

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra systémového inženýrství



Diplomová práce

Agilní přístup v řízení projektu softwarové firmy

Bc. Ksenia Kirsanova

© 2020 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Ksenia Kirsanova

Systemové inženýrství a informatika

Projektové řízení

Název práce

Agilní přístup v řízení projektu softwarové firmy.

Název anglicky

Agile project approaches at the software company.

Cíle práce

Cílem této diplomové práce je ohodnotit uplatnění agilních přístupů v řízení projektu v softwarové firmě. V průběhu práce bude popsán přístup k vybranému projektu a poté navrženy změny, které by mohly zefektivnit agilní procesy v závislosti na potřebách týmu a firmy. Práce se bude skládat ze dvou částí. V teoretické budou shrnuty základní pojmy, týkající se agilního řízení. V praktické části bude podrobněji popsán proces agilního řízení ve vybraném projektu a také navržené vhodné změny.

Metodika

V teoretické části budou vysvětlené základní pojmy a koncepty agilního řízení jak obecně tak i konkrétních jednotlivých metod. Budou zohledněny přístupy Scrumu, Kanbanu a taky XP. Agilní přístup bude také porovnán s klasickým přístupem k řízení projektu – Waterfall. Podkladem pro teoretickou část budou knižní zdroje, věnující se této problematice.

V praktické části bude popsán projekt, probíhající v softwarové firmě a řízený za použití agilních metod. Podrobněji bude zanalyzováno, zda projekt odpovídá zásadám agilního řízení. Po detailnějším rozboru jednotlivých částí projektu budou navrženy změny. Tyto změny budou zaměřeny na zefektivnění práce v týmu v rámci agilního způsobu řízení projektu.

Doporučený rozsah práce

60 – 80 stran

Klíčová slova

Agilní řízení, Scrum, XP, Kanban, Waterfall, projektové řízení, metody řízení projektu.

Doporučené zdroje informací

DERBY, Esther. 7 rules for positive, productive change: micro shifts, macro results. Oakland, CA: Berrett-Koehler Publishers, [2019]. ISBN 9781523085798.

KERZNER, Harold. Project management: a systems approach to planning, scheduling, and controlling. Twelfth edition. Hoboken, New Jersey: Wiley, 2017. ISBN 9781119165354.

MEDINILLA, Ángel. Agile Management: Leadership in an Agile Environment. Heidelberg: Springer, [2012]. ISBN 9783642289095.

MYSLÍN, Josef. Scrum: průvodce agilním vývojem softwaru. Brno: Computer Press, 2016. ISBN 9788025146507.

ŠOCHOVÁ, Zuzana a Eduard KUNCE. Agilní metody řízení projektů. 2. vydání. Brno: Computer Press, 2019. ISBN 9788025149614.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

doc. Ing. Tomáš Šubrt, Ph.D.

Garantující pracoviště

Katedra systémového inženýrství

Elektronicky schváleno dne 29. 8. 2019

doc. Ing. Tomáš Šubrt, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 14. 10. 2019

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 22. 03. 2020

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Agilní přístup v řízení projektu softwarové firmy" jsem vypracovala 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 autorka uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 4.4.2020

Poděkování

Ráda bych touto cestou poděkovala doc. Ing. Tomášovi Šubrtovi, Ph.D. za vedení diplomové práce, odborné rady a cenné připomínky. Dále bych ráda poděkovala své rodině za podporu během celého studia.

Agilní přístup v řízení projektu softwarové firmy

Abstrakt

Předmětem této diplomové práce je agilní řízení projektu softwarové firmy a metoda „Scrumban“. Cílem diplomové práce je ohodnocení uplatnění agilních přístupů v praxi vybrané společnosti a návrh změn, které mohou pozitivně ovlivnit současný stav. V teoretické části jsou popsány základní pojmy z oblasti klasického a agilního projektového řízení, metod agilního řízení a Lean přístupu. V praktické části je provedena analýza používání principu agility na základě agilního manifestu, metodik Scrum a Kanban, a to z důvodu použití obou zvolených metodik najednou v jejich kombinaci - „Scrumban“. Byly identifikovány 3 problémy při práci vybraného týmu a k nim byla navržena řešení, která mohou současnou situaci změnit.

Klíčová slova: Projektové řízení, agilní přístup, Scrum, Kanban, Lean, extrémní programování, agilní manifest.

Agile project approaches at the software company.

Abstract

The subject of this thesis is agile approach in software company projects management and method "Scrumban". The aim of the thesis is to evaluate the application of agile approach in practice of a selected company and to propose changes that can positively affect the current situation. The theoretical part describes the basic concepts of classical and agile project management, agile management methods and Lean approach. The practical part contains analysis of agility approach based on agile manifesto, Scrum and Kanban methodologies, because of the use of both selected methodologies at once in their combination - "Scrumban". Three problems were identified in the work of the selected team and a solution, that could change the current situation, was proposed.

Keywords: Project management, agile approach, Scrum, Kanban, Lean, Extreme Programming, agile manifest.

Obsah

1 Úvod.....	12
2 Cíl práce a metodika	13
2.1 Cíl práce	13
2.2 Metodika	13
3 Teoretická východiska	14
3.1 Základy projektového řízení	14
3.1.1 Projekt.....	14
3.1.2 Procesy v projektu	15
3.1.3 Projektový board.....	15
3.1.4 Projektový manažer	16
3.1.5 Projektový cíl.....	16
3.1.6 Trojimperativ	17
3.2 Lean přístup.....	18
3.3 Agilní přístup	19
3.3.1 Agilní manifest	19
3.3.2 Zavedení agilního řízení	23
3.3.3 Tři zákony agility.....	26
3.3.4 Scrum	27
3.3.5 Role.....	28
3.3.5.1 Scrum Master.....	28
3.3.5.2 Product Owner.....	29
3.3.6 Ceremonie a artefakty	31
3.3.6.1 Sprint	31
3.3.6.2 Cíl Sprintu	31
3.3.6.3 Prioritizace.....	32
3.3.6.4 Sprint planning	32
3.3.6.5 Daily Scrum.....	33
3.3.6.6 Backlog Grooming	33
3.3.6.7 Produktový a Sprint backlog	33
3.3.6.8 Sprint review	34
3.3.6.9 Retrospektiva.....	35
3.3.6.10 User Story.....	35
3.3.6.11 Scrum tabule.....	36
3.3.6.12 Story points.....	38

3.3.6.13	Produktová Roadmapa.....	38
3.3.6.14	Velocita.....	39
3.3.7	Kanban.....	40
3.3.7.1	Push a Pull systém.....	40
3.3.7.2	Podstata a hodnoty.....	41
3.3.7.3	Lead time.....	42
3.3.7.4	Kanban tabule.....	43
3.3.7.5	Role v Kanbanu.....	44
3.3.8	Scrum tabule vs Kanban tabule.....	45
3.3.9	Extrémní programování.....	46
3.3.9.1	Základní hodnoty.....	47
3.3.9.2	Praktiky.....	48
3.4	Porovnání Lean, vodopádové a agilních přístupů.....	52
4	Praktická část.....	54
4.1	Představení firmy.....	54
4.1.1	Strategie a KPI.....	56
4.1.2	Brandy společnosti.....	56
4.1.3	Nadační fond Avast.....	58
4.2	Projektové řízení v podniku.....	59
4.3	Projekt.....	59
4.3.1	Metodika v rámci projektu.....	60
4.3.2	Důvody zavádění agilního řízení do týmu.....	60
4.3.3	Tým.....	62
4.3.4	Nástroje.....	63
4.3.4.1	Slack.....	63
4.3.4.2	Jira.....	64
4.3.4.3	Zoom.....	65
4.3.5	Ceremonie a artefakty.....	66
4.3.5.1	Sprint.....	66
4.3.5.2	Daily Scrum.....	66
4.3.5.3	Grooming.....	66
4.3.5.4	Retrospektiva.....	67
4.3.5.5	Sprint review.....	67
4.3.5.6	Vizualizace.....	68
4.3.5.7	Roadmapa.....	70
4.3.6	Ukazatele a jejich měření.....	71

4.3.7	Důvodu zvolení metodiky.....	75
4.4	Analýza agility podle agilního manifestu	76
4.5	Analýza podle metodiky Scrum a Kanban.....	78
4.5.1	Hodnoty Scrum	78
4.5.2	Hodnoty Kanban	79
4.6	Srovnání a ohodnocení agilního přístupu a metod.....	81
5	Identifikované problémy a návrh řešení	84
6	Závěr.....	86
7	Seznam použitých zdrojů	87

Seznam obrázků

Obrázek 1	- Fáze v projektu.....	15
Obrázek 2	- Projektový trojimperativ	17
Obrázek 3	- Dynamika organizačních změn.....	23
Obrázek 4	- Cynefin model.....	24
Obrázek 5	- Rozpad úkolu	36
Obrázek 6	- Scrum tabule	37
Obrázek 7	- Příklad Burndown grafu.....	40
Obrázek 8	- Littlův zákon	43
Obrázek 9	- Kanban tabule	43
Obrázek 10	- Extrémní programování	47
Obrázek 11	- Proces doručení přidané hodnoty ve vodopádovém přístupu	52
Obrázek 12	- Proces doručení přidané hodnoty v agilním přístupu	52
Obrázek 13	- Logo společnosti	54
Obrázek 14	- Světové pobočky	55
Obrázek 15	- AVG logo.....	56
Obrázek 16	- CCleaner logo	57
Obrázek 17	- HMA logo	57
Obrázek 18	- Nadační fond logo.....	58
Obrázek 19	- Příklad VPN produktu.....	60
Obrázek 20	- Slack aplikace	64
Obrázek 21	- Jira příklad vzhledu.....	65
Obrázek 22	- Logo Zoom.....	65
Obrázek 23	- Kanban tabule	69
Obrázek 24	- Pracovní tok	69
Obrázek 25	- Roadmapa	70
Obrázek 26	- Dokončené úkoly	72
Obrázek 27	- Průměrná doba zpoždění.....	72
Obrázek 28	- Nevyřešené úkoly.....	74
Obrázek 29	- Průměrná doba čekání.....	74

Seznam tabulek

Tabulka 1 - Dokončené úkoly	71
Tabulka 2 - Nevyřešené úkoly	73
Tabulka 3 - Hodnocení agilních hodnot	81
Tabulka 4 - Hodnocení Scrum metodiky	82
Tabulka 5 - Hodnocení Kanban metodiky	83

1 Úvod

Projektové řízení je pojem vzniklý během šedesátých let 20. století, se kterým se v poslední době můžeme setkat čím dál tím více. Původní způsob řízení rozsáhlých a velmi komplexních úloh už dávno není určen pouze pro dlouhodobé a finančně náročné projekty, ale setkáváme se s ním i v krátkodobých projektech u malých firem. Řízení, založené na klasickém pojetí projektového trojimperativu, na síťových diagramech a finanční analýze potkáváme již i mimo klasické oblasti, kde jsme na podobný přístup zvyklí. S postupem času se rozšiřují nejenom oblasti, kam zasahuje projektové řízení, ale také se mění pohled lidí na definice projektu, na přístup k práci nad projektem, hodnotami a cíli, které stojí v pozadí práce jak týmu, tak i jednotlivců. Často můžeme pozorovat, že lídři a vedoucí pracovníci cítí odpovědnost nejenom za uskutečnění projektu, ale také za lidi, kteří se na projektech podílí a přispívají ke kvalitně odvedené práci. Od roku 2001 se teprve začaly objevovat prvky takzvaného agilního řízení, které v sobě neslo úplně jiný přístup, než na který jsme zvyklí z tradičního projektového pohledu.

Tato práce se primárně bude zabývat agilním přístupem v oblasti softwarového vývoje. Budou vysvětleny základní myšlenky agility a cíle, kterých chce dosáhnout agilní vedoucí projektu a celý tým, což je důležité v agilním procesu řízení, dále bude vysvětleno, jak se tento přístup liší od klasického projektového manažera. Základními agilními metodikami, popsány v této práci budou „Scrum“ a „Kanban“, a také jejich společné využití v rámci jednoho týmu, tzv. „ScrumBan“.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem této diplomové práce je ohodnocení uplatnění agilních přístupů v řízení ve vybrané softwarové společnosti. Z tohoto důvodu se v práci nebude zabývat konkrétní vývojovou činností, ale přístupem, který využívají členové týmu ve své práci. Poté bude vznesen návrh, který by mohl práci v týmu zefektivnit. Cílem teoretické části je seznámit čtenáře se základy problematiky klasického projektového řízení a agilního projektového řízení. Cílem praktické části je analýza současného stavu v týmu a míry využití agilního přístupu v každodenní práci vývojového týmu a z něj vyplývající doporučení na zlepšení z hlediska agilního přístupu.

2.2 Metodika

Teoretická část bude zpracovaná na základě zpracování odborné literatury z oblasti projektového řízení. Budou definovány základní pojmy projektového řízení, následně budou popsány principy agilního řízení rozšířené o agilní metodiky, jako jsou Scrum, Kanban a Extreme Programming (XP). Tyto dva pohledy na projektovou činnost budou porovnány mezi sebou. Teoretická část poslouží jako podklad pro praktickou část. Praktická část se bude zabývat agilním řízením ve vybrané softwarové společnosti. Vývojová činnost v týmu probíhá za sloučení dvou nejpopulárnějších metodik „Scrum“ a „Kanban“. Budou popsány probíhající procesy, jednotlivé role zastoupené v rámci týmu a také vysvětleno které prvky které metodiky jsou používány při agilním způsobu práce.

Hodnocení dat bude probíhat na základě agilního manifestu a obecných agilních principů, kde na základě splnění agilních předpokladů se mohou procesy v týmu považovat za agilní. Data k hodnocení jsou sbírána osobním pozorováním agilních ceremonií týmu po dobu několika sprintů. Na základě analýzy a výsledného vyhodnocení bude vznesen návrh, který má zlepšit agilní procesy v týmu.

3 Teoretická východiska

3.1 Základy projektového řízení

Jednou z definic projektového řízení je definice od H.Kerznera, která říká, že projektové řízení je plánováním, organizováním, řízením a kontrolou zdrojů společnosti pro relativně krátkodobý úkol, který byl stanoven pro dosažení konkrétního cíle. V pozadí projektového přístupu stojí systémový přístup k řízení s využitím vertikální a horizontální hierarchií. [59]

3.1.1 Projekt

Projekt je podstatou projektového řízení, a proto je důležité rozlišovat co projekt je a co není. Podle H.Kerznera projektem je sled aktivit které:

- mají konkrétní cíl, který musí být naplněn;
- mají definovaný začátek a konec;
- mají finanční limity;
- vyžaduje použití určitých zdrojů;
- propojuje více business úrovní.

To znamená, že projekt je jedinečná činnost, která má své omezení a limity, a to nejenom finanční, časové ale také limity lidských zdrojů, při jeho zahájení víme, čeho chceme dosáhnout a spojuje to lidi napříč několika podnikových úrovní.

Projekt vyžaduje správné lidi na správném místě, které reflektují zájmy všech stran do projektu zapojených – z businessové strany, ze strany uživatele a také dodavatelé.

[6]

3.1.2 Procesy v projektu

Procesy v projektu se dělí na několik fází (Obrázek 1).

Obrázek 1 - Fáze v projektu



Zdroj: Vlastní zpracování

1. Fáze iniciace obsahuje uvědomění přínosu projektu, stanovení limitu, volba projektového manažera a příprava projektové dokumentace.
2. Fáze plánování obnáší definování požadavku, rozsahu práce a zdrojů, rozvrh aktivit a hodnocení rizik.
3. Fáze uskutečnění vyžaduje vyjednávání o členech týmu, řízení práce a spolupráce s týmem na jejich rozvoji.
4. Fáze kontrola obsahuje monitoring, porovnávání aktuálního výstupu a předpokládaného výstupu, analýzu vlivu a možnosti, provedení korekce.
5. Fáze uzavírání má v sobě ověření kompletnosti odvedené práce, uzavření smlouvy a finanční vypořádání, zdokumentování ukončení projektu. [8]

3.1.3 Projektový board

Projektový board se skládá z projektového manažera, volitelně asistenta projektového manažera a projektového týmu.

Projektový tým se může skládat ze dvou částí:

- zaměstnanců, kteří jsou pevně přiřazeni k projektu a věnují se pouze práci nad ním (project office);
- zaměstnanců, kteří pracují nad vyhrazenými prvky projektu a tráví jen relativně malé procento svého času na konkrétním projektu (project team).

Rozsáhlost projektového boardu vždy závisí na velikosti konkrétního projektu. Nejsou výjimkou projekty, které se skládají pouze z několika lidí a v tom případě neexistuje potřeba rozsáhlého týmu a jednotlivci zastupují více pozic najednou. [6]

Projektový tým by se měl skládat z odborníků, určených projektovým manažerem. A to z toho důvodu, že tým by měl splňovat kritéria, stanovená projektovým manažerem, protože tým by měl skládat z lidí schopných spolupráce, profesně zdatných, schopných rozhodovat. Tyto osobní charakteristiky by měly podporovat kvalitní spolupráci v týmu a umožňovat vytváření hodnotného výstupu práce. [7]

3.1.4 Projektový manažer

Projektový manažer je osoba, která v projektu odpovídá za koordinaci a integraci aktivit napříč oblastmi rozhodování v organizaci. Hlavními aktivitami projektového manažera jsou integrace a řízení aktivit potřebných pro vývoj projektového plánu, uskutečnění projektového plánu a také aktivit nápomocných k aplikaci změn v plánu.

Projektový manažer je zároveň postavou, pro kterou jsou důležité nejen technické znalosti ale také zkušenost s vedením lidí a schopnosti spolupracovat s lidmi, vést tým, být lídrem. [62]

Hlavními funkcemi projektového manažera vůči týmu je sestavení týmu, sestavení plánu činností, projednávání s týmem, organizace týmové práce, doporučení postupu vhodných pro práci týmu, hodnocení členu týmu a udržování dobré pracovní atmosféry. [7]

3.1.5 Projektový cíl

Projekt existuje za účelem naplnění konkrétního cíle, ale tento cíl by měl být správně definován. Cíl, který je projektovým manažerem vhodně formulován je jedním z klíčových faktorů úspěchu tohoto projektu. Při stanovení cílů jsou důležité nejenom technické popisy toho, co má být dosaženo ale aby všechny strany, které se na uskutečnění tohoto cíle podílejí, porozuměly, co za ním stojí a jak bude vypadat po dokončení.

Pro usnadnění stanovení projektových cílů existuje technika SMART.

Jednotlivá písmena této techniky nám říkají, jaký by měl cíl být:

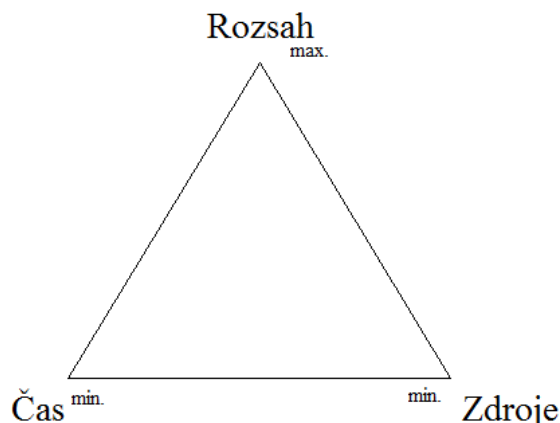
- S (specific) – specifický, konkrétní, potřebujeme vědět co chceme mít na výstupu;
- M (measurable) – měřitelný, je možné přesně určit, zda byl cíl dosažen;
- A (attainable/achievable/appropriate) – dosažitelný a vyhovující oběma stranám;
- R (relevant/realistic) – reálný, víme, že je možné dosáhnout cíle;
- T (timed) – cíl má určitý časový rámec, vymezený na jeho uskutečnění.

Můžeme se také setkat s doplňkem „i“, který vyjadřuje míru integrování cíle do cíle a strategii celého podniku. [8]

3.1.6 Trojimperativ

Pokud mluvíme o projektu, je podstatné říct, jaké jsou základní proměnné, se kterými pracuje projektový manažer. Nejlépe jsou zachycené v projektovém trojimperativu (Obrázek 2).

Obrázek 2 - Projektový trojimperativ



Zdroj: Vlastní zpracování

Důležité je, jakým způsobem a v jakém poměru se tyto proměnné kombinují. Pokud dochází k značné převaze jedné z proměnných, ostatní proměnné budou zmenšené o určité procento pro zachování poměru proměnných, poněvadž jsou mezi sebou provázané.

Rozsah (anglický „Scope“) zobrazuje objem našich výstupů, nejenom kvantitativně, ale také kvalitativně. Čas (anglický „Time“) označuje čas, který se vyhrazuje na projekt. Zdroje (anglický „Cost“) jsou náklady, které v projektu jsou. Mezi náklady zařazujeme nejenom finanční náklady, které vznikají v projektu ale také lidské zdroje.

Čas a zdroje neboli náklady na zdroje, se snažíme minimalizovat, u rozsahu se snažíme o maximalizaci (největší možný výstup při nejmenším možném vstupu). [8]

3.2 Lean přístup

Lean je jeden z moderních přístupů k řízení a vývoji, který vzešel z principu „štíhlé výroby“, která byla vynalezená a použita v Japonsku ve výrobě automobilu. Cílem lean přístupu je maximalizace hodnoty, doručované podnikem při minimalizaci času a „odpadu“. Tim odpadem se myslí věci, které vyžadují naše úsilí a náklady, ale na konci nepřinášejí žádnou konečnou hodnotu a nejsou žádané. Lean přístup se také vyznačuje soustředěním na kontinuální zlepšení. [11][12][1]

Hlavní principy lean myšlení se dají shrnout do pěti bodů:

1. Stanovení hodnoty z pohledu zákazníka.

Určení toho, co má opravdovou cenu pro zákazníka a za co bude ochoten platit.

2. Identifikace aktivit v procesu, které přinášejí hodnotu.

Je potřebné určit všechny kroky v procesu vytvoření hodnoty.

3. Přispojit procesy vytváření hodnoty celému výrobnímu procesu.

Nastavování výroby směrem ke vytvoření hodnoty zákazníkům.

4. Vytváření hodnoty pro zákazníka pouze když existuje aktuální poptávka.

Minimalizace výroby, která požaduje určité vstupy, ale nepřinášejí žádnou reálnou hodnotu na výstupu.

5. Stálé zlepšení.

Místo opakování stanovených procesů nalezení způsobu, jak procesy zlepšit.[10]

Lean je zaměřen na efektivní procesy v podniku, které přinášejí určitou hodnotu, přes neustálé zlepšení a motivaci zaměstnanců, které tvoří jeden tým.

