebo Reporting, zodpovídají subproduct ownery, kteří jsou v kontaktu s uživateli daných oddílů. Na základě toho vznášá návrhy na zlepšení a vytváří požadavky (user stories) na nový vývoj. Ty přidávají do sekce Ideas v rámci product backlogu. Následně jsou takto vzešlé požadavky od všech subproduct ownerů prioritizovány a přesouvány do sekce „Backlog“ product backlogu k dalšímu zpracování na základě rozhodnutí hlavního product ownera. Celý proces je nakreslen na následujícím diagramu (Obrázek 18). Ve výsledku tedy směřování a vývoj celé aplikace skutečně určuje jeden člověk (product manager), ale proces je ve srovnání s teorií výrazně složitější, protože musí zpracovávat podklady od subproduct ownerů, kteří jsou jediní v přímém kontaktu s uživateli. Vzhledem k rozsahu aplikace, o kterou se produkční tým stará, zřejmě neexistuje způsob, jakým by požadavky ze všech oblastí mohla pojmout jedna osoba, a proto je zde nastavené dělení role product ownera žádoucí. Nese to však značně zvýšené nároky na komunikaci a synchronizaci mezi ownery. V kapitole zabývající se technickými nedostatky budou na tuto oblast položeny otázky za účelem zjistit, zda v týmu existují nedostatky v komunikaci, řízení požadavků apod. Ty se budou snažit konkrétně rozkrýt současnou situaci, případně nastínit prostor pro zlepšení.



Obrázek 18: Popis práce product ownerů v produkčním týmu, zdroj: Archiv Autora

Pozice scrum mastera by podle teorie měla usnadňovat ostatním členům týmu průběh práce, viz kapitola 7.5.2. V produkčním týmu však pozice scrum mastera není ustanovena. Část z běžných kompetencí scrum mastera přebírají ostatní členové týmu. Product manager vysvětluje nově příchozím členům týmu způsob implementace Agile a vede schůzku Backlog Grooming. Konzultanti vedou ostatní schůzky a snaží se odstraňovat překážky pro další členy týmu, jsou však často limitováni nedostatkem času na tuto aktivitu. Neexistence dedikované role scrum mastera tak vede ke skutečnosti, že pokud produkční tým narazí při vývoji na překážku, musí si s ní poradit sám, což neumožňuje stoprocentní koncentraci na dodávku přídatku. Členové produkčního týmu tak ztrácí čas, a dokonce mohou být i demotivováni od současné práce. Toto je jednoznačně slabé místo implementace Agile v produkčním týmu a v rámci této diplomové práce budou rozpracovány návrhy na zlepšení.

Pozice členů produkčního týmu by podle teorie měla zahrnovat řadu různých technických rolí a být stoprocentně dedikovaná a odpovědná za práci na přídatcích, viz kapitola 7.5.2. V produkčním týmu jsou zastoupeny všechny technické role tak, jak nabádá teorie implementace Agile. Na rozdíl od teorie však není možné zajistit stoprocentní zaměření na dodání přídatku, protože neexistence role scrum mastera neumožňuje delegovat problémy nesouvisející s prací na přídatku na jeho osobu. To však není vina nastavení práce v produkčním týmu. Tato záležitost je řešena v předcházejícím odstavci. Vzhledem k problémům s testováním a zdržením softwarových dodávek, jak bylo nastíněno v kapitole

8.6, bude do dotazníkového šetření zahrnuta skupina otázek, které se budou věnovat nastavení pracovních procesů, organizaci práce a technologickému zázemí produkčního týmu. Cílem těchto otázek bude získat detailní zpětnou vazbu na nastavení Agile v této oblasti, která bude moci být použita pro návrh zlepšení.

4.7.2 Porovnání týmových schůzek s teorií Agile

V rámci teorie Agile se vyskytuje 6 schůzek: Sprint Planning One, Sprint Planning Two, Backlog Grooming, Daily Scrum, Sprint Review, Sprint Retrospective. Tyto schůzky mají jasně danou audienci a účel. V produkčním týmu se pravidelně opakuje následujících 5 schůzek: Backlog Pre-Review, Backlog Review, Backlog Grooming, Daily Standup, Sprint Retrospective. Náplň schůzek se oproti teorii zásadně liší, především z důvodu přizpůsobit proces tvorby user stories situaci s více product ownery – Backlog Pre-Review a Backlog Review jsou schůzky zaměřené čistě na definování přídavek v patřičné kvalitě pro následující sprint.

V rámci schůzky v týmu zvané Backlog Grooming můžeme z teorie najít obsah schůzek Sprint Planning One, Sprint Planning Two a Backlog Grooming, kde dochází k ohodnocení náročnosti zpracování user stories, domluvě na rozsahu práce v následujícím sprintu a případnému detailnímu rozplánování postupu. Schůzka bývá poměrně dlouhá, ale přehledně plní svůj účel.

Schůzka Daily Standup plní stejný účel jako schůzka definovaná v teorii Agile pouze s tím rozdílem, že tady se jí mohou účastnit i členové širšího týmu (subproduct ownery), což může negativně ovlivňovat výkon produkčního týmu (nerelevantní témata probíraná v rámci Standupu, narušování „intimity týmu“ nečleny apod.). Toto lze považovat za slabé místo implementace Agile a zlepšení bude rozvedeno v následující kapitole.

Schůzka Sprint Retrospective slouží podobně jako v teorii Agile k definování míst pro zlepšení v příštím sprintu. Jedná se o způsob získávání vzájemné zpětné vazby v produkčním týmu. Na schůzky jsou však zváni též členové širšího produkčního týmu, což má podobné důsledky, jaké jsou uvedeny v předchozím odstavci. To se často projevuje neochotou produkčního týmu se do dění v této schůzce zapojovat.

Schůzka Sprint Review se v produkčním týmu vůbec nekoná. To vyplývá z nastavení, že definice dokončeného je určována ad-hoc na základě domluvy člena produkčního týmu s autorem user story (subproduct ownerem). Vzhledem k povaze

spolupráce, kde jednotlivé user stories vycházejí z návrhů uživatelů a ne z požadavků zákazníka, je toto rozdělení pochopitelné. Na druhou stranu existence této schůzky by týmu umožnila v rámci firmy lépe propagovat svou práci.

4.7.3 Porovnání týmových technik s teorií Agile

V rámci produkčního týmu jsou využity následující techniky: product backlog, sprint backlog, user story, definice dokončeného a definice připraveného. Oproti teorii Agile nevyužívá produktový cíl a Burnup/Burndown chart.

Product backlog je u partnera z praxe na rozdíl od klasického backlogu rozšířen o pole Ideas, do nějž vkládají nové položky všichni subproduct ownři tak, aby je následně product manager mohl podle priorit schválit. Je udržován paralelně vedle sprint backlogu, ve kterém se udržuje přehled o stavu zpracování user stories v aktuálním sprintu. Vzhledem k většímu množství kategoriích, které jsou vedle sebe v obou backlozích, a mnoha různým nálepkám, kterými se označují jednotlivé user stories, se stává takto komplexní backlog poněkud nepřehledným. V průběhu dotazníkového šetření bude v kategorii řízení požadavků očekáváno bližší zhodnocení funkčnosti takového nastavení.

User stories v rámci týmu fungují obdobně jako v teorii Agile. V případě týmu označují jeden konkrétní návrh (množství práce), který je potřeba vykonat pro doručení přídatku. V rámci kategorie řízení požadavků následujícího dotazníkového šetření bude možné zjistit, zda nastavená pravidla pro tvorbu user stories vyhovují všem členům produkčního týmu.

Definice dokončeného a definice připraveného se používají obdobným způsobem, jako předesílá teorie Agile. Definici dokončeného v produkčním týmu stanovuje autor dané user story. Tato definice určuje splnění podmínek, při kterých je možné označit user story jako dokončenou. Definici připraveného určují pravidla pro tvorbu user stories a její splnění posuzuje product manager. Případné nedostatky v této oblasti bude možné zjistit v rámci dotazníkového šetření v kategorii Řízení požadavků.

Produktový cíl není v produkčním týmu využíván, protože není jednoznačně definován. Produkční tým nestaví na zakázku jednu konkrétní aplikaci s předem jasně danými funkcionalitami. Spíše rozvíjí současnou aplikaci a vhodně ji doplňuje. Pojem produktový cíl tedy není relevantní, za důležitější je v rámci týmu možné považovat cíl sprintu, který je na jeho začátku vždy jasně stanoven.

V produkčním týmu též nejsou využívány techniky jako Burnup a Burndown chart, protože do sprintu nebývá zahrnutých tolik user stories, aby tento typ techniky pomohl v orientaci mezi nimi. Tyto techniky bývají využívány spíše v případech týmu s delšími sprinty.

4.8 Návrh změn na základě porovnání s teorií Agile

Na základě porovnání současného stavu s teorií Agile byly navrženy následující návrhy na zlepšení. V této kapitole budou stručně představeny. Jejich detailní popis je k nalezení v příloze, která bude sloužit jako návrh změn partnerovi z praxe.

1) Zavedení dedikované role scrum mastera

Cílem této změny je vytvořit roli scrum mastera, která v týmu na rozdíl od teorie Agile zásadně chybí, což negativně ovlivňuje pracovní atmosféru v produkčním týmu. Vzhledem k existenci pouze jednoho produkčního týmu, jeho velikosti a způsobu práce (týdenní sprinty, zaběhnuté procesy) se nedá předpokládat, že by role dedikovaného scrum mastera zabírala v tomto týmu konstantně 100 % pracovního času, a proto se očekává využití přibližně 50 % kapacity přesně jednoho ze členů týmu (ideálně působícího na pozici konzultanta, který již v současné době roli scrum mastera částečně supluje).

2) Úprava audience zavedených schůzek Daily Standup a Sprint Retrospective

Cílem této změny je vytvořit příznivější prostředí na schůzkách, kterých se účastní členové produkčního týmu. Audience těchto schůzek bude omezena tak, aby se jich účastnil co nejmenší počet relevantních členů týmu (product manager, scrum master, vývojáři). Z audience budou vypuštěni subproduct owneri, kteří nejsou členy týmu a jejichž přítomnost schůzky prodlužovala a narušovala pocit sounáležitosti a „intimitu“ technického týmu.

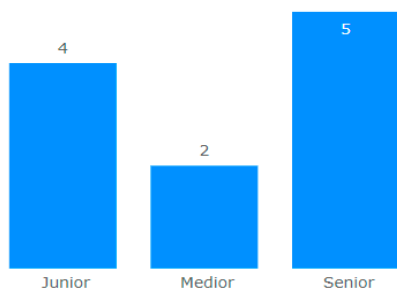
4.9 Identifikace technických nedostatků v procesu implementace Agile – dotazníkové šetření

V následující kapitole jsou uvedeny vizualizace a popis výsledků dotazníkového šetření, které mělo za cíl identifikovat konkrétní technické nedostatky ve způsobu, jak je v týmu provedena implementace agilního projektového řízení. Vizualizace jsou doprovázeny vysvětlujícím a kontext doplňujícím textem.

