



Bakalářská práce

Agilní metodika ve vývoji softwaru desktopových aplikací

Studijní program:

B0688P140002 Informační management

Autor práce:

Jan Otmar

Vedoucí práce:

Ing. Michal Dostál, Ph.D.

Katedra informatiky

Liberec 2023



Zadání bakalářské práce

Agilní metodika ve vývoji softwaru desktopových aplikací

<i>Jméno a příjmení:</i>	Jan Otmar
<i>Osobní číslo:</i>	E20000182
<i>Studijní program:</i>	B0688P140002 Informační management
<i>Zadávající katedra:</i>	Katedra informatiky
<i>Akademický rok:</i>	2022/2023

Zásady pro vypracování:

1. Principy a problematika agilní metodiky
2. Představení a zhodnocení vybraných metod
3. Analýza fungování metody SCRUM a Kanban ve vývojářském týmu
4. Dotazníkové šetření v rámci týmu
5. Zhodnocení a závěr

Rozsah grafických prací:
Rozsah pracovní zprávy: 30 normostran
Forma zpracování práce: tištěná/elektronická
Jazyk práce: Čeština

Seznam odborné literatury:

- DOLEŽAL, Jan, Pavel MÁCHAL a Branislav LACKO, 2012. *Projektový management podle IPMA*. 2., aktualiz. a dopl. vyd. Praha: Grada. ISBN 978-80-247-4275-5.
- RAD, Nader K. a Frank TURLEY, 2013. *The Scrum Master Training Manual: A Guide to Passing the Professional Scrum Master (PSM) Exam*. Management Plaza. ISBN 978-90-821147-1-3.
- ŠOCHOVÁ, Zuzana a Eduard KUNCE, 2019. *Agilní metody řízení projektů*. 2. vydání. Brno: Computer Press. ISBN 978-80-251-4961-4.
- MYSLÍN, Josef, 2016. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press. ISBN 978-80-251-4650-7.
- BENNETT, Jason a Jennifer BOWEN, 2018. *Lean: 8-in-1 book ultimate collection – lean startup, lean analytics, lean enterprise, kaizen, six sigma, agile project management, kanban, scrum*. England: Independently published. ISBN 9781720216292.
- PROQUEST, 2022. *Databáze článků ProQuest* [online]. Ann Arbor, MI, USA: ProQuest. [cit. 2022-09-26]. Dostupné z: <http://knihovna.tul.cz>

Konzultant: Milan Zimola - Scrum master ve společnosti JABLOTRON CLOUD Services s. r. o.

Vedoucí práce: Ing. Michal Dostál, Ph.D.
Katedra informatiky

Datum zadání práce: 1. listopadu 2022
Předpokládaný termín odevzdání: 31. srpna 2024

L.S.

doc. Ing. Aleš Kocourek, Ph.D.
děkan

Ing. Petr Weinlich, Ph.D.
vedoucí katedry

V Liberci dne 1. listopadu 2022

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

Anotace

Tato bakalářská práce se zabývá problematikou používání agilních metodik pro řízení vývoje desktopových aplikací. Teoretická část práce vysvětluje pojem metodiky řízení vývoje softwaru a projektové řízení a utváří důležitá počáteční stanoviska pro další části této práce. Teorie dále pojednává o klasických a agilních metodikách, přehledně shrnuje jejich základní principy, poukazuje na pozitiva a negativa vybraných metodik obou směrů a předává sadu doporučení, kdy a za jakých podmínek je vhodné danou metodiku používat. V následné praktické části je na základě dotazníkového šetření provedena analýza praktického využití agilních metodik ve firmě zabývající se z větší části vývojem softwaru desktopových aplikací. Cílem šetření bylo získat od zaměstnanců reálný pohled na práci v agilním prostředí, jaké metodiky jsou využívány, které nástroje jim pomáhají tvořit agilní přístup, jaké osobní pocity mají z agilních metodik, popřípadě popsat výhody a nevýhody dané metodiky, ve které se zaměstnanec pohybuje. V závěru pak bylo provedeno celkové shrnutí nabytých poznatků z obsahu celé bakalářské práce, objektivní zhodnocení práce a byl popsán její ekonomický přínos.

Klíčová slova

Vývoj softwaru

Klasické metody

Agilní metody

Aplikovaná teorie

Scrum

Kanban

Anotation

This bachelor thesis deals with the issue of using agile methodologies to manage the development of desktop applications. The theoretical part of the thesis explains the concept of software development project management methodologies and forms important initial positions for the later parts of this thesis. The theory then discusses classical and agile methodologies, summarizes their basic principles, highlights the positives and negatives of selected methodologies of both directions, and conveys a set of recommendations for when and under what conditions it is appropriate to use a given methodology. In the subsequent practical part, an analysis of the practical use of agile methodologies in a company dealing mainly with the development of desktop software applications is carried out on the basis of a questionnaire survey. The aim of the survey was to get a realistic view from the employees about working in an agile environment, what methodologies are used, which tools help them to create an agile approach, what personal feelings they have about agile methodologies, or to describe the advantages and disadvantages of a given methodology in which the employee is working. Finally, an overall summary of the knowledge gained from the content of the whole bachelor thesis was made, an objective evaluation of the thesis was made and its economic contribution was described.

Keywords

Software development

Classical methods

Agile methods

Applied theory

Scrum

Kanban

Poděkování

Rád bych zde vyjádřil poděkování panu Ing. Michalu Dostálovi, Ph.D. za jeho ochotu, profesionální rady a včasnou komunikaci při konzultaci a tvorbě této bakalářské práce, učitelskému sboru Katedry Informatiky za nabití potřebných vědomostí v průběhu studia, svým kamarádům a spolužákům za morální oporu a především celé nejbližší rodině, která při mně stála v dobrém i ve zlém, a která mě stále motivovala svou neutuchající podporou.

Obsah

Úvod.....	15
1 Metodiky řízení softwarového vývoje.....	17
2.1 Klasické metody.....	18
1.1.1 Waterfall.....	19
1.1.2 Spirálový model	23
2.2 Agilní metody.....	26
1.2.1 Scrum	27
1.2.2 Kanban	34
2 Analýza reálné aplikace metody Scrum a Kanban.....	39
2.1 Dotazníkové šetření.....	40
2.2 Vyhodnocení dotazníkového šetření	44
3 Doporučení.....	67
4 Přínosy a ekonomické zhodnocení práce.....	68
5 Závěr	69
Seznam použité literatury	71

Seznam ilustrací

Obr. 1 - Schéma modelu Waterfall.....	20
Obr. 2 - Schéma Spirálového modelu	25
Obr. 3 – Struktura událostí a artefaktů Scrumu	32
Obr. 4 - Virtuální Kanban board v nástroji JIRA	35
Obr. 5 – Procentuální zastoupení pracovních pozic	45
Obr. 6 – Složení primárních firemních rolí.....	46
Obr. 7 – Zastoupení členů jednotlivých týmů	47
Obr. 8 – Hlavní pracovní náplň týmu	48
Obr. 9 - Prvotní pocity členů týmů na agilní metody.....	49
Obr. 10 - Používání agilních metod v rámci týmů	50
Obr. 11 - Dodržujete v týmu striktně zásady vybrané agilní metodiky?.....	51
Obr. 12 - Vyhovuje Vám osobně metodika, kterou tým aktuálně používá?	52
Obr. 13 - Usnadňuje Vám řízení dle agilních metodik práci?	52
Obr. 14 - Myslíte si, že je zvolená agilní metodika pro Váš tým přínosná?	53
Obr. 15 - Kde jste poprvé na vlastní kůži zažili práci v agilním prostředí?.....	63
Obr. 16 - Co ve Vás evokují obecně metodiky řízení SW vývoje?	64

Seznam tabulek

Tab. 1 - Výhody a nevýhody metody Waterfall.....	22
Tab. 2 - Výhody a nevýhody Spirálového modelu.....	26
Tab. 3 - Výhody a nevýhody použití metody Scrum.....	33
Tab. 4 - Hlavní rozdíly metody Scrum a Kanban.....	37
Tab. 5 - Výhody a nevýhody metody Kanban.....	38
Tab. 6 – Výhody Kanbanu uvedené členy týmů.....	54
Tab. 7 - Výhody Scrumu uvedené členy týmů.....	55
Tab. 8 - Nevýhody Kanbanu uvedené členy týmů.....	56
Tab. 9 - Nevýhody Scrumu uvedené členy týmů.....	57
Tab. 10 – Prvotní pohled na agilní metodiky jiných rolí.....	61
Tab. 11 – Výhody a nevýhody metod Scrum a Kanban dle Jiných rolí.....	62
Tab. 12 – Ovlivnění dovedností zaměstnanců agilními metodikami.....	65

Seznam použitých zkratk

CEO	Chief executive officer
CTO	Chief Technical Officer
FDD	Feature Driven Development
IT	Informační technologie
JCS	JABLOTRON CLOUD Services s.r.o.
JIT	Just In Time
LeSS	Large-Scale Scrum
PO	Product Owner
SAFe	Scaled agile framework
SRS	Software requirements specification
SW	Software
TDD	Test Driven Development
TL	Team Lead
WIP	Work In Progress

Úvod

Současná moderní doba postupně stále vytváří větší tlaky na digitalizaci stávajících analogových systémů. Krásným příkladem takového tlaku je automobilový průmysl. Interiéry aut byly dříve zaplněny velkým množstvím různých tlačítek a ovladačů. V současné době ale většinu těchto funkcí zastávají dotykové plochy a obrazovky. Stejně tvrzení o digitalizaci by šlo ale použít například i na bankomatech, doručovacích boxech, či pokladnách. Všechny tyto jmenované věci mají ale jednu společnou vlastnost. Tou je potřeba kvalitního softwaru (SW), který těmto nástrojům všedního života vdechuje život a umožňuje lidem tyto věci ovládat dle svých potřeb.

I přes fakt, že poslední trend spíše nahrává tvorbě mobilních aplikací, základní stavební kámen stále tvoří nespočet desktopových nástrojů. Jednoduše lze říci, že bez počítačů a jeho SW nástrojů by se aktuálně lidstvo neobešlo. Je tedy nutné stále v této oblasti vyvíjet nové produkty, udržovat stávající a inovovat ty zastaralé. Aby takový vývoj ale mohl správně a efektivně fungovat, je potřeba ho řídit vhodným způsobem. K tomu slouží takzvané metody řízení softwarového vývoje. V zásadě existují dva směry a to: klasické metody a agilní metody. A právě agilním metodikám se bude věnovat tato závěrečná práce.

Na začátku teoretické části bude vysvětleno, co se skrývá obecně pod pojmem metodika řízení vývoje, jaké směry metodik existují, a proč je důležité za každou cenu jakoukoliv metodiku přijmout, používat a řídit se dle principů, které definuje.

První polovina teoretické části dále naváže popisem klasických, potažmo tradičních, metod řízení SW vývoje. Bude se snažit celkově shrnout principy a procesy, na kterých klasický přístup staví a charakterizovat je. Zároveň budou představeni dva vybraní zástupci tohoto směru, konkrétně Waterfall a Spirálový model. Práce shrne jejich klíčové vlastnosti, silné a slabé stránky, a uvede příklady, kdy je danou metodiku vhodné implementovat a následně používat. Toto prvotní představení klasických metod řízení vývoje vytvoří v této závěrečné práci kontrastní základ pro lepší pochopení agilních metodik.

Druhá polovina teoretické části se zaměří na agilní metodiky. Vysvětlí jejich odlišný přístup k řízení vývoje SW, který sice dědí některé vlastnosti klasických metodik, ale vysvětluje je po svém. Jako příklad agilních metodik budou představeni dva nejnámější zástupci tohoto směru a to Scrum a Kanban. Teoretická část přehledně

shrne základní principy, které dělají tyto agilní metodiky agilními a jaké výhody, či omezení tento směr přináší. Jako v případě klasických metod, i pro zástupce agilního přístupu k vývoji budou uvedena doporučení a varování týkající se vhodnosti pro projekty zabývající se vývoje SW.

Praktická část bude tvořena analýzou fungování agilních metodik v praxi. Tato analýza se bude snažit přinést reálný pohled do prostředí firmy, která se zabývá vývoje softwaru desktopových aplikací, a která aktivně využívá k řízení projektů agilní metodiky. Za účelem získání potřebných dat, která vytvoří objektivní zhodnocení funkčnosti agilních metodik, bude sestaveno dotazníkové šetření, které bude následně předloženo, se souhlasem vedení firmy, zaměstnancům. Všichni účastníci šetření budou mít anonymní charakter, aby bylo zachováno jejich soukromí a pravidla GDPR. Na konci praktické části budou všechny odpovědi vyhodnoceny a budou učiněny potřebné závěry a představena doporučení testovanému subjektu.

1 Metodiky řízení softwarového vývoje

Metodiky řízení softwarového vývoje jsou sady procesů, které udávají, jakým způsobem se bude postupovat, jaká bude posloupnost jednotlivých činností, jak bude probíhat plánování projektu, jak budou definovány jednotlivé role a co všechno budou mít ve své kompetenci, včetně výběru vedoucího projektu a způsobu, jakým bude projekt řídit. Za slovem metodika tedy není potřeba hledat cokoliiv myšlenkově složitého a cizího. Bez použití jakékoliv metodiky řízení vývoje je, i se špičkovým týmem vývojářů, designerů a grafiků, velmi malá šance na to, aby byl projekt dotažen do zdárného konce. Ano, jednoduché projekty je možné realizovat i bez použití metodiky řízení vývoje. Problém však nastává u větších projektů. Velké projekty jsou totiž životně závislé na týmové práci a je nutné tým vést, koordinovat a řídit. Zároveň je nezbytné nastavit jasná pravidla komunikace se zákazníkem, definovat pravidla pro projektovou dokumentaci a mnoho dalších procesů. Jen tímto přístupem je možné docílit kvalitně vyvinutého softwaru.

Člověk však musí mít na mysli, že metodika řízení není náboženství a nemusí být dodržována do posledního detailu. Tímto způsobem metodiky nebudou a nemůžou fungovat, protože software se vyvíjí a nikoliv vyrábí. Není možné tedy vytvořit přesný univerzální postup. Metodika je lépe popsitelná jako průvodce procesem vývoje softwaru, který tým vede k úspěšnému dokončení projektu.

K dnešnímu dni se můžeme setkat s velkým množstvím nejrůznějších metodik vývoje softwaru. Z tohoto důvodu je náročné vybrat tu, která bude nejvhodnější pro realizaci projektu. Všechny metodiky jsou ale stále relativně novým nástrojem, který není starší více než pár desítek let. Je to z toho důvodu, že první moderní počítače, na kterých by mohl vývoj v takovém prostředí probíhat, nejsou na světě zase až tak dlouho (Myslín, 2016).

V zásadě lze metodiky vývoje rozdělit na klasické, nebo tradiční metodiky, a agilní metodiky. V následujících kapitolách teoretické části této práce jsou jednotlivé přístupy popsány a přestaveni vybraní zástupci obou směrů.