Mezi věci, které by se měly odstranit patří například:

- Znovuobjevení

Nedostatečně kvalitní řešení, která nemohou být použita znova ani modifikována. Pokaždé se musí vypracovat nové řešení.

- Zbytečné schůzky

Schůzka, která nemá jasný a konkrétní účel a slouží pouze pro neefektivní diskuzi.

- Nedostatečný zájem o představu klienta

Vyrábět něco, co nevyhovuje představa klienta je neefektivní a je často důsledkem nedostatečného zájmu o potřeby a představy klienta.

- Ignorování projektového managementu

Velké a komplexní úlohy potřebují pečlivé plánování zdrojů, nákladu a času. Zanedbání těchto projektových procesů a principů povede k chybám a neúspěšnému průběhu a bude potřeba celý projekt připravit znovu. [14]

3.3 Agilní přístup

Pokud mluvíme o agilním přístupu, máme hlavně na mysli dynamický a přizpůsobivý přístup, rychle reagující na změnu. Je to zároveň jiným pohledem jak na projekt, tak i na život obecně. Agile není striktním procesem, přesto není neuspořádaným chaosem. Definuje hranice, ale nechává prostor pro vytváření vlastních pravidel, které vyhovují všem členům týmu za účelem dosažení největší produktivity, efektivity a kvality. [1]

3.3.1 Agilní manifest

Agilní manifest byl vytvořen zástupci jednotlivých metodik, které se začali vytvářet v devadesátých letech dvacátého století. Účelem bylo shrnout základní principy, které jednotlivé metodiky mají a vytvořit jednotnou základnu pro oblast agilních metodik.

Principy agilního manifestu jsou:

1. *„Naší nejvyšší prioritou je vyhovět zákazníkovi časným a průběžným dodáváním hodnotného softwaru.“ [2]*

V agilním vývoji přístup je inkrementální, to znamená, že hodnota produktu se zvyšuje postupně od začátku, to znamená že zákazník průběžně dostává části produktu a může navrhovat úpravy postupně.

2. *„Vítáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.“ [2]*

Změny jsou způsobem, jak dodat zákazníkovi konkurenceschopný produkt nejvíce vyhovující jejích představě. I když změny, zvláště v posledních fázích mohou být mohou být náročné, zvyšuje to výstupní hodnotu produktu.

3. *„Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.“ [2]*

Dodání v agilním prostředí provádí se v iteracích, velikost, kterých si určuje každý tým sám, ale nejčastějším intervalem jsou dva týdny až měsíce.

4. *„Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.“ [2]*

Pro přesnější definici požadavku a kontrolu v průběhu vývoje je potřebné zohlednit názory a představy nejenom vývojářů ale i zadavatelů.

5. *„Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.“ [2]*

Agilní prostředí poskytuje členům týmu možnost samostatně rozvrhovat svojí práci a jím poskytováno více důvěry a prostoru pro jejich pracovní výkon. Týmy jsou samoorganizující a nepotřebují nepřetržitý monitoring.

6. *„Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.“ [2]*

Osobní komunikace je způsob, jak sdělit informace rychle a efektivně, a tím docílit rychlejšího řešení problému bez zbytečného odkládání.

7. *„Hlavním měřítkem pokroku je fungující software.“ [2]*

Fungující software znamená konkrétní hmatatelný výstup, a v případě že funguje, znamená kladný přírůstek na výstupní hodnotě.

8. *„Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.“ [2]*

Trvalé tempo znamená, že tým je schopný pracovat se stejnou efektivitou bez výkyvu intenzity práci a sponzoři, uživatelé a další zájmové skupiny se tomu tempu přizpůsobí.

9. *„Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.“ [2]*

V agilním přístupu se klade důraz na kvalitně odvedenou práci, která má výborné charakteristiky jak z technické stránky, tak i ze vzhledové.

10. *„Jednoduchost--umění maximalizovat množství nevykonané práce--je klíčová.“ [2]*

Podstatné je dělat opravdu efektivní věci přinášející přidanou hodnotu projektu, bez zaměření na detaily, které zdržují dobu dodání a nemusí být nutně provedené. Množství zbytečné práci by mělo být minimalizováno.

11. *„Nejlepší architektury, požadavky a návrhy vzejdou ze samoorganizujících se týmů.“ [2]*

Samoorganizující se tým je tým motivovaný, schopný samostatně rozvrhovat svojí práci, přichází s vlastními návrhy, protože mají v této oblasti přehled a mají prostor pro vyjádření nápadu.

12. *„Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování a zvyklosti.“ [2]*

Agilní tým sdělí filozofii agility a snaží se o zlepšení procesu, které vedou ke kvalitně odvedené práci a spokojenosti všech členů týmu a zájmových stran. V případě, že existující procesy nefungují, snaží se o změnu za účelem dosažení lepšího fungování a jsou schopni se změnám přizpůsobit. [2]

Z principu, vyplývajících z agilního manifestu, došlo k formování čtyř hlavních hodnot:

1. *„Jednotlivci a interakce před procesy a nástroji“ [2]*

Týmy, kde členy spolupracují, mají měřitelně lepší výsledky, než týmy, kde každý jednatel pracuje sám za sebe. Dodržováním této hodnoty tým směřuje k dosažení kvalitnějších výsledků za kratší dobu.

2. *„Fungující software před vyčerpávající dokumentací“ [2]*

Dokumentaci se nelze vyhnout, ale lze jí neupřednostňovat před fungujícím výstupem práce. Pokud více času a námahy se bude věnovat sepsání na papíře, výsledek s velkou pravděpodobností nebude splňovat očekávání zákazníka, výsledek nebude uspokojivý.

3. *„Spolupráce se zákazníkem před vyjednáváním o smlouvě“ [2]*

Smlouva je důležitou částí vztahu se zákazníkem, ale spolupráce a komunikace mohou poskytovat důležité informace o produktu, které vedou k vytvoření produktu, který přesně odpovídá požadavkům. Dlouhodobou spoluprací můžeme také prospět k přesnější formulaci vlastností produktu. Tím docelíme maximální spokojenosti.

4. *„Reagování na změny před dodržováním plánu“ [2]*

Svět kolem nás se mění rychle. Technologií se vyvíjí, problémy zákazníku se mění spolu se změnou požadavků na naši práci. Pokud dodržení plánu bude přednější před rychlou

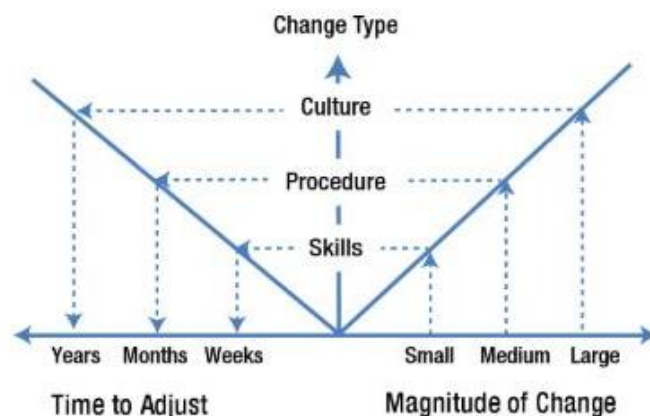
změnou s ohledem na měnící se prostředí, naše konkurenceschopnost značně klesá, a spolu s tím i klesá zájem zákazníka o spolupráci. Držet krok nám pomáhá udržovat možnost přicházet s nejlepším řešením pro svého zákazníka. [2]

3.3.2 Zavedení agilního řízení

Přestože vrcholový management si často uvědomuje benefity, které přináší zavedení agilního přístupu, a sledují trendy v odvětví, dochází k pouhému vysvětlení technické stránky tohoto přístupu, agilního přístupu pouze jako mechanického nástroje pro organizace práci. Zaměstnanci nedostávají informace o tom, proč to dělají, ale jsou pouze seznámeni s tím, jak to budou vykonávat dále. Nemají uvědomělý přístup a nesdílí agilní filozofii.

Zavedení agilního přístupu by mělo procházet napříč celou organizací a ovlivnit i celou firemní kulturu. Protože agilní přístup není jen o pochopení procesu, ale primárně o přijetí sady hodnot, názorů a principů vyžaduje to změnu v chování a myšlení lidí. Proto přijetí tohoto přístupu nemůže se konat ze dne na den, ale mělo by být trvat určitou dobu, a být cestou za transformací (Obrázek 3).

Obrázek 3 - Dynamika organizačních změn



Zdroj: [3]

Na začátku by mělo být uvědomění, že skutečně existuje problém a mělo by hledat konkrétní řešení. Tímto řešením je právě vstup na cestu celofiremních změn.

Hlavními problémy podniku jsou většinou neschopnost doručit produkt v požadované kvalitě, včas a s dodržением rozpočtu.

Dochází k tomu často kvůli tomu, že:

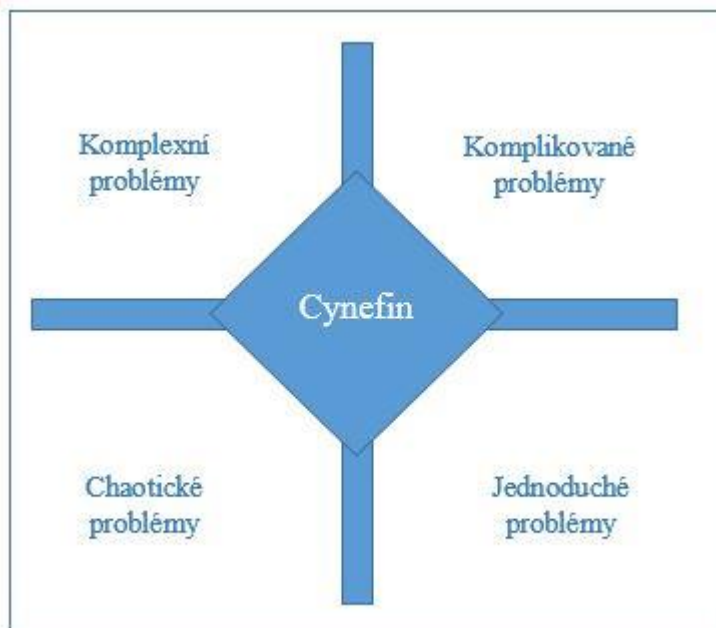
- zákazník a trh vyžadují změny v procesu vytváření hodnoty,
- požadavky nebyli dostatečně objasněné při zahájení,
- finální testy produktu jsou zanedbávané z důvodu nedostatku času,
- procesy jsou zdlouhavé a komplikované,
- rozsah práce je málo definován,
- nejde předvídat všechny situace, kterou mohou nastat v projektu, na jeho

začátku.[3]

Pravě této příčiny často stojí za uvědoměním, že existuje potřeba situaci v podniku změnit. Ale ne všechny problémy mají jako optimální řešení agilní přístup. Pro rozhodování, či při existujících problémech v podniku je vhodné si zvolit agilní přístup, mělo by se zamyslet nad celým kontextem problému firmy a jejího prostředí

Pro rozhodování je vhodné použít rozhodovací Cynefin model složitosti (Obrázek 4), vytvořený Davidem Snowdenem pro zjednodušení rozhodování.

Obrázek 4 - Cynefin model



Zdroj: Vlastní zpracování

Podle tohoto modelu se problémy dělí na jednoduché, komplikované, komplexní a chaotické.

1. Jednoduché problémy

K jednoduchým problémům řadíme problémy, se kterými jsme se již v minulosti mnohokrát setkali a víme, jak je budeme řešit. Zamýšlíme se, kategorizujeme a reagujeme. V oblasti jednoduchých problémů očekáváme také použití takzvaných „best practices“, tzn. používáme již hotové a fungující řešení.

2. Komplikované problémy

Komplikované, obsahují takové problémy, u kterých neznáme řešení, ale víme že můžeme řešení najít, pokud provedeme analýzu tohoto problémů, a najdeme vztah mezi příčinou a následkem. Takže postup v této oblasti je zamyslet se, zanalyzovat a pak reagovat.

3. Komplexní problémy

U komplexních problémů si nejsme jisti, zda můžeme určit příčiny, ale můžeme provést řadu experimentu, a zjistit, jak se celý systém chová. Existují nepředvídatelné vstupy a tím komplikují možnost problém řádně zanalyzovat. Experimenty mohou probíhat paralelně, protože definitivně správné řešení neexistuje. Zkoušíme, zamýšlíme se a reagujeme.

4. Chaotické problémy

U tohoto druhu problému nemůžeme odvodit vztah mezi příčinou a následkem. Problém nezkoumáme, nevíme, jak se chová a nedokážeme to pomocí experimentu předvídat, a proto pouze reagujeme v závislosti na konkrétní situaci. Často je to pro nás nový a neočekávaný problém, na který jsme se nikdy nepřipravovali. [36]

Takže prvním krokem by mělo být rozhodnutí, které problémy podnik anebo konkrétní tým pravidelně řeší. Pokud se primárně jedná o jednoduché problémy, dají se řešit procesně. V případě chaotických problémů saháme spíše do oblasti krizového řízení, kde potřebujeme rychlé reagovat na nepředvídané změny. V oblasti komplikovaných problémů se nejčastěji setkáváme s projekty, kde víme, co by mělo být výstupem práci a prostředí je relativně stabilní. Nevíme, jak to řešení bude vypadat ve chvíli kdy se dozvíme problém zákazníka, ale po provedení analýzy dospějeme k řešení tohoto problému. Pro tento typ problému nejvhodnějším řešením je klasické neboli vodopádové, projektové řízení, kde

v rámci projektu problém zanalyzujeme, navrháme správné řešení a pak ho provedeme. Pro komplexní problémy je ale nejvhodnějším řešením agilní přístup. Důvodem je, že pokud nevíme, co se skrývá za problémem a analýza není způsobem, jak to zjistit, návrh a následné uskutečnění nemohou přinést správný výsledek. Právě iterativní agilní přístup, založený na prototypch a pravidelné zpětně vazbě, umožňuje postupné hledání správného řešení, a tak do chvílí, až ten „neznámý“ problém bude vyřešen pomocí iterací-experimentu.

Ve chvíli, když se rozhoduje o zavedení agilního přístupu, tým anebo podnik celkově by měl si uvědomit, které problémy řeší a ve které doméně se nachází, a v závislosti na tom si zvolit konkrétní přístup. [4] [5]

3.3.3 Tři zákony agility

Tři zákony agility byly definovány Stevem Denningem jako poklad pro pochopení, jaké organizace by mohly být úspěšné v zavedení agilního přístupu. Na základě pozorování firem, kde se dobře uplatnily agilní přístupy, vyčlenily se tři základní společné charakteristiky těchto firem, které S. Denning pojmenoval zákony agility.

„Zákon malého týmu“ říká, že většina agilních metodik se používá v malých, autonomních, samoorganizujících se týmech, pracujících v kratších cyklech na menších úkolech. Pokud se tým setkává s většími úkoly, rozdělí se to do menších a pracuje s ním po částech. Důvodem je, že pokud malý tým pracuje na nějakém úkolu, je pro členy jednodušší dosáhnout synergického efektu a působit jako jediný organizmus, a proto jsou schopni dosáhnout lepších výsledků a hladšího průběhu spolupráce.

Druhou charakteristikou je „Zákon zákazníka“. Agilní týmy jsou soustředěny na vyprodukování měřitelné hodnoty svým zákazníkům. Hodnota zákazníka je jednou z nejkomplicovanějších bodů celé koncepce, i přes to, že měla být nejočividnější z nich. Na rozdíl od klasického přístupu k zákazníkovi, tým vidí, jak se zvyšuje výsledná hodnota produktu, a pokud není možné navýšení hodnoty zaznamenat, měla by se najít příčina tohoto nesouladu a najít nové řešení. Hodnoty, cíle a plány se přizpůsobují nepřetržitému procesu inkrementálního kontinuálního přidání hodnoty produktu pro zákazníka.

Poslední charakteristikou je „Zákon sítí“. Organizace by měla představovat síť vzájemně komunikujících a spolupracujících jednotlivců, které se spolu podílí na dosažení společného cílů – uspokojení zákazníku. V agilním prostředí komunikace hraje podstatnou roli pro sdílení požadavků, představ, problémů, se kterými se jednotlivci setkávají. To ale vyžaduje zavedení agilního přístupu napříč celou organizací, a sjednocení organizace do jednoho organismu, složeného s malých vysoce efektivních týmů, které díky široké komunikační síti mohou komunikovat napříč celou organizací. [9]

3.3.4 Scrum

Podle Agile Alliance [15] Scrum je agilní procesní framework pro řízení produktového vývoje a dalších znalostních oblastí. Scrum je empirický v tom, že poskytuje jim možnost stanovit hypotézu o tom, jak něco funguje, zkusit to, poučit se ze zkušeností a na základě poučení se zlepšit.

Scrum se zakládá na několika základních hodnotách:

1. Oddanost

Oddanost jako hodnota vyjadřuje ochotu a angažovanost týmů ve vytváření agilní kultury jako celku a v dosažení firemních a týmových cílů.

2. Odvaha

Tým by měl projevovat odvahu v průběhu práce, a to tak, že se jeho účastníci se nebojí pustit do náročných nebo neznámých úkolů a dokážou přiznat, že potřebují pomoc.

3. Soustředěnost

Tým by se měl soustředit na úkoly, určené v rámci sprintu a na cíle, kterých tým chce dosáhnout. Dokážou se plně věnovat aktuálním úkolům a prioritám.

4. Otevřenost

Členy týmu jsou otevření ohledně práce, kterou provádí jak mezi sebou, tak i se zájmovými stranami. Jsou schopni přiznat, že mají nějaký problém a požádat o pomoc.

5. Respekt

Členové týmu se navzájem respektují a uznávají schopnosti a samostatnost ostatních.

Tým v agilním prostředí by měl splňovat dvě hlavní zásady, a to být:

1. Samoorganizující se

Klasický tým je řízen jedním manažerem, který určuje úkoly v rámci týmů, vertikálně. Samoorganizující se tým má prostor, proto aby mohl vybrat, jak nejlépe dělat svojí práci. Nejsou řízení shora, všichni členové týmů se aktivně zúčastňují agilních aktivit a praktik. Činí rozhodnutí o své práci a jsou odpovědní za důsledky, a proto jsou svědomitější, odpovědnější a jsou více motivované ke odvádění kvalitní práci. Všichni v týmu by měli sdílet stejnou vizi a brát tým jako celek.

2. Cross-funkcionální/ Multifunkční

Cross-funkcionální tým se skládá z odborníků z různých odborných oblastí, divizi, které pracují společně na společných úkolech za účelem splnění určitých cílů. Je důležité, aby v rámci týmu existovala nejvyšší možná vzájemná zastupitelnost. A to neznamená, že nikdo nemá stanovené vlastní naplnění práce nebo neví co má dělat, ale vzájemnou zastupitelnost v případě že někdo z týmu je z nějakého důvodu nepřítomný. Práce nezůstává do navracení příslušného zaměstnance, ale přerozděluje se mezi jinými členy týmu. Je to prospěšný z důvodu značně vyšší flexibility. Jednotlivci v týmu by se měli snažit o rovnoměrné vytížení v průběhu sprintu. [18] [17] [19]

3.3.5 Role

3.3.5.1 Scrum Master

Členové týmu by měly být mezi sebou zastupitelní, však konkrétní role by měly být odlišené od ostatních a nikdo by neměl zastupovat najednou. Tyto role jsou charakteristické pro agilní přístup a v klasickém přístupu se s nimi nesetkáme.

Scrum master je rolí, která nahlíží na práci celého týmu a směřuje ho k agilně procesu a přístupu. Tato role je často vnímána jako role koučů, který pomáhá týmu v jejich práci bez zásahu do konkrétních činnosti. Cílem Scrum mastera je směřovat tým k samostatnosti, efektivitě a spokojenosti, které povedou ke splnění týmových cílů a cílů organizace. Scrum Master by měl pomoci týmu v procesu agilní transformace. Měl by naslouchat jeho potřebám a navrhopvat změny, které práce týmu usnadní a nasměruje správným směrem k dosažení agilního efektivního prostředí. Mezi úkoly Scrum Mastera patří vedení agility

v týmu, moderování a plánování ceremonií, ochrana týmu před nežádoucími vnějšími vlivy, motivování člena týmu. Osobnost Scrum Mastera by měla být chápavá, komunikující a důsledná. [35]

Stručně náplň práce Scrum mastera se dá shrnout do:

- Pomoc týmu ve svém agilním fungování vytvořením vhodného prostředí pro jeho každodenní práci
- Eliminace překážek, se kterými se tým potýká
- Podpora pozitivních vztahů jak uvnitř týmu, tak i mimo tým
- Omezení rozptýlení a podpora soustředění v týmu.

Scrum master je součástí týmu a měl by být členům k dispozici po celou dobu práci, být pozorný, neřešit problémy týmu za něj ale směřovat je k nalezení řešení, na základě, kterých se poučí a příště budou schopné vyřešit problém samostatně. Měl by pracovat s týmem, vztahy uvnitř a mimo tým a celou organizaci pro zachování komplexního pohledu na problém a souvislosti.

V praxi se často role Scrum mastera slučuje s jinou, buď s Product Ownerem nebo dalším členem týmu. Mělo by to být tou nejméně preferovanou možností, protože v tom případě dochází k konfliktu cílů anebo zájmu. Problém nastává ve chvíli, když Scrum master, pro kterého je hlavní prioritou je tým, je zároveň Product Ownerem, prioritou, kterého je produkt a spokojenost zákazníka. V případě sloučení s rolí člena týmu dochází k prioritizaci své práce nad úkolem před práci Scrum mastera, a tak tým ve chvíli zvýšeného vytížení může zůstat bez potřebné podpory a pozornosti. [20] [1]

3.3.5.2 Product Owner

Product Owner, jak napovídá název rolí, je vlastníkem produktu. Určuje produktovou vizi, určuje priority a co se v rámci produktu bude vyvíjet za účelem dosažení uspokojivého výsledku. Měl by rozumět produktu, zákazníkovi, komunikovat skrz celou organizaci a dokázat vysvětlit cíl a porozumět požadavkům. [22] [23]

Hlavní náplň role je:

1. Definování vize

Product Owner zodpovídá za komunikaci se všemi zájmovými stranami, včetně zákazníku, manažeru, týmů a ujištění že vize naplňuje zároveň i obchodní cíle.

2. Produktový backlog

Jedním z hlavních úkolů Product Owenera je správa produktového backlogu, který obsahuje všechny požadavky na tým. V rámci backlogu Product Owner vytváří priority jednotlivých úkolů na základě aktuálního stavu aplikace a taky strategií.