4.9.1 Identifikace respondentů

Do výzkumu se zapojilo všech 11 členů produkčního týmu pro vývoj interní aplikace ITSONe. K identifikaci respondentů byla zvolena pro téma relevantní kritéria, tedy seniorita (oficiální postavení zaměstnance v pracovní hierarchii), doba, po jakou se podílí na vývoji aplikace, zkušenosti s prací v Agile v jiném prostředí a množství těchto zkušeností.

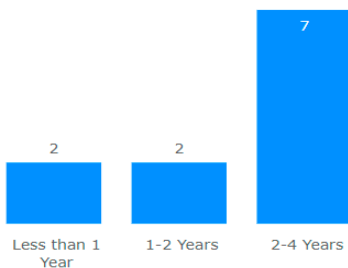
Respondenti podle seniority



Obrázek 19: Identifikace nedostatků – Respondenti dle seniority, zdroj: Archiv autora

Většina vývojového týmu pracuje na seniorních pozicích, znamená to tedy, že se jedná o zkušené a oborem prověřené zaměstnance (Obrázek 19). Zkušení zaměstnanci by měli být efektivnější, s větší odborností v oboru a též zvyklí na zaběhlé pořádky. Tento trend je zároveň patrný v rámci celé firmy.

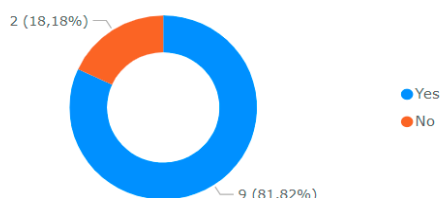
Respondenti podle délky práce u týmu



Obrázek 20: Identifikace nedostatků – Respondenti dle délky práce u týmu, zdroj: Archiv autora

Většina zaměstnanců ve vývojovém týmu působí již delší dobu, měli by tedy být velmi dobře seznámeni s požadavky na vývoj a vším, co vývoj ITSoOne obnáší (Obrázek 20). Tento fakt koresponduje s jejich vyšší průměrnou senioritou.

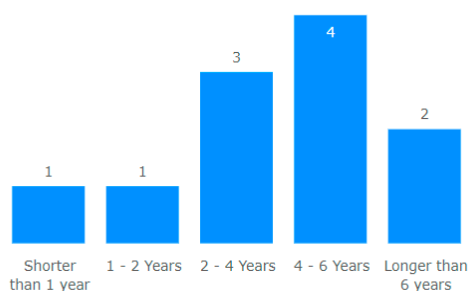
Respondenti podle zkušeností s prací v jiném týmu



Obrázek 21: Identifikace nedostatků – Respondenti dle zkušeností s prací v jiném týmu, zdroj: Archiv autora

Většina zaměstnanců má zároveň zkušenosti s prací v jiných vývojových týmech, mají tedy možnost porovnání současných nastavených pořádků se zavedenými pořádky v jiných týmech (Obrázek 21). Zvyšuje to u nich tedy obeznámenost s různými podmínkami pro vývoj softwaru.

Respondenti podle roků zkušeností s Agile



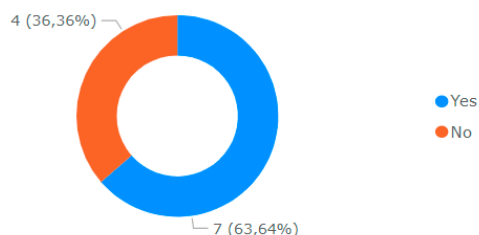
Obrázek 22: Identifikace nedostatků – Respondenti dle roků zkušeností s Agile, zdroj: Archiv autora

Většina zaměstnanců má minimálně 4 roky zkušeností s prací v agilním vývojovém procesu, mají tedy velmi dobrou praktickou zkušenost s agilními metodikami (Obrázek 22).

4.9.2 Existuje prostor pro zlepšení implementace Agile?

Existují nedostatky v řízení projektů při agilní transformaci ve vybrané organizaci (Obrázek 23)? 64 % respondentů odpovědělo, že nedostatky existují. Pokud se na výsledek podíváme detailněji, je vhodné zmínit, že většina respondentů (75 %), kteří nedostatky v implementaci Agile nevidí, je na seniorních pozicích a pracuje v týmu v průměru déle než respondenti, kteří odpovídali, že nedostatky v procesu vidí.

Existuje prostor pro zlepšení implementace Agile v týmu?



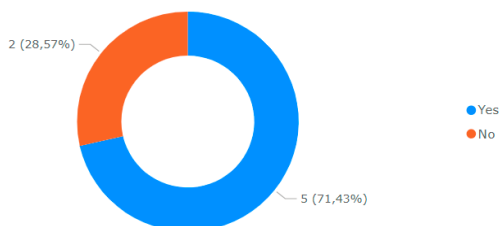
Obrázek 23: Identifikace nedostatků – Prostor pro zlepšení Agile, zdroj: Archiv autora

4.9.3 Doplnující dotazníkové otázky

Respondenti, kteří odpověděli, že vidí nedostatky v agilním projektovém řízení, byli následně dotazováni na konkrétní kategorie, ve kterých vidí problém. V každé kategorii byly otázky pro identifikaci problému v dané oblasti a následně otázky pro identifikaci dopadu problému na úspěch projektu. Kategoríí bylo celkem 6 – Organizace týmu a práce, Komunikace, Technologie, Řízení požadavků, Procesní nedostatky, Ostatní nedostatky

4.9.3.1 Organizace týmu a práce

Existuje prostor pro zlepšení ve způsobu, jak je organizován tým, či práce?



Obrázek 24: Identifikace nedostatků – Zlepšení organizace, zdroj: Archiv autora

71 % dotázaných respondentů vidí nedostatky ve způsobu, jakým je organizován tým či práce (Obrázek 24). Při rozpadu na kategorie seniority se jedná o juniornější členy týmu.

Co přesně by mohlo být zlepšeno?

All the priorities management, decision making and further development of the app are lying on one person

Following Process, Testing and equal contribution from every Dev

More testers could be allocated to the team which will have time to create test scenarios to ensure all requirements are 100% met.

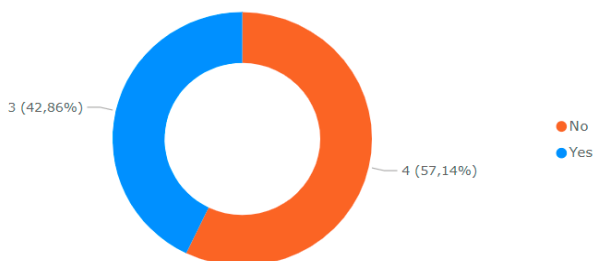
Software testing

Work can be distributed evenly for everyone. Also, not all information communicated to the whole Dev team

Obrázek 25: Identifikace nedostatků – Zlepšení organizace – Detail, zdroj: Archiv autora

Z konkrétních popsaných nedostatků vyčnívá testování softwaru (Obrázek 25). Podle respondentů nedochází k dostatečnému testování softwaru, díky čemuž se ve výsledné aplikaci objevují chyby. Dále byl kritizován způsob rozdělování práce, kdy na některé developery nezbývají úkoly nebo ne všichni developeři jsou zapojováni do komunikace. Další odpověď zmiňuje přílišnou centralizaci dění okolo product managera.

Ovlivňuje současná situace negativně pracovní výsledky?



Obrázek 26: Identifikace nedostatků – Zlepšení organizace – Pracovní výsledky, zdroj: Archiv autora

Z respondentů, kteří odpověděli, že vidí nedostatky v oblasti organizace práce, vnímá tyto nedostatky 43 % jako natolik závažné, že ohrožují úspěch projektu (Obrázek 26).

Jak přesně ovlivňuje současná situace pracovní výsledky?

Sometimes some areas are not paid enough attention because the main person doesn't have time and capacity to care of them. Therefore it causes delays.

When multiple stories are finished at the same time, the tester may be overloaded and needs to prioritize and test some things later. Once the deliverables are tested, if they need adjustments, the developers may be working on something else at the time.

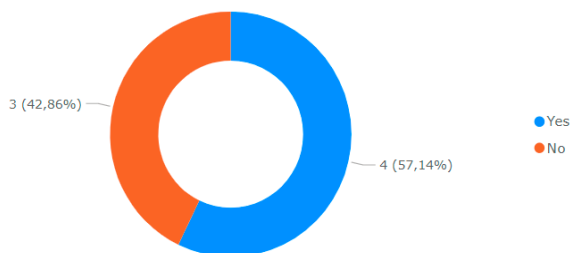
Without proper integration testing, the bugs usually found during UAT which we should avoid this.

Obrázek 27: Identifikace nedostatků – Zlepšení organizace – Pracovní výsledky – Detail, zdroj: Archiv autora

Konkrétně se jedná o proces testování (Obrázek 27). Pro patřičné testování softwaru kapacita jednoho testera nedostačuje, což vede k tomu, že chyby jsou často odhalovány až během uživatelského testování. Zároveň je tu uvedeno, že občas jsou některé oblasti vývoje přetíženy, protože hlavní product manager je velmi zaneprázdněný a nestíhá se všemu adekvátně věnovat.

4.9.3.2 Komunikace

Existuje prostor pro zlepšení ve způsobu, jak probíhá komunikace v týmu?



Obrázek 28: Identifikace nedostatků – Zlepšení komunikace, zdroj: Archiv autora

57 % respondentů vidí nedostatky ve způsobu komunikace na projektu (Obrázek 28).

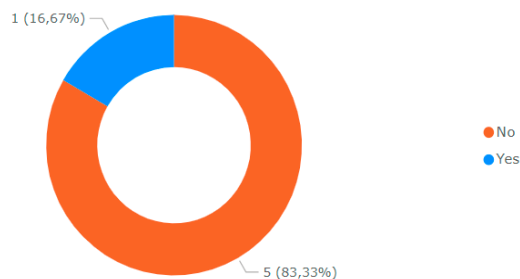
Co přesně by mohlo být zlepšeno?

As the main product owner is extremely busy, he doesn't have time to take care of all the communication needed to facilitate development.
Keeping the whole team in cc incase any task related communication happens
More team building
When is a new feature deployed, the owners should be notified always

Obrázek 29: Identifikace nedostatků – Zlepšení komunikace – Detail, zdroj: Archiv autora

V oblasti komunikace se jeví jako největší nedostatek skutečnost, že o důležitých věcech nejsou informováni všichni, kteří by měli, ať už se jedná o subproduct ownery či další členy týmu, kteří pak nemají přehled o dění (Obrázek 29). Celkově jsou patrné nedostatky v komunikaci.

Ovlivňuje současná situace negativně pracovní výsledky?



Obrázek 30: Identifikace nedostatků – Zlepšení komunikace – Pracovní výsledky, zdroj: Archiv autora

83 % respondentů uvedlo, že nedostatky v komunikaci zásadně neovlivňují úspěch projektu (Obrázek 30).

Jak přesně ovlivňuje současná situace pracovní výsledky?