2.1 Klasické metody

Klasické neboli tradiční metodiky se označují tímto termínem, aby je bylo snadné odlišit od dnešních moderních agilních metodik (Myslín, 2016). Vývoje softwaru pomocí těchto metodik jsou založeny na předem organizovaných fázích životního cyklu vývoje softwaru. Zde je tok vývoje jednosměrný: od požadavků k návrhu, vývoji a následně k testování a údržbě. V klasických přístupech, jako je například vodopádový model, má každá fáze specifické výstupy a podrobnou dokumentaci, která prošla důkladným procesem kontroly. Tradiční přístupy jsou vhodné, pokud jsou požadavky jednoduché a srozumitelné, například v odvětvích, jako je stavebnictví, kde všichni jasně rozumí konečnému produktu. Na druhou stranu v rychle se měnících odvětvích, jako je IT, mohou tradiční vývojové postupy selhat při dosahování cílů projektu. Celý soubor požadavků zákazníka a následná projektová dokumentace musí být stanovena v počáteční fázi bez možnosti jejich změny nebo úpravy po zahájení vývoje projektu. Dokumentace tak nabývá vysoké priority a její tvorba se stává nákladnou a časově náročnou. Uživatel zároveň nemůže provádět průběžná hodnocení, aby se ujistil, zda je vývoj produktu sladěn tak, aby konečný produkt splňoval obchodní požadavky. Tradiční metodiky vývoje jsou vhodné pouze tehdy, když jsou požadavky přesné, tj. když zákazník přesně ví, co chce, a může s jistotou říci, že v průběhu vývoje projektu nedojde k žádným zásadním změnám rozsahu. Nehodí se pro velké projekty, např. projekty údržby, kde jsou požadavky mírné a existuje velký prostor pro průběžné změny. (Mavuru, 2018).

Zajímavostí je, že klasická metodika zahrnuje i další metody, jako je PRINCE2, která už vykazuje znaky agilního přístupu, a kterou se řídí různé organizace spadající pod britskou vládu a soukromé společnosti jako Vodafone a Siemens (Kashyap, 2020).

V příštích dvou podkapitolách si představíme nejskloňovanější klasické metody, kterými jsou Waterfall a Spirálový model.

1.1.1 Waterfall

Waterfall neboli vodopádový model je jeden z prvních formálních modelů procesu vývoje softwaru, který se používá již více než 50 let. Poprvé ho popsal v článku z roku 1970 odborník na počítačovou techniku, doktor Winston W. Royce, jako jeden z prvních formálních modelů procesu vývoje softwaru (Whyatt, 2023). Doktor Royce představil Waterfall s těmito slovy: „*Na základě svých zkušeností z oblasti vývoje softwaru jsem získal předsudky a některé z nich uvedu v této prezentaci. Zároveň popíšu své osobní názory na řízení vývoje rozsáhlého softwaru*“ (Walker Royce, 1970). Tento model byl přímým důsledkem jednoduchého přizpůsobení hardwarově orientované vývojové metody v době, kdy neexistoval žádný formální model pro vývoj softwaru. Tento způsob řízení vývoje bylo možné pozorovat ve zpracovatelském nebo stavební průmyslu (Pratt, 2021).

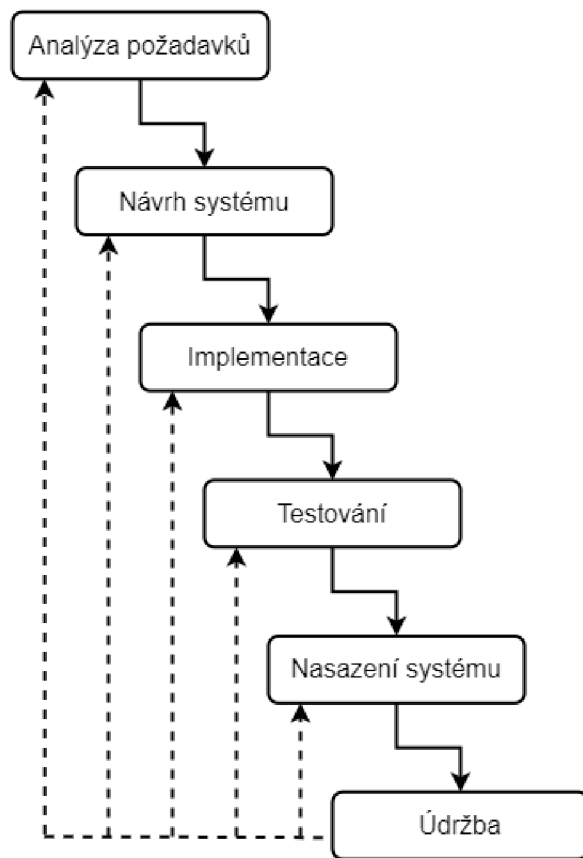
Klasický vodopádový model rozděluje svůj životní cyklus do několika fází. Počítá se, že jedna fáze může být zahájena až po dokončení fáze předchozí. To znamená, že výstup jedné fáze bude vstupem do fáze následující. Proces vývoje lze tedy ve vodopádu považovat za sekvenční tok. Fáze se při tomto procesu vzájemně nepřekrývají (Kumar Pal, 2022).

Vodopádový model lze dělit na 5 až 7 fází. Tato skutečnost závisí na typu projektu a konkrétní metodice. Takto vyčleněné fáze se typicky neopakují a realizují se kompletně pro celý projekt najednou, a ne zvlášť pro každou klíčovou součást (Šimůnek, 2018).

Nejčastěji se základní fáze modelu dělí tímto způsobem:

- Analýza požadavků
- Návrh systému
- Implementace
- Testování
- Nasazení systému
- Údržba

Posloupnost jednotlivých fází graficky znázorňuje obrázek č. 1.



Obr. 1 - Schéma modelu Waterfall

Zdroj: vlastní tvorba dle (Notepub, 2021)

1. **Analýza požadavků** - Účelem první fáze vodopádového modelu je přesně pochopit a řádně zdokumentovat požadavky zákazníka na produkt. Nejprve jsou od zákazníka shromážděny všechny požadavky na software. Ty jsou následně analyzovány. Cílem analytické části je odstranit neúplnost a nekonzistenci. Neúplný požadavek představuje skutečný požadavek, ve kterém byly vynechány některé části. Za nekonzistenci se považuje požadavek, ve kterém je část požadavku v rozporu s jinou částí. Takto analyzované požadavky jsou popsány v dokumentu specifikace softwarových požadavků – SRS (Software Requirements Specification). Dokument SRS slouží jako smlouva mezi vývojovým týmem a klientem. Případný budoucí spor lze vyřešit na základě prozkoumání právě zmiňovaného dokumentu SRS (Kumar Pal, 2022).

2. **Návrh systému** - V druhé fázi se připravují vysokoúrovňové a nízko úrovňové návrhy, které jsou považovány za jednu z kritických fází, která je užitečná při určování požadavků na hardware, požadavků na software, návrhu všech souvisejících operací softwarového projektu a databáze (Yahya, 2022).
3. **Implementace** - Následujícím krokem je fáze implementace, také známá jako fáze kódování. V této fázi se na základě návrhové dokumentace vytvořené v předchozí fázi vyvíjí softwarový produkt. V této fázi by už vývojáři měli být obeznámeni, která konkrétní vývojová prostředí, programovací jazyky a typy databází budou použity při realizaci projektu (Art Of Testing, 2022). Vývojáři vytváří logické modely pomocí malých programů, které označujeme jako jednotky. Tyto malé programové jednotky jsou integrovány do softwaru a dohromady tvoří celek.
4. **Testování** - Ve čtvrté fázi je vyvinutý software nejprve testován softwarovými inženýry a poté validován skupinou uživatelů. Testováním se ověřuje, zda vyvinutý software splňuje požadavky zákazníka, či nikoli. Tato fáze by měla zajistit, že se klient při instalaci a používání softwaru neseťká s žádnými problémy a nedostatky (Neha T, 2020).
5. **Nasazení systému** - V předposlední fázi je po dokončení funkčního a nefunkčního testování produkt nasazen v reálném prostředí zákazníka nebo uvolněn přímo na trh (Tutorials Point, 2013).
6. **Údržba** - Poslední a nejdůležitější fází životního cyklu softwaru je provoz a údržba. Úsilí vynaložené na údržbu představuje 60 % celkového úsilí vynaloženého na vývoj celého softwaru. V zásadě existují tři typy údržby: korektivní údržba, špičková údržba a adaptivní údržba.

Korektivní údržba se provádí za účelem opravy chyb, které nebyly odhaleny ve fázi vývoje produktu. Špičková údržba se zaměřuje na rozšíření funkcí systému na základě dodatečných požadavků zákazníka. Adaptivní údržba je obvykle nutná pro portování softwaru, aby byla zajištěna funkcionality v novém prostředí, například na nové počítačové platformě nebo s novým operačním systémem (Kumar Pal, 2022).

Z principů, na kterých metoda Waterfall staví, plynou samozřejmě i pozitivní a negativní vlastnosti, které pro větší přehlednost znázorňuje tabulka 1.

Tab. 1 - Výhody a nevýhody metody Waterfall

Výhody	Nevýhody
nízká komplexita	absence možnosti zpětné vazby
snadno pochopitelný a implementovatelný	nemožnost opravy chyb v předešlých fázích
každá fáze životního cyklu má určené výstupy	vysoké riziko selhání finálního softwaru
snadná delegace jednotlivých úkolů	úprava požadavků v průběhu není možná
detailní projektová dokumentace	
snadné vytváření projektové dokumentace	

Zdroj: (Art Of Testing, 2022), (Smartiqhub, 2021)

Problém absence jakéhokoliv zpětného přechodu mezi fázemi v klasickém vodopádovém modelu se snaží řešit jeho modifikovaná verze, zvaná také jako iterativní vodopádový model. Hlavním rozdílem mezi klasickým a modifikovaným vodopádovým modelem je přidání systému zpětné vazby. V každé fázi tak vznikne možnost reportovat případné chyby a žádat o opravu v předchozí fázi (Smartiqhub, 2021). Tato skutečnost je vyznačena na obrázku 1 v podobě přerušovaných čar.

Vodopádový model je vhodné používat pouze pro krátkodobé nebo menší projekty, u nichž jsou požadavky zákazníka dokonale zdokumentovány a jasně pochopeny, a pro projekty, u nichž nejsou vyžadovány nebo předpokládány časté změny (Art Of Testing, 2022).

1.1.2 Spirálový model

Spirálový model poprvé definoval softwarový vývojář Barry William Boehm ve svém článku s názvem "A Spiral Model of Software Development and Enhancement" z roku 1986 (William Boehm, 1986). Jedná se o model procesu vývoje softwaru, který se snaží kombinovat prvky vodopádového modelu a modelu iteračního vývojového procesu v jednotlivých fázích, čím se snaží o spojení výhod přístupu shora dolů a zdola nahoru (Alshamrani, 2015).

Přístup shora dolů se spoléhá na to, že vyšší autority určí větší cíle, které se promítnou do úkolů zaměstnanců na nižších úrovních. Oproti tomu styl komunikace zdola nahoru se vyznačuje rozhodovacím procesem, který dává všem zaměstnancům možnost vyjádřit se k cílům společnosti. Jednotlivé úkoly zůstávají plynulé, protože zaměstnanci dosahují svých cílů (Eby, 2018).

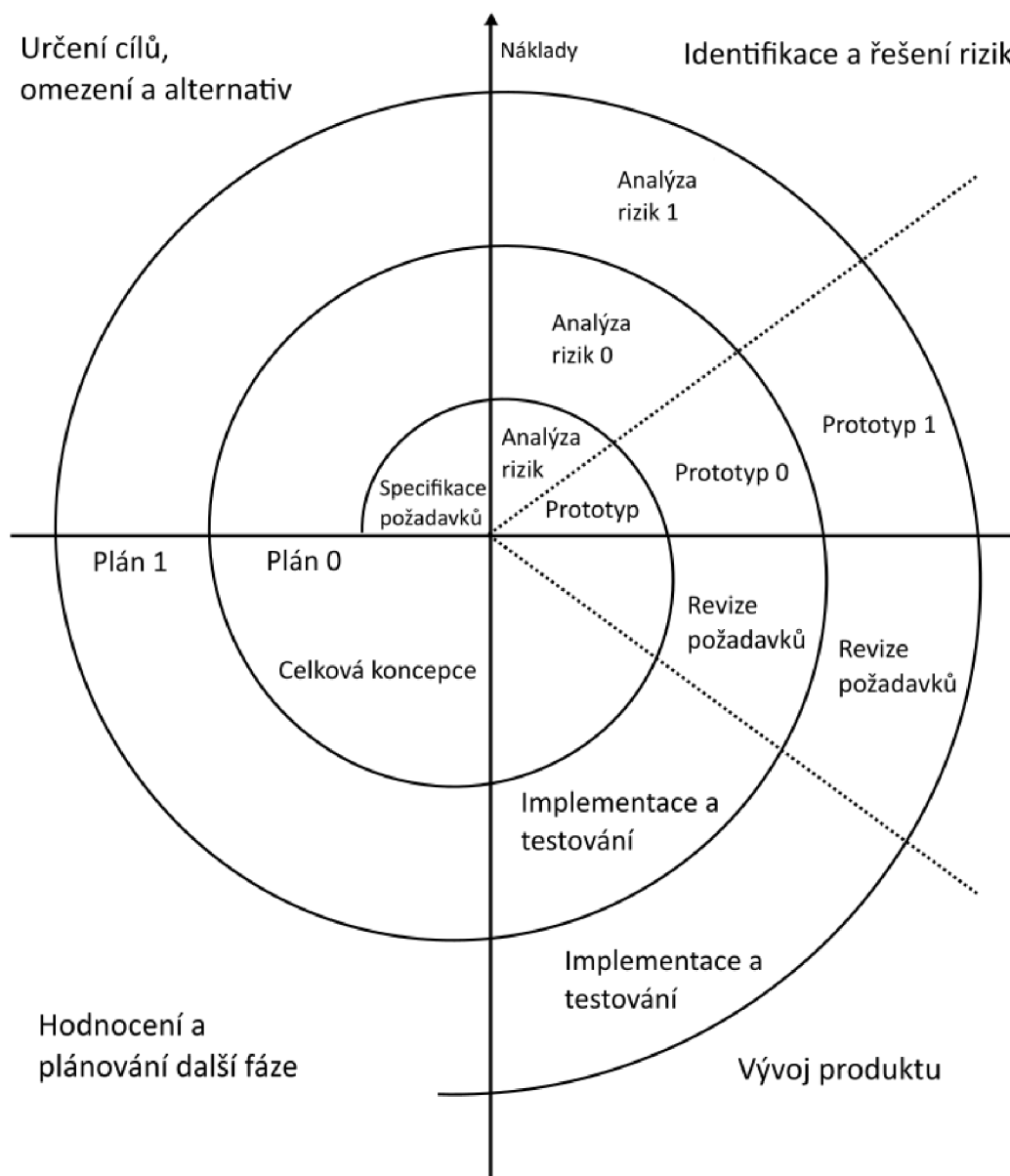
Spirálový model mohou využívat i jiné modely. Tuto skutečnost nazýváme metamodel. Jedním z hlavních rysů zmiňovaného spirálového modelu je zaměření se na hodnocení a minimalizaci rizik projektu, kterého se dosahuje rozdělením projektu na menší segmenty, které pak poskytují v průběhu celého životního cyklu snadnější implementaci změn během procesu vývoje a také možnost přehodnotit znovu rizika a zvážit, zda se i nadále vyplatí v projektu pokračovat. Tuto skutečnost nazýváme iterativní přístup. (Alshamrani, 2015).

Spirálový model tvoří tedy několik iterací, které jsou následně rozděleny do 4 hlavních fází:

1. **Určení cílů, omezení a alternativ** - První čtvrtina se zabývá se určením cílů, omezení a alternativ začíná shromažďováním obchodních požadavků tvořících základní spirálu. Po provedení komplexních systémových studií různých podnikových procesů se shromažďují a definují nové systémové požadavky. Tento postup obvykle zahrnuje i rozhovory s interními a externími uživateli, přípravu podrobných vývojových diagramů zobrazujících proces nebo procesy, pro které má být systém vyvinut, vstupy a výstupy ve smyslu způsobu zaznamenávání nebo zadávání dat a formy, v níž mají být výsledky prezentovány.