3. Prioritizace

Product Owner by se měl věnovat prioritizaci úkolů v rámci týmu, zároveň při dodržení priorit týmu a podniku jak celku, při dodržení rozsahu, rozpočtu a času. Zároveň upřednostňuje funkcionality produktu, které budou vyvinuté jako první a které například budou vůbec vynechané z produktu.

4. Monitoruje vývoj

Po stanovení vize a prioritizace nastává vývoj v průběhu kterého Product Owner průběžně monitoruje proces vývoje a konzultuje napříč iteraci. Poté ujasňuje s týmem procesy a oblastech ve kterých existuje prostor pro zlepšení.

5. Zohledňuje potřeby klienta

Expertní oblasti Product Owenera je porozumění potřebám klientů a reflexe jeho přání ve výsledném produktu. Proto potřebuje mít výborné komunikační schopnosti a porozumění trhu pro dosažení co nejvhodnějších výsledků, který uspokojí jak klienta, tak i zájmové strany.

6. Je kontaktní osobou

Product Owner je prostředníkem mezi týmem a zájmovými stranami, který předává požadavky na produkt týmu, aby členové týmu mohli pracovat nad opravdu potřebnými věcmi a zároveň jim nechal prostor pro práci tím, že sběr požadavků a jejich interpretaci udělá sám.

7. Hodnocení vývoje produktu

Product Owner je zodpovědný za finální produkt a za jednotlivé fáze vývoje.

Prostřednictvím vyhodnocení každého jednotlivého cyklu dokáže průběžně korigovat nedostatky produktu a pozměňovat ho podle aktuálních potřeb pro dosažení očekávaného výsledku. [21] [23]

3.3.6 Ceremonie a artefakty

3.3.6.1 Sprint

Sprint je agilním vyjádřením práce pomocí iterací. Celý proces v rámci Scrumu je rozdělený do jednotlivých sprintů. Sprint je časový úsek, délka, které po celou dobu trvání vývoje zůstává neměnná. V rámci sprintu by měla být dokončena práce, která byla naplánovaná a taky poté kontrolována. Délka sprintu může být individuální pro každý tým. Stanovuje se podle naplnění práce týmu, protože na konci sprintu by měla existovat funkcionality připravená k prezentaci zákazníkovi. To znamená, že v případě příliš krátkého sprintu na konci pravděpodobně nebude existovat žádný prezentovatelný výstup, a v případě příliš dlouhého tým začne pracovat na dalším úkolu a se tým okrádá o zpětnou vazbu a často se již tak podrobně nevěnuje tomu, co dělali na začátku sprintu. Kratší sprint je výhodný tím, že poskytuje týmu možnost rychle reagovat na změny, a tak přizpůsobovat svoji práci co nejvíce požadavkům a potřebám týmu a zájmům stran.

Každá iterace by měla sloužit jako způsob zlepšení pro tým, který se něco naučí, a v dalších iteracích budou schopni lepšího výkonu. Univerzální délkou sprintu je 14 dní. Zároveň je doporučováno nepřesahovat délku 1 měsíce. Pokud se v rámci sprintu nestihá dodat potřebná funkcionality má se zvážit jeho naplnění, je možné že úkoly zahrnuté do sprintu jsou až příliš velké a měli by být rozdělené do menších částí, které již budou splnitelné v rámci jednoho sprintu. Také je vhodné zvážit počet úkolů, přiřazených do sprintu. Při zbytečně velkém počtu úkolů se délka sprintu může zdát nevhodná, ale musí se to zvolit podle reálných kapacit, které mají v týmu k dispozici. [24] [1]

3.3.6.2 Cíl Sprintu

Sprint Goal, neboli cíl Sprintu, je cílem, ke kterému tým směřuje po celou dobu sprintu. Může to být bráno jako malá vize po dobu sprintu. Cíl Sprintu by měl být zaměřen na hodnotu, která bude dodána v průběhu sprintu, neměl by být zaměřován na konkrétní funkcionality produktu. Jedná se o přidanou hodnotu, která vzniká dokončením úkolu v rámci Sprintu. Pro tým je to možnost lépe pochopit proč a na čem budeme pracovat v průběhu celé iterace. [1] [40]

3.3.6.3 Prioritizace

Prioritizace je procesem určení posloupnosti úkolu z backlogu, které se dostanou do dalšího Sprintu. Je to umění rozhodnout, zda je konkrétní úkol v současnou chvíli důležitý anebo je možné nechat splnění úkolu do dalšího Sprintu bez ohrožení hodnoty výsledného produktu. Product Owner by měl určovat co je v současnou chvíli prioritou za základě představ zákazníka a zájmových stran, aby výsledný produkt byl úspěšný a návratnost investic splňovala očekávání. Protože se backlog v průběhu času mění, přicházejí nové požadavky, je důležité být neustále v obraze. Podstatné je si uvědomit, že je hodnota výsledného produktu často nelineární, a větší část hodnoty se vytváří na začátku, to znamená že je důležité úkoly prioritizovat podle toho, jakou přidanou hodnotu přidají do produktu. [1]

3.3.6.4 Sprint planning

Sprint Planning je ceremonií v rámci Scrumu, smyslem, které je naplánování budoucího sprintu, v průběhu, kterého se přiřazují úkoly do sprintu, na kterých celý další sprint členové týmu budou pracovat a probíhá diskuze nad postupem pro dokončení všech úkolu. Hlavním základem plánování je stanovení cílů sprintu, kterého prostřednictvím jednotlivých úkolu chceme dosáhnout. Při porozumění si definujeme, co od sprintu očekáváme a co by mělo být jeho výstupem a přínosem. Na základě produktového backlogu a prioritizaci se pečlivě volí naplň Sprintového backlogu.

Product Owner navrhuje položky, které by měli dostat do sprintu a také navrhuje priority a Cílu Sprintu. Tým navrhuje množství úkolu, které si myslí že jsou schopni dodat v průběhu sprintu. Scrum Master vede celou tu událost, kontroluje, zda komunikace je efektivní a všichni se shodly na cíli Sprintu a nahlížet, zda se volí opravdu vhodné úkoly do sprintu. [25]

Hlavními otázkami, které by měli být v procesu plánování sprintu zodpovězené, jsou „Jaká funkcionality, přinášející přidanou hodnotu, by měla být doručena v průběhu sprintu?“ a „Jak by měla být provedená práce, aby byla přidaná hodnota skutečně doručena?“. Teprve po odpovězení na této otázky tým a Product Owner plánují nadcházející sprint. Celý proces je moderován Scrum Masterem při plné účasti týmu. [1] [29] [33]

3.3.6.5 Daily Scrum

Daily Scrum neboli Stand-up je jednou z důležitých ceremonií, které se používají v rámci Scrum metodiky. Hlavním cílem ceremonií je pravidelní sdílení informací v rámci týmu, a tím lepší pochopení průběhu sprintu, hodnocení výkonů a sledování průběhu dosažení cílů sprintu. Každý den trvání sprintu se tým schází ve stejný čas u Scrum tabulí, kde každý člen týmu sdílí, co dokončil za dobu od posledního Stand-upu za účelem dosažení cílů sprintu, na čem momentálně pracují za účelem dosažení cílů sprintu a co v současnou chvíli tvoří překážku mezi týmem a dosažením cílů sprintu prostřednictvím naplní sprintu. Stand-up jak už vyplývá z názvu neprobíhá u počítače, ale ve stojí, a tím pomáhá lidem vyjadřovat se stručně a věcně, protože Daily Scrum by neměl trvat příliš dlouho a být stručnou formou reportingu a sdílení mezi kolegy a zainteresovanými stranami. Product Owner a Scrum master jsou převážně v pozorovací roli a přímo se nezúčastňují Stand-upu, ale poznamenávají důležité informace a mohou pokládat otázky vzhledem k obsahu sdílení. [1] [29] [30] [31] [36]

3.3.6.6 Backlog Grooming

Backlog Grooming je jednou z možných aktivit, kterou může použít agilní tým v rámci práci s backlogem. Jedná se o pravidelnou schůzku, která může probíhat jednou nebo i několikrát v průběhu jednoho sprintu. Hlavním cílem této aktivity, které se zúčastňuje jak tým, tak i Product Owner, je porozumění jednotlivým položkám v rámci aktuálního sprintu.

Jedná se nejen o rozbor konkrétní naplní úkolu a požadavku v rámci zadání ale také o přínos úkolu, souvislosti a taky stanovení kritérií. Lepší porozumění položkám backlogu vede nejenom k lépe odvedené práci ale také k jednoduššímu a efektivnějšímu plánování Sprintu. [3] [1]

3.3.6.7 Produktový a Sprint backlog

Produktový backlog je seznamem věci, které by měli být udělané v rámci projektu. Naplní produktového backlogu můžou tvořit manažeři, samotným týmem, zákazníky.

Není statický, ale jeho obsah se pravidelně mění a rozšiřuje se o nové úkoly, priorit, kterých se mění v závislosti na změnách v backlogu. [1]

Backlog by měl být jediným místem, kde se nacházejí všechny úkoly pro tým. Neznamená to, že všechny úkoly v backlogu budou udělané, ale všechny úkoly, na kterých bude tým pracovat se budou nacházet v backlogu. Souhrnně položky v backlogu by měli:

1. Vytvářet hodnotu pro zákazníka
2. Být uspořádané na základě prioritizace
3. Mít odhad doby trvání
4. Být v průběhu projektů rozšiřované o další
5. Být podrobnější, pokud jdou do sprintu.

Sprint backlog je součástí Produktového backlogu a obsahuje hlavní hodnoty, které by měly být v průběhu Sprintu dodané. Je podmnožinou produktového backlogu, obsahující konkrétní úkoly, které potřebujeme udělat v rámci aktuální iteraci. Napln sprint backlogu si volí tým na základě priorit Product Ownera a cílů, který by měl být dosažen na konci sprintu. Je to způsob, jak z velkých úkolů v rámci projektu dostat malé a líc zpracovatelné části. [26] [1] [29]

3.3.6.8 Sprint review

Sprint review je ceremonií, která celému týmu, Product Ownerovi a Scrum Masterovi jako dodavatelovi umožňuje získat zpětnou vazbu na výsledek své práci v průběhu předchozího sprintu. Tuto zpětnou vazbu získávají bezprostředně od zákazníka, který bude ten produkt používat, a tak je schopen to vyhodnotit z hlediska svých potřeb. Požadavky a názory zákazníka se sbírají před zahájením práci týmu, aby se mohla definovat představa budoucího produktu, ale právě sběr pravidelných zpětných vazeb na již vytvořené funkcionality umožňuje ověřovat, zda představy členu týmu a Product Ownera se shodují se zákazníkem a přidaná hodnota produktu, která se vytvořila prostřednictvím iteraci, odpovídá aktuálním požadavkům zákazníka a má pro něj stejnou hodnotu, jakou si představujeme. [1] [29] [34] [36]

3.3.6.9 Retrospektiva

Retrospektiva stejně jako Sprint Review je ceremonií, hlavním cílem, které je získání zpětné vazby. Zásadním rozdílem je však to, že zpětnou vazbu nezískáváme od našich zákazníků ale od samotného týmu, Product Ownera a Scrum Mastera. V procesu retrospektivy každý člen týmu analyzuje, jak probíhal předchozí sprint, co se jim na tomto sprintu líbilo a co bych chtěl dělat i nadále, co nebylo úplně podle představ a chtěl by to dělat jinak a co by rád začal dělat nově. Hodnotit se může úkoly, práce členu týmu, způsob práce apod. [36]

Retrospektiva může mít kromě úvodu několik kroků:

- Sběr informací

Výměna zpětné vazby mezi členy týmů, Produkt Ownerem a Scrum Masterem. Všichni si sdělují názory na aktuální problémy a silné stránky.

- Porozumění informacím

Cílem je pochopit podstatu problémů a také příčiny, které problém způsobili.

- Kolektivnímu návrhu možnosti

Orientační návrhy, vzniklé prostřednictvím skupinového myšlení a analytických technik jako například Brainstorming, Brainwritting apod.

- Návrh konkrétních kroků

Formulace konkrétních kroků ke zlepšení, které budou implementované v průběhu dalšího sprintu.

Výstupem retrospektivy by měli být konkrétní návrhy všech účastníků na zlepšení práci týmu v průběhu sprintu se zahrnutím toho co má zůstat stejné, co se má zlepšit, co změnit a co by se mělo zavést.[1] [29] [32]

3.3.6.10 User Story

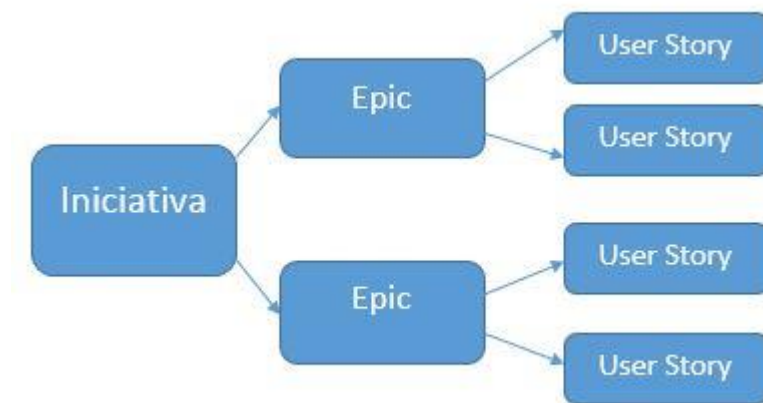
User Story je způsob, jak stručně vyjádřit představu o produktu. Zachycuje pohled na produkt očima zákazníka a definuje jejich očekávání od produktu. User Story umožňují šetření času na psaní zdlouhavých dokumentací a věnovat se doručení produktu.

V důsledku tým má možnost doručit klientovi produkt za kratší čas a tím více uspokojit jeho potřeby. [27] [1]

Při existenci více User Stories občas se vyskytuje potřeba je držet v jednom kontextu, a právě v tomto případě se používá Epic, který je představuje velký rámeček pro uchování jednotlivých User Stories se stejným kontextem (Obrázek 5).

Několik Epicu se da seskupit do většího celku, který nazýváme Iniciativou. Slouží k naplnění dlouhodobějších cílů prostřednictvím určení několika Epicu, které té cíle naplní a mají větší dopad na společnost. [28] [1]

Obrázek 5 - Rozpad úkolu

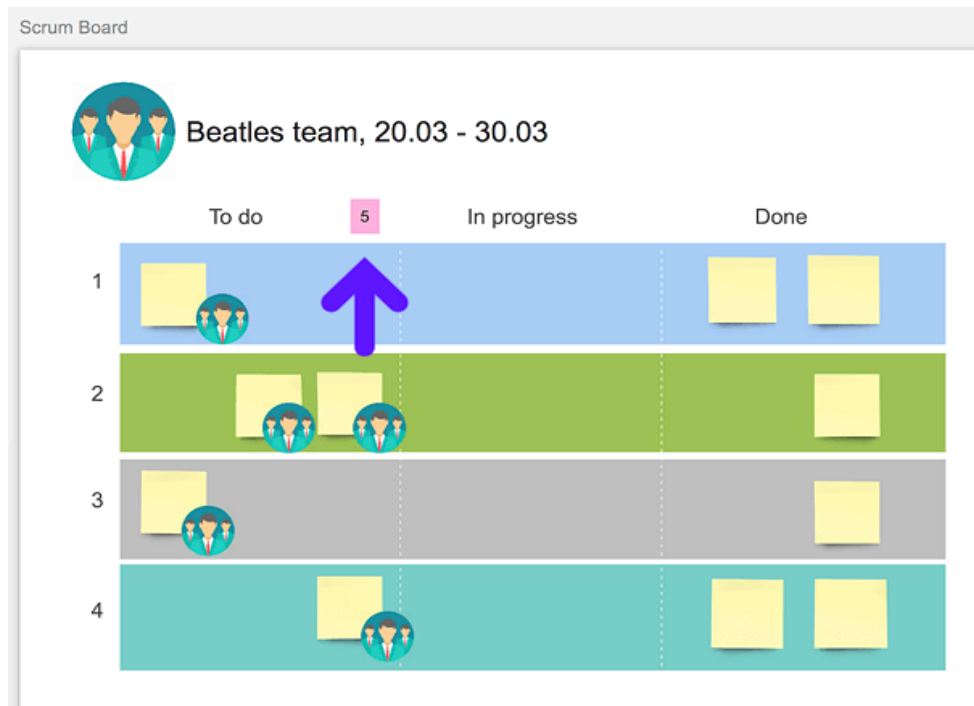


Zdroj: Vlastní zpracování

3.3.6.11 Scrum tabule

Tabule ve Scrumu je způsobem, jak využít vizualizaci v rámci každodenní práci v průběhu Sprintu. Scrum tabule je způsob, jak jednoduše a zároveň efektivně sdílet informace mezi členy týmu. Může existovat jak ve elektronické podobě, tak i ve fyzické, pro které vystačí klasická psací tabule nebo i zeď, na kterou se lepí lístky s jednotlivými úkoly (Obrázek 6). Tabule ve fyzické podobě je levným, flexibilním, přehledným a zároveň jednoduchým způsobem vizualizace pro Scrum tým. [1] [38]

Obrázek 6 - Scrum tabule



Zdroj: [39]

Scrum tabule musí obsahovat tři základní sloupce:

- „To do“, neboli „Udělat“ popisující všechny nezapočaté úkoly, které se nacházejí ve Sprint Backlogu a bude se na nich pracovat v rámci aktuálního sprintu.
- „In progress“, neboli „V procesu“, kde jsou zobrazené všechny úkoly, nad kterými již členové týmu pracují.
- „Done“, neboli „Hotovo“, kde se nachází všechny úkoly, které byli v rámci týmu řádně dokončené.

Každý člen týmu si bere úkol, napsaný na lístku, ze sloupce „To do“ jakmile začne pracovat na úkolu a přesouvá do sloupce „In progress“, kde zůstává až po dobu ukončení práci nad úkolem. Po ukončení příslušný člen týmu přesouvá ten lístek do sloupce „Done“.

Je možné podle potřeb týmu rozšířit sloupce o další, jako například „Nebylo začato“, „Specifikováno“, „K ověření“, „Testování“ apod. Předpokladem je, že každý člen týmu v rámci týmu pracuje pouze nad vyhrazenými úkoly a počet rozpracovaných úkolu členem týmu v rámci sprintu není omezen. [1] [39]

3.3.6.12 Story points

Story point je měrnou jednotkou, kterou v agilním prostředí měříme náročnost jednotlivých úkolů, která by měla zobrazovat reálnou složitost úkolu pro konkrétní tým. Právě proto, v každém týmu v rámci jedné organizace se hodnoty story pointu mohou mezi sebou lišit. Hlavním účelem hodnocení náročnosti pomocí story pointu je jednodušší a přesnější plánování sprintu.

Story pointy nejsou konstantní v čase a mohou se měnit v průběhu času v závislosti na složení týmu a jejich pracovních zkušenostech. Hodnoty, které patří ke stejnému úkolu budou se lišit v závislosti na tom, zda konkrétní tým se s podobnými úkoly setkává často nebo jsou to unikátní činnosti, na odborných znalostech a praxi jednotlivých členů týmu. S nárůstem zkušenosti a zralosti agilního týmu zároveň roste schopnost členů týmu dělat přesnější odhad story pointu pro konkrétní úkol. [41]

Pro zjednodušení procesu určení story point u úkolu se může používat Fibonacciho posloupnost - 1, 2, 3, 5, 8, 13, 21. Číslem jedná v tomto případě se hodnotí nejjednodušší úkol a číslem 21 nejsložitější. Jinou existující možností je číslování podle „triček“, kde se náročnost vyjadřuje velikostí oblečení – XS, S, M, L, XL, kde XS je nejjednodušší úkol a XL nejsložitější.

Story Points se může využívat v procesu Sprint Groomingu a plánování, kde pomáhá Product Ownerovi udělat představu o objemu práce v průběhu sprintu, v procesu měření výkonnosti, kde se může porovnávat objem dokončených story pointu mezi různými sprinty a později i pro plánování dalších projektů, kde délka trvání může být odhadovaná pomocí porovnání počtu Story Points na předcházejících projektech. [41] [1]

3.3.6.13 Produktová Roadmapa

Produktová Roadmapa je způsobem vizualizace a následného monitorování produktové vize a směřování vývoje produktu v čase podle předem stanovené strategií ve formě grafu.

Roadmapa se tvoří pro:

- definování strategií a produktové vize
- vytvoření plánu pro realizaci strategií
- zprostředkování možnosti pro diskuzi ohledně vývoje produktu
- snadnou prezentaci vize před zájmovými stranami.