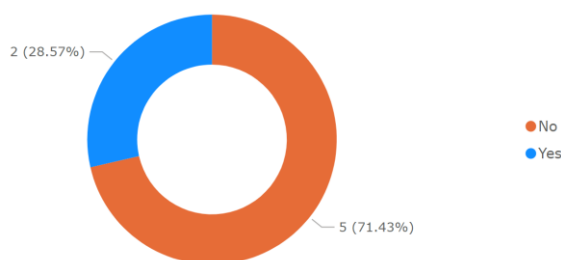
Not affecting project delivery, but it will be easy to work as a team
Sometimes there aren't clear requirements.

Obrázek 31: Identifikace nedostatků – Zlepšení komunikace – Pracovní výsledky – Detail, zdroj: Archiv autora

Z odpovědí o vlivu komunikace na úspěch projektu je patrné, že zásadní vliv na úspěch nemají, nicméně členové týmu přiznávají, že pokud by probíhala lepší komunikace, pracovalo by se jim lépe (Obrázek 31).

4.9.3.3 Technologie

Existuje prostor pro zlepšení ve způsobu, jak jsou využívány technologie?



Obrázek 32: Identifikace nedostatků – Zlepšení technologií, zdroj: Archiv autora

29 % respondentů vidí nedostatky ve způsobu využití technologií v procesu agilního vývoje (Obrázek 32)

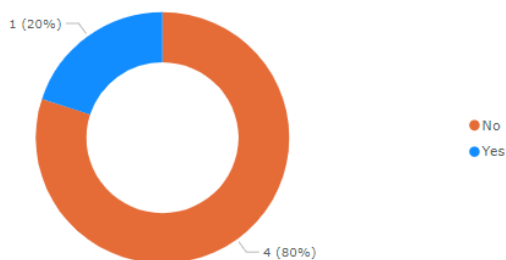
Co přesně by mohlo být zlepšeno?

Cant say since Im not a technical person
CI/CD

Obrázek 33: Identifikace nedostatků – Zlepšení technologií – Detail, zdroj: Archiv autora

Co se týká konkrétních nedostatků, tak byl zmíněn CI/CD proces (jedná se o DevOps praktiku Continuous Improvement – Continuous Deployment), která by v podobě své implementace u partnera z praxe měla být vylepšena (Obrázek 33).

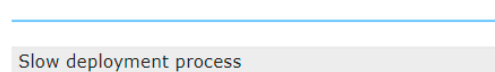
Ovlivňuje současná situace negativně pracovní výsledky?



Obrázek 34: Identifikace nedostatků – Zlepšení technologií – Pracovní výsledky, zdroj: Archiv autora

20 % respondentů uvedlo, že nedostatky v užívaných technologiích negativně ovlivňují úspěch projektu (Obrázek 34).

Jak přesně ovlivňuje současná situace pracovní výsledky?

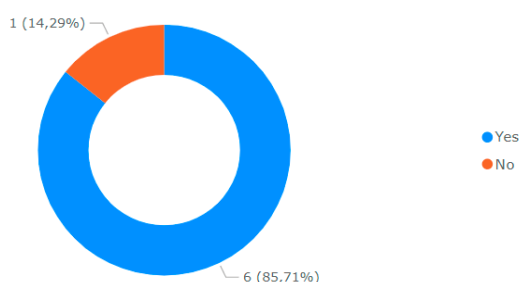


Obrázek 35: Identifikace nedostatků – Zlepšení technologií – Pracovní výsledky – Detail, zdroj: Archiv autora

Konkrétně se jedná o pomalý proces nasazování změněných částí aplikace, což souvisí s předchozím bodem týkajícím se CI/CD procesu (Obrázek 35).

4.9.3.4 Řízení požadavků

Existuje prostor pro zlepšení ve způsobu, jak jsou řízeny požadavky v týmu?



Obrázek 36: Identifikace nedostatků – Zlepšení řízení požadavků, zdroj: Archiv autora

86 % respondentů odpovědělo, že vidí nedostatky v řízení požadavků v týmu (Obrázek 36).

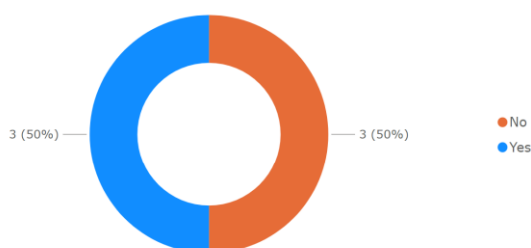
Co přesně by mohlo být zlepšeno?

Better product backlog handling
Better Story with Design and style guides :)
Estimations from developers got much better over the last year and a half, but rating the story difficulty sometimes still does not match the real time spent working on it.
Main product owner should take care more about the defining the requirements.
So far, everything is great
The pipeline is not always transparent and new features are prioritized by the main product owner without letting know the other (sub)owners, so their backlogs might be delayed for a while

Obrázek 37: Identifikace nedostatků – Zlepšení řízení požadavků – Detail, zdroj: Archiv autora

Polovina respondentů zmínila nedostatky v řízení produktového backlogu (Obrázek 37). Prioritizace práce není vždy jasná, neboť o ní rozhoduje především hlavní product manager, což někdy zdržuje naplnění požadavků sub-ownerů. Další zmiňovanou výtkou je obsahem nedostatečná příprava user stories, což vede k ne vždy jasné definici požadavků pro developera. Zároveň bylo zmíněno i nejasné hodnocení náročnosti user stories, které ne vždy odpovídá reálnému času strávenému prací nad nimi.

Ovlivňuje současná situace negativně pracovní výsledky?



Obrázek 38: Identifikace nedostatku – Zlepšení řízení požadavků – Pracovní výsledky, zdroj: Archiv autora

50 % respondentů odpovědělo, že nedostatky v řízení požadavků negativně ovlivňují úspěch projektu (Obrázek 38).

Jak přesně ovlivňuje současná situace pracovní výsledky?

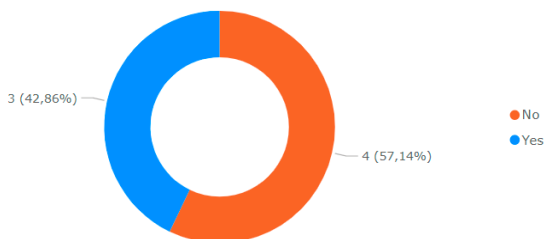
As mentioned, the sub products and their further improvements can be postponed
In some cases it affects time.
Not affect anything
Once the requirements aren't set sufficiently, it delays the delivery with extra communication needed.

Obrázek 39: Identifikace nedostatků – Zlepšení řízení požadavků – Pracovní výsledky – Detail, zdroj: Archiv autora

Konkrétně respondenti zmínili, že díky nejasnému definování požadavků v user stories vznikají zpoždění, protože kvůli nejasnosti je potřeba další komunikace navíc, která zdržuje obě strany v řešení problému (Obrázek 39). Též bylo zmíněno, že díky nejasné prioritizaci jsou pozdržovány práce na sub-aplikacích.

4.9.3.5 Procesní nedostatky

Existuje prostor pro zlepšení ve způsobu, jak jsou nastaveny procesy?



Obrázek 40: Identifikace nedostatků – Zlepšení procesů, zdroj: Archiv autora

43 % respondentů uvedlo, že existují nedostatky v nastavení procesů (Obrázek 41).

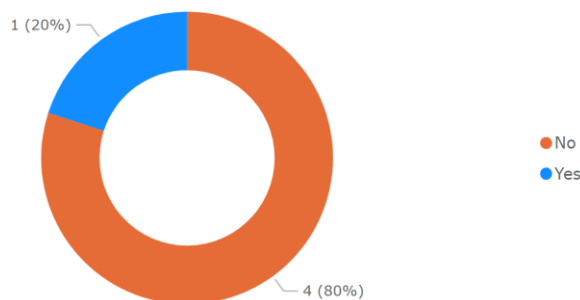
Co přesně by mohlo být zlepšeno?

More Robust Testing
The testing is not well taken care of.

Obrázek 41: Identifikace nedostatků – Zlepšení procesů – Detail, zdroj: Archiv autora

Konkrétně bylo zmíněno testování, na které není vyhrazená dostatečná kapacita, tudíž se nestíhají otestovat všechny vydávané funkcionality, a na chyby se tak často naráží až při uživatelském akceptačním testování, což z dlouhodobého hlediska například podryvá důvěru ve vývojový tým jako takový (Obrázek 42).

Ovlivňuje současná situace negativně pracovní výsledky?



Obrázek 42: Identifikace nedostatků – Zlepšení procesů – Pracovní výsledky, zdroj: Archiv autora

20 % respondentů uvedlo, že procesní nedostatky negativně ovlivňují úspěch projektu (Obrázek 42).

Jak přesně ovlivňuje současná situace pracovní výsledky?

There are deployed to production parts of app which have bugs.

Obrázek 43: Identifikace nedostatků – Zlepšení procesů – Pracovní výsledky – Detail, zdroj: Archiv autora

Konkrétně se jedná o již výše popsany problém nedostatečné kapacity pro testování (Obrázek 43).

4.9.3.6 Ostatní nedostatky

V kapitole dotazníku „ostatní, dosud nezmíněné nedostatky“, nebyly žádné uvedeny, proto tato kapitola nebude dále rozepisována.

4.9.4 Shrnutí dotazníkového šetření

Nejčastěji zmiňovaným problémem v rámci dotazníku bylo testování přídavek. Ať už se jedná o nepostačující kapacitu pro testování, která vede k jeho odkládání, nebo neorganizovanost testování, které ústí ve vydávání chybami zatíženého softwaru. Dalším řešeným nedostatkem je komunikace v rámci týmu. Problémem je nefungující zapojení všech, kteří by se komunikace účastnit měli, či nejasná definice komunikačních kanálů. Třetím silně zmiňovaným nedostatkem je neuspokojující kvalita příprava a zpracování user stories, které vedou k nedostatečně jasné definici požadavků.

V následující tabulce (Tabulka 1) jsou jednotlivé části dotazníku seřazeny podle četnosti, s jakou byly označeny za problematické.

Dotazníková otázka	Ano	Ne
Existuje prostor pro zlepšení implementace Agile?	7	4
Existuje prostor pro zlepšení ve způsobu, jak jsou v týmu řízeny požadavky?	6	1
Existuje prostor pro zlepšení ve způsobu, jak je organizován tým či práce?	5	2
Existuje prostor pro zlepšení ve způsobu, jak probíhá komunikace v týmu?	4	3
Existuje prostor pro zlepšení ve způsobu, jak jsou nastaveny procesy?	3	4
Existuje prostor pro zlepšení ve způsobu, jak jsou využívány technologie?	2	5
Existuje prostor pro zlepšení v jiné, dosud nezmíněné oblasti?	0	7

Tabulka 1: Četnosti odpovědí v prvním dotazníkovém šetření, zdroj: Archiv autora

4.10 Prvotní návrh zlepšení implementace Agile

Díky vyhodnocení dotazníku, který úspěšně identifikoval nedostatky v implementaci Agile principů v každodenní práci týmu, bylo přistoupeno k návrhu funkčních změn, které by měly vést k odstranění těchto nedostatků. Tyto změny jsou roztrženy do tematických kapitol totožným způsobem, jakým byl členěn vstupní dotazník. Cílem těchto navržených změn je reagovat na konkrétní nedostatky uvedené v jednotlivých kategoriích. Navržené změny budou následně předány k posouzení produkčnímu týmu v rámci druhého dotazníku. Výsledky dotazníku tak budou sloužit k inspiraci při detailním návrhu vlastního opatření. Tuto a následující kapitolu lze tedy považovat za studii proveditelnosti.