2. **Identifikace a řešení rizik** - Postup ve druhé fázi je zaměřen na identifikování rizik a alternativní řešení. Proces zahrnuje řešení faktorů, které mohou ohrozit úspěšné dokončení celého vývoje projektu, včetně nabídky alternativních strategií a omezení. Diskutují se problémy týkající se například možnosti nesplnění požadavků uživatele, požadavků na reporting, přecenění schopností vývojového týmu nebo špatnou, či neexistující kompatibilitu a funkčnost hardwaru se softwarem. Analýza rizik a řešení navržená k jejich zmírnění nebo celkovému odstranění se tak stanou součástí finální strategie pro zahájení vývoje. Na konci celé fáze se vytváří prototyp (C. Reddy, 2021).
3. **Vývoj produktu** - Předposlední kvadrant se nese ve znamení samotného vývoje produktu. Na základě revize shromážděných funkčních požadavků zákazníka, analýze rizik a vytvořeném prototypu začíná samotné psaní kódu. Vzniká tak nová verze vyvíjeného softwaru. Po implementaci všech funkcí následuje testování, které ověřuje celkovou stabilitu a spolehlivost programu.
4. **Hodnocení a plánování další fáze** - Celá iterace se zakončí zákaznickým hodnocením výsledků testů nejnovější vyvinuté verze softwaru. Tato analýza umožňuje programátorům se zastavit a pochopit, co fungovalo a nefungovalo, než budou pokračovat v nové sestavě. Zároveň začíná plánování další fáze a celý cyklus se opakuje. Na konci celé spirály je software konečně nasazen na příslušném trhu (Tech Target, 2019).

Jak projekt postupně roste, spirála se s každou novou iterací rozpíná směrem od středu. Vyšší počet iterací má pak za následek i vyšší celkové náklady na realizaci projektu. Tuto skutečnost zachycuje vertikální osa spirály na obrázku 2.



Obr. 2 - Schéma Spirálového modelu

Zdroj: vlastní tvorba dle (Banks, 2016)

Jako i v případě předchozí metody Waterfall, i Spirálový model nabízí řadu neopomenutelných výhod, ale také i znatelné nevýhody, které by mohly odradit manažery vývojových týmů od použití této metody. Hlavní výhody a nevýhody shrnuje tabulka 2.

Tab. 2 - Výhody a nevýhody Spirálového modelu

Výhody	Nevýhody
proces založený na analýze rizik	nezohledňuje míru nejistoty úspěšného projektu
kladen menší důraz na dokumentaci a kódovou implementaci	nezohledňuje míru variability projektu
zuzitkovává silné stránky jiných modelů a snaží se řešit jejich nedostatky	nebere v potaz možnost změny názoru zákazníka a vrácení hotového projektu
podpora zpětné vazby od zákazníka	vývoj se může protahovat stále do nekonečna
realizace změn v zadání v průběhu celého procesu vývoje	s každou další iterací se zvyšují náklady a objem dokumentace
prezentace prototypu zákazníkovi v každé iteraci	nejisté dodržení termínu dokončení
podněcuje zákazníka k větší důvěře a spokojenosti	velká časová náročnost s vyšším počtem iterací

Zdroj: (Narang, 2022)

Vzhledem k popsáným výhodám a limitům lze říci, že spirálový model je ideální pro velké a složité projekty, které mohou být zákazníkem v průběhu celého procesu pozvolna upravovány, protože průběžné vytváření prototypů a vyhodnocování pomáhá snižovat míru rizika. Toto jsou zároveň i hlavní rysy, které odlišují spirálový model od klasického vodopádového modelu, který je spíše vhodný pro menší a jednodušší projekty, které mají už od začátku k dispozici velmi detailní projektovou dokumentaci.

2.2 Agilní metody

Některé typy projektů zaměřené na vývoj nové technologie, jako je například právě vývoj softwaru, se vyznačují značnou mírou neurčitosti. Mohou chybět informace, které by poskytly spolehlivé časové odhady a projekt samotný může být během svého průběhu ovlivněn velkým množstvím změn, které se často promítají do

samotného zadání. Je tedy obtížné sestavit projektový plán, který bude použitelný. Řešení této situace nabízí právě agilní metodiky řízení vývoje SW (Doležal, 2012).

Agilní přístup k vývoji softwaru, zkráceně nazývaný agile, je do značné míry založen na týmové práci, spolupráci se zákazníkem, časovém rozvržení úkolů a pružnosti co nejrychleji reagovat na změny. Agile se řídí iterativním procesem, kdy jsou projekty rozděleny do kratších časových období, takzvaných sprintů. Dále také dělí velké úkoly na menší celky, které jdou lépe a rychleji odbavit. Na rozdíl od klasických přístupů se také věnuje méně času předběžnému plánování a určování priorit, protože agile je v ohledu změn a vývoje specifikací pružnější (Kashyap, 2020). Agilní přístupy jsou z těchto důvodů vstřícnější k zákazníkům, protože jim dávají možnost provádět pravidelné úpravy v průběhu všech fází vývoje projektu a zároveň i možnost průběžně hodnotit jednotlivá řešení a zjistit, zda plní obchodní potřeby, čímž si zajistí kvalitní výsledky (Mavuru, 2018).

Agilní přístup není ale žádný striktní proces a nejedná se ani o žádné chaotické řízení. I agile má určitá pravidla. Zjednodušeně lze říci, že definuje hranice a vymezuje menší pracovní prostor, v němž si tým může nastavit vlastní pravidla tak, aby se členům dobře pracovalo a tím pádem byli produktivní, efektivní a byli schopni dodat kvalitní software v co nejkratším čase (Šochová, 2019).

V následujících dvou podkapitolách jsou podrobněji popsáni vybraní představitelé agilního přístupu, konkrétněji Scrum a Kanban.

1.2.1 Scrum

Hned na začátek je dobré si říct, jaký je význam názvu této agilní metody. Je Scrum nějaký akronym? Odpověď zní ne. Ve skutečnosti je inspirován skrumáží ve sportu ragby. V ragby se tým sejde v takzvané skrumáži a společně se snaží posunout míč vpřed. V tomto kontextu je Scrum místem, kde se tým spojí, aby posunul produkt vpřed (Scrum.org, 2012).

Scrum je jedním z neúspěšnějších frameworků neboli souboru procesů, úkolů a nástrojů, které poskytují vodítka a strukturu pro realizaci projektu, jak být agilní (Šochová, 2019). Framework Scrumu byl poprvé důkladněji definován Kenem Schwabem and Jeffem Sutherlandem, kteří svou ideu představili světu v roce 1995 jako lepší způsob týmové spolupráce při řešení komplexních problémů. Svoji definici

Scrumu dodnes udržují v příručce The Scrum Guide (Schwaber, 2020). Scrum je empirický proces, kde jsou rozhodnutí založena na pozorování, zkušenostech a experimentech. Scrum staví na 3 pilířích: transparentnost, kontrola a přizpůsobení. Tato stavba podporuje koncept iterativní práce. Základ Scrumu ale tvoří důvěra, která spojuje všechny prvky dohromady. Pokud důvěra ve Scrum týmu chybí, je pravděpodobné, že bude docházet k napětí a překážkám v práci.

Rámec Scrumu je poměrně jednoduchý a skládá se ze Scrum týmu, který tvoří Product Owner, Scrum Master a vývojáři, z nichž každý má specifické odpovědnosti. Celý tým se pak účastní pěti událostí a vytváří tři artefakty (Scrum.org, 2012). Není možné definovat žádné další role, protože by došlo k poškození jednoty týmu a zároveň by vytvoření další role nebylo slučitelné s filozofií Scrumu. Je ale možné, aby jedné osobě bylo přiřazeno více standardních rolí. To se však nedoporučuje (Rad, 2013).

V první řadě si společně představíme jednotlivé role Scrumu.

- ❖ **Product Owner** – Jedná se o obchodně orientovanou osobu, jejímž cílem je maximalizovat hodnotu produktu a práci vývojového týmu. Ve Scrumu se tato osoba nazývá Product Owner, zkráceně PO. Tato role patří jedné osobě. V projektech vývoje softwaru nemají PO povinnost být sami vývojáři. Stačí mít pouze obecné povědomí o procesu vývoje. Hlavně musí být dobře obeznámeni s fungováním podniku.

PO je zodpovědný za tvorbu produktového backlogu, a že každý zadaný úkol v backlogu je snadno srozumitelný pro tým a další zúčastněné strany. PO by měli efektivně komunikovat se zákazníkem a využívat informace k aktualizaci produktového backlogu o všechny změny. Položky backlogu jsou seřazeny na základě jejich hodnoty, takže čím výš jsou na seznamu, tím dříve je bude vývojový tým vyvíjet. PO měří také výkonnost projektu, předpovídají datum dokončení a tyto informace transparentně poskytují všem zúčastněným stranám.

- ❖ **Scrum Master** - Scrum Masteři jsou ti, kteří plně rozumí Scrumu a pomáhají týmu tím, že ho koučují a zajišťují správnou implementaci všech Scrum procesů. Scrum Master je manažerská pozice, která řídí proces Scrum, nikoli Scrum Team. Je jakýmsi služebníkem vedoucího vývojového týmu.

Scrum Master se snaží také odstraňovat překážky, které vývojovému týmu brání v práci, usnadňuje jeho akce, školí ho a trénuje. Scrum Masteři pomáhají také PO tým, že s nimi konzultují různé techniky, předávají si informace a zprostředkovávají související události. Povinnosti Scrum Masteře se neomezují pouze na Scrum Team. Měli by také pomáhat lidem mimo Scrum tým porozumět vhodným interakcím se členy týmu, aby se maximalizovala hodnota vytvořená týmem.

Je možné, aby jedna osoba byla zároveň Scrum Masterem a členem vývojového týmu, i když se to nedoporučuje. Být Scrum Masterem projektu nemusí nutně zabrat 100 % času dané osoby. V takovém případě je nejlepším řešením přidělit tutéž osobu jako Scrum Masteře ve více projektech, než ji učinit členem vývojového týmu.

- ❖ **Vývojový tým** – Ve vývojovém týmu jsou odborníci z oblasti vývoje softwarových aplikací, kteří jsou zodpovědní za kompletní položek z produktového backlogu a řízení vlastního pracovního nasazení. Tvorba vývojového týmu by měla být v souladu s cílem projektu.

Úkol může být v průběhu sprintu přidělen jednomu členovi, ale za splnění tohoto úkolu je zodpovědný celý vývojový tým, protože žádný jednotlivec nevlastní žádný úkol. Vývojový tým dodává konečný produkt v postupných krocích, jak je definováno v produktovém backlogu.

Členům vývojového týmu se doporučuje pracovat pouze na jednom projektu, aby zůstali soustředění a agilní. Složení vývojového týmu by se nemělo tak často měnit, protože při změně složení týmu dojde ke krátkodobému snížení produktivity. Scrum je většinou efektivní, když má vývojový tým 3 až 9 členů. U velkých projektů můžeme použít škálovaný model s více Scrum týmy. Použití více týmů však není ve Scrumu běžné.

V předchozích odstavcích byl představen první základní prvek rámce Scrumu, tedy Scrum tým. Nyní bude popsán druhý, a to artefakty, které tým vytváří společnými silami. Jde o produkty řídicích činností a jsou navrženy tak, aby zvyšovaly transparentnost informací souvisejících s realizací projektu a poskytovaly možnosti kontroly a přizpůsobení. Ve Scrumu se v zásadě vyskytují 3 artefakty:

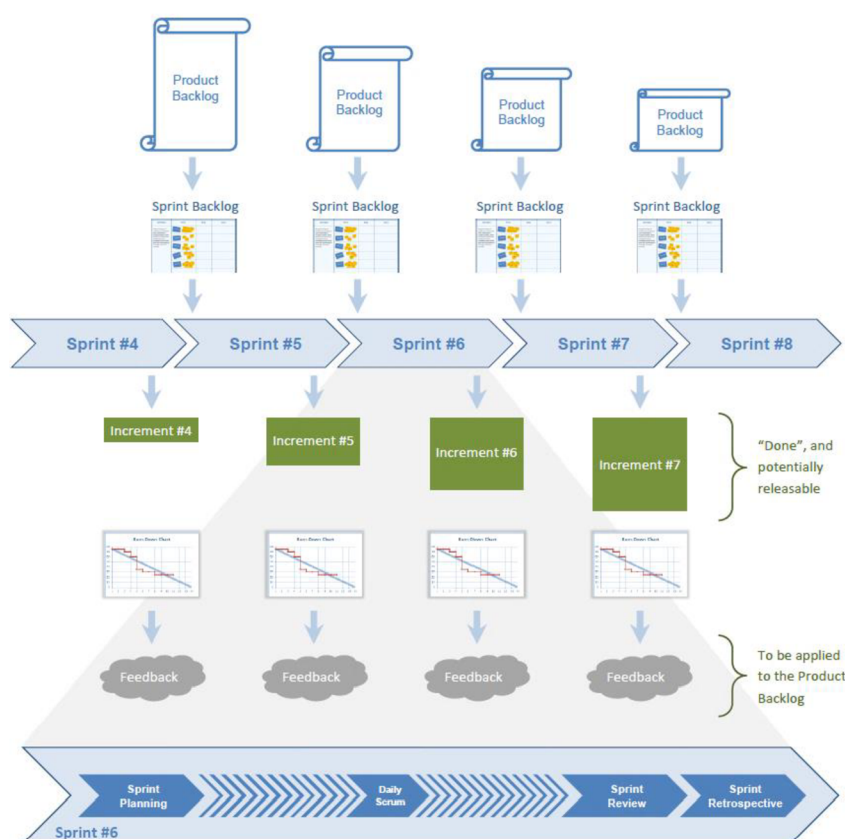
1. **Produktový Backlog** - Uspořádaný seznam úkolů, takzvaných příběhů, které by mohlo být v konečném produktu potřeba zapracovat. Je tvořen především PO, ale může být průběžně upravován i členy vývojového týmu. Všechny položky jsou popsány jednoduchým obchodním jazykem, aby jim rozuměla každá zúčastněná strana. Každý požadavek a každá změna v projektu se odrazí v produktovém backlogu.
2. **Sprint Backlog** - Vybrané příběhy z produktového backlogu, které mají být vyřešeny během sprintu. Vytváří se během Sprint Planningu. Po plánování se Sprint Backlog zmrazí a vývojový tým se na základě tohoto plánu zaměří na dodání takzvaného Inkrementu. V průběhu Sprintu nelze přidávat ani odebírat příběhy. Nutnost získat další informace, zdůvodnit nebo vymazat některé položky v průběhu Sprintu, by mělo být provedeno za přítomnosti Product Ownera.
3. **Inkrement** – Jde o přírůstek, tedy soubor všech položek produktového backlogu, které byly v projektu dosud dokončeny, v rámci určitého sprintu. Každý přírůstek musí být hotový a musí být možno ho vydat, aby zákazník mohl zhodnotit průběh práce. PO ale může a nemusí určitý přírůstek vydat, jestliže se tak všechny zúčastněné strany dohodnou. Produktový Backlog se Sprint od Sprintu snižuje s tím, jak se zvyšuje počet odbavených příběhů v Inkrementu.

Do artefaktů lze zařadit i Definition of "Done", tedy společnou shodu všech členů Scrum týmu na tom, co znamená, že je část práce hotová. Může jít například o situaci, kdy Done znamená, že se implementuje změna, která následně projde kontrolou, a nakonec se otestuje na reálném případě.