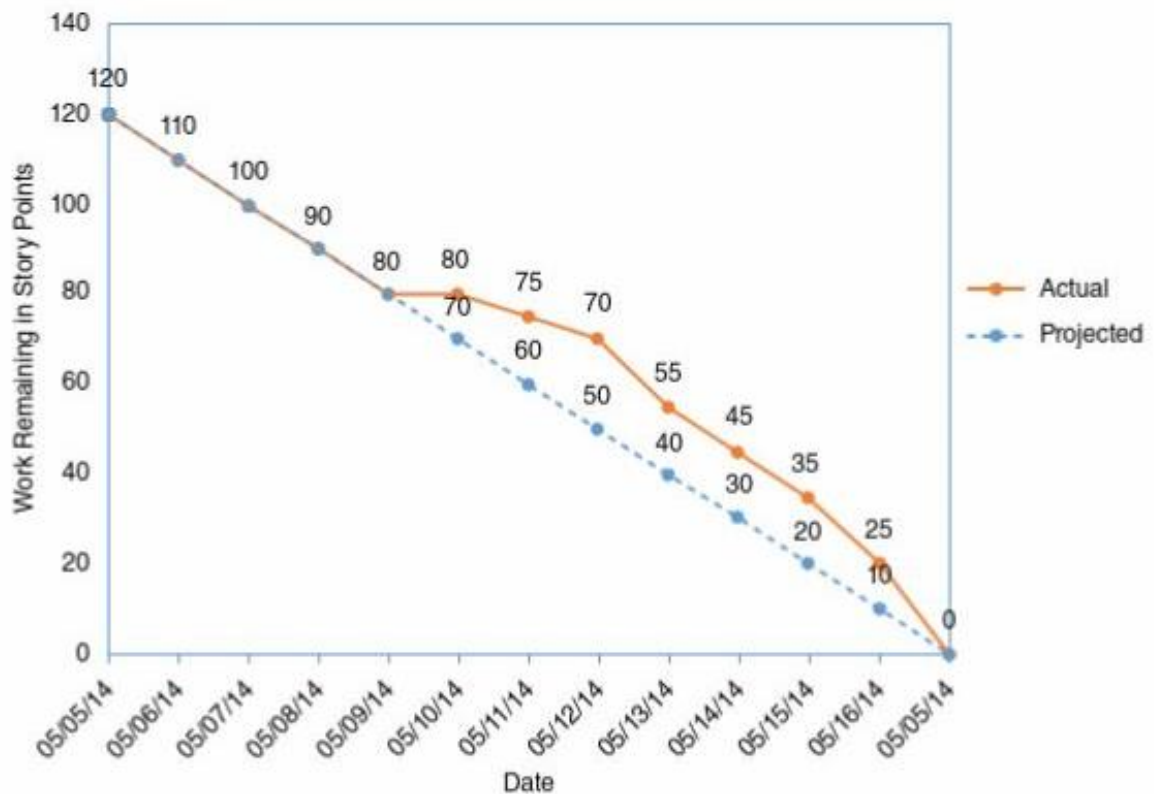
Při tvorbě Roadmapy by se mělo primárně přemýšlet nad tím, čeho chceme dosáhnout a jak se k dosažení té představy můžeme dojit prostřednictvím práce nad produktem a jakou hodnotu to bude mít pro našeho zákazníka. V průběhu práce nad produktem neustále probíhá komunikace na základě, které vznikají zpětné vazby a návrhy. Roadmapa neomezuje změny v její naplně a je možné pravidelně na základě těchto návrhů aktualizovat, což umožňuje tam mít vždy aktuální a přínosné informace. [43] [44]

3.3.6.14 Velocita

Velocita je metrikou, která vyjadřuje, kolik práce je agilní tým schopen doručit v průběhu jednoho sprintu. Nejedná se o odpracované hodiny, ale o objem doručené přidané hodnoty. Velocita je tak vyjádřena pomocí Story Points a času.

Velocita slouží jak v predikci budoucího výkonu týmu, tak zároveň k hodnocení již proběhlých sprintů. K měření velocity již dokončených sprintů se využívá Burndown graf neboli Burndown report. V Burndown grafu jsou dvě osy, X – čas, ve kterém sledujeme velocitu, a Y – počet dokončených Story Points v tomto čase (Obrázek 7- Příklad Burndown grafu). [41] [29]

Obrázek 7 - Příklad Burndown grafu



Zdroj: [41]

3.3.7 Kanban

Samotné slovo Kanban je původem z japonštiny, kde v doslovném překladu znamená „označovací tabule“ nebo „informační tabule“. Metodika Kanban tak vznikla v automobilové manufaktuře Toyota v Japonsku v 50 letech 19. století, ale začala vyvolávat zájem až v 70. letech 19. století. Původní zamýšlené k snížení nákladů a řízení opotřebovávání strojů, Kanban se dodnes používá ale také k určení překážek ve výrobě a neustálému zlepšení. [42]

3.3.7.1 Push a Pull systém

Jedním ze základů pochopení metodiky Kanban je Push a Pull systém, kde pod pojmem Push se rozumí klasický přístup k plánování na základě odhadu budoucí poptávky. Podnik dělá odhad toho, jakou poptávku bude mít konkrétní produkt. Pak se provádí výpočet pracovních a na jejím základě se provádí alokace a zajištění zdrojů.

Pravým opakem toho je Pull systém, který je primárně řízen zákaznickou poptávkou. Děláme pouze to, na co existuje reálná poptávka, a na jejím základě plánujeme práci a rozvrhujeme zdroje. [41]

Kanban je metodikou, která se staví na základě Pull systému, rozvrhování na základě poptávky. Produkt by v tomto případě měl být vytvořen v jednom ze dvou případů:

- jako náhrada produktu, který byl již využit zákazníkem
- jako reakce na jasně vyjádřený zájem zákazníka o produkt. [42]

3.3.7.2 Podstata a hodnoty

Kanban se staví na několika základních principech:

- začínat s již rozpracovanými úkoly
- ochota následovat vývojové změny
- respekt k existujícím rolím a odpovědnostem
- podporovat vůdcovství bez rozlišení úrovně jeho výskytu.

Kanban není procesem, proto se používá hlavně pro optimalizaci a zlepšení procesu, které již v rámci podniku existují. [45]

Důležitými body v Kanbanu jsou:

1. Vizualizace práce

Vizualizace pomáhá k lepšímu přehledu o práci, o jednotlivých úkolech a jejich aktuálním stavu. Podstatou je využití Kanban tabule nebo softwaru, který tuto tabuli nahrazuje a umožňuje sledování úkolu v elektronické podobě.

2. Omezení rozpracovaných úkolu

Omezení rozpracovaných úkolu (WiP – work in progress) podporuje stanovení limitu na počet rozpracovaných úkolu jedním členem týmu na objem práce, kterou schopen zvládat najednou. To znamená, že každý člen týmu může mít pouze několik aktivních úloh a může začít pracovat na dalším pouze po dokončení jednoho z rozpracovaných úkolu. Tato zásada je preventivním krokem proti vytvoření nepřiměřeně velké fronty započatých ale neukončených úkolu.

3. Řízení dodací lhůty

V Kanban přístupu dodací lhůta by měla být měřená a řízena za účelem její optimalizaci a snižovaná na minimálně přípustné hranice. Průměrný čas dodání funkcionalit by měl být předvídatelný a dodací cyklus malý. Měřit se můžou rozdílně úseky výrobního cyklu, ale základem je měření doby od začátku práci nad úkolem po nasazení jeho výstupu.

4. Transparentnost

Tým by měl udržovat transparentnost ohledně procesu a zásad, které stojí za práci týmu. Napomáhá to porozumění tomu, co celý tým dělá a umožňuje jednoduché následování pracovních postupů a umožňuje navrhnout případné zlepšení.

5. Získávání zpětné vazby

Při doručování je důležitá nejenom rychlost ale spokojenost zákazníku s dodáním výstupem. Spokojenost zákazníku zajišťuje výnos firmy a jeho úspěšnost, proto je podstatné dostávat zpětnou vazbu nejenom od kolegu ale také od koncových uživatelů budoucího systému, poskytující informace, zda je doručovaný produkt je v příslušné kvalitě a naplňuje očekávání zákazníka.

6. Kolektivní implementace zlepšení

Návrh změn, které mohou být čerpané jak ze zkušenosti, tak i existujících teorií – teorií chaosu, teorií her, teorií omezení apod. Mají vycházet od členu týmu v pracovním procesu a být použité pro zlepšení existujících procesu. Nastávající změny by měli být přijaté po souhlasu všech zúčastňujících se stran. [47] [45] [46] [50]

3.3.7.3 Lead time

Lead time neboli dodací lhůta je pojmem, který v Kanban prostředí označuje časový úsek mezi vznikem konkrétního úkolu a jeho ukončením. V praxi se často za lead time označuje časový úsek od zahájení členem týmu práci nad úkolem po jeho ukončení.

Dodací lhůta se používá pro měření tempa doručení, pro které platí Littlův zákon, který říká, že tempo doručení se rovna rozpracované práce dělená doručovací lhůtou (Obrázek 8- Littlův zákon). [50]

Obrázek 8 - Littlův zákon

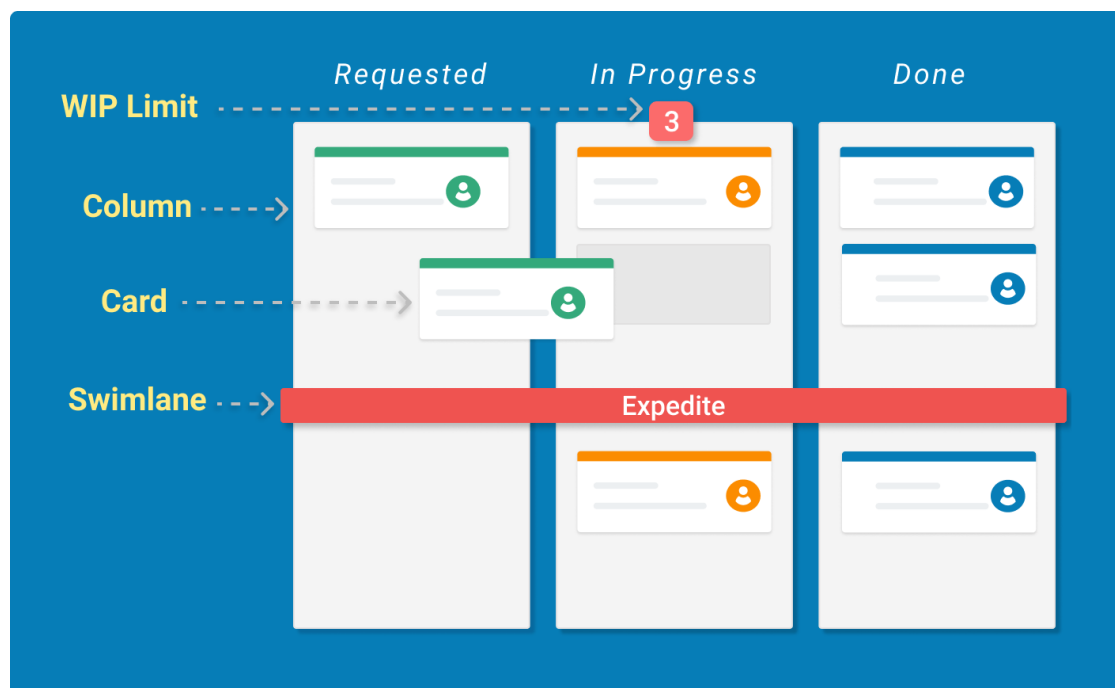
$$\overline{Delivery\ Rate} = \frac{\overline{WIP}}{\overline{Lead\ Time}}$$

Zdroj: [50]

3.3.7.4 Kanban tabule

Kanban tabule neboli Kanban board je způsobem, jak práci vizualizovat. Může vypadat jako obyčejná tabule nebo být zastoupena softwarem, který tu tabuli imituje. Softwarové řešení je flexibilní a má možnosti implementování více funkcionalit, zároveň fyzická tabule více zapojuje členy týmu do analýzy současného stavu, protože musí psát jednotlivé úkoly na tabuli, posouvat je v závislosti na úrovni rozpracovanosti a tím vede k větší pozornosti a uvědomování fungování procesu. [48]

Obrázek 9 - Kanban tabule



Zdroj: [49]

Kanban tabule se skládá z:

- Karet

Karty ilustrují úkoly. Každá karta obsahuje informaci o úkolu a po započnutí je přidělena jednotlivým členům týmu.

- Sloupců

Každý sloupec ilustruje stav rozpracovanosti úkolu, a při každé změně stavu se posouvá dál, až bude ve sloupci s dokončenými úkoly.

- WIP

Omezení rozpracovaných úkolu neboli WIP ukazuje, kolik rozpracovaných úkolu se může nacházet najednou v konkrétním sloupci. Pokud je kapacita naplněná, další úkol může tam být posunut až po dokončení jednoho z již rozpracovaných úkolu.

- Rozdělovací linie

Linií rozdělují úkoly, které se nachází ve stejné fázi rozpracovanosti, ale liší se typem aktivity, týmy apod. a tím zpřehlednit probíhající práci. [49]

3.3.7.5 Role v Kanbanu

Kanban v své původní podobě nevyžaduje žádné definování rolí a odpovědnosti a vytváření nově pojmenovaných pozic. Z uplatnění metodiky v praxi, ale vznikly dvě hlavní role, které byly zanesené do metodiky a mohou být definovány za místo určení konkrétní role pro každého členu týmů.

- Správce požadavku na služby

Odpovídá za porozumění požadavku zákazníka a za prosazování jeho zájmu v průběhu práci nad produktem, volbu a uspořádání úkolu. Je to alternativní rolí Product Ownera v metodice Scrum.

- Správce dodávek služeb

Odpovídá za dodání funkcionalit zákazníkovi a řízení průběhu práci včetně potřebných schůzek a plánování. Alternativními názvy pro tuto roli může být manažer dodávek nebo manažer toku. [50]

3.3.8 Scrum tabule vs Kanban tabule

Kanban tabule je nástrojem pro měření a regulování pracovního toku s omezením počtu rozpracovaných úkolů. Omezení počtu rozpracovaných úkolů snižuje objem front a vede k rychlejšímu dodání. Scrum tabule je nástrojem pro měření práce v průběhu sprintu. Délka sprintu zajišťuje možnost se soustředit a zároveň postupně dodávat přidanou hodnotu.

1) Omezení

Ve Scrumu existuje omezení v počtu úkolů, vykonávaných v průběhu sprintu. Při naplnění objemu možné práce, není možné pracovat nad jinými úkoly, které se do sprintu nedostali. Kanban neomezuje počet úkolů v backlogu, ale neumožňuje mít v konkrétním stavu rozpracovanosti – sloupci – více úkolů, než je stanovený limit pro tento sloupec a další úkol může být započat pouze po přesunutí úkolů, který je již ve sloupci do dalšího sloupce.

2) Vlastnictví

Scrum tabule patří jednomu Scrum týmu. To znamená, že úkoly, které se nachází ve Scrum tabule bude zpracována právě tím týmem, který ten úkol vlastní.

Kanban tabule nemusí patřit konkrétnímu týmu. Často patří určitému toku práce, který sdružuje vše úkoly na konkrétním produktu nebo skupině produktu od několika týmů.

3) Úprava obsahu vlastníkem produktu

Ve Scrumu Product Owner nemůže zasahovat do již odsouhlaseného týmem obsahu Sprintu, když v Kanbanu se obsah může upravovat někým, kdo rolí produktového vlastníka zastupuje.

4) Obsah iterací

Obsah iterací ve Scrumu se určuje před jeho začátkem a jeho kapacita následně se nemůže. Kanban neomezuje počet úkolů, které se mohou dostat do tabule, protože existující omezení rozpracovaných úkolů. Odhady doby trvání a objemu dokončených úkolů se stanovují na základě pozorování trendu práce týmu v čase.

5) Naléhavé úkoly

V rámci Kanban tabuli existuje možnost rozšíření o sekci, kam se mohou dostat úkoly, které mají vysokou prioritu a za nějakých výjimečných podmínek potřebují být odbavené co nejdříve je to možné. Podobné úkoly se zobrazují pod rozdělovací linií. Ve Scrumu se nepočítá s nastáním podobné situace, poněvadž tým klade důraz na analýzu, prioritizaci, plánování iteraci a predikci, a produkt je adaptivní.

6) Backlog

Ve Sprintu úkoly se dostávají do Sprint backlogu z produktového backlogu, a odtud se posouvají po tabuli. Kanban může mít backlog, ale sklada se ze všech úkolů bez rozlišení na sprint a produktový backlog.

7) Velikost úkolu

V rámci Scrumu, se velké úkoly by měli být rozdělené do co nejmenších částí, a až pak té jednotlivé části by měli být přidáné do tabule. Kanban neklade žádné omezení na velikost jednotlivých úkolů, co souvisí s tím, že nemá časové ohraničené iterace.

8) Obsah tabuli

Scrum porozumívá, že na konci každého sprintu, všechny úkoly se budou nacházet v posledním sloupci a budou řádně dokončené, jinak sprint může být považován za neúspěšný. Po analýze chyb se nedodělané úkoly přesouvají do dalšího sprintu, který je naplňován novými úkoly na základě vytvořených priorit.

Kanban tabule se nemusí zásadně měnit v průběhu času, protože nemá konkrétní časové ohraničení. Cyklus práce je nepřetržitý a úkoly se přidávají a odebírají v závislosti na potřebách v průběhu celého projektu. [39]

3.3.9 Extrémní programování

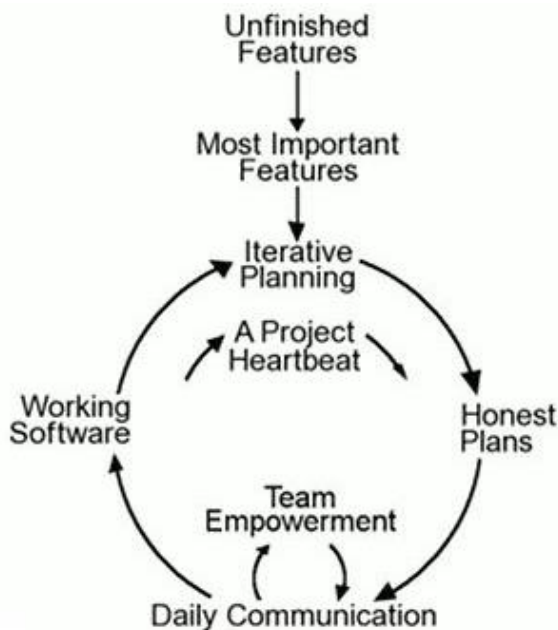
Extreme Programming neboli extrémní programování je agilní metodikou, nejvíce rozšířenou na začátku roku 2000, založenou na souboru hodnot, postupu a principu, podporující efektivní softwarový výstup. Základním kamenem úspěchu extrémního programování je doručování produktu a funkcionalit, které maximálně uspokojí zákazníka. Důležitým bodem je, že tým se nesnaží dodávat maximálně možný objem práce, ale soustředí se na dodávání právě toho, co klient potřebuje a bude využívat, toho, co bude

opravdu využívat. Charakteristickým pro extrémní programování je akcent na spolupráci v týmu a rovnocennosti jak členu týmu, tak manažeru a zákazníku. [51]

3.3.9.1 Základní hodnoty

Extrémní programování obsahuje sadu hodnot, které slouží jako podklad pro principy a praktiky v rámci metodiky.

Obrázek 10 - Extrémní programování



Zdroj: [51]

1) Komunikace

Jedním z nejdůležitějších faktorů pro úspěšnost týmu je efektivní komunikace, která je pravidelná a probíhá v otevřeném prostředí. V průběhu komunikace se probírají aktuální problémy a navrhuje se řešení, na které bez efektivní komunikace týmy nemusí přijít.

2) Jednoduchost

Dalším způsobem, jak dosáhnout úspěchu ve vývoje je dělat věci jednoduše, čím zároveň bude maximalizovaná kvalita výsledného produktu. Týká se to jak kódu, tak používaných procesů a dokumentace. Jednoduchost je způsob, jak dodat to, co zákazník potřebuje, a

zároveň předejít vytvoření přidané hodnoty, kterou zákazník nepotřebuje a pravděpodobně nebude využívat.

3) Zpětná vazba

Pravidelní zpětná vazba pomáhá týmu zjistit, jak se jim daří. Zpětná vazba by měla být nejenom pravidelní ale taky často opakovatelná, aby se zajistilo, že doručovaný produkt splňuje očekávání a je v dostatečné kvalitě. Zároveň nemusí být poskytována zákazníkem nebo odborníky, ale taky systematické sbíranými informacemi v podobě grafu, výsledku testu apod.

4) Respekt

Každý člen týmu a jeho práce by měla být respektovaná. Každý přínos je považován za podstatný. Respekt by měl být mezi týmem, managementem a zákazníkem, a to v otázkách profesní odbornosti, odpovědnosti a autority.

5) Odvaha

Pro každého člena týmu je důležité mít dostatečně odvahy na to, aby se mohl pouštět do nových a doposud nevyzkoušených věcí. Důvodem, proč je to tak náročné, je strach před něčím neznámým, který tým musí překonat, aby mohl dosahovat zřetelných výsledků. Kromě toho, musí umět přiznat, že potřebuje pomoc, když se setká s problémem, který neumí vyřešit samostatně a taky být upřímní ohledně skutečného pokroku a odhadu pracovní. [52] [53]

3.3.9.2 Praktiky

Základem metodiky extrémní programování jsou tyto 12 praktik, se kterými se můžeme v setkat. Mohou být používány samostatně, anebo společně, protože se navzájem doplňují a tím poskytují týmu více možností.

1) Plánovací hra

Hlavní myšlenkou je rychle pochopení představy zákazníka a hrubého odhadu nákladu na její uskutečnění. Zákazník vytvoří plán požadavku, pak tým vytváří odhad doby trvání a nákladu, následně zákazník určuje, zda disponuje potřebnými prostředky a je ochoten do

toho investovat. Předěje to ukončení rozdělaného projektu, na který buď nestačí finanční prostředky nebo přestane být aktuální. [54] [53]

2) Vydávání malých verzí

Pravidelní vydávání verzí produktu, které ilustrují přidanou hodnotu pro zákazníka, jsou více uspokojující pro zákazníka než nepravidelné vydávání s velkými časovými odstupy. Při menších a častějších vydáváním zákazník je schopen pravidelně ohodnocovat vynaložené investice, protože vidí změny výsledného produktu. Kromě toho tým je schopen dostávat pravidelní zpětné vazby a pravidelně je implementovat. [54] [53]

3) Metafora

Popis systému prostřednictvím metafor poskytuje možnost vyjádřit komplexnost systému jednoduše. Členové týmu nemusí na každé schůzce kompletně vysvětlovat schování a funkcionality systému, a místo toho použije metaforu, která je rychlá, jednoduchá a zároveň dokáže uvést jiného člověka do představy fungování produktu. Měla by být pochopitelná, a jakmile je zavedená v projektu, měla by být po celou dobu trvání projektu zachovaná a používána napříč celým týmem. [54] [53]

4) Jednoduchý design

Návrh designu by měl jednoduchý, a uspokojovat pouze definitivně vyjádřené požadavky zákazníka na produkt. Jednoduchý design je zároveň více flexibilní vůči změnám, které v průběhu projektu mohou vzniknout. Zároveň produkt by měl být doručen v nejvyšší možné kvalitě. [54] [53]

5) Testování

Všechny úkoly by měli být testované pomoci automatického testování. Pravidelní a prvotní testování je klíčové pro doručení kvalitního produktu, protože chyby a nesrovnalosti se podchycují průběžně. [54] [53]

6) Refaktorizace

Pod pojmem Refaktorizace se rozumí proces změny existujícího softwaru, který zlepšuje vnitřní uspořádání kódu, ale není zřetelný zvenku při použití softwaru. V procesu refaktorizace dochází k zlepšení struktury kódu a taky k lepší a jednodušší udržitelnosti. Je to procesem zlepšení a zjednodušení, které zvyšuje kvalitu kódu. V rámci refaktorizace se

provádí odstranění nepoužívaných funkcionalit, zlepšení struktury a designu kódu, modernizace zastaralých částí kódu, odstranění duplicit a opakujících se částí. [41]

7) Párové programování

Párové programování je proces, kterého se zúčastňují dva vývojáři pracující společně na jednom kódu. Hlavním důvodem, proč se toho procesu zúčastňují zároveň dvoje vývojáři je, že se šetří čas potřebný k revizi kódu a zároveň se snižuje zátěž na testování. V procesu psaní kódu jeden vývojář se věnuje přímo promyšlení a napsání a druhý vývojář na hledání možných chyb, hodnocení toho, jak ten kód může ovlivňovat jiné související systémy, nad testovacími scénáři a podobně. [54] [53]

8) Společné vlastnictví

V procesu vývoje se nejčastěji setkáme s tím, že jeden vývojář vlastní nějaký produkt, systém, část kódu, na které pracuje výhradně on. Společné vlastnictví umožňuje ostatním vývojářem zasahovat do kódu, na kterých běžné nepracují. Hlavní myšlenka tkví v teorii omezení – pokud každý vývojář bude pracovat pouze na vlastním úseku kódu, vytvoří se „úzké místo“, a tak celý systém bude náchylnější. V případě, že se kód udržuje více vývojáři najednou, roste pravděpodobnost, že kód bude méně specifický, a tak jeho údržbu bude moct provádět i někdo kromě vlastníka systému. [54] [53]

9) Nepřetržitá integrace

Continuous integration neboli nepřetržitá integrace spočívá v pravidelné integraci nového kódu do produkčního softwaru. Minimální interval integrace by měl činit 1 den. Po každé integraci by měly být spuštěné testy, které odhalují, zda po integraci došlo k chybám a nesouladům v kódu, které je potřeba okamžitě upravit.