4.10.1 Kategorie Organizace týmu a práce

V kategorii Organizace týmu a práce byly definovány následující nedostatky:

- 1) Organizace testování softwaru
- 2) Způsob rozdělování práce
- 3) Přílišná centralizace dění okolo product managera

V reakci na zmíněné nedostatky byly v této kategorii navrženy následující úpravy procesu.

Vytvoření přehledu pro týmové kapacitní plánování

Návrh této změny spočívá ve vytvoření nástroje, v němž by jednotliví členové týmu zaznamenávali úkoly, na kterých v daný den pracují, díky čemuž by bylo možné operativně upravovat pracovní náplň jednotlivých členů vývojového týmu. Každý den ráno by každý člen týmu uvedl výčet úkolů, na kterých má v plánu pracovat, a zároveň uvedl jejich prioritu pro úspěch projektu. Zodpovědností product managera by bylo dohlížet na tento přehled a v případě potřeby jej upravovat – úkoly od přetížených pracovníků přesouvat na méně vytížené, více prioritizovat a v případě nutnosti (například zvýšené potřebě testovat software před nasazením do produkce) úkolem pověřit více pracovníků – nejen testera, ale třeba i nevytížené konzultanty nebo developery.

Primárním cílem této změny je přímo zlepšit způsob rozdělování práce v týmu. Bude dosaženo větší přehlednosti a operativní optimalizace se stane snazší. Zároveň tato změna nepřímou povede i ke zlepšení v problematické oblasti organizace testování softwaru, protože díky větší přehlednosti v rozdělení práce bude možné udělit testování vyšší prioritu či do něj snadno zapojit více lidí.

Zvýšení početního stavu o jednoho testera

Návrh této změny spočívá v začlenění nového člověka do týmu, který by měl v popisu práce výhradně testování. Provedení této změny je komplexnější a vyžaduje součinnost několika zúčastněných stran v čele se seniorním managementem. Pro úspěšné rozšíření týmu o testera je potřeba nejprve rozhodnout, zda tato změna bude dočasná či trvalá, zda bude přijat interní nebo externí zaměstnanec a na jaký úvazek.

Primárním cílem této změny je vyřešit situaci s nedostatečným testováním softwaru. Přítomnost nového testera umožní daleko detailnější testování softwaru, vytváření komplikovanějších testovacích scénářů a přímo tak zvýší kvalitu vydávaného softwaru.

Přidání testování do kompetencí developerů a konzultantů

Tento návrh je z hlediska náročnosti jednodušší na splnění než návrh předchozí a může být tedy považován za jeho alternativu v případě, že výše uvedený návrh nebude možné splnit.

Návrh této změny spočívá ve vytvoření systému, v němž by se do testování softwaru dle potřeby zapojovali i konzultanti a developéři. Tím by bylo možné doplnit chybějící pracovní kapacitu testerů o kapacitu dosud nezúčastněných členů týmu. Výhodou tohoto řešení je zároveň skutečnost, že konzultanti i developéři mají povědomí o očekávaných funkcích systému a lze tedy očekávat, že testování softwaru budou schopni efektivně a erudovaně provést.

Primárním cílem této změny je vyřešit situaci s nedostatečným testováním softwaru.

Striktnější a organizovanější způsob rozdělování práce

Návrh této změny spočívá v reorganizaci způsobu rozdělování práce mezi developery. V současnosti si na začátku pracovního týdne developéři práci rozdělují na základě svých zájmů a preferencí. Z výsledků dotazníku však vyplývá, že někteří jsou se stávajícím stavem nespokojeni, protože na ně nezůstávají zajímavé a naplňující úkoly, vzhledem k tomu, že jiní členové týmu jsou v procesu rozdělování práce průbojnější a aktivnější. Smyslem této změny je dát řád rozdělování práce, případně jej přidat přímo do zodpovědnosti product managera, který by tím mohl dohlížet na spravedlivé a smysluplné rozdělení povinností.

Primárním cílem této změny je zlepšení stávajícího způsobu rozdělování práce, který je dle výsledků dotazníku nespravedlivý.

4.10.2 Kategorie Komunikace

V kategorii Komunikace byly identifikovány následující nedostatky:

- 1) Nedostatečná komunikace v týmu
- 2) Vážnoucí komunikace a synchronizace mezi jednotlivými subproduct ownery
- 3) Nedostatek týmových mimopracovních aktivit

V reakci na nedostatky byly v této kategorii navrženy následující úpravy procesu.

Využití aplikací, které umožňují třídit komunikaci dle témat

Návrh této změny spočívá v zařazení nové komunikační platformy do komunikačních proudů nebo ve změně způsobu, jakým jsou komunikační platformy v současné době využívány. V týmu je nejvíce využíván e-mail, individuální zprávy na platformě Skype for Business či výjimečně MS Teams nebo pravidelné týmové schůzky v rámci platformy MS Teams. Účelem této změny je zařadit do komunikačních proudů platformy umožňující třídění komunikace dle témat. Těmito aplikacemi jsou například již zmíněný MS Teams nebo Slack. Výhodou tohoto způsobu komunikace je, že vše odpovídající danému tématu je přehledně organizované na jednom místě pro všechny, kteří k danému tématu mají přístup. Tento postup odbourává současnou nepřehlednou organizaci komunikace v rámci emailových vláken, která jsou obtížně organizovatelná a náchylná k chybám, kdy jsou například někteří pracovníci omylem vynecháni ze zpráv, o kterých by měli mít přehled.

Primárním cílem této změny je zlepšení nedostatečné komunikace v týmu, tato změna však může pomoci i ke zlepšení komunikace mezi jednotlivými subproduct ownery.

Vytvoření oddělené schůzky pro jednotlivé subproduct ownery

Návrh této změny spočívá v ustanovení nové pravidelné schůzky, jíž by se účastnili pouze subproduct ownery a product manager. Cílem této schůzky by byla synchronizace postupů v jednotlivých částech aplikace ITSSOne, ujasnění priorit na aplikační úrovni a sjednocení komunikace za jednotlivé části. Ve výsledku budou moci výstupy z této schůzky fungovat jako vstupy do plánování vývojových cyklů. Celý proces bude ucelenější a lépe prioritizovatelný.

Primárním cílem této změny je zlepšení synchronizace jednotlivých subproduct ownerů.

Častější nepracovní online setkávání

Návrh této změny spočívá ve snaze o vytvoření zvyku častěji se nepracovně scházet v rámci týmu. Součástí návrhu může být vytvoření pravidelného času (například jednou za dva týdny), kdy se tým sejde, věnuje se společné aktivitě nebo si dá například společný oběd. Tímto je možné snáze navázat přátelství, utužit vztahy, zlepšit náladu v rámci týmu a celkově zvýšit angažovanost jednotlivých pracovníků.

Primárním cílem této změny je napravit pocit z nedostatku nepracovních týmových akcí.

4.10.3 Kategorie Řízení požadavků

V kategorii Řízení požadavků byly identifikovány následující nedostatky:

- 1) Řízení produktového backlogu
- 2) Nedostatečná příprava user stories
- 3) Přesnost ohodnocení náročnosti user stories

V reakci na zmíněné nedostatky byly v této kategorii navrženy následující úpravy procesu.

Přidání grafiky a detailnějšího popisu do user stories

Návrh této změny spočívá ve vytvoření přesnější interní metodiky pro psaní user stories. User stories by měly mít lépe definovanou strukturu, od uvedení do problému přes popis požadované změny po návod, jak do user stories přidávat relevantní podpůrnou grafiku. Developeři v dotazníku zmiňovali, že postrádají návodné obrázky či wireframes v rámci user stories, které by doplnily text a umožnily vývojářům snáze pochopit kontext a požadované změny.

Primárním cílem této změny je vylepšit nedostatečnou přípravu user stories.

Oddělení product backlogu a sprint backlogu pro zlepšení přehlednosti

Návrh této změny spočívá ve fyzickém oddělení místa, na kterém se udržuje tzv. sprint backlog a product backlog. V současné době jsou oba udržovány v rámci jedné stránky, což je dle dotazníku pro některé členy týmu matoucí a nemají tak přehled o dlouhodobých prioritách v rámci vývoje. Oddělení obou backlogů by zároveň dávalo smysl v kontextu výše navržené změny ohledně zavedení oddělených schůzek jednotlivých sub-product ownerů, kde by obstarávání odděleného product backlogu bylo náplní nově zavedené schůzky, díky čemuž by vývojové plány jednotlivých sub-aplikací získaly na přehlednosti.

Primárním cílem této změny je zlepšení řízení produktového backlogu.

Hodnocení přesnosti odhadu náročnosti user story v rámci Sprint Retrospective

Návrh této změny spočívá v přidání nové procedury porovnávání odhadnutého času stráveného na vypracování user story s reálně stráveným časem do pravidelných schůzek produkčního týmu Sprint Retrospective. V dotazníku bylo řečeno, že během plánování sprintu často neodpovídá odhadnutý čas vypracování user story reálně strávenému času potřebnému pro jeho vypracování. Přidání této procedury může pomoci získat jednotlivým členům produkčního týmu přehled nad tím, v jakém typu user stories mají tendenci podceňovat čas potřebný k vypracování a v jakém typu jej naopak přeceňují. Následně budou poznatky těchto porovnání využívány k přesnějším časovým odhadům, což povede k lepší optimalizaci pracovní náplně. Tato změna se může potenciálně vhodně doplňovat s výše navrhnutou změnou o kapacitním plánování. Hodnocení reálně stráveného času na tvorbě user stories může pomoci zpřesnit celkové týmové kapacitní plánování.

Primárním cílem této změny je zlepšení odhadu náročnosti user stories.

4.10.4 Kategorie Procesní nedostatky

V kategorii Procesní nedostatky byly identifikovány následující nedostatky:

1) Proces testování softwaru

V reakci na tento procesní nedostatek bylo navrženo následující zlepšení.

Zavedení developerského testování

Návrh této změny spočívá v zavedení takzvaného developerského akceptačního testování, které by předcházelo uživatelskému akceptačnímu testování. Dle dotazníku se v současné době z důvodu nedostatečně otestovaného softwaru stává, že k uživatelskému akceptačnímu testování se dostane software, který má zásadní vady a samotné testování následně postrádá jakýkoliv smysl. Smyslem developerského akceptačního testování by bylo zabránit tomu, aby se k uživatelům dostal nedokonalý software.

Primárním cílem této změny je zlepšení procesu testování softwaru.