Posledním prvkem rámce Scrumu jsou události. Tyto akce mají umožnit kritickou transparentnost, kontrolu, pravidelnost a přizpůsobení. Dáváme přednost těmto předem definovaným schůzkám s pevně stanovenými cíli a maximální dobou trvání před schůzkami ad hoc, které s největší pravděpodobností plýtvají naším časem (Rad, 2013). Metodika Scrum pro události využívá následující pojmy:

- ❖ **Sprint** – To jsou krátká, pevně stanovená období práce trvající měsíc nebo méně, během nichž se pracuje na položkách z produktového backlogu. Tvoří konzistence a zajišťují krátké iterace pro zpětnou vazbu s cílem zkontrolovat způsob práce a to, na čem se pracuje. Pokud jsou cykly delší, může se vytratit duch časté zpětné vazby. Sprint je zároveň jakýmsi kontejnerem pro ostatní události Scrumu. Nový sprint začíná okamžitě po ukončení předchozího Sprintu.
- ❖ **Sprint Planning** – Doslova jde o plánování prací, které budou probíhat během sprintu. Tento výsledný plán je výsledkem spolupráce celého Scrum týmu. PO zajistí, aby účastníci byli připraveni diskutovat o nejdůležitějších položkách produktového backlogu a o tom, jak se vztahují k produktovému cíli. Scrum tým může k účasti na plánování sprintu přizvat i další osoby, aby poskytly rady. Délka schůzky by neměla přesáhnout 8 hodin.
- ❖ **Daily Scrum** - Účelem Daily Scrumu je zkontrolovat pokrok směrem k cíli sprintu a podle potřeby upravit sprint backlog a navzájem se informovat o průběhu práce každého vývojáře. Tato akce je představována 15minutovou schůzkou a pro snížení složitosti se koná každý pracovní den sprintu ve stejný čas a na stejném místě.
- ❖ **Sprint Review** – Sprint Review má za úkol zkontrolovat výsledek sprintu a určit budoucí úpravy. Scrum tým prezentuje výsledky své práce klíčovým zúčastněným stranám a diskutuje se o pokroku při plnění produktového cíle. Událost je časově omezená na maximálně čtyři hodiny v rámci jednoměsíčního sprintu.

- ❖ **Sprint Retrospective** - Retrospektiva sprintu se snaží naplánovat způsoby, jak zvýšit kvalitu a efektivitu. Scrum tým zkontroluje, jak proběhl poslední sprint s ohledem na jednotlivce, interakce a procesy. Identifikují se předpoklady, které je sváděly na scestí, a zkoumá se jejich původ. Tým diskutuje o tom, co se během Sprintu povedlo, s jakými problémy se setkali, a jak byly nebo nebyly tyto problémy vyřešeny. Následně se identifikuje nejužitečnější změna pro zlepšení celkové efektivity a ta se co nejdříve implementuje. Retrospektiva sprintu uzavírá sprint. Je časově omezena na maximálně tři hodiny pro jednoměsíční sprint (Scrum.org, 2012).



Obr. 3 – Struktura událostí a artefaktů Scrumu

Zdroj: (Rad, 2013)

Ač je Scrum poměrně propracovaná agilní metodika řízení vývoje SW, je dobré si i v jejím případě zhodnotit, jaká nabízí pozitiva a negativa. Ta nejcitelnější shrnuje tabulka 4.

Tab. 3 - Výhody a nevýhody použití metody Scrum

Výhody	Nevýhody
efektivní využití času a peněz	plíživý nárůstu rozsahu, díky absenci definitivního data ukončení
rozdělení velkých projektů do snadno zvládnutelných sprintů	vysoké riziko neúspěchu projektu v moment, kdy členové aktivně nespolupracují
kódování a testování během revize sprintu	náročná implementace ve velkých týmech
tým i zákazník mají jasný přehled o průběhu práce	pravidelné schůzky mohou být pro členy frustrující
skvělá podpora zpětné vazby od zákazníků a zúčastněných stran	rámec může být úspěšný pouze se zkušenými členy týmu
možnost provádění změn na základě zpětné vazby	odchod člena může značně zkomplikovat celý vývoj
viditelné úsilí každého člena týmu	pomalejší aplikace změn zadání díky sprintům
zákazník dostává průběžné verze SW	

Zdroj: (Chandana, 2022)

Metodu Scrum můžeme použít pro různě velké SW projekty, které nemají od začátku jasně stanovené požadavky, tudíž zadání se může v průběhu měnit. Zároveň je dobré tuto metodu využít pro projekty, které umožňují kreativní design a inovace. Musíme mít ale na paměti, že celá metodika Scrum stojí na odhodlání lidí, kteří jí používají. Z toho důvodu není dobré tuto metodu používat v situaci, kdy ji lidé odmítají a nerespektují, nebo nechápou její průběh (Scrum.org, 2012).

1.2.2 Kanban

Kanban je jednoduchá agilní metoda, která se zaměřuje na vytváření kontinuálního pracovního postupu a jeho přidanou hodnotu pro zákazníka. Jejím cílem je vizualizovat a zlepšit jakýkoli proces vývoje softwaru. Konečným výsledkem je vývojový kanál, který předvídatelně a efektivně dodává práci s vysokou hodnotou (Kanban tool, 2014). Metodika Kanban je stará více než 50 let. Byla vytvořena ve 40. letech 20. století, kdy Toyota potřebovala optimalizovat své inženýrské procesy. Jedná se o princip JIT, kdy si výroba zažádá o dané množství surovin, které musí být dodáno ve správný čas na správné místo, aby byla zaručena maximální produktivita a efektivita. Základní principy tohoto rámce jsou nadčasové a použitelné téměř v každém odvětví. Proto i agilní týmy vyvíjející software dnes mohou využívat stejné principy JIT tím, že přizpůsobují množství rozpracované práce (WIP) kapacitě týmu. To týmům poskytuje flexibilnější možnosti plánování, rychlejší výstupy, jasnější zaměření a transparentnost v celém vývojovém cyklu (Atlassian, 2014).

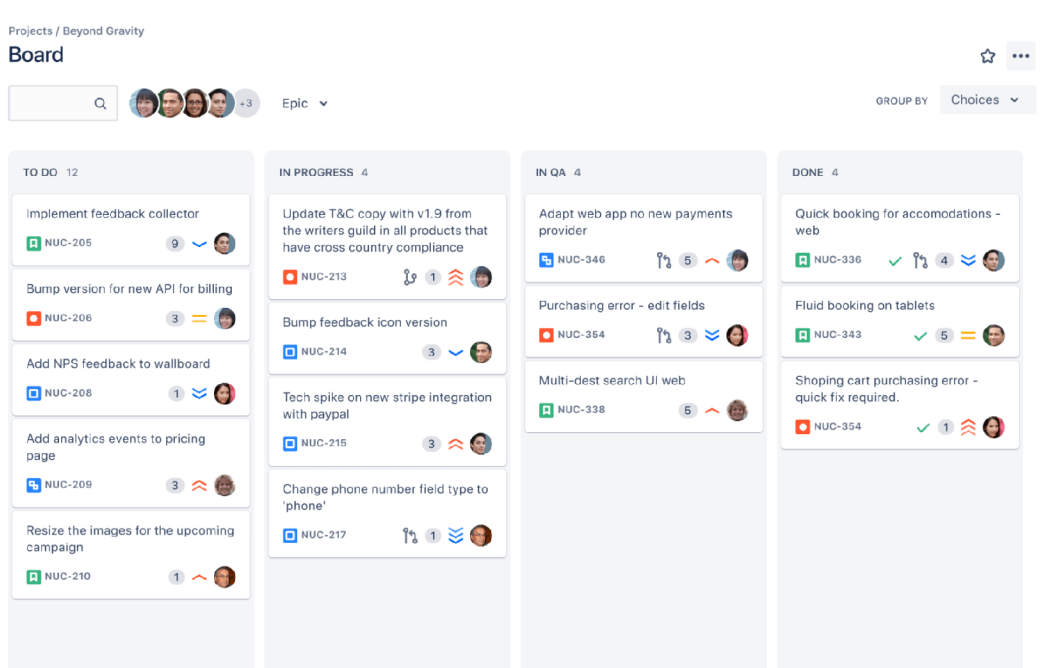
Metoda Kanban má 3 základní principy, kterými by se měl každý řídit, aby dosáhl zlepšení výkonnosti týmu vyvíjejícího software.

1. **Znázornění pracovního postupu** - Vizualizace poskytuje sdílený přehled o procesu, a tím pomáhá identifikovat a řešit případné problémy s pracovními postupy. Navíc zjednodušuje spolupráci mezi lidmi a napříč různými projektovými týmy. Běžnou praxí je proto Kanban board, na kterém jsou vizualizovány konkrétní pracovní fáze.
2. **Omezení WIP** - Jedná se o princip, který je s Kanbanem asi nejvíce ztotožňován. Smyslem limitů rozpracovanosti je snaha zajistit, aby se nikdy nezačal nebo nedokončil úkol, který nelze plynule předat do další fáze, aniž by se zablokoval celý pracovní postup. Množství rozpracované můžeme omezovat pro jednotlivé fáze, osoby nebo typy pracovních položek.
3. **Měření doby realizace** - Po zavedení systému Kanban je třeba kontrolovat a optimalizovat pracovní postupy. Důležitou metrikou je pro vás doba realizace, což je průměrná doba, za kterou se úkoly dostanou ze zadaného počátečního bodu do zadaného konečného stavu. Můžeme například měřit, kolik času zabere provedení kroků v rámci vývoje, správy defektů a řešení problémů s

podporou. Identifikací a odstraněním úzkých míst můžete dobu realizace snadno zkrátit (Kanban tool, 2014).

Principy Kanbanu jsou nyní jasné. Můžeme se tedy podívat na jádro Kanbanu, kterým je Kanban board, dále už jen board.

Kolem boardu se točí práce všech členů týmu a je vnímána jako ultimátní zdroj pravdy. Základní board pro tým zabývající se vývojem softwaru se skládá ze 3 sloupců: Backlog, In Progress a Done. Každý tým si ale může tabuli upravit dle svých potřeb. Běžně se tedy můžeme setkat i se sloupci jako To Do, Code Review nebo In Test. Používají se jak fyzické nástěnky, tak i virtuální, které jsou pro svou sledovatelnost, snazší spolupráci a dostupnost z více míst mnohem výhodnější pro agilní vývoj softwaru (Atlassian, 2014). Bez ohledu na formu nástěnky, její funkcí je vždy zajistit vizualizaci práce týmu, standardizaci jeho pracovního postupu a okamžitou identifikaci a řešení všech blokád a závislostí (Bennett, 2018). Příklad virtuálního boardu je vidět na obrázku 4.



Obr. 4 - Virtuální Kanban board v nástroji JIRA

Zdroj: (Atlassian, 2021)

Samotný board je pak naplněný kartami označovanými jako Kanban cards, které můžeme také pozorovat na obrázku 4. Každá pracovní položka je reprezentována jako samostatná karta na boardu. Karty umožňují členům týmu sledovat postup práce v pracovním procesu velmi názorným způsobem. Uvádějí důležité informace o dané položce práce, takže celý tým má plný přehled o tom, kdo je za danou položku práce zodpovědný, stručný popis práce, jak dlouho bude daná práce trvat a další technické podrobnosti, které jsou pro zadavatele cenné. Karty by se měly vždy posouvat pouze zleva doprava, tedy z Backlogu postupně až do Done (Atlassian, 2014).

Samotný backlog je tvořen zúčastněnými stranami v projektu a je potřeba jej udržovat v sémantickém pořadí, kterého by mělo být dosaženo s co nejmenším úsilím. Způsobem, jak toho dosáhnout, je vytvoření různých úrovní priority pro jednotlivé karty (Bennett, 2018). V případě potřeby shromáždit velký soubor nápadů, lze vytvořit samostatný board, který bude sloužit jako backlog. Nebude se však jednat o Kanban board, protože zde nebude probíhat tok práce. Bude sloužit pouze jako způsob, jak lépe vizualizovat a prioritizovat hrubé nápady a také zpětnou vazbu od zákazníků (Tsonev, 2019).

Kanban jako takový definuje pouze 2 role:

- Service Delivery Manager
- Service Request Manager

Je dobré si ale vyjasnit, že mají pro Kanban mnohem menší význam než například role ve Scrumu. Tým si může a nemusí tyto role určit, protože nejsou potřebné pro celkové fungování Kanbanu. Proto není nutné si je více rozebírat (Novkov, 2019).

Velmi často se můžeme setkávat s názory, že Kanbanu a Scrumu jsou to samé. Sdílí sice společné koncepty, ale jejich přístupy jsou velmi odlišné. Neměly by se tedy vzájemně zaměňovat. Hlavní rozdíly shrnuje tabulka 4.

Tab. 4 - Hlavní rozdíly metody Scrum a Kanban

	Scrum	Kanban
Kadence	Pravidelné sprinty s pevnou délkou	Kontinuální tok
Způsob vydání SW	Na konci každého sprintu, pokud to schválí PO	Průběžné vydávání nebo dle uvážení týmu
Role	PO, Scrum Master, Vývojový tým	Žádné nutné role.
Klíčový ukazatel	Rychlost	Délka cyklu
Aplikace změn	Při plánování nového sprintu.	Změny se můžou aplikovat kdykoli

Zdroj: (Atlassian, 2014)

Některé týmy však spojují ideály Kanbanu a Scrumu do "Srumbanu". Ze Scrumu přebírají pevně stanovené sprinty a role, a díky Kanbanu se zaměřují na limity rozpracované práce a dobu cyklu. Týmům, které s agilními metodami teprve začínají, je však důrazně doporučeno vybrat si jednu nebo druhou metodiku a nějakou dobu se jí řídit (Atlassian, 2014).

Jaké citelné výhody a nevýhody má však samotný Kanban? Tato otázka je objasněna v tabulce 5.

Tab. 5 - Výhody a nevýhody metody Kanban

Výhody	Nevýhody
flexibilní struktura	možnost chybné interpretace informací o úkolu
velmi rychlé implementování změn	neexistující časový rozvrh
přehlednost a transparentnost pracovního postupu	možné zpoždění v každé fázi
nedochází k přetěžování členů týmu	koncepční složitosti projektu mohou být zastíněny jednoduchým vizuálním zpracováním
průběžná dodávka produktu	zaměření na výsledek zastiňuje lidské potřeby

Zdroj: (Jadealm, 2021), (Kaur, 2022)

Srovnání Scrumu a Kanbanu se nabízí i v případě, kdy se tým rozhoduje, kterou metodiku použít. Scrum dobře funguje v týmech, které mají relativně stálé priority, a které se v průběhu času příliš často nemění. Například při vytváření jednoho typu produktu. Priority a úpravy se mohou v průběhu projektu měnit, ale projekt má konečný cíl. Kanban se na druhou stranu hodí spíše pro procesy s velmi proměnlivými prioritami jako IT Service Desk, IT Operations nebo jiné podpůrné práce, kde se postupně vyskytuje nespočet požadavků od více zainteresovaných stran (Insight Spotter, 2020). Neexistuje ale přesná definice situací, kdy použít Scrum, a kdy Kanban. Toto rozhodnutí bude vždy záviset na lidech, kteří ve vybrané metodice budou poté pracovat. To ostatně platí pro všechny agilní metodiky řízení vývoje SW.

2 Analýza reálné aplikace metody Scrum a Kanban

Agilní metodiky pomáhají s řízením vývoje softwaru desktopových aplikací už pár desítek let. Avšak do širšího povědomí se dostávají pouze vybrané z nich. Pokud se člověk například ponoří do světa internetu s cílem se edukovat v oblasti agilního vývoje SW, v drtivé většině narazí mezi uvedenými příklady na metodiku Scrum, a to platí jak pro domácí zdroje, tak i zahraniční. Při bližším zkoumání této metodiky však často dochází ke srovnávání, či porovnávání s metodikou Kanban, a ne méně častěji i na popis spojení obou metod do hybridního celku nazývaného Scrumban. Množství informací o obou metodách pak může vzbudit dojem absolutní převahy, co se reálného užívání týče.

Proč se ale Scrum a Kanban tolik skloňují ve vývoji SW? Proč zastiňují jiné agilní metody jako Six Sigma nebo Extrémní Programování? V čem je jejich síla?

Na všechny tyto otázky lze celkem jednoduše odpovědět i z pohledu laika, který zná pouze standardní popis většiny agilních metod. Scrum a Kanban jsou totiž poměrně jednoduché, versatilní nástroje, které jdou aplikovat téměř na každý projekt vývoje SW bez nutnosti vysokých investic nebo drastických změn ve složení firmy. Zároveň i přes svou malou komplexitu nabízejí zvýšení produktivity a zlepšení komunikace mezi zaměstnanci i směrem ke klientům. Tyto teoretické výhledy jsou více než lákavé. Historie však už několikrát dokázala diametrální odlišnost teorie od praxe.