Zásadní výhodou integrace malých částí kódu je zjednodušení hledání chyby, pokud při integraci dojde k rozladěnosti uvnitř systému. Je značně jednodušší najít chybu v menší části kódu než při integraci jednou měsíčně. Zároveň oprava menší chyby trvá méně, a tak produkt se udržuje stále aktuální a v nejlepší funkčnosti. [55] [41]

10) 40 hodin týdně

Pro zajištění stálého tempa a maximálního výkonu tým by měl dodržovat určité množství práce odvedené za časový úsek. Množství plánované práce by mělo být adekvátní a za plánovaný časový úsek proveditelné. Správné rozvrhování by mělo předejít vzniku přesčasu a tím pravidelného přetížení, které při dlouhodobém opakování vede k celkovému snížení efektivity práci a zřetelně nižší kvalitě produktu. [54] [53]

11) Účast zákazníka

Při vývoje produktu je základním cílem uspokojit potřeby zákazníka. Dosáhnout maximálně uspokojivého výsledku je možné v tom případě, že zákazník, ten člověk, který bude opravdu využívat budoucí funkcionality nebo člověk, který má hluboké znalosti o požadavcích a specifikacích produktu, bude účastníkem vývojového procesu. Získávání pravidelné zpětné vazby snižuje možnost toho, že budoucí produkt nebude naplňovat očekávání. [54] [53]

12) Kódovací standarty

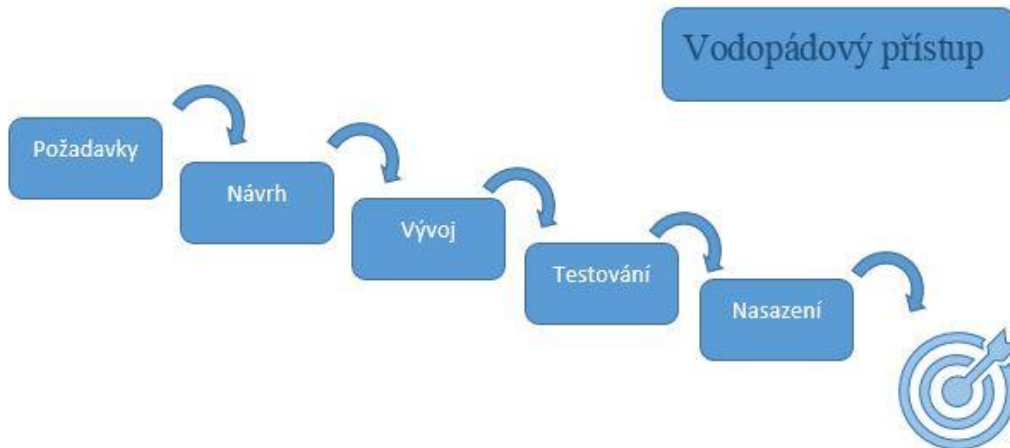
Standart kódu by měl být jediný pro všechny vývojáře které pracují společně v rámci vývoje produktu. Individuální formát kódu u vývojářů, které pracují na společném produktu vede k nekonzistentnímu stavu kódu a v dlouhodobém hledisku znamená postupné snížení přehlednosti a udržitelnosti kódu. Dodržování standartu zároveň umožňuje refaktorizace, protože kód je jednotný a srozumitelný pro všechny, kdo s ním pracuje. [54] [53]

Refaktoring, nepřetržitá integrace a párové programování jsou procesy, které i přes to, že se řadí do metodiky extrémního programování, jsou využívány v agilním prostředí i mimo tuto metodiku, nejsou na ní specificky vázané a nevnášejí do jiných metodik žádné restrikce. [41]

3.4 Porovnání Lean, vodopádové a agilních přístupů

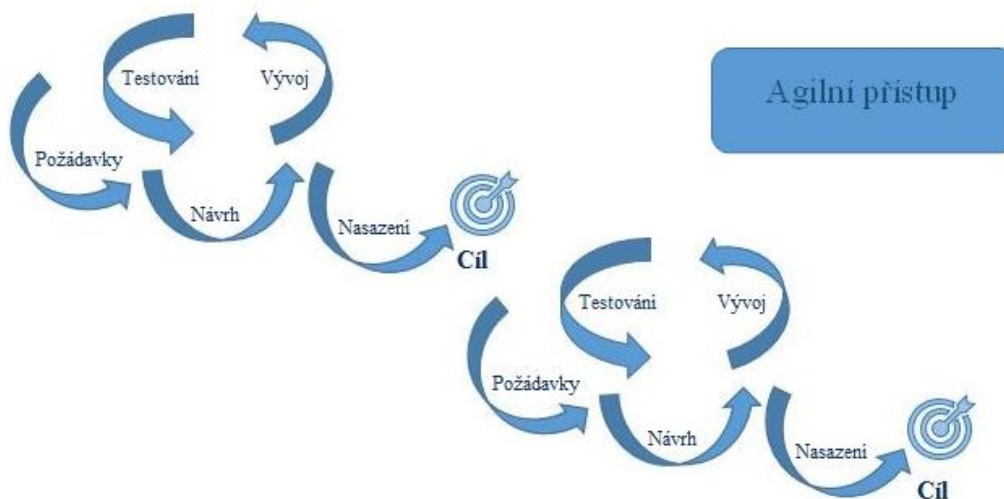
Oboje vodopádový (Obrázek 11) a agilní (Obrázek 12) přístupy k řízení jsou srovnatelně uplatnitelné pro projekty zvláště v oblasti softwarového vývoje. Není možné porovnávat, který z přístupů je lepší, protože mají rozdílný pohled na celý proces.

Obrázek 11 - Proces doručení přidané hodnoty ve vodopádovém přístupu



Zdroj: Vlastní zpracování

Obrázek 12 - Proces doručení přidané hodnoty v agilním přístupu



Zdroj: Vlastní zpracování

Hlavní rozdíl mezi vodopádovým a agilním přístupem spočívá v průběhu vytvoření přidané hodnoty produktu. Ve vodopádovém přístupu přidaná hodnota vzniká na konci projektu, jakmile většina práce už je odvedená, po celou dobu trvání projektu se zásadně nemění. Zcela opačný průběh má vytvoření přidané hodnoty v agilním prostředí.

Zásadou je postupné, pravidelní přidání hodnoty produktu. Přidaná hodnota ve formě nových funkcionalit se v malých iteracích vzniká, testuje se, implementuje se a následně se ověřuje se zákazníkem.

V shrnutí se da říct, že zásadními rozdíly mezi vodopádovým a agilním přístupem jsou:

- míra strukturovanosti
- úroveň před definovanosti před začátkem projektu
- míra flexibility ke změnám v průběhu projektu
- způsob vytváření přidané hodnoty
- proces získávání zpětné vazby. [56] [41]

Při porovnání agilního přístupu a Lean přístupu je možné identifikovat některé společné prvky:

- Kontinuální zlepšení
- Vysoká kvalita výstupního produktu
- Efektivita práci
- Spolupráce v týmu a motivovanost.

Avšak mají i pouze své charakteristické prvky. Lean přístup je primárně zaměřen na optimalizaci výrobního procesu, doručení hodnoty, minimalizaci odpadu. Agilní přístup se zaměřuje na zákazníka, rychlou adaptaci, iterativní a inkrementální vývoj a dodání produktu podle nejlepších představ zákazníka. [11] [12] [13]

4 Praktická část

V praktické části bude představená firma a následně bude popsána metodika, která je používána v rámci práce týmu nad projekty v oblasti vývoje softwaru. Kromě toho bude ohodnocené využití týmem agilních metodik v praxi a navrženo doporučení pro zlepšení současného stavu.

V praktické části budou používány odborné pojmy v anglickém jazyce z důvodu jejich použití bez překladu v praxi vybrané společnosti, kde je angličtina jazykem firemní komunikace a dokumentace.

4.1 Představení firmy

Název firmy: Avast Software s.r.o.

Datum vzniku: duben 1988

Sídlo: Píkrtova 1737/1a, Nusle, 140 00 Praha

Právní forma: Společnost s r.o.

Generální ředitel: Ondřej Vlček

Zakladatele: Pavel Baudiš a Eduard Kučera

Logo společnosti (Obrázek 13):

Obrázek 13 - Logo společnosti



Zdroj: [60]

Avast je jedním největších poskytovatelů zabezpečení na světě, poskytující software pro více než 435 milionů uživatelů po celém světě. Ve 59 zemích počet uživatelů přesahuje 1 milion. Avast je česká firma, která byla založena v roce 1988 ale pod původním názvem Alwil Software. Kromě klasického antivirového zabezpečení firma nabízí software, který zvyšuje bezpečnost na internetu, software předcházející sledování na internetu a taky software zvyšující a optimalizující výkon počítače.

Některými z produktu firmy Avast jsou:

- Avast Premium Security
- Avast SecureLine VPN
- Avast Battery Saver.

Rostoucí portfolio firmy se zaměřuje na:

- Detekci a zamezení vírových napadení
- Umělou inteligenci a strojové učení
- Internet věci
- Technologií se zaměřením na lokaci.

Centrála společnosti se nachází v Praze, ale pobočky firmy se nachází po celém světě a počet zaměstnanců přesahuje 1700 lidí (Obrázek 14).

Obrázek 14 - Světové pobočky



Zdroj: [57]

Některými z cílů společnosti jsou:

- Soukromí a bezpečnost uživatelů pomocí pokročilého softwaru na ochranu před kybernetickými útoky.
- Bezpečnost rodin, kde rodiči mají dostatečný přehled o poloze a aktivitách svých dětí.
- Ochrana malých a středních podniku před poškozením a ztrátami v následku kybernetického útoku.

- Výkonnost uživatelských zařízení pomoci softwaru, které zvyšují efektivitu používání zařízení a optimalizaci jejich fungování.
- IoT bezpečnost pro všechna chytrá zařízení v domácnosti, které díky svým širokým funkcionalitám a přístupu k internetu potřebují dostatečnou ochranu. [57]

4.1.1 Strategie a KPI

Jednou z již naplněných strategií firmy bylo získání většího podílu uživatelů, duševního vlastnictví a vstupu na nové trhy formou akvizici. Akvizice proběhla v roce 2016 se společností AVG Technologies.

V současné době hlavním strategickým zaměřením firmy je investice do technologií a rozvoje nových způsobů detekce a prevence hrozeb uživatelům. Investice do výzkumu a vývoje, kde současně pracuje 45 procent zaměstnanců firmy, umožňuje rozvoj nových a moderních produktů, které zvyšují zájem uživatelů, který je zpeněžován jak u stávajících, tak i u nových uživatelů.

Klíčové ukazatele (KPI) jsou v souladu s dlouhodobými cíli společnosti ohledně růstu, a jimi jsou:

- Finanční ukazatele: Upravené příjmy, EBITDA v hotovosti, EBITDA, čistý příjem a FCF.
- Provozní ukazatele: Počet zákazníků, průměrný příjem na zákazníka (ARPC) a průměrný počet produktů na zákazníka (APPC). [69]

4.1.2 Brandy společnosti

Obrázek 15 - AVG logo



Zdroj: [61]

Značka Avast je vedoucí a základní značkou firmy, pod kterou prodává svoje produktu. Po akvizici v roce 2016 firma převzala vedení nad českou firmou AVG Technologies, která je v současnosti dceřinou společností firmy a prodává svoje produkty stále pod značkou AVG (Obrázek 15). V současné době značka AVG nabízí zákazníkům produkty zaměřené na antivirovou ochranu, na optimalizaci výkonu a ochranu soukromí na internetu.

Některé z produktů značky AVG jsou:

- AVG Ultimate
- AVG AntiTrack
- AVG TuneUp.

Obrázek 16 - CCleaner logo



Zdroj: [63]

Další značkou, která je zároveň dceřinou společností firmy Avast je CCleaner, která byla zakoupená v roce 2017 (Obrázek 16). Firma se zaměřuje na vývoj softwaru pro optimalizaci výkonu počítače. CCleaner nabízí produkty pro různé platformy, CCleaner prohlížeč.

Obrázek 17 - HMA logo



Zdroj: [62]

HMA je firma, poskytující VPN (Virtual private network – Virtuální privátní síť) software, která je dceřinou společností Avast od roku 2016 (Obrázek 17). Hlavním posláním firmy je chránit bezpečnost uživatelů na internetu a pečlivě šifrovat veškerou jejich činnost, a tím

je činí plně anonymními před zneužitím dat o uživateli. Podmínky používání však zakazují použití softwaru pro ilegální účely.

Všechny značky také nabízí softwarové řešení pro firmy, přizpůsobené firemním účelům a podmínkám použití.

4.1.3 Nadační fond Avast

Obrázek 18 - Nadační fond logo



Zdroj: [64]

Nadační fond Avast vznikl v roce 2010 pomocí finančních prostředků, věnovaných společností Avast a taky práci jejich zaměstnanců jako jeden ze způsobů prosazování odpovědného chování. Fond se snaží o zlepšení života jednotlivců a komunit, které zapadá do představ firmy o bezpečnějším a lepším světě.

Hlavní misí fondu je zlepšení kvality lidského života a taky o právu jednotlivce rozhodovat o tom, jak si prožije svůj život. Hlavními oblastmi, ve kterých nadační fond působí jsou:

- Podpora hospiců a rozvoj péče o starší a nemocné lidé v ČR
- Podpora rodin s dětmi se zdravotním omezením
- Podpora neziskových organizací zaměstnanci firmy
- Vzdělávání dětí v oboru kybernetické bezpečnosti
- Podpora místních dobrovolnických a neziskových organizací v ČR. [58]

4.2 Projektové řízení v podniku

V rámci společnosti probíhá celá řada projektů, část, kterých je zaměřená na softwarový vývoj. V současné době společnost se nachází v procesu agilní transformaci, která začala na konci roku 2018 a většina projektu, které se týkají softwarového vývoje se momentálně řídí agilním přístupem.

Před zahájením agilní transformaci se projekty řídili vodopádovým modelem, však bez použití konkrétní projektové metodiky a standartu.

Projektový manažer inicioval projekt na základě požadavku zákazníka a za podpory vedoucího divize kam spadal. Organizační struktura se tak skládala z vrcholového managementu, který schvaloval iniciace projektu, projektového manažera, který se věnoval vypracování, realizaci a kontrole projektu, a projektového týmu. Kromě projektového manažera projekt by měl být schválen vedoucím divize, kam patří projektový manažer a projektový tým. Projekt tak procházel klasickými fázemi projektu, kde se vypracovávali příslušné studie, potřebná dokumentace, plánování a rozvrhování zdrojů, odhady dob trvání a nákladu na projekt.

V současné době projekty, zaměřené na softwarový vývoj jsou řízené agilně, pomocí jak základních myšlenek agilního přístupu, tak i použití konkrétních metodik. Každý tým ale má možnost si přístup zvolit samostatně bez jednotného nařízení ze strany vedení společnosti. Agilní hodnoty však by měli být dodržovány napříč celou firmou.

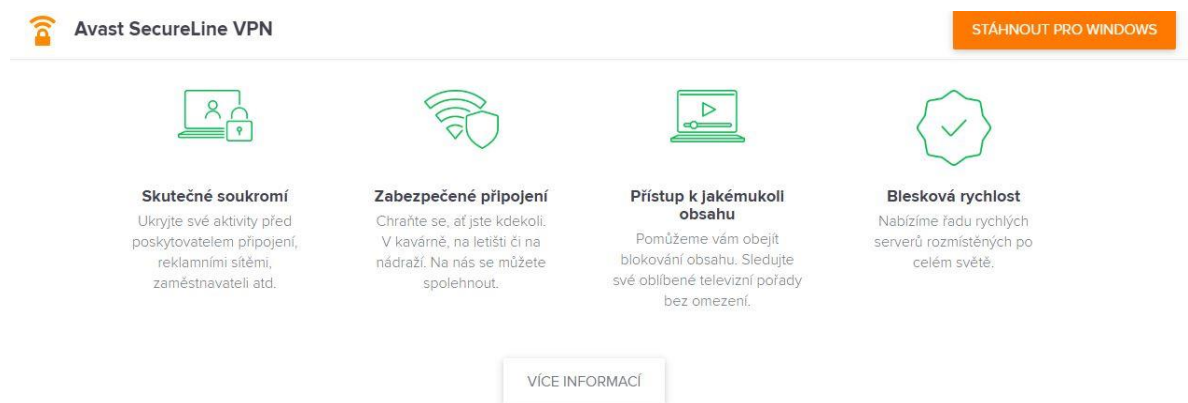
4.3 Projekt

V rámci této práci byl sledován jeden konkrétní projekt ve zaměření na softwarový vývoj. Hlavním cílem sledovaného projektu je vývoj nového produktu, který rozšíří stávající portfolio produktu značky Avast. Funkcionalitou budoucího produktu je poskytování bezpečného použití internetu na zařízeních uživatelů, které bude shrnovat více podstatných funkcionalit již existujících produktu a poskytovat uživatelů větší uspokojení potřeb v oblasti bezpečnosti na internetu (Obrázek 19). Doba trvání projektu je 18 měsíců.

Klíčovým nápadem projektu je sdílení znalosti mezi dceřinými společnostmi a společností Avast v oblasti vývoje softwaru pro používání technologií virtuální privátní sítí, na základě, které jsou nabízené produkty značek Avast, AVG a HMA. Tak na práci se podílí zaměstnanci nejenom z mateřské společnosti a tím bude docíleno efektivnějšího využití znalosti v této oblasti.

V současné době již proběhla tvorba prototypu produktu, na základě, kterého bylo rozhodnuto o zahájení vývoje produktu. Důležitým bodem je možnost částečného použití kódu z již existujících produktu.

Obrázek 19 - Příklad VPN produktu



Zdroj: [65]

Cílovým stavem je kompletní dokončení softwarového produktu za základě již existujícího prototypu a jeho následného uvedení na trh.

4.3.1 Metodika v rámci projektu

V rámci projektu tým využívá agilní přístup k vývoji softwaru. Metodikou ale je nedávno vznikly a zatím není široce popsány hybrid dvou klasických agilních metodik Scrumu a Kanbanu – Scrumban. Podstatou této metodiky je využití rolí a procesu, používaných ve Scrumu a vizualizaci, používané v Kanbanu.

4.3.2 Důvody zavádění agilního řízení do týmu

Současný tým po provedení akvizici je tvořen členy různých týmů, které pracovali jak v mateřské společnosti, tak i v dceřiných společnostech. Po začátku spolupráci ale začalo

docházet k nespokojenosti společnosti s výkonem a rovněž k nespokojenosti týmu s průběhem jejich práci.

Hlavními důvody, které byly identifikované před zaváděním agilního řízení do týmů bylo:

- Nepravidelnost release funkcionalit a produktu

Tým měl značný problém s pravidelností releasu. Naplánované termíny se poměrně často posouvaly, a proto počet releasu ročně nepřesahoval 5.

- Komplikaci při práci nad rozsáhlými projekty

Kvůli geografické roztržitosti týmu vznikali komplikace při práci nad rozsáhlými projekty. V případě rozsáhlého projektu, který trvá rok a půl roku a více, docházelo k časovým prodlevám kvůli velkému počtu činnosti a jejich návaznosti a přirozenosti ke zdrojům. Výsledný produkt i přes to, že byl dodán, nesplňoval termín ukončení nebo nesplňoval požadovaný rozsah.

- Netransparentnost

Z důvodu, že členové týmu se nenachází ve společném pracovním prostředí docházelo k nepochopení činnosti svých kolegu. Při společné práci na jednom produktu nepochopení práci ostatních znamená duplicitu práci, nekompatibilitu v kódu a tím časové prodlevy a komplikaci při vývoje.

- Konflikty v rámci týmu

Členové týmu se v průběhu vývoje skoro nepotkávali, neměli pravidelnou interakci ohledně současného stavu, dalších kroku, představ o vývoje, stejně tak neměli pravidelnou komunikaci s projektovým manažerem. Proto každý člen týmu pracoval v podstatě na společném produktu ale zároveň samostatně. Po nějaké době začali vznikat konflikty uvnitř týmu, a to ovlivňovalo termíny dodání a kvalitu.

- Absence sjednoceného systému kooperace a koordinace

Kvůli absenci systému, který práci v týmu sjednotí a zpřehlední docházelo k nedostatečné koordinaci jednotlivých členu a procesu celkové, výsledky nebyli dostatečně zjevné a jednoduše kontrolovatelné.

4.3.3 Tým

Tým se skládá z 8 lidí, zastupujících jednotlivé role podle metodiky Scrum. Tým byl vytvořen čerstvě před zahájením agilní transformace a skládá se z lidí, které pracovali v jak mateřské, tak i dceřině společnosti a dříve neměli zkušenost spolupráce na více podobných produktech. Jedním z důležitých a zároveň omezujících faktorů je to, že se jedná o tým, kde členové se nachází v Praze, Brně a na Slovensku, a tak není možné, aby se osobně setkávali a komunikovali na denní bázi. Tým těsně spolupracuje s ostatními týmy napříč společností hlavně kvůli společnému vlastnictví některých kódů, který využívají i ve svých dalších produktech kvůli vzájemné integraci produktu apod.