4.11 Vhodnost navržených řešení pro implementaci – dotazníkové šetření

Cílem této kapitoly je získat zpětnou vazbu od členů týmu na deset výše navržených zlepšení agilního projektového řízení. Podle této zpětné vazby bude následně vypracován detailní proces pro implementaci zlepšení. Zpětná vazba bude získána pomocí druhého

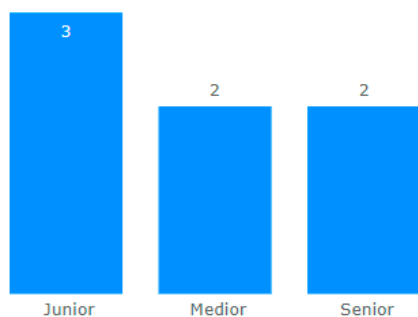
dotazníkového šetření na stejném vzorku – účastnit se tedy budou opět členové produkčního týmu.

V rámci dotazníkového šetření bylo položeno 10 otázek, každá sestávající z odpovědi ano/ne na otázku, zda je popsání řešení vhodné pro implementaci v týmu, a prostoru pro přidání komentáře k volbě ano/ne. Na základě výsledků tak budou změny navrženy k implementaci. V rámci implementace též budou vzaty v úvahu případné doplňující komentáře poskytující relevantní zpětnou vazbu. Účast v tomto šetření byla nižší než v předcházejícím šetření.

4.11.1 Identifikace respondentů

V rámci identifikace bylo přistoupeno ke stejným otázkám jako v předchozím dotazníku, neboť jsou relevantní i pro tento výzkum.

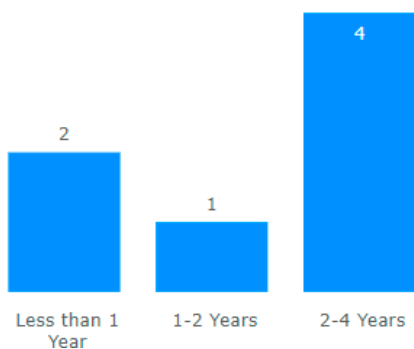
Respondenti podle seniority



Obrázek 44: Vhodnost navrhnutých řešení – Respondenti dle seniority, zdroj: Archiv autora

Většina respondentů uvedla, že pracuje na juniorních pozicích, které mají menší zodpovědnost a rozhodovací váhu při provádění hypotetických změn (Obrázek 44).

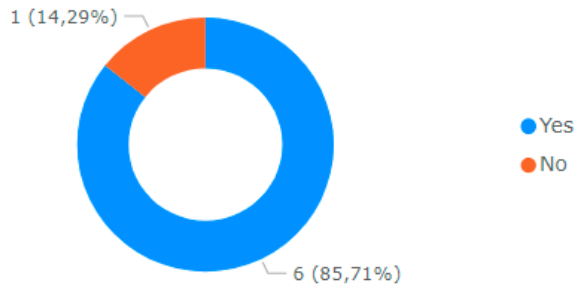
Respondenti podle délky práce u týmu



Obrázek 45: Vhodnost navrhnutých řešení – Respondenti dle délky práce u týmu, zdroj: Archiv autora

Více než polovina respondentů pracuje s týmem mezi dvěma až čtyřmi roky (Obrázek 45). Znájí tedy velmi dobře zaběhlé interní procesy a mají značný přehled o potenciálních nedostacích a představu o jejich možném odstranění.

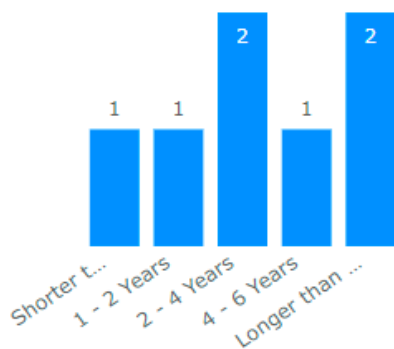
Respondenti podle zkušeností s prací v jiném týmu



Obrázek 46: Vhodnost navrhnutých řešení – Respondenti dle zkušeností s prací v jiném týmu, zdroj: Archiv autora

Většina zde zúčastněných zaměstnanců má zkušenosti s prací v jiných vývojových týmech a má tedy možnost srovnání svých současných zkušeností se svým minulým uplatněním (Obrázek 46).

Respondenti podle roků zkušeností s Agile



Obrázek 47: Vhodnost navrhnutých řešení – Respondenti dle roků zkušeností s Agile, zdroj: Archiv autora

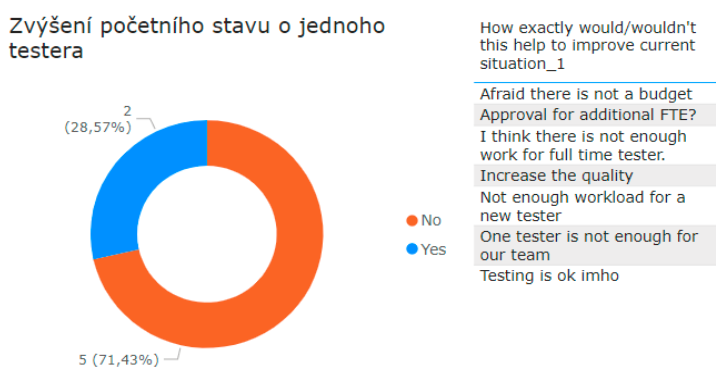
Co se týká rozdělení respondentů podle roků zkušeností s prací v agilních týmech, převažují v tomto výzkumu zaměstnanci s víceletými zkušenostmi, mají tedy kromě zkušeností s prací v jiných týmech často také zkušenosti právě s prací v Agile (Obrázek 47).

4.11.2 Dotazníkové otázky

Před respondenty bylo předloženo 10 konkrétních návrhů, jak zlepšit současnou situaci implementace agilního přístupu v týmu. Návrhy byly členěny do tematických celků podobně, jako byl členěn předchozí dotazník. Respondenti měli uvést, zda podle jejich názoru daný návrh na zlepšení povede ke zlepšení situace nebo ne, a tuto svou volbu následně slovy rozvést a odůvodnit.

4.11.3 Kategorie organizační zlepšení

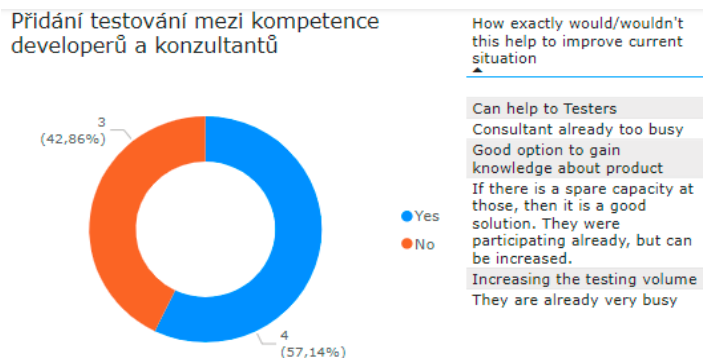
Zvýšení početního stavu týmu o jednoho testera



Obrázek 48: Vhodnost navržených řešení – Organizační zlepšení – Přidání testera, zdroj: Archiv autora

Výše popsaný návrh na rozšíření týmu o jednoho testera, čímž by byl vyřešen problém s nedostatečnou kapacitou na testování softwaru, se podle výsledků dotazníku neseťkal s kladnou odezvou (Obrázek 48). Většina respondentů jej odmítla jako nevhodný, nejčastější výtkou vůči návrhu byla pochybnost, zda je vůbec možné získat souhlas pro zvýšení početního stavu za současné situace ve firmě. Někteří se též domnívají, že v rámci týmu není dostatek práce pro plnou alokaci jednoho testera, takže by nová posila potenciálně zůstala částečně nevyužita. Dotazníkové šetření tedy neprokázalo vhodnost tohoto opatření.

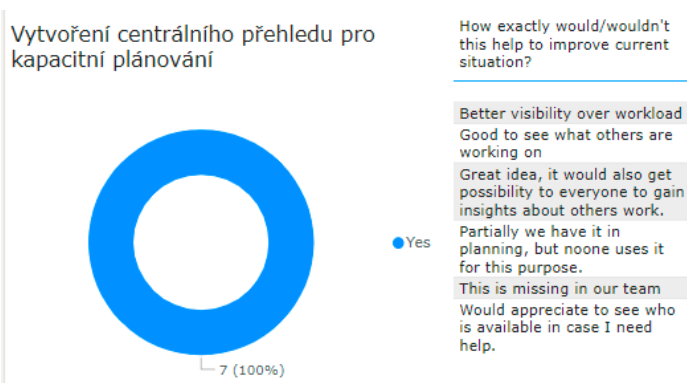
Přidání testování mezi kompetence developerů a konzultantů



Obrázek 49: Vhodnost navržených řešení – Organizační zlepšení – Testující developři a konzultanti, zdroj: Archiv autora

Výše popsany návrh na metodické zapojování i jiných členů týmu do testování softwaru se setkal převážně s kladnou odezvou (Obrázek 49). Účastníci šetření považují tento návrh za vhodnější pro zlepšení situace s testováním než najmutí testera, protože se nejedná o tak nákladný krok. Zároveň například vítají možnost rozšířit si díky testování znalosti o produktu. Nejčastější výtkou vůči tomuto opatření je konstatování, že zmíněné role (konzultant, developer) jsou již příliš vytížené a jejich systematické zapojování do procesu testování softwaru by jejich situaci potenciálně ještě zhoršilo. Dotazníkové šetření tedy prokázalo vhodnost tohoto opatření, zároveň díky němu byly získány doporučení pro jeho implementaci.

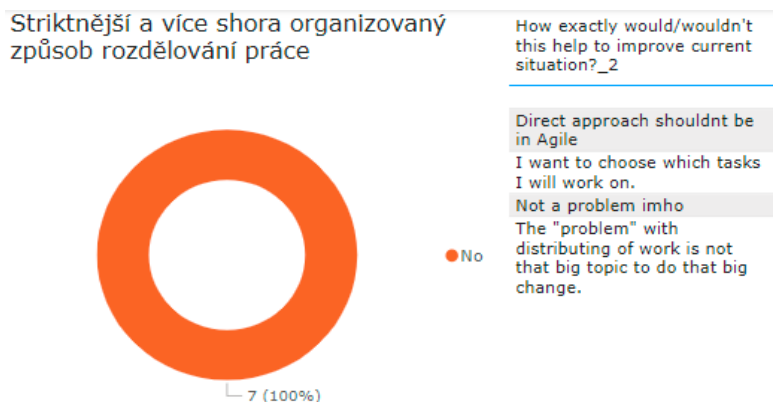
Vytvoření centrálního přehledu pro kapacitní plánování



Obrázek 50: Vhodnost navržených řešení – Organizační zlepšení – Centrální přehled pro kapacitní plánování, zdroj: Archiv autora

Výše popsany návrh na vytvoření se setkal s plně kladnou odezvou (Obrázek 50). Účastníci často zmiňují, že takovýto přehled v týmu chybí, že by rádi měli přehled o tom, na čem ostatní pracují, aby si mohli například navzájem pomáhat. Dotazníkové šetření tedy jednoznačně doporučilo jeho implementaci.