Praktická část této závěrečné práce byla proto zaměřena na analýzu reálné aplikace agilních metodik Scrum a Kanban ve firmě zabývající se vývojem desktopových aplikací, a která tyto metodiky aktivně využívá. Cílem bylo zjistit, kterou z metodik vývojové týmy používají raději, jak jejich působení vnímají jednotliví členové týmu a zaměstnanci spolupracující s týmy, a zhodnotit celkový přínos používané metodiky.

Za účelem vytvoření této analýzy agilních metodik bylo provedeno dotazníkové šetření, které bylo vytvořeno s ohledem na věcnost, soukromí zaměstnanců, jejich postavení ve firmě a celkovou časovou náročnost na vyplnění, aby délka dotazníku neodrazovala potenciální respondenty.

Vybraným subjektem k tomuto dotazníkovému šetření se stala společnost JABLOTRON CLOUD Services s.r.o. (JCS) se sídlem v Jablonci nad Nisou, která společně s dalšími sesterskými společnostmi spadá do skupiny JABLOTRON GROUP a.s. Vybrána byla z důvodů splnění všech kritérií, které stanovila již zmiňovaná

analýza: aktivní užívání agilních metodik Scrum a Kanban, vývoj desktopových aplikací a přítomnost vícero vývojových týmů. K této skutečnosti velice napomohl i fakt, že právě tato společnost se stala místem výkonu roční řízené praxe autora, v rámci které byla vytvořena i tato závěrečná práce.

Dotazníkové šetření bylo vytvořeno pomocí online aplikace Google Forms, která je veřejně dostupná online, má velmi jednoduchý uživatelský interface, a která díky svému systému automatického třídění odpovědí značně ulehčuje práci tvůrci dotazníku, přičemž zároveň minimalizuje faktor lidské chyby.

2.1 Dotazníkové šetření

Celé dotazníkové šetření tvořilo 15 až 18 otázek. Tento rozdíl byl dán faktem, že zaměstnanci JCS zastávají různé role a některé položené otázky by mohly působit zcestně. Celkový počet oslovených čítal 67 lidí, kterým byl dotazník prezentován v interním firemním kanálu aplikace Slack, který tvoří hlavní komunikační spojení mezi vedením, zaměstnanci a externisty. Dotazníkové šetření se během 3 týdnů nakonec zúčastnilo 21 osob z celkem 12 různých týmů, ve kterých buď zaměstnanci působí, nebo s ním úzce spolupracují.

I přes fakt, že většina oslovených má povědomí o práci v agilním prostředí, byly položené otázky obecného charakteru, aby se nevytvořila zbytečná znalostní bariéra, která by mohla mít za následek nevyplnění dotazníku. Každá otázka sestávala ze zadání a dle vyžadovaného typu odpovědi byli respondenti žádáni o krátkou vlastní odpověď, či výběru z možností. U výběru z možností byla ve většině případů přítomna možnost „Jiná“, kdyby si oslovený člověk nedokázal vybrat z nabízených odpovědí. Jak už bylo řečeno v úvodu této kapitoly, velký důraz byl kladen na časovou náročnost dotazníku. Jeho vyplnění mělo v nejkrajnějším případě zabrat 10 minut, což mělo opět motivovat oslovené osoby k zapojení se do analýzy.

V následující části bude uvedeno, jaké informace se dotazníkové šetření snažilo od respondentů získat, a jaký byl jejich smysl pro analýzu používání agilních metod ve firmě. Celý popis šetření bude rozdělen do bloků, jelikož některé informace plynuly z otázek, které se otevíraly dle odpovědí na určité mezní otázky. Kompletní dotazník s přesným zněním a rozdělením otázek lze nalézt v příloze této práce.

❖ **Základní otázky**

První blok tvořil sadu 2 základních otázek, jejichž smyslem bylo zjistit, jakou pracovní pozici účastník šetření zastává a jaká je jeho primární firemní role. Každá pozice a role má totiž svá specifika a agilní metodika, ve které se zaměstnanec pohybuje, může a nemusí být vždy tou nejlepší volbou pro jeho pracovní nasazení. Takovéto rozdělení bylo nutné, neboť vývojář, který je členem týmu může vnímat agilní metodiku jinak než například PO, který sice s týmem úzce spolupracuje, ale není jeho plnohodnotnou součástí.

Zodpovězení otázky ohledně firemní role bylo stěžejní, protože zvolená odpověď měla vliv na zbytek struktury dotazníkového šetření, kdy se dle zvolené možnosti odemkla respondentovy určitý blok otázek. Konkrétně se jednalo o otázky určené pro členy týmu, Product Ownery, Team Leady a ostatní specializované role.

❖ **Otázky pro členy týmu**

Členové jsou nejcennějším zdrojem informací z celého dotazníkového šetření, protože má na ně zvolená agilní metodika největší vliv. Nejvíce otázek bylo proto položeno právě jim. Celkem bylo členům položeno 12 otázek.

Začátek tohoto bloku tvořily 2 otázky týkající se názvu a hlavní pracovní náplně týmu, do kterého člen patří. Cílem bylo třídit členy dle týmů a samotné týmy dle zaměření práce, čímž bylo možno lépe porovnat nadcházející odpovědi týkající se využívání metodik, aby se utvořil celistvější obrázek o celkovém vlivu agilního prostředí.

V dalším kroku následovalo získání informace o aktuálně používané metodice v týmu, do kterého dotazovaný člen patří a poté bylo přistoupeno k rozboru individuálních pocitů, které v členech týmů vyvolávalo používání dané agilní metodiky v rámci týmu. Konkrétně šlo o otázky týkající se stylu používání metodiky, její působení na osobu samotnou, názoru na fungování metodiky v týmu a popisu pozitivních a negativních prvků metodiky.

Poslední část bloku otázek pro členy byla věnována používanému SW pro sledování pracovního postupu týmu a interní komunikaci. Otázky v této oblasti

byly zaměřené přímo na vybrané SW produkty, se kterými zaměstnanci během svého pobytu ve firmě přicházejí do styku.

❖ **Otázky pro Product Ownery**

I přes fakt, že role Product Ownera je úzce spjatá s metodikou Scrum, ve společnosti JCS se označení této role obecně zažilo pro popis pracovní náplně zaměstnanců, kteří tvoří a starají se o určitý produkt a spolupracují s týmy, či týmy, na jeho realizaci. Nejsou však jeho plnohodnotnou součástí. Zároveň sami stojí prakticky na opačném konci toku informací, než členové týmů, co se zadávání práce týče a jejich pohled na práci dle agilní metodiky má i v tomto případě vysokou informační hodnotu. Toto tvrzení podporuje i fakt, že role PO má ve firmě četné zastoupení.

Jak už bylo řečeno, PO netvoří součást týmu, pouze s ním spolupracuje. Z tohoto důvodu bylo tedy nutné vytvořit modifikovanou sadu otázek, aby jejich skladba dávala potenciálnímu respondentovi smysl. To platilo i pro účely analýzy.

V první řadě bylo potřeba zjistit, s kolika týmy se PO podílí na vytváření produktu, či produktů. Ne zřídka se stává, že PO spolupracuje s více týmy. Následovala otázka, kde měl účastník šetření vyjmenovat jednotlivé týmy.

Následující otázky byly již skoro totožné jako v případě členů týmů, tedy zjistit jaké pocity v PO vyvolává agilní metodika, která z metodik mu nejvíce vyhovuje při plnění pracovních povinností, jaký vliv si myslí, že má zvolená metodika na tým a zkusit popsat jaké výhody a nevýhody mají jednotlivé metodiky, tedy Scrum a Kanban.

Závěr bloku otázek se pak opět zaměřil na získání informací o používanému SW pro sledování pracovního postupu a interní komunikaci, a byly opět zaměřeny přímo na vybrané SW produkty.

❖ **Otázky pro Team Leady**

Team Lead je specializovaná role s organizačními a vůdčími schopnostmi, která tvoří pomyslný střední management a stojí v čele týmu, kde zodpovídá za plnění cílů, které si tým sám vytyčil, nebo které byly týmu zadány. Snaží se objektivně hodnotit práci členů týmu, motivovat a přizpůsobovat pracovní prostředí za účelem dosažení optimálních výsledků. Mezi další kompetence Team Leady dále patří finanční ohodnocení všech členů týmu, schvalování dovolené, nábor nových zaměstnanců a komunikaci požadavků týmu na úroveň top managementu.

Tato role se velice podobá Product Ownerovi. Team Lead není přímou součástí týmu a spíše stojí opět na opačném konci toku informací, než tým samotný. Proto byl v tomto případě využit stejný blok otázek, jako u role PO.

❖ **Jiná role**

Poslední blok otázek byl připraven pro ostatní role vyskytující se ve skladbě společnosti JCS, jejichž pracovní náplň se odlišuje od ostatních rolí, které byly vyjmenovány v předešlých částech a jejichž výskyt není tolik frekventovaný. Tuto volba byla určena například pro vrcholový management, sekretariát nebo SW architektky.

Tyto specializované role nemusí být nutně každý den v kontaktu s jednotlivými týmy a jejich členy, tudíž otázky ohledně práce týmu byly v tomto bloku otázek vynechány. Hlavní zaměření se tedy týkalo pocitů z používání agilních metod, jejich výhod a názoru na jejich přínos pro týmy, které v nich pracují.

Poslední dvě otázky se týkaly opět využívání SW pro sledování pracovního postupu týmů a interní komunikaci v rámci firmy.

❖ **Doplňující otázky**

Poté, co respondenti prošli popsáním třídícím mechanismem dotazníkového šetření, jim nakonec byly položeny poslední 4 doplňující otázky, které měli pro všechny stejné znění.

Cílem bylo zjistit, kde dotázaný člověk poprvé na vlastní kůži zažil práci v agilním prostředí a jestli díky jeho působení došlo ke zlepšení osobnostních

dovedností. Dále bylo žádoucí zjistit, co v zaměstnancích obecně evokují metodiky řízení SW vývoje. Na začátku teoretické části byl dle myšlenek pana Josefa Myslína přijmut fakt, že metodiky řízení jsou spíše příručky, jak dělat věci lépe, a že se nejedná v žádném případě o doslovné návody. Proto byli všichni respondenti požádáni o projevení osobního názoru v této oblasti.

Na samém závěru dotazníkového šetření padla poslední otázka týkající se širšího povědomí o agilních metodách, kdy měli dotázaní zaměstnanci zkusit vyjmenovat alespoň jednu další agilní metodiku kromě již zmíněného Scrumu a Kanbanu.

2.2 Vyhodnocení dotazníkového šetření

Dle číselných údajů zmíněných na začátku podkapitoly 2.1 lze usoudit, že dotazníkové šetření bylo, co se týče počtu odpovědí, úspěšné, protože se ho zúčastnila téměř třetina všech oslovených lidí. Toto tvrzení podporuje i fakt, že byly získány poznatky z působnosti lidí z 12 různých týmů. Díky přítomnosti těchto skutečností bude více podpořena objektivnost.

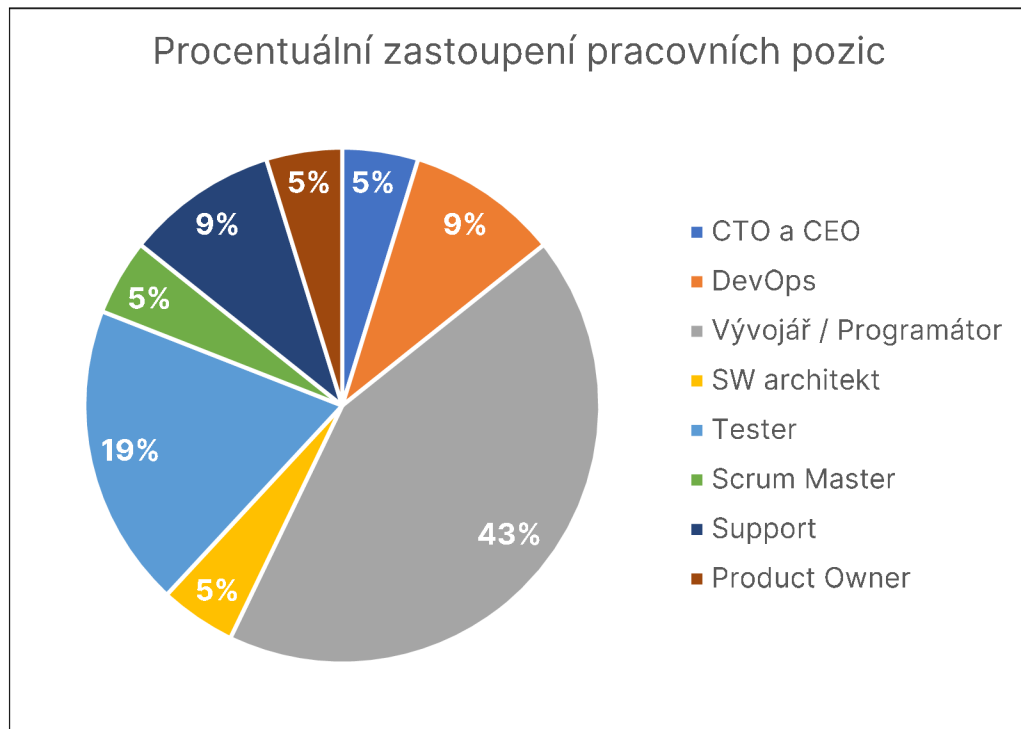
Stanovený cíl pro dotazníkové šetření byla nejméně čtvrtina oslovených lidí, konkrétněji alespoň 14 odpovědí. 21 je tedy v tomto případě přijatelné číslo. Vedlejším cílem bylo získat tyto odpovědi od různých osob z co možná nejvíce týmů, alespoň však 3. I tento cíl byl splněn.

Jelikož byly všechny položky z dotazníkového šetření povinné, nebylo možné odeslat nekompletní seznam odpovědí. Tím bylo zamezeno případným chybám a odchyškám, kdy by nevyplněné odpovědi mohli zkreslovat celkový obraz šetření.

Vyhodnocení a analýza celého dotazníkového šetření bude rozdělena stejným způsobem, jako v případě předchozího popisu položených otázek. Nejprve se zaměří na úvodní otázky, následně rozebere odpovědi dle firemních rolí, zhodnotí doplňující otázky a na závěr se pokusí spojit poznatky všech účastníků do výsledného obrazu.

❖ Základní otázky

Skladba pracovních pozic, která byly v odpovědích zaznamenány, pokryla velkou část všech existujících pozic ve firmě. Konkrétní pozice a jejich procentuální četnost lze ilustrovat grafem na obrázku 5.