Product Owner: Vlastník produktu, nahrazující roli projektového manažera v klasickém vodopádovém přístupu. Hlavním cílem Product Owenera je řízení průběhu vývoje produktu a komunikace se zákazníkem. V průběhu komunikace se provádí sběr požadavků, které se mohou tvořit i v průběhu vývoje produktu.

Scrum Master: Scrum Master je rolí moderátora a facilitátora v průběhu práci týmu nad produktem. Původní úlohou Scrum Mastera v týmu bylo zavedení agilního přístupu do práce týmu, průvod agilní transformaci, postupná implementace prvku metodiky Scrum. Je hlavní odpovědnou osobou v průběhu agilní transformaci. V současné době hlavní úlohou Scrum Mastera je facilitace agilních ceremonií a řešení problému, které v průběhu práce týmu nastávají. Kromě toho provádí zápis z retrospektivy, který slouží k analýze fungování týmu, provádí analýzu ukazatelů výkonnosti, motivuje tým k pozitivním změnám a k agilnímu myšlení. Jedním z důležitých faktorů je, že Scrum Master současně spolupracuje s několika týmy z důvodu nedostatečných kapacit. Za občasně nastávajících podmínek rolí Scrum Mastera plní Product Owner.

Vývojový tým: Tým se skládá z vývojářů a testerů, které vzájemně spolupracují na vývoje produktu. Před začátkem agilní transformaci neměli zkušenosti s agilním přístupem ve vývoji produktu.

4.3.4 Nástroje

4.3.4.1 Slack

Slack je softwarovým nástrojem na základě cloud technologií poskytující možnost komunikaci ve firemním prostředí (Obrázek 20).

- Kanály

V rámci produktu nabízí možnost vytvoření skupin nebo týmu, ve kterých se mohou sdílet informace pouze pro omezený počet účastníků konverzací, dostat se do skupiny může pouze člověk, který dostal pozvání od členu skupiny. Je tak alternativou pro skupiny při používání emailu.

- Osobní konverzace

Kromě možnosti využití společných konverzací Slack poskytuje možnost osobních konverzací s jednotlivými členy komunity.

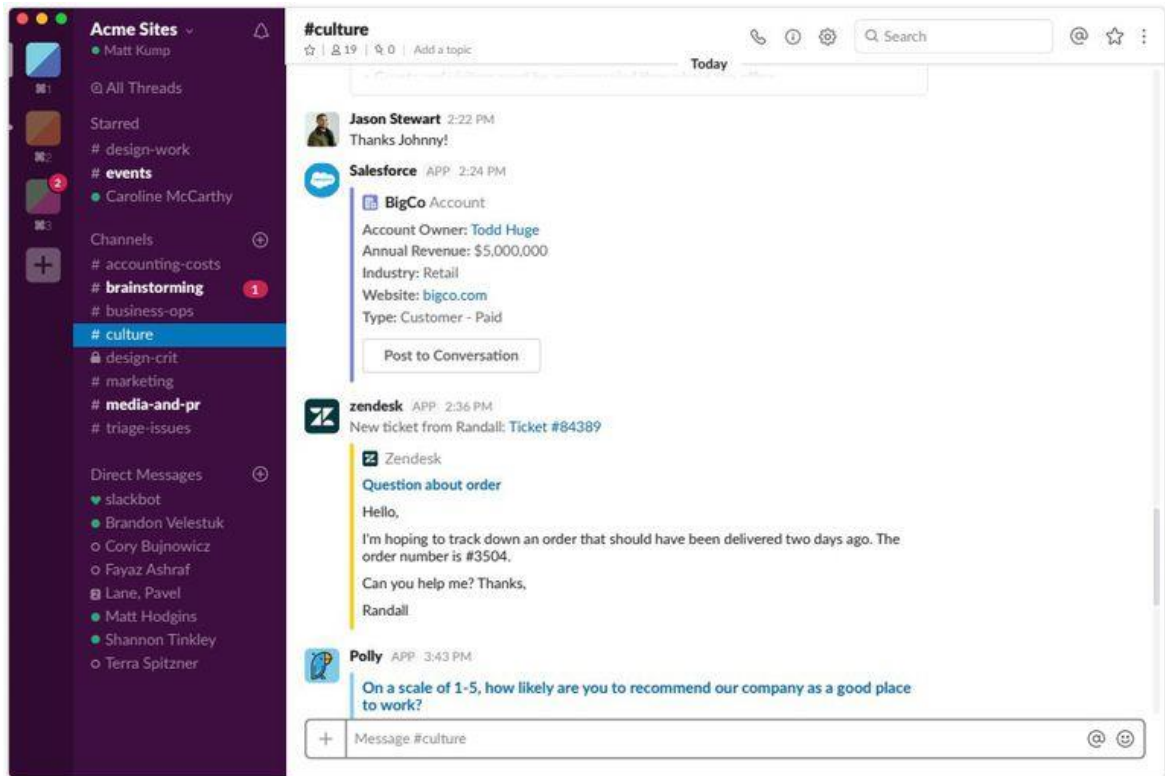
- Komunity

Komunity jsou možností, jak v rámci jedné aplikace rozlišit komunikační prostředí pro více firem. Tak je možné používat Slack jak pro komunikaci v rámci firmy, tak i pro komunikaci s partnerskými firmami.

- Integrace

Součástí aplikace je možnost integraci s několika dalšími aplikacemi jako například Google Kalendář apod.

Obrázek 20 - Slack aplikace



Zdroj: [66]

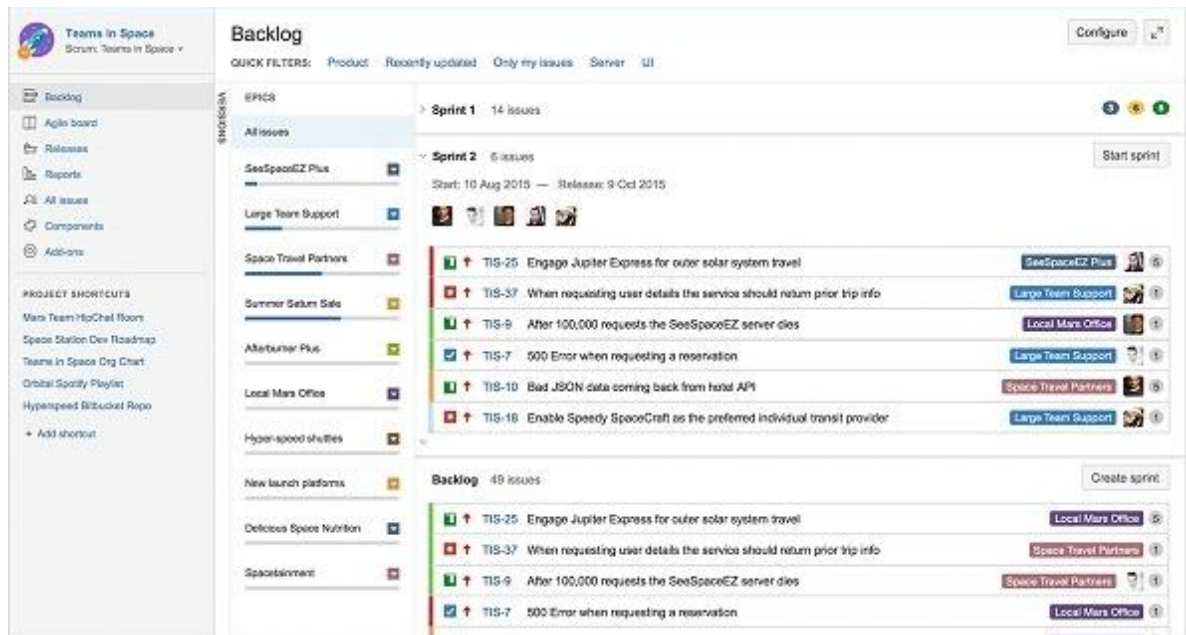
4.3.4.2 Jira

Jira je softwarový nástroj od společnosti Atlassian, který slouží k řízení projektu a sledování vývoje softwaru. Nástroj umožňuje také evidenci chyb a sledování operativních úkolů v rámci projektu. Hlavní výhodou tohoto nástroje je taky podpora agilních metodik řízení projektu. V rámci softwaru je možné plně přizpůsobit vzhled a funkcionality ke konkrétním potřebám týmu.

Jira umožňuje nastavení:

- Pracovního toku (Nastavení jednotlivých kroků u úkolu)
- Informací o úkolu (Produkt, vliv a finanční ukazatele, status rozpracovanosti, prioritita apod.)
- Informace o průběhu plnění (Zaznamenané kroky, komentáře apod.)
- Statistiku o provedených úkolech a souhrn různých pracovních prostředí.

Obrázek 21 - Jira příklad vzhledu



Zdroj: [67]

4.3.4.3 Zoom

Obrázek 22 - Logo Zoom



Zdroj: [68]

Zoom je nástroj na základě cloudové technologií umožňující video konference. Hlavním účelem je provádění video meetingu. Výhodou je možnost generování unikátního odkazu, který uživateli umožňuje se jednoduše připojit ke konkrétnímu meetingu a taky možnost připojení až 15 členu najednou.

4.3.5 Ceremonie a artefakty

4.3.5.1 Sprint

Tým pracuje ve omezených iteracích o délce 1 měsíců. Na konci sprintu se pokaždé provádí release vyvinutých funkcionalit. Před zavedením agilního přístupu release funkcionalit se prováděl pětkrát do roku. Minimální délka sprintu může být stanovena na jeden týden, však náklady na jeden release jsou poměrně vysoké (implementace, testování funkcionalit, testování kompatibility apod.), a proto není finančně výhodně to provádět v kratších časových intervalech. Délka sprintu větší, jak jeden měsíc je zároveň dostatečně velká a přehled změn a získávání zpětné vazby se komplikuje.

4.3.5.2 Daily Scrum

Daily Scrum neboli Stand-up se provádí v týmu každý pracovní den ve stejný čas. Všichni členové týmu by se měli zúčastňovat této ceremonii, však většiny Stand-upu se zúčastňuje Product Owner a několik členů vývojového týmu. Hlavním účelem Daily Scrumu je sdílení toho, na čem momentálně pracují a jestli vznikl nějaký problém, který existuje potřeba řešit. Pro propojení všech členů týmu se používají Zoom konference. V některých případech dochází k tomu, že velká část týmu se nemůže zúčastnit, v tom případě přítomné členové týmu využívají konverzaci ve Slacku, kde ve krátké formě popisují současný stav jejich práce. Doba trvání je přibližně 15 minut.

4.3.5.3 Grooming

Grooming produktového backlogu probíhá jednou za 2 týdny na konci týdnu. V procesu groomingu se probíhá analýza položek, které se přidali za předchozích 14 dnů do backlogu. U každého probíraného úkolu se provádí odhad náročnosti pomocí „tričkové“ metody, kde se nejjednodušší úkol označuje se XS a nejkomplicovanější jako XL. Tým nemá předepsaný rozsah toho, kam jednotlivé kategorie spadají a hodnotí úkol subjektivně, na základě zkušenosti s podobnými úkoly v rámci jak současného, tak i předchozích projektu. Hlavním cílem toho odhadu je budování podobného vnímání náročnosti úkolu. Grooming probíhá pomocí Zoom konference. Kromě toho se diskutuje, jakým směrem by se mohl produkt vyvíjet, které položky backlogu jsou podstatné a které v současném

kontextu dávají smysl a budou později zařazené do budoucích sprintu. Sprint Grooming se koná na začátku a v polovině měsíce, tak na začátku měsíce je ceremonie používaná pro analýzu položek a plánování sprintu. V polovině měsíce představuje spíše odhad úspěšnosti průběhu sprintu a vyjasnění komplikací a případných nejasností, které pomohou k úspěšnému dokončení sprintu.

Plánování sprintu se jako samostatná ceremonie se v rámci týmu nepoužívá, a proto Sprint Grooming je způsobem, jak položky backlogu zanalyzovat a zároveň naplánovat položky, které mohou být zpracované v průběhu dalších týdnů. Groomingu se primárně zúčastňuje Product Owner a vývojový tým. Doba trvání je přibližně 1 hodina.

4.3.5.4 Retrospektiva

Retrospektiva probíhá každých 14 dní den před Sprint groomingem a probíhá formou Zoom konverzace. Cílem je sdílení názoru na předchozí sprint. Všichni sdílí své návrhy na zlepšení do dalšího sprintu a návrhy na řešení identifikovaných problémů. V rámci retrospektivy se kromě toho analyzují zpětné vazby, které tým dostává od beta uživatelů, které mají možnost průběžně testovat produkt, ke kterému se postupně přidávají funkcionality. Retrospektivy se zúčastňují Produkt Owner, Scrum Master a vývojový tým. Z celé retrospektivy se vede zápis, aby se po časovém odstupu se dalo vrátit k problémům, identifikovaným dříve a ohodnotit, zda některé z těchto problémů zůstali nevyřešené. Doba trvání je přibližně 1 hodina.

4.3.5.5 Sprint review

Sprint review neboli „Weekly update“ se provádí na začátku každého týdne. V rámci této ceremonie se sdílí aktuální informace o produktu, kde se každý člen týmu vyjádří k tomu, jaký je současný stav produktu, sdílí se názoru na proces vývoje. Sprint review probíhá formou Zoom konferenci. Přibližná doba trvání je 45 minut. Přestože délka sprintu je 1 měsíc, sprint review se provádí jednou týdně proto, že se aktualizace stavu produktu může probíhat častěji než doba trvání sprintu. V průběhu ceremonie se dá odhadnout, jaký průběh má současný sprint, a zda vývoj produktu odpovídá představám.

4.3.5.6 Vizualizace

Při sledování agilních principů, zastoupení rolí a provádění Scrum ceremonií, tým používá vizualizaci, která patří do metodiky Kanban. Z důvodu roztržitého charakteru týmu vizualizace je představená formou digitální Kanban tabule ve softwaru Jira.

Z důvodu možnosti personálního nastavení byli zvoleny tyto sloupce:

- „To do“

Všechny úkoly, které se nachází v backlogu a budou postupně odbavované v průběhu vývoje.

- „In progres“

Úkoly, na kterých již někdo z týmu pracuje.

- „Quality Assurance review awaiting“

Úkol je dokončen a očekává se začátek testování v úkolu dokončených funkcionalit.

- „Quality Assurance review“

Úkol se již nachází v procesu testování.

- „Done“

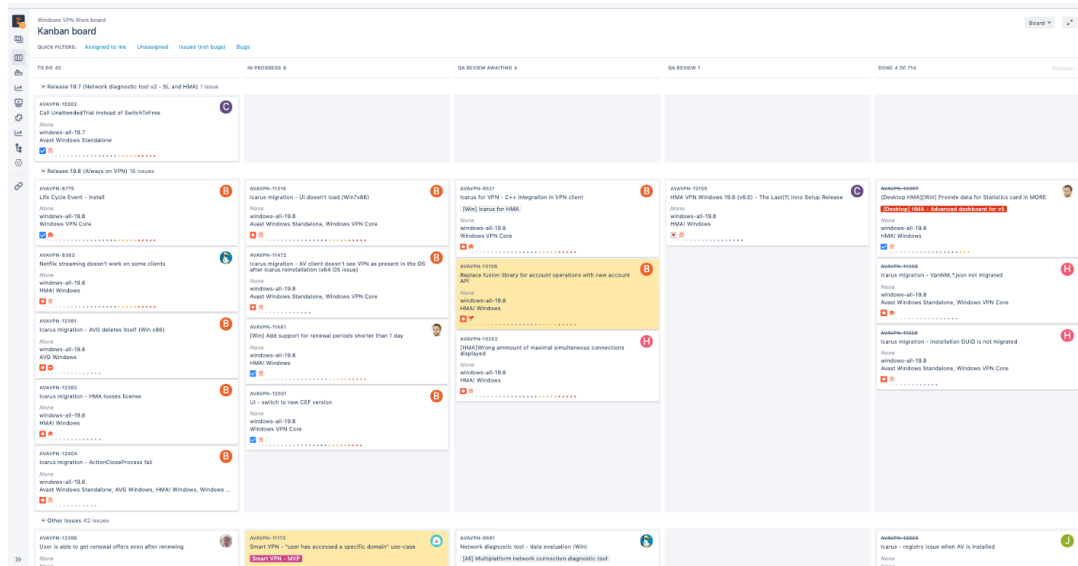
Úkol je otestován a je možné se přesunout k dalším úkolům.

V rámci Kanban tabule se používá rozdělovací linie, která odděluje jednotlivé releasy. Pod posledním rozdělením se nachází úkoly, které nebyli do jednotlivých releasu přiřazené.

Každý člen týmu má vlastní účet, a tak je schopen úkoly z tabule přerazovat k sobě, a po přiřazení každý člen týmu je schopen identifikovat, kdo má na starosti konkrétní úkol.

Počet rozpracovaných (Work in Progres) v současné době není přesně stanoven, tak každý člen týmu se snaží o redukci rozpracovaných úkolů sám bez použití konkrétního omezujícího množství.

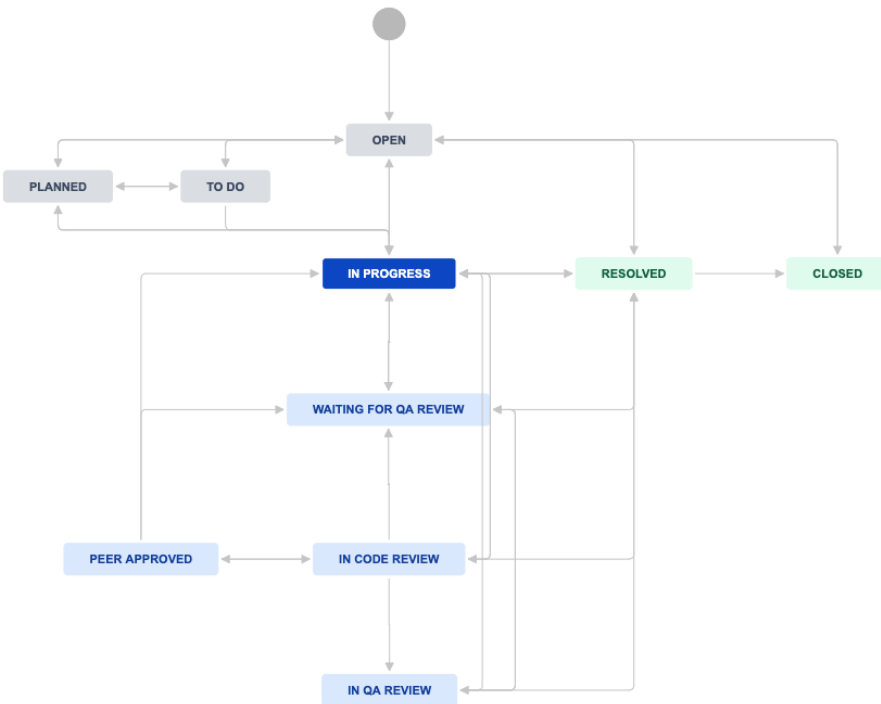
Obrázek 23 - Kanban tabule



Zdroj: Avast Software

Pro sledování předchozích a současného stavu úkolu a také informaci o budoucích krocích v rámci projektu je možné použít personalizovaný pracovní tok. Na obrázku (Obrázek 24) můžeme vidět jednotlivé kroky, kterými prochází každý úkol v rámci týmu.

Obrázek 24 - Pracovní tok



Zdroj: Avast Software

4.3.5.7 Roadmapa

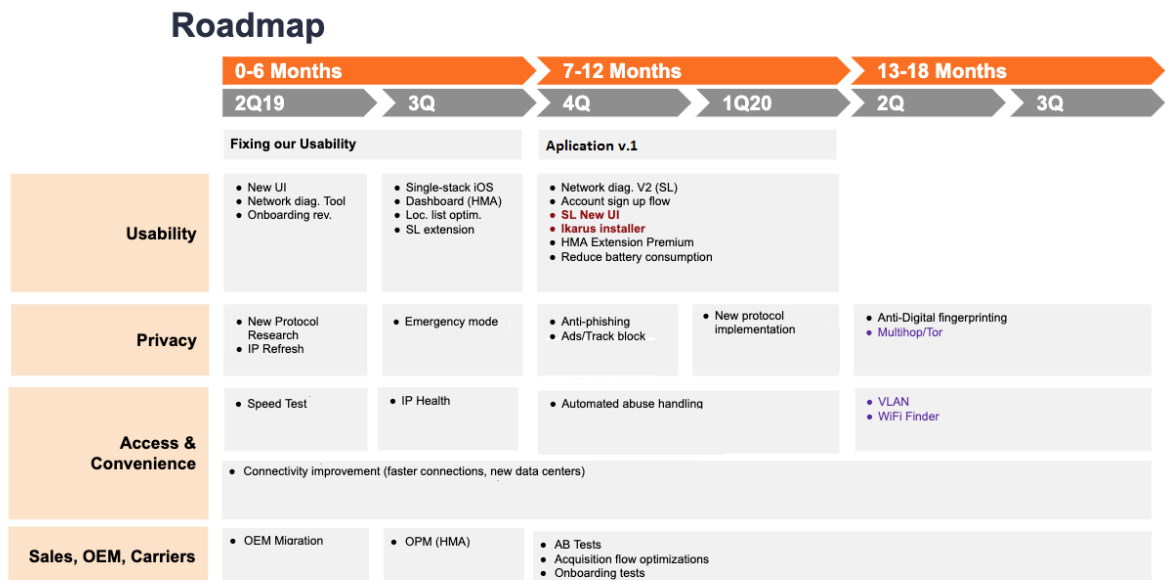
Tým ve své práci používá produktovou roadmapu, která se používá pro vizualizaci produktové vizi, budoucí směřování produktu a odhad doby implementace určitých zásadních funkcionalit.

Časové úseky, které jsou používané pro plánování, představují čtvrtletí (Obrázek 23 nahoře). Každé čtvrtletí má určené hlavní úkoly, u kterých se předpokládá dokončení. Jednotlivé úkoly jsou rozdělené do specifických kategorií, rozlišující oblast, do které funkcionalita by měla být implementována (Obrázek 23 vlevo):

- Použitelnost
- Ochrana soukromí
- Přístup a pohodlí
- Prodej a doručování.

Roadmapa neslouží jako přesný a definitivní plán a je možné jí v průběhu vývoje upravovat podle měnících se požadavků na výsledný produkt.

Obrázek 25 - Roadmapa



Zdroj: Avast Software

4.3.6 Ukazatele a jejich měření

V rámci sledování práce a pracovního výkonu se v týmu používají data, která jsou sbírána prostřednictvím Jira analytiky.