Striktnější a více shora organizovaný způsob rozdělení práce

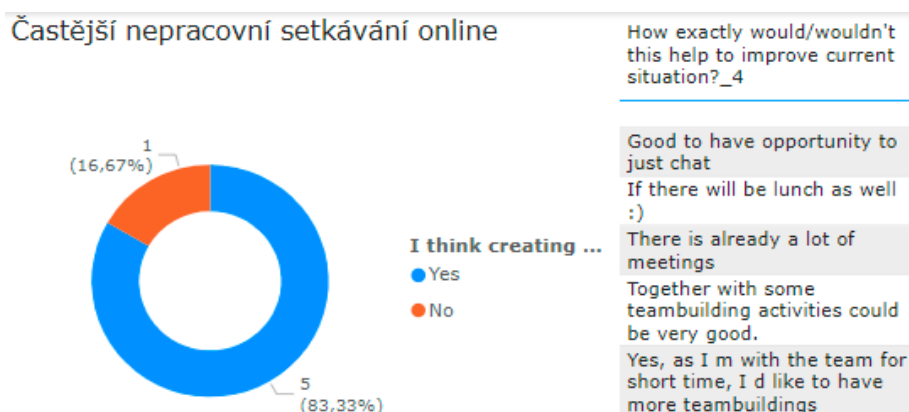


Obrázek 51: Vhodnost navrhnutých řešení – Organizační zlepšení – Striktnější organizace práce, zdroj: Archiv autora

Výše popsany návrh pro striktněji organizované dělení práce se setkal se zcela negativní odezvou (Obrázek 51). Účastníci buď nevidí danou věc jako problém, nebo nechápou princip tohoto opatření v kontextu zavedeného Agile. Dotazníkové šetření tedy jednoznačně nedoporučilo jeho implementaci.

4.11.4 Kategorie komunikace

Častější nepracovní setkání online

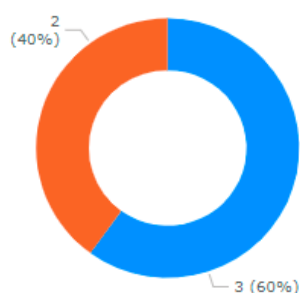


Obrázek 52: Vhodnost navrhnutých řešení – Komunikační zlepšení – Nepracovní setkávání, zdroj: Archiv autora

Výše popsany návrh na častější nepracovní online setkávání napříč lokacemi se setkal s převážně kladnou odezvou (Obrázek 52). Členové týmu by byli rádi za více teambuildingových aktivit, ve kterých by lépe poznali své kolegy. Zazněl však i názor, že už teď je zde mnoho meetingů. Dotazníkové šetření doporučilo implementaci návrhu.

Vytvoření meetingu jednotlivých product ownerů

Vytvoření meetingu pro jednotlivé product ownery



How exactly would/wouldn't this help to improve current situation?_5

"Scrum of Scrums" is a good idea

Better coordination among owners can definitely help during planning of future changes

Can't say

Don't know as I am not product owner

Not my problem

Yes, but not too often

● Yes

● No

Obrázek 53: Vhodnost navrhnutých řešení – Komunikační zlepšení – Meetingy product ownerů, zdroj: Archiv autora

Výše popsáný návrh na nový druh meetingu, během nějž by se setkávali subproduct ownery, se setkal s převážně kladnou odezvou (Obrázek 53). Hlasující často podle svých slov nemají k tématu co říci. Obecně však panuje shoda, že díky novému meetingu by se mohla zlepšit koordinace jednotlivých vyvíjených částí aplikace. Zároveň padl i návrh na název meetingu – Scrum of Scrums – který vychází z nově zaváděného stylu meetingu v teorii Agile. Dotazníkové šetření tedy doporučilo implementaci návrhu.

Přesun komunikace do aplikací, které umožňují třídění dle témat

Přesun komunikace do aplikací, které umožňují třídít dle témat (Teams, Slack)



How exactly would/wouldn't this help to improve current situation?_3

Better than email

Have all the communication in one place would be a huge benefit and would save a lot of time.

Would appreciate a lot

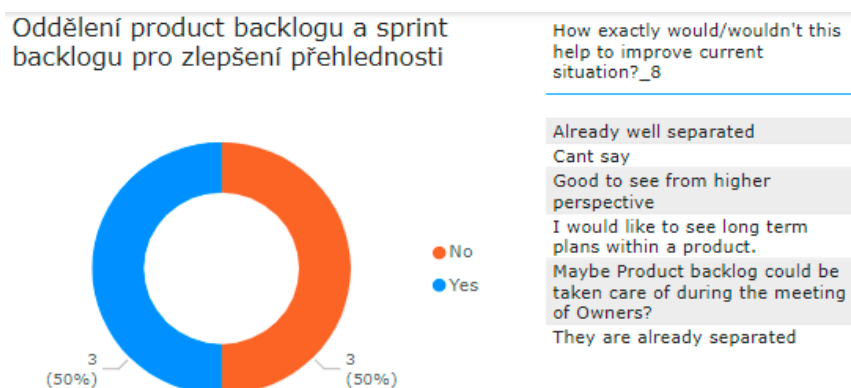
● Yes

Obrázek 54: Vhodnost navrhnutých řešení – Komunikační zlepšení – Tematické komunikátory, zdroj: Archiv autora

Výše popsáný návrh na zavedení nových moderních komunikačních kanálů se setkal s velmi kladnou odezvou (Obrázek 54). Obecně k němu nebyly ani doplňující komentáře, lze tedy výsledek dotazníkového šetření pro tento návrh považovat za jednoznačně doporučující.

4.11.5 Kategorie Řízení požadavků

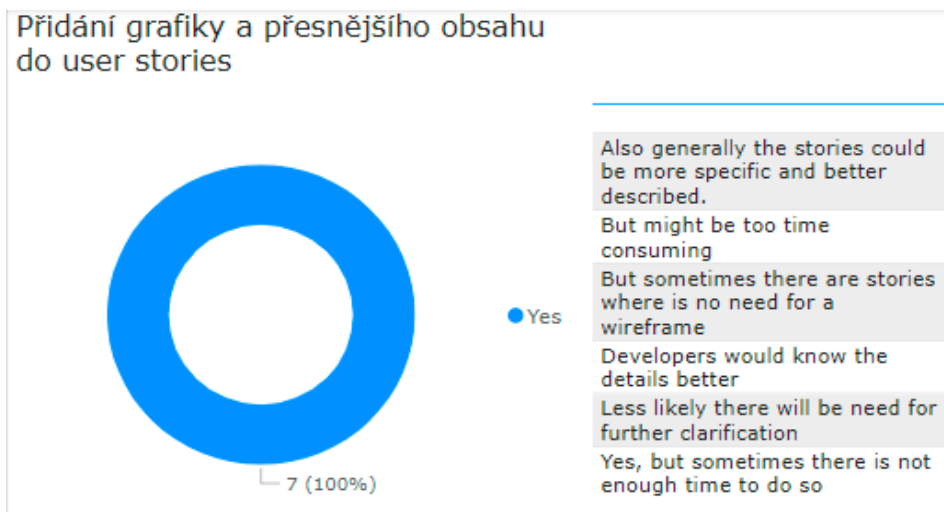
Oddělení product backlogu a sprint backlogu pro zlepšení přehlednosti



Obrázek 55: Vhodnost navrhnutých řešení – Zlepšení řízení požadavků – Oddělení product backlog a sprint backlog, zdroj: Archiv autora

Výše popsany návrh na vizuální a logické oddělení product a sprint backlogu byl přijat rozpačitě (Obrázek 55). Výtkou proti tomuto opatření je skutečnost, že pro některé jsou tyto dvě části backlogu již logicky odděleny a nepotřebují další úpravy. Z hlasů pro zařazení úpravy zazněl návrh, že by dávalo smysl přenést product backlog pryč ze současné lokace a projednávat jej v rámci výše navrhnutého meetingu jednotlivých subproduct ownerů, kteří by tak měli k dispozici prioritizační nástroj na úrovni produktu. Na základě dotazníkového šetření nelze určit, zda je návrh doporučen jako samostatný návrh nebo ne, ale po modifikaci by mohl sloužit jako doplněk k výše zmíněnému návrhu o meetingu product ownerů.

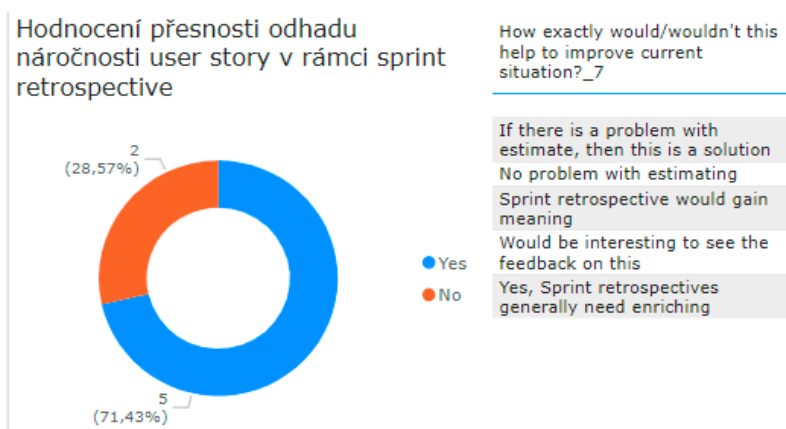
Přidání grafiky a přesnějšího obsahu do user stories



Obrázek 56: Vhodnost navržených řešení – Zlepšení řízení požadavků – Grafika v user stories, zdroj: Archiv autora

Výše popsaný návrh na vytvoření dodržovaných standardů pro tvorbu user stories se setkal s absolutně kladnou odezvou (Obrázek 56). V záplavě kladné odezvy však zaznělo i několik doplňujících poznámek k potřebě upravit standardy tak, aby vytváření i banálních user stories nebylo příliš časově náročnou činností, případně k obavám o časovou náročnost některých úprav vzhledem k vyčerpání daných pracovníků. Dotazníkové šetření doporučilo tento návrh k implementaci.

Hodnocení přesnosti odhadu náročnosti user stories v rámci Sprint Retrospective



Obrázek 57: Vhodnost navržených řešení – Zlepšení řízení požadavků – Zlepšení odhadu pracovních, zdroj: Archiv autora

Výše popsaný návrh na zavedení hodnocení přesnosti původních odhadů pracovních user stories se setkal s převážně kladnou reakcí (Obrázek 57). Většina respondentů se

vyslovila pro zavedení této změny. Zazněly též názory, že toto hodnocení přesnosti by mohlo dát nějaký smysl jinak neoblíbeným schůzkám Sprint Retrospective. Někteří však zároveň nevidí přesnost odhadu jako problém, který si zasluhuje nějaké řešení. Dotazníkové šetření však doporučilo návrh k implementaci.