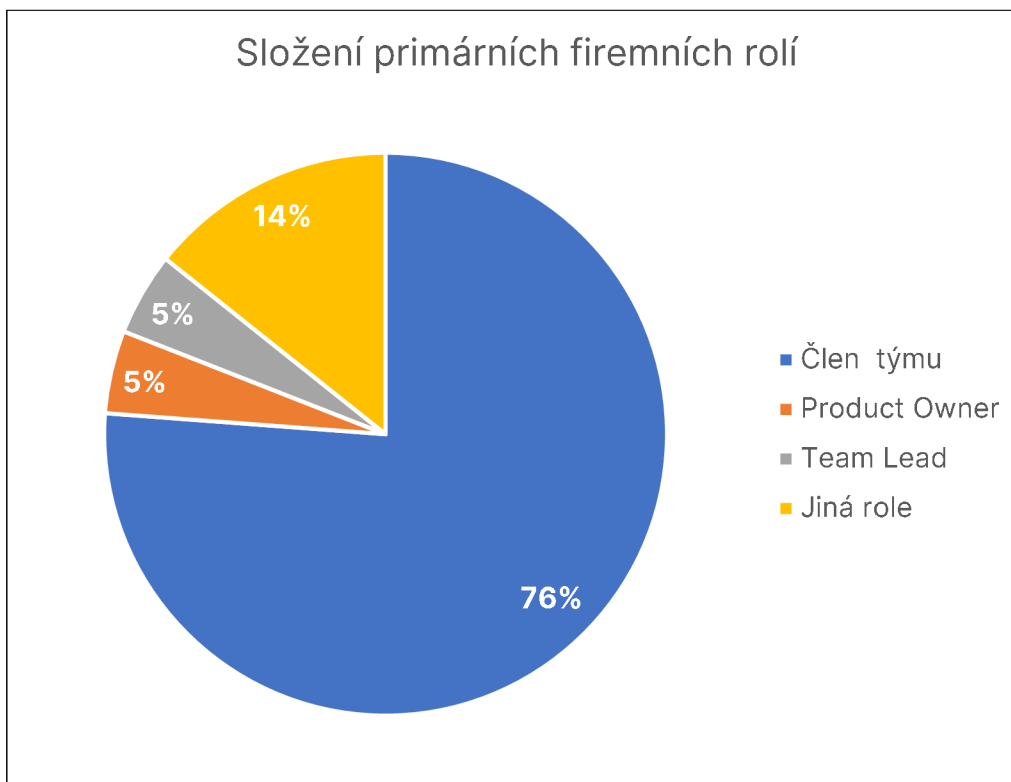


Obr. 5 – Procentuální zastoupení pracovních pozic

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Více jak třetina respondentů tvořili vývojáři / programátoři. To je vzhledem k povaze firmy pochopitelné a z hlediska analýzy i žádané, protože vývojáři tvoří ve většině případů páteř každého týmu, a agilní metodiky na ně mají největší dopad. Druhé nejčetnější zastoupení tvořili testeři, kteří také tvoří nedílnou část každého týmu a bez nichž by výsledný produkt nemusel dosahovat požadovaných kvalit. Ostatní pracovních sice už nebyly v tak hojném zastoupení, i tak ale poskytly vitální informace z jejich úhlu pohledu.

Co se týče rozdělení firemních rolí, jež úzce souvisí s pracovními pozicemi, jejich složení z procentuálního hlediska lze opět pozorovat v grafu na obrázku 6.



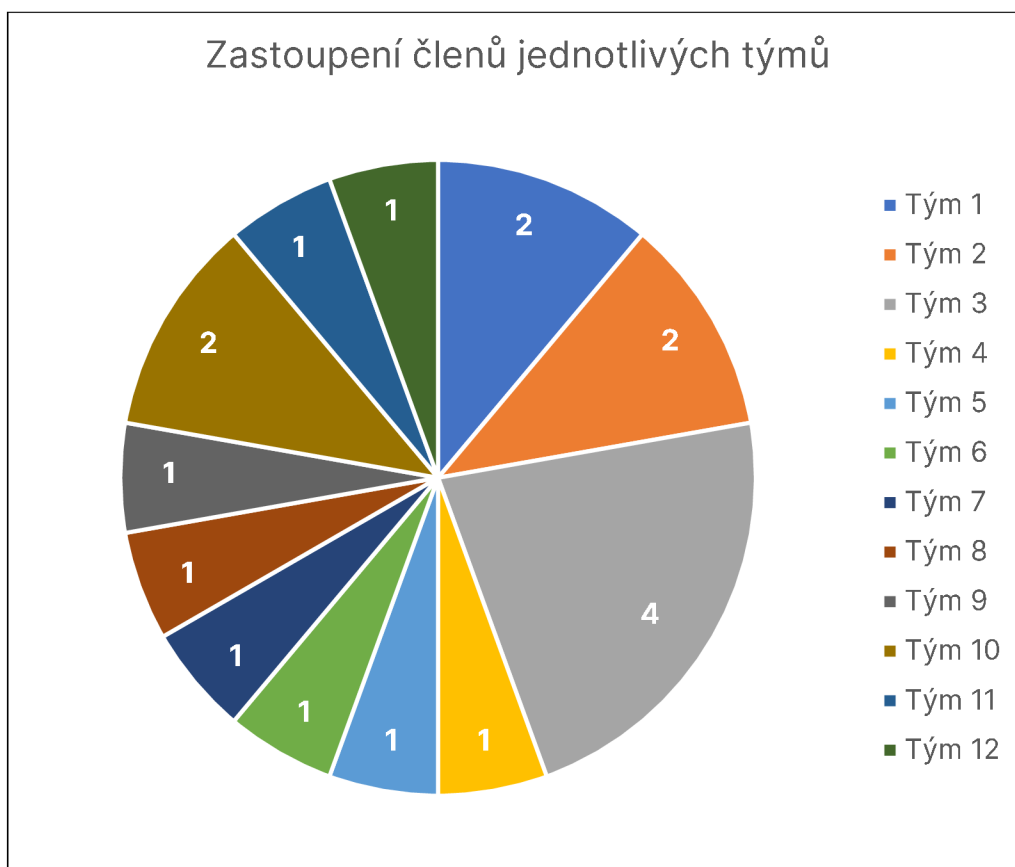
Obr. 6 – Složení primárních firemních rolí

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Je vidět, že členové týmů tvořili tři čtvrtiny všech odpovědí, následovaly jiné role se 14 % a Product Owneři s Team Leady (TL) zastupovali každý 5 %. Počet členů týmu byl více než dostatečný k objektivní analýze. V oblasti rolí, které spolupracují s týmy, tedy PO a TL, už počty razantně klesly, tudíž pohled na agilní metodiky z toho úhlu bude už spíše orientačního rázu. Jiné role v tomto šetření byly zastoupeny také velmi nízkými počty, ale v tomto případě byl takovýto výsledek žádaný, jelikož jim kladené otázky byl velice obecné a mohly by vytvořit zkreslený závěr.

❖ Členové týmů

Členové týmu tedy tvořily největší část respondentů, přesněji 16 odpovědí. Toto množství navíc zahrnovalo jednotlivce z celkem 12 týmů. Přesnou účast z každého týmu vyjadřuje graf na obrázku 7.

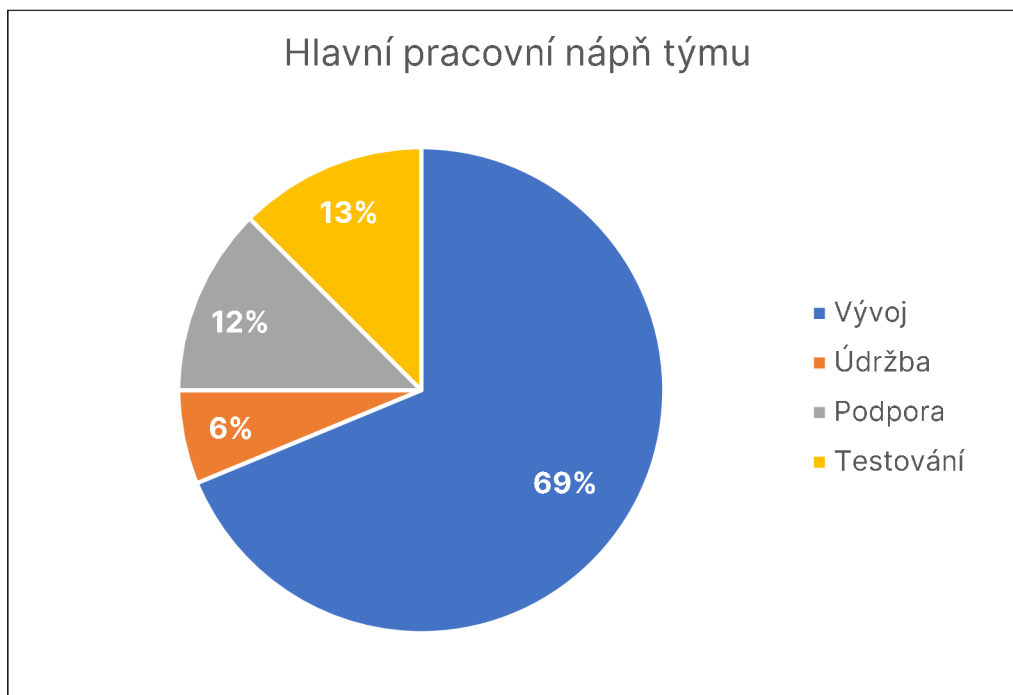


Obr. 7 – Zastoupení členů jednotlivých týmů

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Nejpočetnější skupinou se stali členové z Týmu 3. K tomu velmi přispěl fakt, že tento tým se stal místem pro výkon roční řízené praxe v rámci studia. Účast ostatních týmů se pak pohybovala mezi jedním až dvěma členy.

Všichni členové měli zároveň popsat, jaká je hlavní pracovní náplň jejich týmu. V zásadě se vyskytovaly tyto odpovědi: vývoj, údržba, podpora, testování. V žádném případě, kdy odpovídalo více členů z jednoho týmu, nedošlo v této otázce k neshodě v tom, co je hlavním posláním celého týmu.



Obr. 8 – Hlavní pracovní náplň týmu

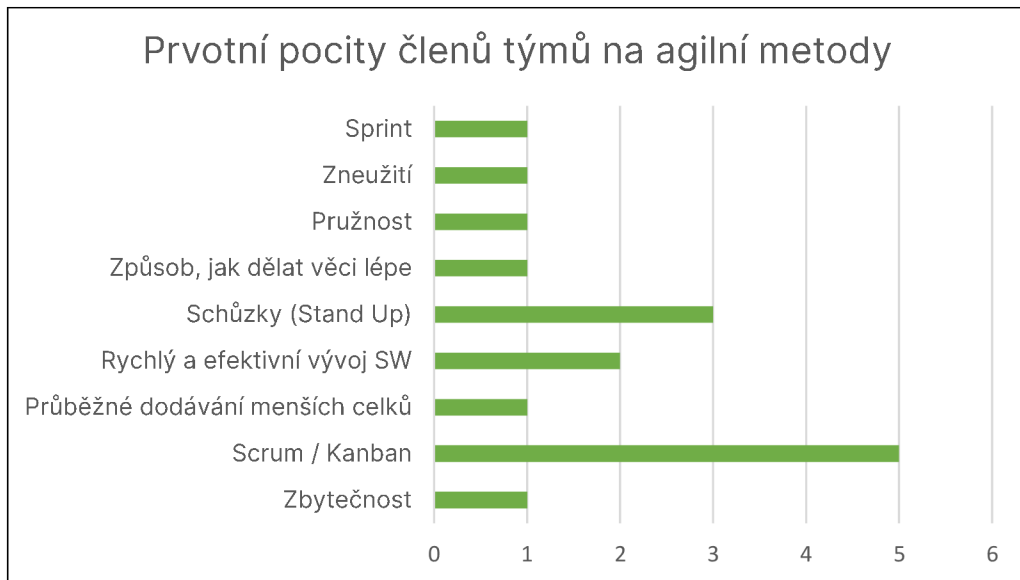
Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Na obrázku 8 je vidět, že převážná část týmů se věnuje vývoji, tedy tvorbě nových produktů. Na druhou stranu údržba je zde ve značné menšině.

V tuto chvíli se dotazník ubírá své zaměření směrem k samotným agilním metodám. První náhled do myšlenek členů týmů měla poskytnout otázka, kdy měl respondent napsat první věc, která ho napadne, když se mluví o agilní metodice. Nejčtenější odpovědi se staly právě názvy zkoumaných metodik: Scrum, Kanban. Je vidět, že přítomnost těchto metodik je mezi členy často skloňovaným tématem a jejich názvy se staly postupem času synonymem k agilnímu řízení. Druhou nejčastější odpovědí zaujímaly obecně schůzky, například Stand Up. Při agilní vývoji jsou interakce mezi lidmi důležité, a proto je zmínění schůzek přirozenou reakcí.

Ostatních odpovědi s četností 1 až 2, pak zmiňovaly: průběžné dodávání menších celků, rychlý a efektivní vývoj SW, způsob, jak dělat věci lépe, průběžné dodávání menších celků, zneužití, zbytečnost.

Zneužití a blbost se v tomto ohledu staly prvními zmíněnými zápory. I když se každá z těchto odpovědí vyskytla pouze jednou, evokují potenciální nespokojenost u 2 dotázaných členů týmu.



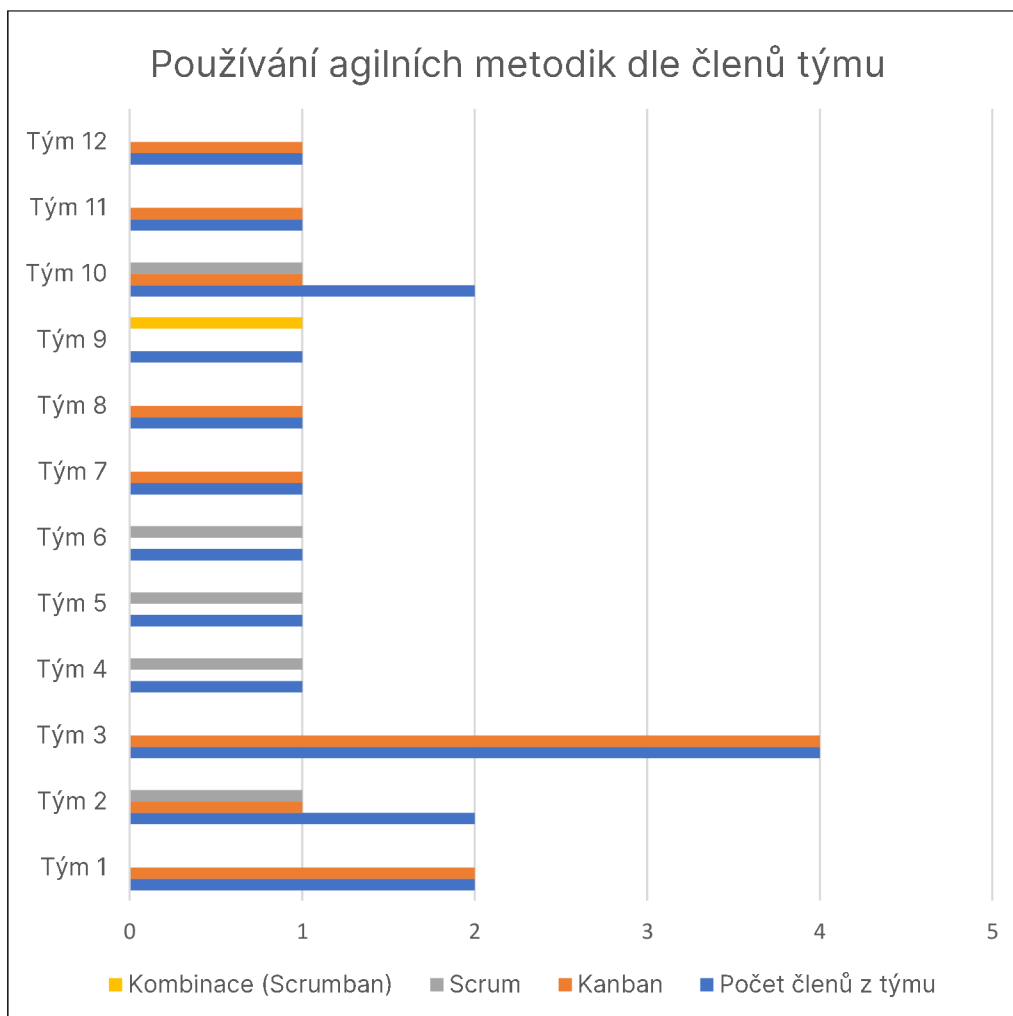
Obr. 9 - Prvotní pocity členů týmů na agilní metody

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Šetření dále pokračovalo zásadní otázkou a to, jakou metodiku týmy používají. Zde bylo nejen záhodno zjistit, která ze zmiňovaných 2 metodik je oblíbenější, ale odhalit i případ, kdy si jednotliví členové stejného týmu nejsou jistí, kterou metodiku používají.