Hlavním ukazatelem je Průměrná doba zpoždění (Average Lead time) – průměrná doba trvání dokončení jednoho úkolu, která je vypočítaná jako celková doba dokončení všech úkolů vydělená počtem dokončených úkolů (Tabulka 1).

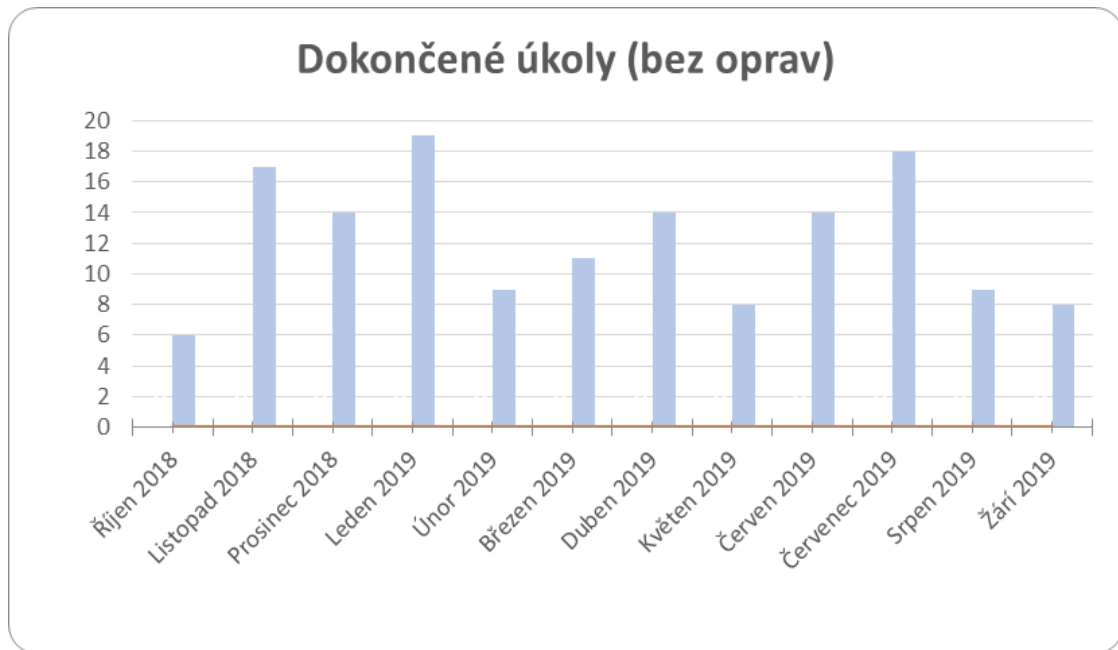
Tabulka 1 - Dokončené úkoly

Období	Dokončené úkoly	Průměrná doba zpoždění
Říjen 2018	6	129
Listopad 2018	17	60
Prosinec 2018	14	23
Leden 2019	19	214
Únor 2019	9	105
Březen 2019	11	38
Duben 2019	14	92
Květen 2019	8	46
Červen 2019	14	71
Červenec 2019	18	34
Srpen 2019	9	27
Žáří 2019	8	22
Celkový průměr	12	72

Zdroj: Vlastní zpracování

Následně, pro větší přehlednost se jednotlivé sloupce vizualizují do sloupcového grafu, kde graf „Dokončené úlohy“ (Obrázek 26) představuje průběh ukončení úkolu v intervalech jednoho měsíce.

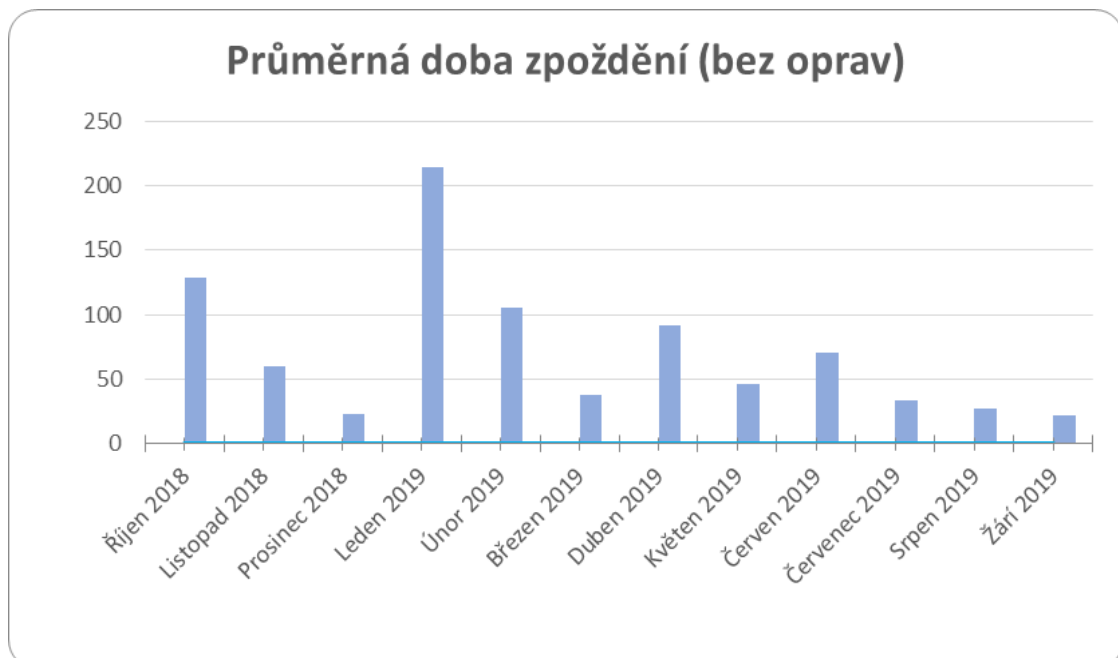
Obrázek 26 - Dokončené úkoly



Zdroj: Vlastní zpracování

Průměrná doba zpoždění (Average Lead time) znázorňuje průběh doby trvání práci nad jedním dokončeným úkolem, sledovaný v intervalech jednoho měsíce (Obrázek 27).

Obrázek 27 - Průměrná doba zpoždění



Zdroj: Vlastní zpracování

Dalším ukazatelem, který se pravidelně měří na základě dat z Jíry je Průměrná doba čekání (Average Age), která vyjadřuje, jak dlouho v průměru zůstává jeden úkol nevyřešen, a je měřena v úsecích jednoho měsíce (Tabulka 2).

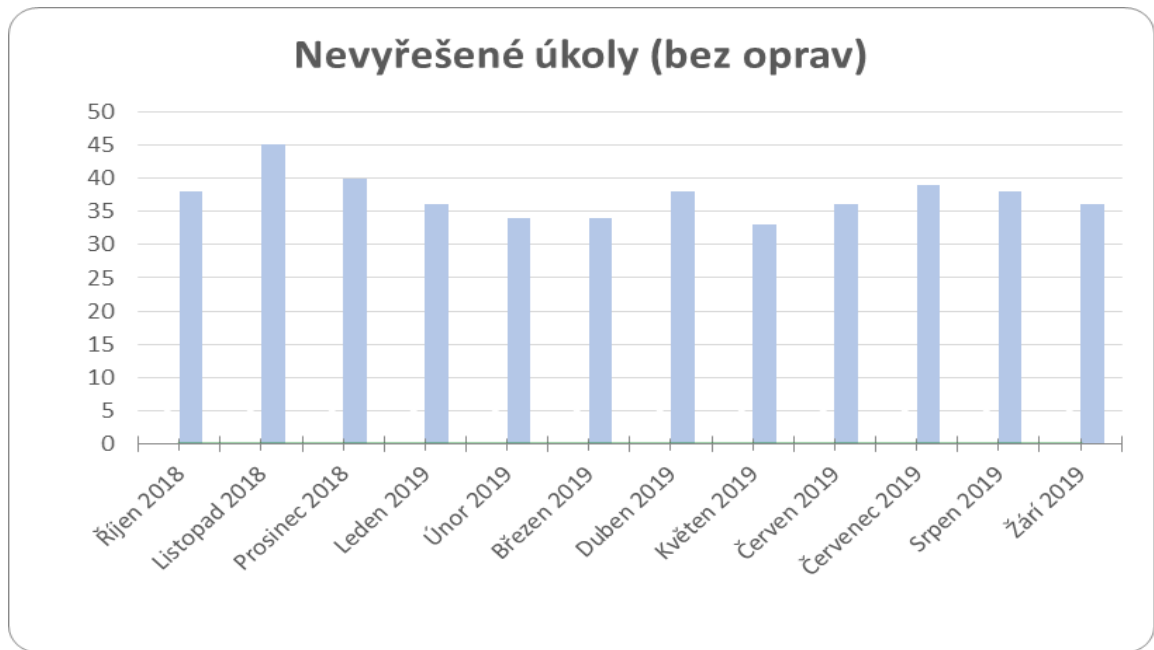
Tabulka 2 - Nevyřešené úkoly

Období	Nevyřešené úkoly	Průměrná doba čekání
Říjen 2018	38	139
Listopad 2018	45	121
Prosinec 2018	40	130
Leden 2019	36	157
Únor 2019	34	62
Březen 2019	34	55
Duben 2019	38	58
Květen 2019	33	50
Červen 2019	36	53
Červenec 2019	39	39
Srpen 2019	38	43
Žáří 2019	36	62
Celkový průměr	37	81

Zdroj: Vlastní zpracování

Graf „Nevyřešené úkoly“ (Obrázek 28) znázorňuje množství nedokončených úkolů, které byli evidované za popisované období.

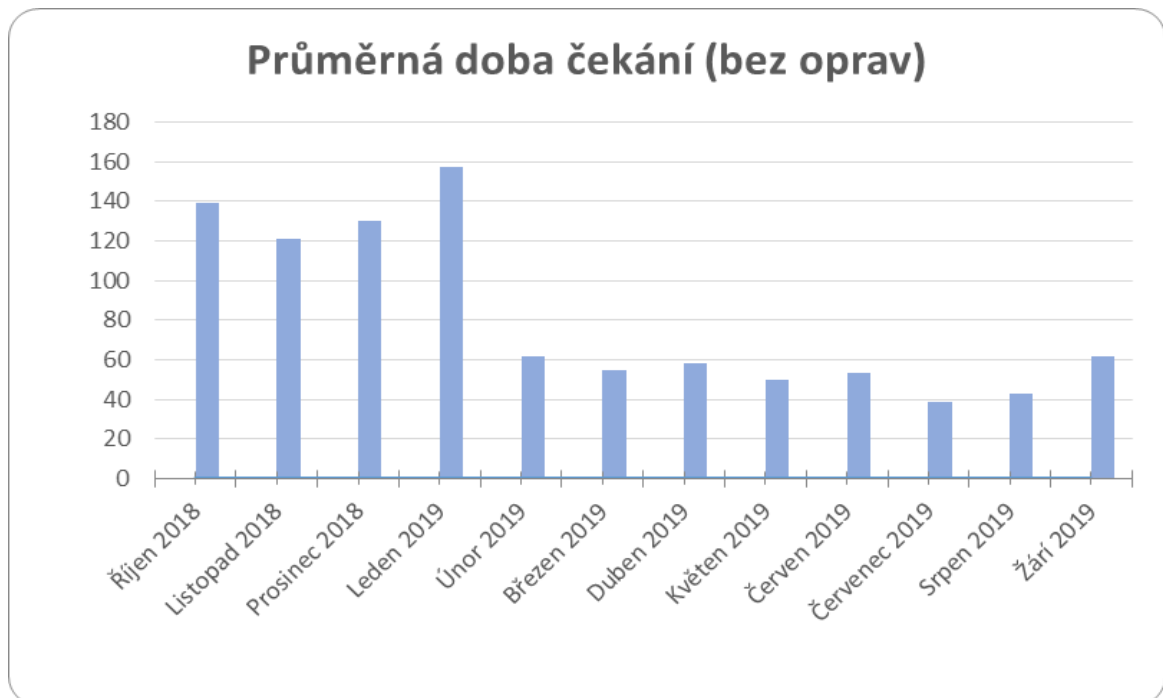
Obrázek 28 - Nevyřešené úkoly



Zdroj: Vlastní zpracování

Průměrná doba čekání (Average Lead time) znázorňuje, kolik v průměru dní zůstává existující úkol nevyřešený (Obrázek 29).

Obrázek 29 - Průměrná doba čekání



Zdroj: Vlastní zpracování

4.3.7 Důvodu zvolení metodiky

Volba metodiky v rámci společnosti není přesně určena, a tak každý tým má možnost si konkrétní metodiku zvolit na základě potřeb. Důležitým důvodem pro volbu metodiky Scrum ve zvoleném týmu je absence zkušenosti s jakoukoliv agilní metodikou. Cílem, který byl stanoven bylo zavedení metodiky Scrum, která odpovídá požadavkům na velikost týmu a rozsah projektech, na kterých tým pracuje. Je to komplexní metodika, která je vhodná pro softwarový vývoj a díky své rozšířenosti existuje spousta možností pro vzdělávání lidí v týmu a okolo něj.

Zásadním problémem při zavádění metodiky je její náročnost a požadavky na tým. Scrum je dobrou volbou pro týmy, které v nějaké podobě se již setkali s agilním přístupem nebo s méně náročnou metodikou. Kanban je zároveň jednou z metodik, které nevyžadují velké změny a je vhodné pro nezkušený tým. Jsou to hlavní důvody pro zahájení agilní transformaci, která má vést k plnému přechodu na Scrum metodiku, pomocí metodiky Kanban.

Jako první krok se provedlo zavedení vizualizačního prvku – Kanban tabule. Jejich hlavním účelem bylo zpřehlednění úkolu, které v rámci týmu jsou, odpovědnost za jejich dokončení a taky jasné informace o tom, v jakém stavu se současně nachází a kdy je možné očekávat dodávky funkcionalit. Tento krok značně přispěl k řešení problému transparentnosti v týmu a zároveň nevyžadoval velké týmové změny. Po nějaké době, co tým pracoval s vizualizací se začala zavádět metodika Scrum. Byly definované role, a zároveň se začali provádět Stand-upy. Po nějaké době se začali přidávat i další ceremonie jako Grooming, Retrospektiva, Sprint review. V současné době se tým stále nachází v procesu transformace, předpokládaná doba, které je 2 roky. Důležité pro přechod od klasického přístupu je postupná implementace a pomalé stupňování náročnosti požadavku na metodiku. Změnu vyžaduje i celkové myšlení členů týmu, které by se měli ztotožnit s agilními hodnotami a přístupem k práci. Právě proto postupný přechod a jeho součást – metodika Scrum – byli vybrané jako nejlepší možnost v rámci agilní transformaci.

Postupný přechod je způsobem, jak provádět změny, které mají vyřešit existující problémy ale způsobem, který nebude jít v rozporu se zvyky fungování týmu a předejde odporu, který může projevit tým, který najednou stojí před nutností změnit celé zažité fungování v procesu vývoje softwaru.

4.4 Analýza agility podle agilního manifestu

Při pozorování týmu po dobu dvou a půl měsíců a rozhovorech s členy týmu bylo hodnoceno, zda tým svojí práci a postojem k ní naplňuje agilní principy, vyjádřené v agilním manifestu.

1. *„Naší nejvyšší prioritou je vyhovět zákazníkovi časným a průběžným dodáváním hodnotného softwaru.“ [2]*

Tým doručuje pravidelně v intervalu jednoho sprintu. Každá vyvinutá funkcionalita je otestována nejenom členy QA (zajištění jakosti) ale i beta uživateli, které jsou zastoupené reálnými uživateli, které nejsou zaměstnanci firmy a mají možnost vyjádřit svůj názor na funkcionality produktu a spokojenost s jeho užíváním dříve, než to bude doručeno všech uživatelům.

2. *„Vítáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.“ [2]*

Požadavky a návrhy na zlepšení v průběhu vývoje jsou akceptované v každé vývojové etapě, protože jsou vnímané jako možnost doručit produkt, který více naplní představy zákazníka.

3. *„Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.“ [2]*

Dodávání funkcionalit softwaru probíhá v intervalu jednoho měsíce. Zvolená perioda není nejkratší možnou délkou intervalu z důvodu finanční náročnosti a zároveň technické komplexnosti projektu.

4. *„Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.“ [2]*
Product Owner se aktivně zapojuje do práci celého týmu a zároveň se zúčastňuje všech probíhajících ceremonií. Tým přímo nekomunikuje se zákazníkem nebo s uživateli jejich produktu, však dostává informace od Product Ownera, který je ve spojení s byznysovou částí, a tak je schopen té informace předávat.

5. *„Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.“ [2]*

V rámci týmu existuje atmosféra, podporující důvěru v práci jednotlivých členů týmu. Každý člen týmu projevuje zájem o probíhající projekt a také o agilní prostředí, budované kolem toho projektu a týmu.

6. *„Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.“ [2]*

Osobní konverzace probíhá mezi členy týmu, které se v pracovní době se nachází ve stejné kanceláře. Jednou za delší dobu, přibližně jednou za tři měsíce, někdo z členů týmu pracovně dojíždí do centrály v Praze, a tak má možnost si některé pracovní příležitosti probrat osobně. Členové týmu ale osobně nesetkávají se zákazníkem.

7. *„Hlavním měřítkem pokroku je fungující software.“ [2]*

V rámci týmu jsou zavedeny měřítka efektivity práce, vyjádřené dodací lhůtou. Hlavním cílem ale zůstává reálný produkt, který uspokojuje potřeby zákazníka.

8. *„Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.“ [2]*

Intenzita práce se vlivem okolností často mění, a proto dochází k vychýlení v pracovním tempu týmu. Ale průměrné tempo práce týmu se dá hodnotit jako trvale udržitelné.

9. *„Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.“ [2]*

Technická výjimečnost a vysoká kvalita provedení je jedním z nejdůležitějších bodů, na kterých si tým v průběhu práce zakládá. Technickou výjimečnost zajišťuje odborná způsobilost členů týmu a dostatečný čas věnovaný vývoji.

10. „Jednoduchost--umění maximalizovat množství nevykonané práce--je klíčová.“ [2]

Z důvodu velikosti týmu vzhledem k objemu práce se v průběhu vývoje členové týmu primárně zaměřují na hlavní úkoly a základní požadované funkcionality. Funkcionalitám, které se vymykají definovaným požadavkům, se tak z technických a časových důvodu nevěnuje moc času.

11. „Nejlepší architektury, požadavky a návrhy vzejdou ze samoorganizujících se týmů.“ [2]

Členové týmu mají možnost si rozvrhovat práce podle vlastního uvážení a jsou ochotní nést odpovědnost za jejich provedení. Projevují zájem o agilní ceremonie a jejich účast i přes to že není absolutní, je dostatečně vysoká.

12. „Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování a zvyklosti.“ [2]

V rámci retrospektivy se probíhá analýza současného stavu týmu a vzházejí návrhy od členu týmu, zaměřené na zlepšení současných procesu a výsledku jak jednotlivých iterací, tak i celkových výsledku.

4.5 Analýza podle metodiky Scrum a Kanban

4.5.1 Hodnoty Scrum

Cílem agilní transformaci v týmu je zavedení metodiky Scrum. Proto bude provedená analýza, která ohodnotí, zda tým splňuje předpoklady, které metodika má.

Do metodiky Scrum patří základní sada hodnot, která by měla být dodržovaná členy týmu:

1. Oddanost

Oddanost týmu se má projevovat jako dostatečné zapojení členu týmu do vytváření agilní kultury. Ve sledovaném týmu existuje atmosféra, podporující agilní myšlení. Členové týmy se ochotně zapojují do ceremonií, a v případě, že něco brání jejich normálnímu průběhu, využívají alternativní možnosti naplňování jejich účelu.

2. Odvaha

V průběhu projektu tým se setkává s novými a netradičními úkoly, se kterými se nemohli setkat dříve ale i přes to dokážou s tím pracovat a překonávat této bariéry. Kromě toho

pomocí pravidelných Stand-upu členové sledovaného týmu sdílí, zda potřebují s nějakým úkolem pomoci a jsou ochotní o tom diskutovat.

3. Soustředěnost

Tým na začátku sprintu určuje priority a cíle budoucího sprintu a v rámci některých groomingů se určují úkoly, na kterých budou po dobu sprintu pracovat. V průběhu sprintu tým se zaměřuje na dosažení postavených cílů a na doručení předpokládaných funkcionalit.

4. Otevřenost

Pravidelní ceremonie poskytují prostor, pro sdílení aktuálních informací o úkolech, na kterých pracují a o stavu jejich rozpracovanosti. V rámci nich se provádí diskuze o tom, v jakém stavu se současné nacházejí aktuální úkoly. K větší transparentnosti týmu přispívá použití prvky vizualizací, díky kterým je možné sledovat průběh práci v týmu i bez účasti týmových ceremonií.

5. Respekt

Podstatou práci v týmu, zvláště pokud se jedná o roztržštěný tým, je dostatečná důvěra v to, že tým je schopen rozvrhovat a odvádět kvalitní práci, a zároveň respekt kvůli ostatním členům týmu a jejich práci. Ve sledovaném týmu existuje důvěra v samostatnost a kvalitu práci.

4.5.2 Hodnoty Kanban

Tým ve své práci používá metodiku Scrumban, a proto v její práci existují i prvky metodiky Kanban.

Hlavními hodnotami této metodiky jsou:

1. Vizualizace

Tým široce využívá a podílí se na vytváření vizualizačních prostředků, jako Kanban tabule, které se mohou rozpracovávat do separátních vizuálních prvků.

2. Omezení rozpracovaných úkolů

I přes to, že tým se pokouší o minimalizaci rozpracovaných úkolů, nemá v Kanban tabuli přesně definované hranice a počet úkolů se tak může v závislosti na okolnostech měnit.

3. Řízení dodací lhůty

Tým má na konci každého měsíce pravidelně nastavené releasy. Přibližná doba dodání funkcionalit je závislá na jejich rozvrhování do jednotlivých iterací, a dodací cyklus není stanoven na minimální dolní hranice.

4. Transparentnost

V rámci své práce tým se pokouší o nejvyšší možnou transparentnost prostřednictvím Kanban tabule, stránek na intranetu, kde je sdílen aktuální stav projektu, cíle a prostředky, prototyp, umožňující lepší představu o budoucím produktu apod.

5. Zpětná vazba

Tým získává většinu zpětných vazeb na produkt od Product Ownera, který je odpovědný za komunikaci mezi týmem a zájmovými stranami. Členové týmu nejsou aktivně zapojené do komunikaci se zadavatelem.