4.11.6 Kategorie Procesy

Zavedení schůzek pro developerské testování před uvedením změn pro uživatelské testování



Obrázek 58: Vhodnost navrhnutých řešení – Zlepšení procesů – Developerské akceptační testování, zdroj: Archiv autora

Výše popsany návrh na zavedení developerského akceptačního testování, které by předcházelo uživatelskému akceptačnímu testování, se setkal s převážně kladnou odezvou (Obrázek 58). Někteří členové týmu se však postavili proti tomuto návrhu z důvodu, že již teď jsou velmi vytížení a nedovedou si představit svou účast ještě v dalším procesu. Někteří též navrhnou spojení tohoto návrhu s výše popsáným návrhem na zavedení nástroje na kapacitní plánování, se kterým by se mohl vhodně doplňovat. Dotazníkové šetření doporučilo tento návrh k implementaci.

4.11.7 Shrnutí

V tomto dotazníkovém šetření byly členům produkčního týmu nabídnuty konkrétní návrhy na zlepšení implementace Agile, ke kterým se měli možnost vyjádřit. V následující tabulce (tabulka 2) je možné najít seznam posuzovaných návrhů seřazený podle počtu příkladů, kdy s tímto návrhem členové týmu souhlasili.

Dotazníková otázka	Pro	Proti
Komunikaci v týmu je možné zlepšit pomocí použití tematických komunikačních aplikací.	7	0
Organizaci práce či týmu je možné zlepšit pomocí vytvoření přehledu pro kapacitní plánování.	7	0
Řízení požadavků je možné zlepšit pomocí přidání grafiky do user stories.	7	0
Komunikaci v týmu je možné zlepšit pomocí častějšího nepracovního online setkávání.	5	1
Řízení požadavků je možné zlepšit pomocí hodnocení přesnosti odhadu náročnosti user story.	5	2
Organizaci práce či týmu je možné zlepšit pomocí zapojení konzultantů do testování.	4	3
Procesy v týmu je možné zlepšit pomocí zavedení vývojářského akceptačního testování.	4	3
Komunikaci v týmu je možné zlepšit pomocí vytvoření meetingu pro product ownery.	3	2
Řízení požadavků je možné zlepšit pomocí oddělení product a sprint backlogu.	3	3
Organizaci práce či týmu je možné zlepšit pomocí najmutí dalšího testera.	2	5
Organizaci práce či týmu je možné zlepšit pomocí striktnějšího přístupu k dělení práce.	0	7

Tabulka 2: Výsledky druhého dotazníkového šetření, zdroj: Archiv autora

4.12 Závěrečný návrh technických zlepšení po vyhodnocení dotazníkového šetření

Na základě výsledků dotazníkového šetření jsou v této kapitole navrženy konkrétní změny včetně odůvodnění a očekávaných benefitů. Všechny změny jsou pak společně se změnami vzešlými z porovnání s teorií velmi detailně popsány v příloze, která bude sloužit jako návrh změn partnerovi z praxe.

- 1) Vytvoření přehledu pro týmové kapacitní plánování v nástroji MS Excel + zapojení konzultantů do testování + ustanovení MS Teams jako hlavního komunikačního kanálu týmu:
 - Zde došlo ke spojení tří v původním návrhu oddělených změn do jedné komplexní, protože ve své implementaci na sebe navazují a jsou na sobě závislé. Je proto nadmíru žádoucí, aby byly komunikovány a prezentovány jako jedna komplexní změna.
 - Tyto změny řeší především zefektivnění procesů dělení práce a koncentraci veškeré tematické komunikace do jednoho dostupného kanálu.
- 2) Zavedení schůzek všech product ownerů + řešení product backlogu na těchto schůzkách:
 - Zde došlo ke spojení dvou v původním návrhu oddělených změn, protože se navzájem velmi tematicky doplňují. Je tak přirozené, že budou implementovány současně.
 - Tyto změny se zaměřují na zlepšení komunikace mezi jednotlivými subproduct ownery
- 3) Přidání grafiky a přesnějšího obsahu do user stories:
 - Tato změna bude navržena a komunikována samostatně, protože tematicky nekoresponduje s žádnou jinou změnou. Slouží pro zjednodušení interpretace zadání z user story.
- 4) Zařazení hodnocení přesnosti odhadu user stories v rámci Sprint Retrospective:
 - Tato změna bude navržena a popsána pro implementaci samostatně, protože tematicky a zaměřením s ní žádná jiná změna nekoresponduje. Slouží pro zlepšení odhadování pracnosti v rámci Backlog Grooming a v návaznosti pak pro zlepšení kapacitního plánování.

5) Častější nepracovní setkávání (online):

- Vzhledem ke svému poměrně neformálnímu charakteru bude tato změna popsána samostatně. Povede k posílení týmové spolupráce a vzájemné sounáležitosti.

6) Developerské akceptační testování:

- Protože se jedná o jedinou procesní změnu, bude popsána a navržena k implementaci samostatně. Tato změna povede ke zvýšení kvality softwaru, který je předkládán uživatelům pro akceptační testování.

Původně byly navrženy pro zlepšení i následující změny, ale nakonec nebudou detailněji dále zpracovávány.

1) Striktnější a shora více organizovaný způsob dělení práce:

- Tato změna nebude detailně rozpracována, protože byla v rámci uživatelského dotazníku jednoznačně odmítnuta jako nepřístupná a problém neřešící. Nemá tedy smysl její myšlenku dále rozvádět.

2) Zvýšení početního stavu týmu o jednoho testera:

- Tato změna nebude detailně rozpracována, protože byla v rámci dotazníku vyhodnocena jako nevhodná k implementaci z důvodu nízké pravděpodobnosti úspěchu této změny – není pravděpodobné, že bude k dispozici rozpočet pro najmutí nového testera.

5 Výsledky a diskuze

V této diplomové práci byla provedena analýza a porovnání současného stavu implementace agilního projektového řízení s relevantní teorií. Porovnání vychází z informací získaných díky dlouhodobému začlenění do produkčního týmu. Z něj vyšly najevo dva jasné nedostatky ve srovnání s teorií Agile: nepřítomnost dedikované role scrum mastera a účast širšího týmu na schůzkách určených pro technický tým (Standup, Retrospective). Jako náprava těchto nedostatků bylo navrženo:

- Vyhrazení 50 % kapacity jednoho z členů týmu na roli scrum mastera
- Úprava audience výše zmíněných schůzek

Z porovnání s teorií též vyšly najevo další otevřené body (funkčnost kooperace více product ownerů, způsob používání product a sprint backlogu a způsob implementace definice připraveného), jež byly rozděleny do tematických kategorií a podrobeny dotazníkovému šetření.

V rámci dotazníkového šetření vyšlo najevo, že 64 % členů produkčního týmu spatřuje nedostatky ve způsobu, jakým je implementováno agilní projektové řízení v týmu, nejčastěji v kategorii Řízení požadavků, Organizace týmu a Komunikace. Za problematické bylo především označeno testování přídavků v týmu, způsob komunikace, dělby práce a tvorby a hodnocení user stories. V reakci na výsledky šetření byl vytvořen prvotní návrh a popis 10 změn, které by měly současnou situaci zlepšit. Tento popis posloužil jako podklad pro druhé dotazníkové šetření, jehož cílem bylo získat zpětnou vazbu od členů týmu na tyto návrhy.

Na základě výsledků a připomínek vzešlých z druhého dotazníkového šetření bylo k předchozím doplněno dalších 8 návrhů:

- Vytvoření přehledu pro týmové kapacitní plánování
- Zapojení konzultantů do testování
- Ustanovení MS Teams jako hlavního komunikačního kanálu týmu
- Zavedení schůzek všech product ownerů (Scrum of Scrums) a řešení product backlogu na těchto schůzkách
- Přidání grafiky a přesnějšího obsahu do user stories
- Zařazení hodnocení přesnosti odhadu user stories v rámci Sprint Retrospective
- Častější nepracovní setkávání (online)
- Developerské akceptační testování

Dohromady bylo těchto 10 návrhů detailně rozpracováno v příloze a může posloužit jako podklad pro partnera z praxe či jiné zájemce. Výše uvedené návrhy jsou relevantní čistě na příkladu partnera z praxe. Pro ostatní zájemce by bylo třeba osobně zhodnotit situaci na pracovišti. Mohou ale postupovat dle metody v této diplomové práci. Za výsledek této práce je tak možné považovat návrh řešení a zároveň i použité metody a postupy.

6 Závěr

V této diplomové práci byla provedena sekvence kroků, na jejímž konci stojí návrh a popis potřebných změn, které mají ambici zlepšit implementaci agilního projektového řízení softwarového vývoje u partnera z praxe (firmy DHL IT Services). Tato analýza detailně uvádí motivace k provedení daných změn, způsoby implementace navržených změn, rozebírá případná rizika plynoucí z jejich implementace a též navrhuje způsoby, jak zpětně hodnotit úspěšnost těchto změn.

Tato práce přináší do současného stavu poznání náhled na fungování implementace Agile na konkrétním případě v korporátním prostředí. Práce obsahuje konfrontaci praxe s teoretickým rámcem a umožňuje tak rozšířit vžitě teoretické poznání o cennou zpětnou vazbu z praktického užití.

Výstupy z této práce jsou užitečným podkladem pro management u partnera z praxe v případě, že by stál o zlepšení pracovních podmínek a výsledků v daném produkčním týmu. Díky této analýze management nemusí věnovat čas a úsilí podchycení současného stavu a může z výsledků práce a její přílohy vynést patřičné a erudované rozhodnutí.

Pro zájemce z jiných firem a produkčních týmů bude spíše než konkrétní výsledek této práce (s přihlédnutím ke specifikům prostředí, ve kterém byla práce zpracovávána) zajímavá a užitečná použitá metodika a postupy. Pomocí použitých postupů, jako je způsob popisu agilního projektového řízení v organizaci, způsob porovnání použitých postupů s teorií, či přístup k tvorbě dotazníků (otázky, kategorizace), je možné získat výstupy relevantní pro danou firmu. Tato práce tak může poskytnout podklady a teoretický rámec pro provedení analýzy úspěšnosti implementace Agile v prostředí libovolné firmy.

Pro výzkumnou sféru tato práce poskytuje náhled do typického příkladu implementace agilního projektového řízení v rámci jednoho týmu ve větší firmě. Vědeckým řízením s detailním popisem jeho praktického využití a může tak sloužit jako motivace k zamýšlení či započítí prací na nových teoretických konceptech vedoucích k vylepšení současné teorie.