Dominanci zde prokázal Kanban, kdy došlo u 6 týmů ke vzájemné shodě v používané agilní metodice. Naproti tomu pouze u 3 týmů byl jednohlasně označen Scrum a u 1 týmu kombinace obou, tedy Scrumban.

V případě Týmu 2 a Týmu 10 však došlo k vzájemné neshodě v odpovědích, kdy došlo k označení různých metodik. Tuto skutečnost ilustruje graf na obrázku 10, který ukazuje, kolik členů z týmu odpovídalo a jaká metodika byla kolikrát uvedena pro každý tým.

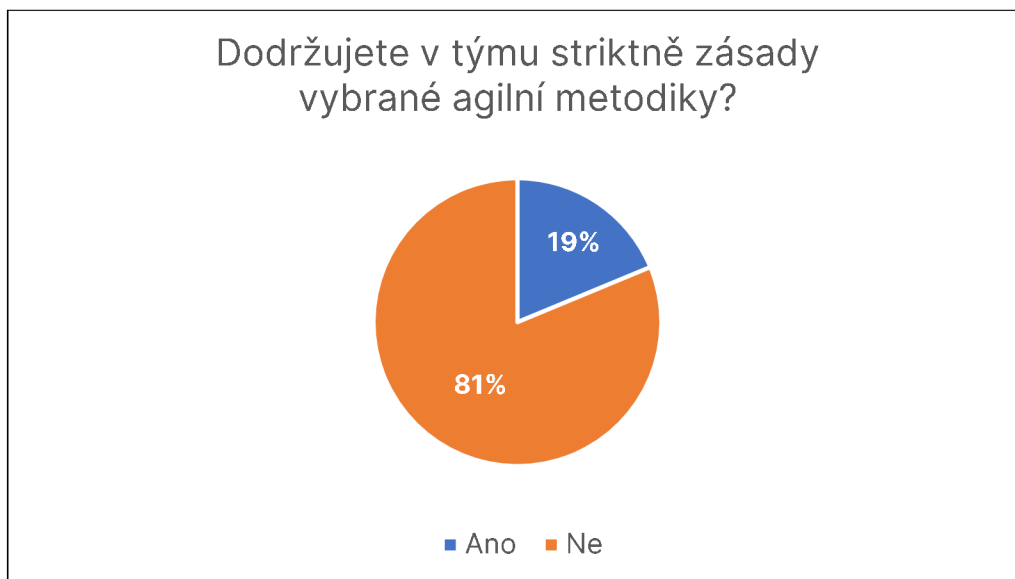


Obr. 10 - Používání agilních metod v rámci týmů

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Fakt, že k takové neshodě došlo může vést k více závěrům. Buď mají někteří členové obtíže rozeznat jednotlivé metodiky, nebo dochází ke špatnému předávání informací v rámci týmu. Lze ale předpokládat, že v případě, kdy bylo k dispozici více odpovědí z týmů, kde participoval pouze 1 člen, došlo by ke smíšeným odpovědím daleko častěji.

Podobně rozporuplné výsledky přinesla i otázka týkající se striktního dodržování zásad používané agilní metodiky. V této části šlo ale spíše o individuální názor jedinců, tudíž rozdílné odpovědi členů ze stejného týmu zde nehráli žádnou roli.

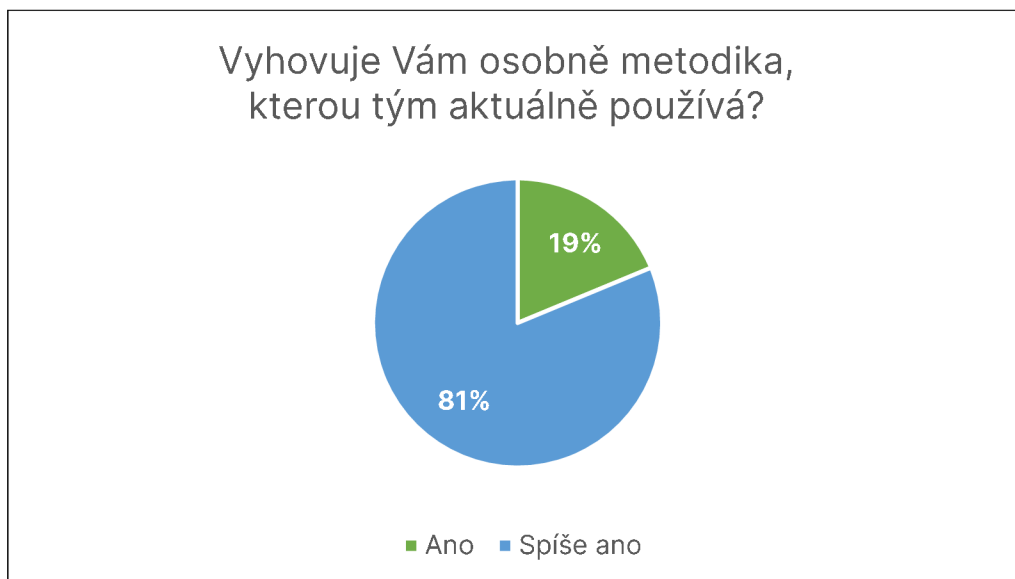


Obr. 11 - Dodržujete v týmu striktně zásady vybrané agilní metodiky?

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Z koláčového grafu na obrázku 11 je vidět, že si 81 % dotázaných členů myslí, že ke striktnímu dodržování agilních metodik nedochází. Takové zjištění je v souladu s přijatým faktem, že metodika není přesný návod. I tak je ale zajímavé, že ve 2 případech se členové stejného týmu v odpovědi neshodli.

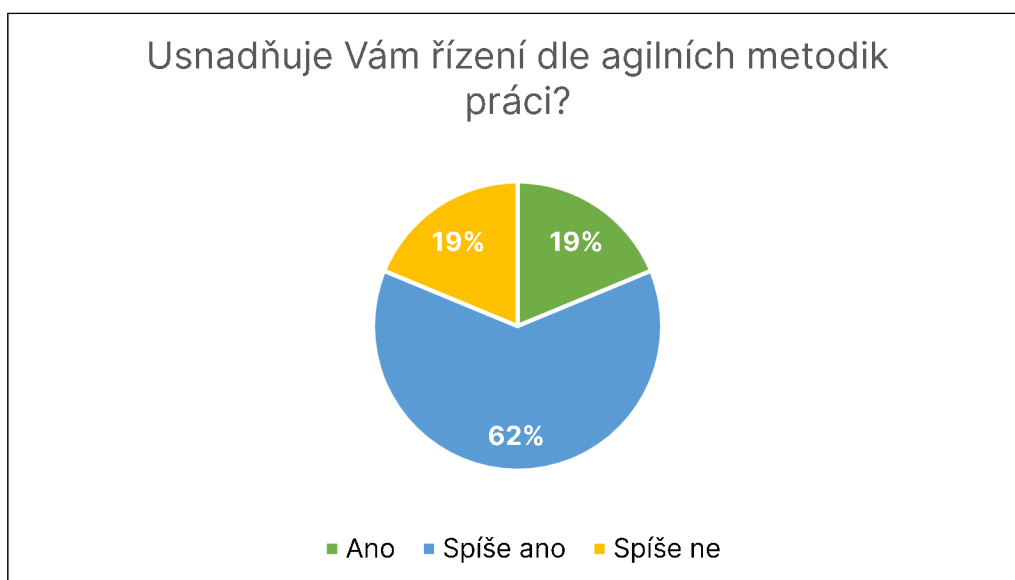
Následující otázka se zaměřila na to, zda dotázaným členům vyhovuje agilní metodika, ve které jejich tým pracuje. Ukázalo se, že pouze 19 % všech dotázaných členů metodika zcela vyhovuje. Zbýlých 81 % tedy stále lehce principy agilních metodik odmítá, či s ní nesdílí stejné názory. Takové rozložení odpovědí bylo očekávané. Dosáhnout pod vlivem agilních metodik absolutní spokojenosti je velice obtížné a spoustu lidí může trápit více, či méně aspektů, které takové řízení vývoje skýtá.



Obr. 12 - Vyhovuje Vám osobně metodika, kterou tým aktuálně používá?

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Dotazníkové šetření dále pokračovalo průzkumem, zda týmem používaná agilní metodiky usnadňuje práci jednotlivcům. Zde už se začaly více objevovat výsledky značící nespokojenost.



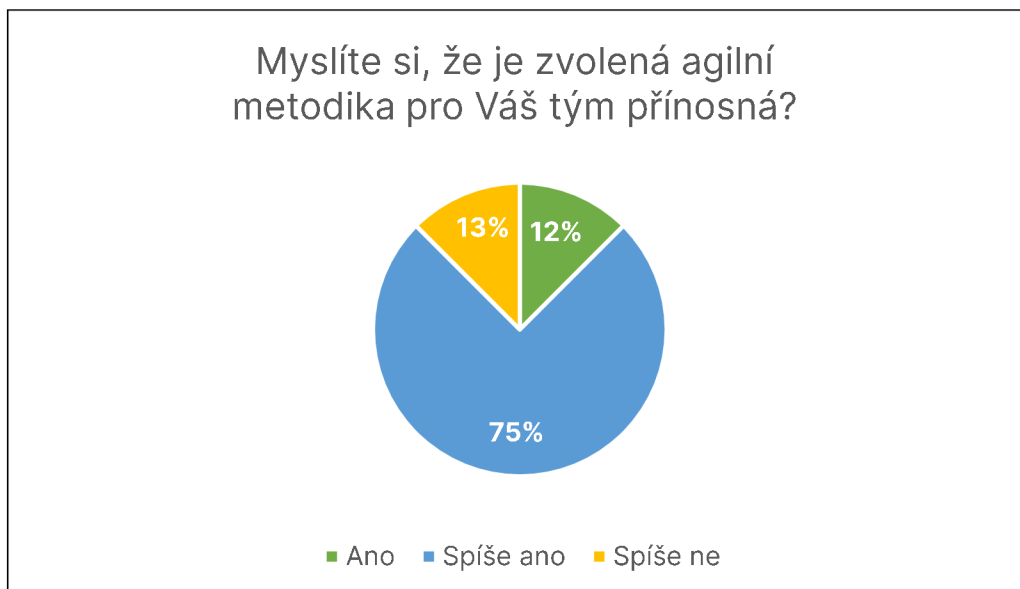
Obr. 13 - Usnadňuje Vám řízení dle agilních metodik práci?

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Obrázek 13 zřetelně ilustruje že 19 % respondentů z řad členů tvrdí, že jim agilní metodika spíše práci ztěžuje, než usnadňuje. Při tom v předchozí otázce bylo vidět, že zvolené agilní metodiky vyhovují v rámci možností všem členům. Vznikl tedy zajímavý kontrast, kdy někomu vyhovuje zvolená agilní metodika, ale nemyslí si, že by mu znatelně usnadnila jeho práci.

I přes tuto skutečnost, vhodnou k zamyšlení, 19 % pocituje, že mu metodika bez debaty usnadňuje pracovní činnost a 62 % to vidím velice podobně, pouze s menšími výhradami.

Další otázka, zda si členové myslí, že je zvolená agilní metodika celkově přínosná pro jejich tým, dosáhla následujících výsledků pozorovatelných na obrázku 13.



Obr. 14 - Myslíte si, že je zvolená agilní metodika pro Váš tým přínosná?

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Celkem 75 % členů má pocit, že Kanban, potažmo Scrum nebo Scrumban, je pro jejich tým v rámci možností přínosem. 13 % pak zastává názor, že agilní metodika má v spíše negativní vliv na tým. 12 % je pak přesvědčeno, že zvolená metodika má pouze pozitivní dopad na celý tým.

V kontrastu k výsledkům předchozí otázky je vidět, že i když některým jednotlivcům agilní přístup spíše práci ztěžuje, tak z pohledu týmu jako celku metodika plní svůj účel dobře.

Aby však analýza neskončila pouze u dat, která hovoří neurčitě o spokojenosti a nespokojenosti, členové byli požádáni, aby uvedli jednu výhodu a nevýhodu pro danou metodiku, kterou jejich tým využívá. Tím je možno identifikovat přesně problémy, které členy trápí, a také zjistit, čeho si naopak si u daných metodik členové naopak cení.

Zaznamenané výhody jednotlivých metodik byly zaneseny do tabulek číslo 6 a 7. Podobné odpovědi byly spojeny vždy v jednu logickou myšlenku.

Tab. 6 – Výhody Kanbanu uvedené členy týmů

Kanban
Možnost rychlé reakce na aktuální vývoj situace
Dobrý přehled o tom, co kdo v týmu dělá a na čem se bude pracovat, jak pro členy, tak pro PO
Průběžné dodávání menší částí
Agilita / Flexibilita
Vidím, které úkoly kolega řeší a mohu mu včas pomoci
Nějaká metodika je lepší, než žádná metodika
Výhodný pro platformní tým, kde nebývá ucelená práce na vývoji
Přenesení odpovědnosti dovnitř týmu

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Jelikož většina členů označila za týmem používanou agilní metodiku Kanban, bylo pro něj zachyceno nejvíce výhod. Všechny odpovědi rámcově zapadají do teorie dané metodiky. I tvrzení, že nějaká metodika je lepší než žádná se shoduje s faktem, který byl popsán Josefem Myslínem v prvním odstavci na začátku teoretické části: bez použití jakékoliv metodiky řízení vývoje je, i se špičkovým týmem vývojářů, designerů a grafiků, velmi malá šance na to, aby byl projekt dotažen do zdárného konce.

Tab. 7 - Výhody Scrumu uvedené členy týmů

Scrum
Přidělování jednotlivých úloh určitým lidem
Přehledné sledování stavu rozpracovanosti
Možnost včasné reakce a převzetí iniciativy
Nedokážu posoudit
Informovanost napříč týmem

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Tabulka výhod Scrumu je sice méně obsáhlejší, ale i přes to většina výroků ze seznamu opět ctí teorii dané agilní metodiky. Jediné tvrzení, které se jeví lehce rozporuplně je možnost včasné reakce. Scrum je postaven na sprintech, které by neměly narušovat žádné náhlé změny a požadavky. Tudíž možnost včasné reakce úplně do tohoto pravidla nezapadá. Fakt, že byla tato odpověď zachycena může utvářet dojem, že zaměstnanec není plně seznámen se zásadami Scrumu, či jeho interpretace dané výhody byla formulována poněkud nešikovně. Pouze v případě krátkých týdenních sprintů by se dalo hovořit o možnosti včasné reakce, ale i přes to jen velmi okrajově.

K popsaným kladům se nyní připojí i nevýhody, které členové popsali ve svých odpovědích. Jádro jejich znění bylo zaznamenáno opět formou tabulek s čísly 8 a 9.

Tab. 8 - Nevýhody Kanbanu uvedené členy týmů

Kanban
Občasný chaos
Správné použití vyžaduje čas a zkušenosti
Je to „agilní“ metodika
Kontinuální práce bez oddechu
Velké množství schůzek zabírá dost času, přičemž některé schůzky by nemusely probíhat tak často
Příliš mnoho změn
Každý pracuje sám za sebe a občas je těžké najít náhradu
Náročnější úkoly se odbavují pomaleji
Než vytvořím úkol do systému, měl bych ho už dávno hotový.

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Zmíněné nevýhody metody Kanban opět velmi korespondují s negativy, která byla zmíněna v teoretické části. Správné použití Kanbanu, ale i ostatních agilních metodik, závisí na zkušenostech, které se tvoří postupem času. Některému týmu může stačit měsíc, další tým se ale například ani za rok nemusí plně sžít s danou metodikou a stále bude při jejím používání přítomen chaos, který zpomalí práci.