6. Kolektivní implementace zlepšení

V týmu se neprovádí kontinuální změny, ale v případě, že od někoho vzejde požadavek na zlepšení nebo implementaci nových nástrojů, měli by být odsouhlasené všemi členy týmu a poté mohou být zavedené. Prostředí v týmu umožňuje flexibilitu.

4.6 Srovnání a ohodnocení agilního přístupu a metod

Tým ve své práci používá agilní zásady a kombinuje metody pro dosažení maximálně flexibilního a efektivního průběhu doručení produktu. Analýza dodržení agilních přístupů a agilních metod byla převedena do hodnoticí tabulky, která je vyjádřena hodnotami ano/ne, kde „ano“ znamená že tým podmínky splňuje, a „ne“ znamená, že podmínky buď nesplňuje nebo splnění podmínek není na dostatečné úrovni (Tabulka 3).

Tabulka 3 - Hodnocení agilních hodnot

Agilní přístup	Dodržení
Naší nejvyšší prioritou je vyhovět zákazníkovi časným a průběžným dodávkám hodnotného softwaru	Ano
Vítáme změny v požadavcích, a to i v pozdějších fázích vývoje. Agilní procesy podporují změny vedoucí ke zvýšení konkurenceschopnosti zákazníka.	Ano
Dodáváme fungující software v intervalech týdnů až měsíců, s preferencí kratší periody.	Ano
Lidé z byznysu a vývoje musí spolupracovat denně po celou dobu projektu.	Ne
Budujeme projekty kolem motivovaných jednotlivců. Vytváříme jim prostředí, podporujeme jejich potřeby a důvěřujeme, že odvedou dobrou práci.	Ano
Nejúčinnějším a nejefektivnějším způsobem sdělování informací vývojovému týmu z vnějšku i uvnitř něj je osobní konverzace.	Ne
Hlavním měřítkem pokroku je fungující software.	Ano
Agilní procesy podporují udržitelný rozvoj. Sponzoři, vývojáři i uživatelé by měli být schopni udržet stálé tempo trvale.	Ano
Agilitu zvyšuje neustálá pozornost věnovaná technické výjimečnosti a dobrému designu.	Ano
Jednoduchost--umění maximalizovat množství nevykonané práce--je klíčová.	Ano
Nejlepší architektury, požadavky a návrhy vzejdou ze samoorganizujících se týmů.	Ano
Tým se pravidelně zamýšlí nad tím, jak se stát efektivnějším, a následně koriguje a přizpůsobuje své chování a zvyklosti.	Ano

Zdroj: Vlastní zpracování

Ze dvanácti principů, popsaných v agilním manifestu, tým nesplňuje pouze dvě podmínky. Těmi podmínky jsou denní spolupráce lidí z byznysu a týmu a taky efektivní osobní komunikace. Spolupráce je ztížena časovou vytížeností týmu a jeho geografickou roztržitostí, kde neustálý kontakt je nahrazen komunikací s Product Ownerem, který komunikuje se všemi zainteresovanými stranami. Podobný důvod má i komunikační problém, kde členové týmu nedisponují možností osobně sdílet informaci kvůli vysokým nákladům, jak časovým, tak i finančním. Proto většina potřebné komunikace probíhá elektronickou formou.

Tabulka 4 - Hodnocení Scrum metodiky

Podmínky	Dodržení
Oddanost	Ano
Odvaha	Ano
Soustředěnost	Ano
Otevřenost	Ano
Respekt	Ano
Scrum Master	Ano
Product Owner	Ano
Vývojový tým	Ano
Sprint	Ano
Prioritizace	Ano
Sprint planning	Ne
Daily Scrum	Ano
Backlog Grooming	Ano
Product Backlog	Ano
Sprint Backlog	Ne
Sprint Review	Ne
Retrospektiva	Ano
Scrum tabule	Ne
Roadmapa	Ano
Velocita	Ne

Zdroj: Vlastní zpracování

Hlavními podmínkami, které tým v rámci metodiky Scrum nesplňuje jsou Sprint planning, který tým nemá jako samostatnou ceremonii a částečně ji nahrazuje součástí Backlog Groomingu. V průběhu Sprint review se provádí analýza přidané hodnoty, však není hodnocená bezprostředně zákazníkem, ale přínos se hodnotí v rámci týmu a příležitostně v přítomnosti jiných týmů. Sprint Backlog je nahrazen pouze produktovým backlogem,

který obsahuje všechny úkoly v projektu. Tým nepoužívá Scrum tabuli, která je plně nahrazená Kanban tabulí a k měření efektivity se nepoužívá Velocity ale doba dodání (Lead time), která se pravidelně měří a všichni členové týmu mají k tomu přístup a mohou pozorovat probíhající změny. (Tabulka 5).

Tabulka 5 - Hodnocení Kanban metodiky

Podmínky	Dodržení
Vizualizace práce	Ano
Omezení rozpracovaných úkolů	Ne
Řízení dodací lhůty	Ne
Transparentnost	Ano
Získávání zpětné vazby	Ano
Kolektivní implementace zlepšení	Ne
Správce požadavku na službu	Ne
Správce dodávek služeb	Ne
Lead time	Ano
Kanban tabule	Ano

Zdroj: Vlastní zpracování

V rámci hodnocení metodiky Kanban, se dá říct, že tým nedodrží role, které jsou v metodice předepsané, jsou plně nahrazené rolami z metodiky Scrum. Hodnoty týmu se plně neshodují s hodnotami metodiky Kanban, některé z nich, které tým sdílí jsou podobné hodnotám metodiky Scrum, včetně vizualizaci, která je v rámci metodiky Scrum v podobě Scrum tabulí. Proto se dá říct, že tým podporuje z metodiky Kanban hlavně způsob měření efektivity a vizualizační prvky. (Tabulka 5).

Tým sdílí a podporuje 10 z 12 agilních hodnot, stanovených agilním manifestem, primárně z důvodu existujících geografických a finančních omezení.

Základní metodikou, ze které tým čerpá je Scrum, poněvadž tým má dodržené hodnoty, role a část existujících ceremonií, které existují buď ve správné podobě anebo v modifikované. Tým to rozšiřuje o vizualizační prvky a také o způsob měření efektivity z metodiky Kanban, a proto ve své práci používá spíše kombinovanou metodiku „Scrumban“, než konkrétní existující metodiku.

5 Identifikované problémy a návrh řešení

V průběhu sledování práci týmu, rozhovoru s členy týmu a také analýze agilního přístupu v práci týmu bylo objeveno několik věcí, které mají negativní dopad na fungování v týmu.

Identifikovanými problémy jsou:

- Komunikace s okolím týmu

Hlavním problémem, který byl objeven jak po sledování týmu, tak po analýze agilního přístupu je problém s komunikací. Komunikace v týmu je pravidelná a efektivní, ale komunikace navenek je náročná. Hlavním problémem je spolupráce s jinými týmy, se kterými v průběhu projektu setkávají a sdílí některé společné části kódu. Kvůli komunikačním šumům dochází k problémům při práci nad společně udržovanými úseky produkčního kódu.

- Nekonzistentní sdílený kód

Právě na komunikační problém navazuje další identifikovaný problém – nekonzistence společného kódu. Problémem je vzájemné rozbíjení fungujícího kódu produktu, následně trvá delší dobu vrátit oba produkty do konzistentního stavu. V případě, že k podobným jevům dochází pravidelně práci týmu to komplikuje a vyžaduje velké časové ztráty, které zpomalují doručení funkcionalit produktu.

- Nedostatečná alokace Scrum Mastera

Dalším bodem, který zpomaluje adaptaci agilního přístupu je malá alokace Scrum Mastera, který z kapacitních důvodů se musí věnovat více týmům najednou. Product Owner tak musí při absenci Scrum Mastera ho částečně nahrazovat, a to snižuje efektivitu a přínos jednotlivých rolí.

Změnami, které mohou pozitivně ovlivnit současný stav jsou:

- Společný komunikační kanál

Jedním z možných způsobů zvýšení informovanosti mezi týmy je vytvoření společného Slack kanálu, kde se budou sdílet důležité aktualizace ohledně produktu obou týmů. Pro každý z týmů to může být přínosné z důvodu nízkých časových požadavků a možnosti monitorovat průběžně stav produktu a také zpětně dohledat příčiny případného nesouladu.

Alternativou možností je zavádění společného Stand-upu, ale vyžaduje to větší časovou flexibilitu a zároveň podporu více účastníků stran.

- Implementace Continuous integration

V případě nekonzistentnosti kódu akceptovatelným řešením pro tým by mohlo být převzatí zásady agilní metodiky extrémního programování – continuous integration. V rámci té zásady implementace nového kódu do společných částí kódu by měli být prováděné v minimálně možném časovém intervalu. Hlavní výhodou této zásady je možnost maximálně rychle najít místo, které celý problém způsobuje. Při větších časových intervalech náročnost nalezení chyby roste a s ní i čas k tomu potřebný. Důležité je, aby stejnou zásadu dodržovali oba týmy, protože změny pocházejí zároveň od obou týmů.

- Scrum Master

Role Scrum Mastera by neměla být zastupována někým jiným z týmu z důvodu snížení objektivnosti pohledu na situaci v týmu a rozptýlení od hlavní rolí, kterou člen zastupuje v týmu. Vhodným doporučením je obsazení nového Scrum Mastera, který je schopen se věnovat týmu po celou dobu jejich práce a pomáhat jim v průběhu ceremonií a v dalších relevantních oblastech. Alternativním řešením je dočasné zrušení rolí Scrum Mastera. Tím se uvolní prostor pro práci Product Ownera, který by nemusel zastupovat obě role po dobu hledání vhodného kandidáta a měl by více času pro naplnění své role. Tým je vysoce namotivovaný a sdílí agilní hodnoty, proto je možné, aby dočasně průběh práce zvládali samostatně.

6 Závěr

Tato diplomová práce se zabývala agilním přístupem v řízení projektu softwarové firmy. Hlavním cílem bylo provedení analýzy používání agilního přístupu a použití metodik Scrum a Kanban v jednom týmu, takzvaný „Scrumban“, při řízení projektu zaměřeného na vývoj softwaru. Sekundárním cílem bylo vytvoření návrhu na změny v týmu.

Při analýze byly použity hodnoty agilního manifestu, metodiky Scrum a Kanban a práce týmu byla ohodnocena podle naplnění jejích hlavních zásad a dodržování rolí a procesů jednotlivých metodik. Byly identifikovány 3 základní problémy, kterým tým v průběhu své práce momentálně čelí. Identifikované problémy jsou důsledkem existujících omezení z geografických a finančních důvodů. Ke každému problému bylo navržené vhodné opatření, které může vést jak ke snížení negativního dopadu na práci v týmu, tak i jeho kompletní eliminaci. Hlavním pozitivem navržených doporučení je jejich jasnost a nenáročnost na prostředky pro jejich zavádění. Nalezené řešení je dostatečně jednoduché a je možné ho reálně implementovat bez velkých investic nebo kardinálních změn.

Zároveň, bylo vyhodnoceno, že tým následuje agilní hodnoty a dá se považovat za agilní. Metodikou práce je „Scrumban“, který kombinuje jednotlivé prvky metodiky Scrum a Kanban. Zvolená metodika je vhodná pro tým, který před začátkem agilních transformací neměl žádné zkušenosti a potřebuje postupný přechod k plně agilnímu fungování a chce zachovat plynulé tempo a angažovanost zaměstnanců. Tým je dostatečně flexibilní, a proto za účelem efektivního výkonu kombinuje možné agilní praktiky tak, aby dodržoval agilní hodnoty a zároveň měl dostatečný prostor pro svoje potřeby a necítil se v nějaké oblasti omezený.

Tým se současně stále nachází v procesu agilní transformace a přístup „Scrumban“ je možností, jak ji dokončit bez tvrdých omezení a ve vhodném tempu pro členy týmu. Podobný postup pomáhá být dostatečně agilními a zároveň neomezuje tým v jeho fungování a udržuje spokojenost a efektivitu členů týmu. Při dodržování aktuálního tempa a agilní angažovanosti existuje vysoká pravděpodobnost úspěšného dokončení transformace.

7 Seznam použitých zdrojů

1. ŠOCHOVÁ, Zuzana a Eduard KUNCE. *Agilní metody řízení projektů*. 2. vydání. Brno: Computer Press, 2019. ISBN 978-80-251-4961-4.
2. *Manifest Agilního vývoje softwaru* [online]. [cit. 2020-03-17]. Dostupné z: <http://agilemanifesto.org/iso/cs/manifesto.html>
3. MOREIRA, Mario E. *Being agile: your roadmap to successful adoption of agile*. New York: Apress, [2013]. ISBN 978-1430258391.
4. BROUGHAM, Greg. *The Cynefin MiniBook*. Toronto: C4Media, 2015. ISBN 978-1329508644.
5. *CYNEFIN pomůže určit, kdy aplikovat agile* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.pmconsulting.cz/2018/01/cynefin-pomuze-urcit-aplikovat-agile/>
6. KERZNER, Harold. *Project management: a systems approach to planning, scheduling, and controlling*. Twelfth edition. Hoboken, New Jersey: Wiley, 2017. ISBN 978-1119165354.
7. MÁCHAL, Pavel, Martina ONDROUCHOVÁ a Radmila PRESOVÁ. *Světové standardy projektového řízení: pro malé a střední firmy : IPMA, PMI, PRINCE2*. Praha: Grada, 2015. Manažer. ISBN 978-80-247-5321-8.
8. DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO. *Projektový management podle IPMA*. 2., aktualiz. a dopl. vyd. Praha: Grada, 2012. Expert (Grada). ISBN 978-80-247-4275-5.
9. DENNING, Stephen. *The age of agile: how smart companies are transforming the way work gets done*. New York: AMACOM, [2018]. ISBN 978-0814439098.
10. SUCCI, Giancarlo a Andrea JANES. *Lean Software Development in Action*. New York: Springer, 2014. ISBN 9783662441787.
11. *Lean and Agile: differences and similarities* [online]. [cit. 2020-03-17]. Dostupné z: <https://twproject.com/blog/lean-agile-differences-similarities/>
12. *Differences Between Lean, Agile and Scrum* [online]. [cit. 2020-03-17]. Dostupné z: <https://brainhub.eu/blog/differences-lean-agile-scrum/>
13. *Lean vs. Agile: What's the Difference?* [online]. [cit. 2020-03-17]. Dostupné z: <https://goleansixsigma.com/lean-vs-agile-whats-the-difference/>
14. RADEKA, Katherine. *The mastery of innovation: a field guide to lean product development*. Boca Raton, FL: CRC Press, 2013. ISBN 978-1439877029.

15. *What is Scrum?* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.agilealliance.org/glossary/scrum/>
16. *The Values of Scrum* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.scrumalliance.org/about-scrum/values>
17. MEYER, Bertrand. *Agile!: the good, the hype and the ugly*. New York: Springer, 2014. ISBN 978-3319051543.
18. *What is a Self-Organizing Team in Scrum?* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.visual-paradigm.com/scrum/what-is-self-organizing-team-in-scrum/>
19. *What is a Cross Functional Team? Its Value and How to Make it Work* [online]. [cit. 2020-03-17]. Dostupné z: <https://perfectial.com/blog/make-a-cross-functional-team-work/>
20. *What is a Scrum Master?* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.agilealliance.org/glossary/scrum-master/>
21. *7 Key Product Owner Responsibilities* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.lucidchart.com/blog/product-owner-roles-and-responsibilities>
22. *What is a Product Owner?* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.agilealliance.org/glossary/product-owner/>
23. *Kdo je to Scrum Master? A kdo Product Owner?* [online]. [cit. 2020-03-17]. Dostupné z: <https://soch.cz/blog/management/agile/scrum-management/kdo-je-to-scrum-master-a-kdo-product-owner/>
24. *Scrum Guide* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.scrumguides.org/scrum-guide.html#events-sprint>
25. *Sprint Planning* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.agilealliance.org/glossary/sprint-planning/>
26. *The Scrum Product Backlog - International Scrum Institute* [online]. [cit. 2020-03-17]. Dostupné z: https://www.scrum-institute.org/The_Scrum_Product_Backlog.php
27. *User Stories with Examples and Template* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.atlassian.com/agile/project-management/user-stories>
28. *Epics, Stories, Themes, and Initiatives* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.atlassian.com/agile/project-management/epics-stories-themes>
29. MYSLÍN, Josef. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press, 2016. ISBN 978-80-251-4650-7.
30. *What is a Daily Scrum?* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.scrum.org/resources/what-is-a-daily-scrum>

31. *5 Scrum Meeting Best Practices: Master the Daily Stand-Up* [online]. [cit. 2020-03-17]. Dostupné z: <https://sprint.ly/blog/scrum-meeting-best-practices/>
32. *Sprint Retrospective* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.mountangoatsoftware.com/agile/scrum/meetings/sprint-retrospective>
33. *What is Sprint Planning?* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.scrum.org/resources/what-is-sprint-planning>
34. *Sprint Review Meeting* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.mountangoatsoftware.com/agile/scrum/meetings/sprint-review-meeting>
35. VANDERJACK, Brian. *The Agile Edge: Managing Projects Effectively Using Agile Scrum*. New York: Business Expert Press, 2015. ISBN 978-1631571862.
36. RUBIN, Kenneth S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Upper Saddle River: Addison-Wesley Professional, 2012. ISBN 978-0137043293.
37. AMBLER, Scott W. a Mark LINES. *Disciplined agile delivery: a practitioner's guide to agile software delivery in the enterprise*. Upper Saddle River, IBM Press, 2012. ISBN 978-0132810135.
38. *Physical vs. Online Scrum Boards* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.smartsheet.com/physical-scrum-board-vs-smartsheet-online-scrum-board>
39. *Kanban vs Scrum: Comparing 11 Board Features* [online]. [cit. 2020-03-17]. Dostupné z: <https://miro.com/blog/scrum-kanban-boards-differences/>
40. *Sprint Goal* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.pmconsulting.cz/slovníkový-pojem/sprint-goal/>
41. COBB, Charles G. *The Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach*. New Jersey: Wiley, 2015. ISBN 978-1118991046.
42. GROSS, John M. a Kenneth R. MCINNIS. *Kanban Made Simple: Demystifying and Applying Toyota's Legendary Manufacturing Process*. Special edition. New York: AMACOM, 2003. ISBN 978-0814438435.
43. *What is a Product Roadmap?* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.productplan.com/what-is-a-product-roadmap/>
44. *Product Roadmaps* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.atlassian.com/agile/product-management/product-roadmaps>
45. BJORKHOLM, Tomas a Jannika BJORKHOLM. *Kanban in 30 Days*. Birmingham: Impact Publishing, 2015. ISBN 9781783000913.

46. LEOPOLD, Klaus a Siegfried KALTENECKER. *Kanban change leadership: creating a culture of continuous improvement*. Hoboken, New Jersey: John Wiley & Sons, 2015. ISBN 9781119019718.
47. *Kanban and Scrum - making the most of both*. Null edition. Toronto: C4Media, 2010. ISBN 978-0557138326.
48. BRECHNER, Eric. *Agile Project Management with Kanban*. Redmond: Microsoft Press, 2015. ISBN 978-0735698956.
49. *What Is a Kanban Board: Basics and Details* [online]. [cit. 2020-03-17]. Dostupné z: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board/>
50. ANDERSON, David J. a Andy CARMICHAEL. *Essential Kanban Condensed*. Seattle: Blue Hole Press, 2016. ISBN 978-0984521425.
51. *Extreme Programming: A gentle introduction* [online]. [cit. 2020-03-17]. Dostupné z: <http://www.extremeprogramming.org/>
52. *Extreme Programming Values* [online]. [cit. 2020-03-17]. Dostupné z: <http://www.extremeprogramming.org/values.html>
53. PEARMAN, Greg a James GOODWILL. *Pro .NET 2.0 Extreme Programming*. Berkeley: Apress, 2006. ISBN 978-1590594803.
54. MEASEY, Peter. *Agile Foundations: Principles, Practices and Frameworks*. Swindon: BCS, The Chartered Institute for IT, 2015. ISBN 978-1780172545.
55. ROSSEL, Sander. *Continuous Integration, Delivery, and Deployment: Reliable and faster software releases with automating builds, tests, and deployment*. Birmingham: Packt Publishing, 2017. ISBN 978-1787286610.
56. *Full Comparison: Agile vs Scrum vs Waterfall vs Kanban* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.smartsheet.com/agile-vs-scrum-vs-waterfall-vs-kanban>
57. *O nás* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.avast.com/cs-cz/about>
58. *Foundation Avast* [online]. [cit. 2020-03-17]. Dostupné z: <https://foundation.avast.com/index>
59. KERZNER, Harold. *Project management: a systems approach to planning, scheduling, and controlling*. 10th ed. Hoboken, New Jersey: Wiley, 2009. ISBN 978-0470278703.
60. *Avast Press* [online]. [cit. 2020-03-17]. Dostupné z: <https://press.avast.com/cs-cz/media-materials>
61. *Zdroje informací Pro média* [online]. [cit. 2020-03-17]. Dostupné z: <https://www.avg.com/cs-cz/resources>

62. *HMA VPN service* [online]. [cit. 2020-03-17]. Dostupné z:
<https://www.hidemyass.com/index>
63. *CCleaner Introduces New 'Easy Clean' Feature* [online]. [cit. 2020-03-17].
Dostupné z: <https://www.prnewswire.com/news-releases/ccleaner-introduces-new-easy-clean-feature-300850503.html>
64. *Ke stažení Nadační fond Avast* [online]. [cit. 2020-03-17]. Dostupné z:
<https://nadacnifond.avast.cz/ke-stazeni>
65. *Free VPN Download* [online]. [cit. 2020-03-17]. Dostupné z:
<https://www.avast.com/cs-cz/secureline-vpn#pc>
66. *Slack vs. Microsoft Teams: Which is the Best Chat App?* [online]. [cit. 2020-03-17].
Dostupné z: <https://www.digitaltrends.com/computing/slack-vs-teams/>
67. *JIRA 7 od Atlassian jako nástroj pro sledování požadavků* [online]. [cit. 2020-03-17]. Dostupné z: <http://www.businessit.cz/cz/jira-7-od-atlassian-jako-nastroj-pro-sledovani-pozadavku.php>
68. *Zoom Available to HMS Users* [online]. [cit. 2020-03-17]. Dostupné z:
<https://it.hms.harvard.edu/zoom-available-hms-users>
69. *Strategy* [online]. [cit. 2020-03-19]. Dostupné z: <https://investors.avast.com/our-story/strategy/>