7 Seznam obrázků, tabulek, grafů a zkratk

Obrázek 1: Ganttův diagram, zdroj: Freepick.com.....	18
Obrázek 2: Metoda Waterfall po zakreslení do Ganttova diagramu, zdroj: teamgantt.com	20
Obrázek 3 Příklad rolí ve Waterfall, zdroj: IT Recruitment Survival Guide (2017)	23
Obrázek 4: Agile project management, zdroj blog.feedbackloop.com.....	27
Obrázek 5: Burndown Chart, zdroj: wikipedia.org.....	33
Obrázek 6: Kanban tabule, zdroj: wikipedia.org	34
Obrázek 7: Kanban v SW Jira, zdroj: Atlassian.....	35
Obrázek 8: Úvodní stránka ITSONe, zdroj: Archiv autora	38
Obrázek 9: Project HUB, zdroj: Archiv autora.....	39
Obrázek 10: Planning Tool, zdroj: Archiv autora.....	40
Obrázek 11: Product Repository, zdroj: Archiv autora.....	40
Obrázek 12: NPS Reporting, zdroj: Archiv autora	41
Obrázek 13: Service Quality Reporting, zdroj: Archiv autora.....	42
Obrázek 14: Product backlog, zdroj: Archiv autora	46
Obrázek 15: Sprint backlog, zdroj: Archiv autora	48
Obrázek 16: JamBoard pro Sprint Retrospective, zdroj: Archiv autora	51
Obrázek 17: Procesní diagram, zdroj: Archiv autora.....	52
Obrázek 18: Popis práce product ownerů v produkčním týmu, zdroj: Archiv Autora	55
Obrázek 19: Identifikace nedostatků – Respondenti dle seniority, zdroj: Archiv autora	59
Obrázek 20: Identifikace nedostatků – Respondenti dle délky práce u týmu, zdroj: Archiv autora.....	59
Obrázek 21: Identifikace nedostatků – Respondenti dle zkušeností s prací v jiném týmu, zdroj: Archiv autora	60
Obrázek 22: Identifikace nedostatků – Respondenti dle roků zkušeností s Agile, zdroj: Archiv autora.....	60
Obrázek 23: Identifikace nedostatků – Prostor pro zlepšení Agile, zdroj: Archiv autora ...	61
Obrázek 24: Identifikace nedostatků – Zlepšení organizace, zdroj: Archiv autora	61
Obrázek 25: Identifikace nedostatků – Zlepšení organizace – Detail, zdroj: Archiv autora	61

Obrázek 26: Identifikace nedostatků – Zlepšení organizace – Pracovní výsledky, zdroj: Archiv autora	62
Obrázek 27: Identifikace nedostatků – Zlepšení organizace – Pracovní výsledky – Detail, zdroj: Archiv autora	62
Obrázek 28: Identifikace nedostatků – Zlepšení komunikace, zdroj: Archiv autora.....	63
Obrázek 29: Identifikace nedostatků – Zlepšení komunikace – Detail, zdroj: Archiv autora	63
Obrázek 30: Identifikace nedostatků – Zlepšení komunikace – Pracovní výsledky, zdroj: Archiv autora	63
Obrázek 31: Identifikace nedostatků – Zlepšení komunikace – Pracovní výsledky – Detail, zdroj: Archiv autora	64
Obrázek 32: Identifikace nedostatků – Zlepšení technologií, zdroj: Archiv autora	64
Obrázek 33: Identifikace nedostatků – Zlepšení technologií – Detail, zdroj: Archiv autora	64
Obrázek 34: Identifikace nedostatků – Zlepšení technologií – Pracovní výsledky, zdroj: Archiv autora	65
Obrázek 35: Identifikace nedostatků – Zlepšení technologií – Pracovní výsledky – Detail, zdroj: Archiv autora	65
Obrázek 36: Identifikace nedostatků – Zlepšení řízení požadavků, zdroj: Archiv autora...	65
Obrázek 37: Identifikace nedostatků – Zlepšení řízení požadavků – Detail, zdroj: Archiv autora	66
Obrázek 38: Identifikace nedostatku – Zlepšení řízení požadavků – Pracovní výsledky, zdroj: Archiv autora	66
Obrázek 39: Identifikace nedostatků – Zlepšení řízení požadavků – Pracovní výsledky – Detail, zdroj: Archiv autora	66
Obrázek 40: Identifikace nedostatků – Zlepšení procesů, zdroj: Archiv autora.....	67
Obrázek 41: Identifikace nedostatků – Zlepšení procesů – Detail, zdroj: Archiv autora	67
Obrázek 42: Identifikace nedostatků – Zlepšení procesů – Pracovní výsledky, zdroj: Archiv autora	67
Obrázek 43: Identifikace nedostatků – Zlepšení procesů – Pracovní výsledky – Detail, zdroj: Archiv autora	68

Obrázek 44: Vhodnost navrhnutých řešení – Respondenti dle seniority, zdroj: Archiv autora	75
Obrázek 45: Vhodnost navrhnutých řešení – Respondenti dle délky práce u týmu, zdroj: Archiv autora.....	75
Obrázek 46: Vhodnost navrhnutých řešení – Respondenti dle zkušeností s prací v jiném týmu, zdroj: Archiv autora	76
Obrázek 47: Vhodnost navrhnutých řešení – Respondenti dle roků zkušeností s Agile, zdroj: Archiv autora.....	76
Obrázek 48: Vhodnost navrhnutých řešení – Organizační zlepšení – Přidání testera, zdroj: Archiv autora.....	77
Obrázek 49: Vhodnost navrhnutých řešení – Organizační zlepšení – Testující developéři a konzultanti, zdroj: Archiv autora	78
Obrázek 50: Vhodnost navrhnutých řešení – Organizační zlepšení – Centrální přehled pro kapacitní plánování, zdroj: Archiv autora.....	78
Obrázek 51: Vhodnost navrhnutých řešení – Organizační zlepšení – Striktnější organizace práce, zdroj: Archiv autora.....	79
Obrázek 52: Vhodnost navrhnutých řešení – Komunikační zlepšení – Nepracovní setkávání, zdroj: Archiv autora	79
Obrázek 53: Vhodnost navrhnutých řešení – Komunikační zlepšení – Meetingy product ownerů, zdroj: Archiv autora	80
Obrázek 54: Vhodnost navrhnutých řešení – Komunikační zlepšení – Tematické komunikátory, zdroj: Archiv autora.....	80
Obrázek 55: Vhodnost navrhnutých řešení – Zlepšení řízení požadavků – Oddělení product backlogu a sprint backlogu, zdroj: Archiv autora	81
Obrázek 56: Vhodnost navrhnutých řešení – Zlepšení řízení požadavků – Grafika v user stories, zdroj: Archiv autora.....	82
Obrázek 57: Vhodnost navrhnutých řešení – Zlepšení řízení požadavků – Zlepšení odhadu pracnosti, zdroj: Archiv autora.....	82
Obrázek 58: Vhodnost navrhnutých řešení – Zlepšení procesů – Developerské akceptační testování, zdroj: Archiv autora.....	83

Tabulka 1: Četnosti odpovědí v prvním dotazníkovém šetření, zdroj: Archiv autora	69
Tabulka 2: Výsledky druhého dotazníkového šetření, zdroj: Archiv autora.....	84

8 Seznam použitých online zdrojů

- ADOBE WORKFRONT 2022. *Project monitoring and controlling phase*. [Online] 2022.
<https://www.workfront.com/project-management/life-cycle/monitoring-controlling>.
- ADOBE WORKFRONT 2022. *Waterfall methodology*. [Online] 2022.
<https://www.workfront.com/project-management/methodologies/waterfall>.
- ATLASSIAN 2022. *Agile project management*. [Online] 2022.
<https://www.atlassian.com/agile/project-management/project-management-intro>.
- ATLASSIAN 2022. *Kanban overview*. [Online] 2022.
<https://www.atlassian.com/agile/kanban>.
- ATLASSIAN 2022. *Microservices vs monolithic architecture*. [Online] 2022.
<https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith>.
- ATLASSIAN 2022. *Scrum of Scrums. Atlassian Agile Coach*. [Online] 2022.
<https://www.atlassian.com/agile/scrum/scrum-of-scrums>.
- BECK, KENT a kol. 2001. *Manifesto for Agile Software Development*. *Agilemanifesto.org*. [Online] 2001. [Citace: 13. 8 2022.] <http://agilemanifesto.org>.
- CAMBRIDGE DICTIONARY 2022. *PROJECT, meaning, definition*. [Online] 2022.
<https://dictionary.cambridge.org/dictionary/english/project>.
- DHL IT SERVICES. 2022. *About DHL IT Services*. [Online] 2022.
<https://dhl.jobs.cz/o-dhl/dhl-it-services/>.
- IBM. 2022. *What is software development?*. [Online] 2022.
<https://www.ibm.com/topics/software-development>.
- KISSFLOW INC. 2021. *Project execution phase*. [Online] 24. 11 2021.
<https://kissflow.com/project/project-execution-phase/>.
- LAWINSIDER. 2022. *IT Project Definition*. [Online] 2022.
<https://www.lawinsider.com/dictionary/information-technology-project>.

LUCIDCHART. 2022. *Advantages of waterfall methodology*. [Online] 2022.
<https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology>.

LUCIDCHART. 2022. *Waterfall project management methodology*. [Online] 2022.
<https://www.lucidchart.com/blog/waterfall-project-management-methodology>.

PROJECT MANAGEMENT.COM. 2022. *Project management Phases*. [Online] 2022.
<https://project-management.com/project-management-phases/>.

PROJECTMANAGER.COM. 2018. *Project Initiation: How to Start Your Project Off Right*. [Online] <https://www.projectmanager.com/blog/project-initiation>.

SCHWABER, KEN A SUTHERLAND, JEFF. 2020. *The Scrum Guide*. 2020, str. 14.

UNIVERSITY OF MASSACHUSETTS BOSTON. 2022. *Project planning phase*. [Online] 2022.
https://www.umb.edu/it/project_management_office/methodology/project_planning_phase

UNIVERSITY OF MASSACHUSETTS BOSTON. 2022. *Project Closing phase*. [Online] 2022.
https://www.umb.edu/it/project_management_office/methodology/project_closing_phase.

9 Seznam použitých knižních zdrojů

AYUB, S. 2012. *Complete IT Recruitment Survival Guide*. místo neznámé : Matador, 2012.

DEEMER, P, a kol. 2010. *The Scrum Primer*. 2010.

KERZNER, H. 2009. *Project Management - A Systems Approach to Planning, Scheduling and Controlling, 10th Edition*. 2009.

LEWIS, J. 1999. *The Project Manager's Desk Reference 2nd Edition*. místo neznámé : McGraw-Hill, 1999.

PHILLIPS, J. 2004. *PMP Project Management Professional Study Guide*. 2004.

POPPENDICK, M. A POPPENDICK, T. 2003. *Lean Software Development: An Agile Toolkit*. místo neznámé : Addison Wesley, 2003.

PROJECT MANAGEMENT INSTITUTE. 2000. *A Guide to the Project Management Body of Knowledge*. 2000.

ŠOCHOVÁ, Z. 2018. *Skvělý scrum master*. místo neznámé : ComputerPress, 2018.

WIBAS GMBH MANAGEMENT CONCULTANCY. 2013. *Scrum Compact*. Darmstadt : autor neznámý, 2013.

10 Seznam použitých odborných článků

ARAFEEN, J. a BOSE, S. 2009 *Improving Software Development Using Scrum Model by Analyzing Up and Down Movements on The Sprint Burn Down Chart: Proposition for Better Alternatives..* 2009, International Journal of Digital Content Technology and its Applications.

DORAN, G. 1981 *There's a S.M.A.R.T. way to write management's goals and objectives*. 1981, Management Review, str. 2.