Zmíněná vysoká četnost ceremonií, které se se opakují s poměrně krátkou periodou, je taky známý problém, se kterým se potýká spousta lidí. Ani JCS tedy nebude výjimkou.

Zajímavým bodem je i tvrzení, že každý pracuje na své práci, která se týká pouze určité oblasti a je pak náročné při výpadku člověka najít adekvátní náhradu. Pokud tým postaven opravdu způsobem, kdy je každý expertem pouze v určité oblasti, pak snaha metodiky o informovanosti celého týmu tvoří jen minimální přínos.

Je to „agilní“ metodika. Toto stanovisko autora je těžké vyložit. Soudě ale dle použití uvozovek, nejspíše naráží na fakt, kdy mu nejspíše Kanban spíše svazuje ruce, než aby mu pomáhal se rychle adaptovat na aktuální situaci.

Tab. 9 - Nevýhody Scrumu uvedené členy týmů

Scrum
Zahlcení informacemi z různých informačních kanálů
Velké úkoly je nutno dělit na menší dílky
Nedokážu posoudit
Náročnější úkoly se odbavují pomaleji

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Velmi skloňovaným tématem v případě Scrumu bylo pak vytváření ticketů/tasků do systému. Při popisu komplexního úkolu může korespondenční práce zabrat mnoho drahocenného času, který by mohl být využit jinde. S tím se pojí pravidlo, které říká, že by se měly tvořit co nejmenší úkoly, aby šli rychle odbavovat a tím se tvořil transparentní postup práce. V některých případech se ale stává, že velký úkol nelze rozdělit. V tu chvíli se papírově zpomaluje tok práce a zákazník pak může nabývat dojmu, že pracovní nasazení upadá.

Zahlcení informacemi je také naprosto adekvátní poukázání na problém. Při práci může zaměstnanec využívat více informačních kanálů a v některých případech mu odepisování a vstřebávání všech dat od kolegů a vedení může zabrat nebývale mnoho času a zbytečně ho vyčerpávat při odpovídání na všechna sdělení.

Bližší pohled členů na Scrum a Kanban ze strany výhod a nevýhod byl analyzován. Pro Scrumban, ale tento proces nenastal. Je to z toho důvodu, že pouze 1 člen uvedl, že tým používá kombinaci agilních metodik a do pole výhod a nevýhod uvedl, že ze své pozice nedokáže posoudit situaci. Z tohoto důvodu tedy nemohla proběhnout žádná analýza.

V posledních dvou otázkách tohoto bloku měli členové týmů uvést, jaké primární nástroje tým používá pro sledování pracovního postupu a jaký nástroj tvoří hlavní komunikační kanál mezi členy a ven z týmu.

Všechny týmy až na Tým 10 používají pro sledování pracovního postupu webový softwarový nástroj JIRA vyvíjený společností Atlassian. Ta ve svém prostředí podporuje jak Scrum, tak i Kanban, či případné kombinace. Tým 10 používá v tomto ohledu primárně sadu webových nástrojů zvanou Phabricator. Ten také podporuje všechny zmíněné metodiky. Není však tak propracovaný a svým vizuálem budí dojem zastaralého SW. Bohužel tento tým je životně závislý na spolupráci s ostatními sesterskými společnostmi spadajícími pod JABLOTRON GROUP a.s., které taktéž využívají Phabricator, a nemá dostatečné slovo, či kapacitu na to, aby přešel také na JIRU nebo jiný propracovanější SW.

V oblasti nástrojů pro hlavní informační tok jednohlasně zvítězila uzavřená obchodní komunikační platforma Slack. V jádru velmi připomíná aplikace Whats Up nebo Facebook Messenger. Co se týče ale ostatních nabízených funkcí, v této oblasti už naprosto zastiňuje veřejně dostupné komunikační nástroje. Tvoří přehledná komunikační vlákna a kanály, tudíž lze různé informace třídit a žádný problém není utopen v množství posílaných zpráv. Zaměstnanec, tedy není vystaven problémům, kdy by mu utekly důležité informace.

❖ **Product Owner**

Role PO byla v dotazníkovém šetření zastoupena pouze jedním člověkem. Analýza jeho odpovědí a názorů tedy v tomto případě proběhne bez pomocných ilustrací.

Dotázaný PO nám sdělil, že jeho primární práce se skládá primárně ze zpracování a analýzy požadavků, prioritizace práce a motivování a úzké spolupráce s týmem. Tento popis naprosto koresponduje s teorií PO, i když otázku motivace by spíš měl v tomto případě zastávat Scrum Master, nebo Team Lead.

Spolupracuje pouze s jedním týmem, konkrétně s Týmem 2. V rámci tohoto týmu odpovídali i 2 členové, tudíž bude možno porovnat pohledy více rolí z jednoho celku.

První věc, která dotázaného PO napadla ve spojitosti s agilní metodikou byl kontakt. Tím nejspíše naráží na, buď na velké množství koordinačních schůzek nebo na nutnost aktivní komunikace.

S týmem používá metodiku Kanban. Vzhledem k tomu, že se členové Týmu 2 v této otázce neshodli znamená, že tedy nejspíše jeden z členů v této otázce nemá jasno. Všichni se ale shodli, že striktně nedodrží zásady dané agilní metodiky. To je, jak už bylo několikrát zmíněno, správný přístup.

V otázkách týkajících se přínosu agilních metodik respondent tvrdí, že mu agilní metodiky v rámci možností jsou pro tým přínosem a jemu samotnému usnadňují práci. Fakt, že v těchto otázkách ale neodpověděl čistě ano značí, že v některých situacích naráží na jisté překážky, či omezení, která mu nevyhovují.

Při bližší specifikaci kladů a záporů Scrumu i Kanbanu uvedl, že jako pozitivum bere kontakt s lidmi. Co se týče negativ, zde odpověděl, že ho žádná nenapadají. Vzhledem k předchozímu vyjádření v otázkách přínosu agilních metodik, bylo očekáváno, že bude uveden konkrétní příklad negativ. Ovšem nestalo se tak.

V poslední řadě PO uvedl, že jako primární nástroj pro sledování pracovního postupu týmu využívá nástroj JIRA a primární kanál pro komunikaci s týmem mu poskytuje aplikace Slack.

❖ **Team Lead**

Stejně jako v případě role PO se podařilo získat pouze jednu odpověď z pohledu této role. Avšak z pohledu skladby JCS se jedná o přibližně třetinu zastoupení této role. Tudiž i přítomnost jednoho Team Leada (TL), lze považovat za úspěch.

Respondent v roli TL uvedl, že pracuje s více jak jedním týme. Mezi týmy, se kterými je v bližším kontaktu patří: Tým 2, Tým 3, Tým 5 a Tým 10. Vzhledem k tomu, že z týmů 2 odpovídal více jak jeden člen a dokonce i PO, bylo v jednom případě dosaženo kompletního pokrytí odpovědí všech rolí pro jeden určitý tým. Týmy 3 a 10 v tomto ohledu ale také převýšili očekávání.

V otázce prvotní myšlenky při zmínění agilních metodik se TL vybavil Scrum. Tato odpověď krátce koreluje s následující otázkou, kdy uvedl, že nejraději pracuje pod záštitou metodiky Scrum. Je tedy vidět, že TL má metodiku Scrum nejspíše pevně zažitou a proto si agilní metody ihned spojuje s tímto názvem. S týmy nedodržují striktně metodiky, se kterými pracují. Opět myšlení správným směrem.

Dále TL ve svých názorech zmínil, že je řízení dle agilních metodik pro týmy z velké části přínosem. Nebyl však vyjádřen plný souhlas s tímto stanoviskem, tudíž s velkou pravděpodobností zaznamenává jistá omezení. Na druhou stranu uvedl, že jeho osobě v roli TL agilní metodiky bezpodmínečně usnadňují práci.

Mezi výhody metodik Scrum a Kanban uvedl: soustředění na zvyšování hodnoty produktu, aktivní týmová spolupráce, názorová otevřenost, postupné dokončování projektů.

Jako nevýhody naopak vnímá: těžké začátky při práci v agilu, změna myšlení, nutnost průběžné motivace k uplatňování principů metodik.

Skutečnost, že respondent uvedl více pozitiv a negativ svědčí o tom, že si je dobře vědom, co řízení dle agilních metodik obnáší a snaží se s vědomím pro a proti pracovat.

Zajímavé stanovisko TL přišlo v otázce na nástroje sloužící ke sledování postupu a komunikace. Uvedl, že interakce celého týmu má pro něj přednost

před nástroji ke sledování pracovního postupu. Zmíněnou JIRU nebo Phabricator tedy bere spíše jako doplňky, než primární zdroj informací.

Co se týče komunikačního kanálu, opět byla uvedena aplikace Slack. Na stejné úrovni však byl zmíněna i fyzická komunikace.

❖ Jiná role

Do kategorie jiných rolí se identifikovali celkem 3 osoby. Jejich popis svých primární pracovní pozic byl: CEO a CTO, SW architekt a Support.

Jelikož tyto role nemají prakticky nic společného, je nutné část jejich odpovědí analyzovat jednotlivě.

Tabulka číslo 10 přehledně shrnuje prvotní myšlenku všech pozic při zmínění agilních metodik.

Tab. 10 – Prvotní pohled na agilní metodiky jiných rolí

Pozice	Prvotní myšlenka
CTO a CEO	Focus na produkt a zákazníka
SW architekt	Vývojář vždycky na prvním místě, je to o lidech
Support	Rychlá reakce na změny

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Dá se říci, že z pohledu ostatních rolí se prvotní reakce ztotožňují s určitými zásadami agilních metodik.

Za nejvhodnější metodiku označil SW architekt a Support shodně Scrumban, tedy kombinaci. CEO/CTO zde zastává názor, že zde velmi záleží na typu projektu, tedy někdy preferuje Scrum a jindy Kanban. Všichni se však shodli, že není rozhodně důležité striktně dodržovat zásady zvolených agilních metod.

V otázce přínosu agilní metodiky týmům CEO/CTO odpověděl s jistotou, že tento způsob je přínosný pro týmy a i jeho osobně zásadně usnadňuje práci. SW architekt a Support zde zvolili možnost, kdy spíše souhlasí s názorem, že jsou agilní metodiky přínosné pro tým, tak pro ně samotné.

Výsledky následných otázek výhod a nevýhod metod Scrum a Kanban zobrazuje tabulka číslo 11. Zde už nebylo nutné rozlišovat odpovědi dle pozice.

Tab. 11 – Výhody a nevýhody metod Scrum a Kanban dle jiných rolí

Výhody	Nevýhody
Přizpůsobivost	Tvorba náboženství
Kontinuální dodávání produktu	Někdy je lepší použít Waterfall
Možnost rychlé reakce	Daná posloupnost úkolů

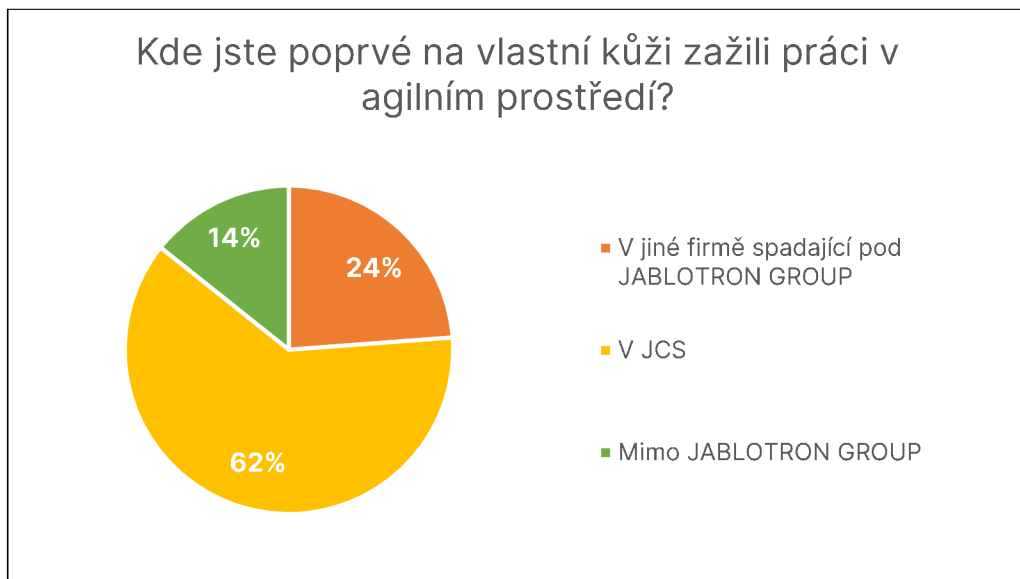
Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Podobné výhody a nevýhody už zazněli v případě členů týmů. Je ale dobře, že jsou v tomto ohledu všichni shodní. To zamezuje pak tvorbě nečekaných konfliktů názorů. Zajímavá byla narážka na použití Waterfallu, kdy chtěl člověk nejspíše poukázat na skutečnost, že někdy agilní metodiky díky častému plánování uměle prodlužují zcela banální úkony.

Sledování pracovního postupu týmů u všech 3 osob zajišťuje JIRA a hlavní komunikační kanál, jako v případě všech ostatních rolí, tvoří Slack.

❖ **Doplňující otázky**

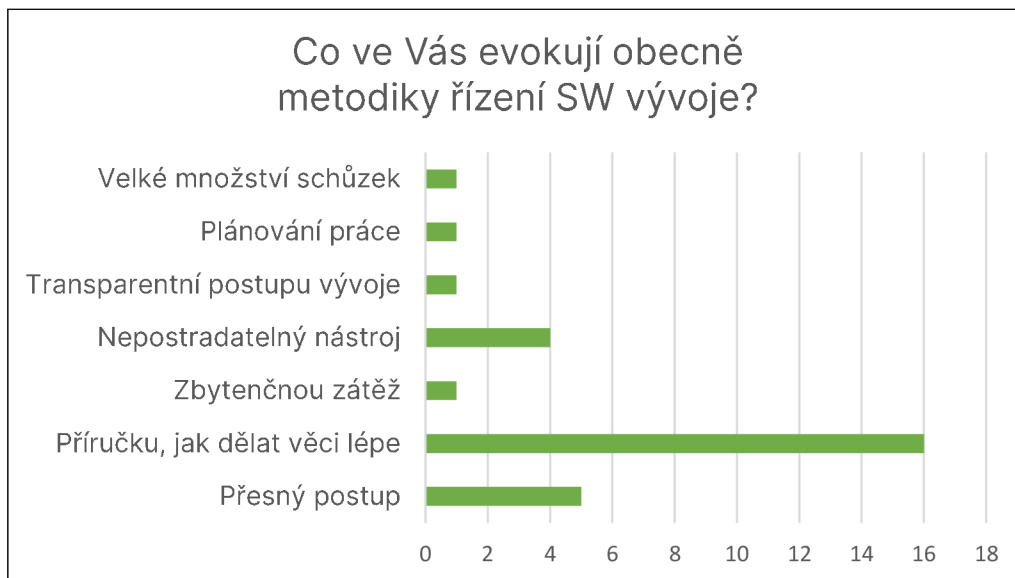
Všem dotázaným bez ohledu na jejich pozici, či roli ve společnosti byla položena otázka, kde se poprvé setkaly s prací v agilním prostředí. 62 % všech lidí odpovědělo, že místem prvního setkání byla právě společnost JCS. Dalších 24 % se s agilem setkala v jiné společnosti spadající pod JABLOTRON GROUP a.s. Pouze u 14 % všech respondentů došlo k tomuto setkání v jiné firmě mimo Jablotron. Výsledky dále ilustruje graf na obrázku číslo 15.



Obr. 15 - Kde jste poprvé na vlastní kůži zažili práci v agilním prostředí?

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Všichni byli také již dotázáni na prvotní myšlenky ohledně agilních metodik. Pro vytvoření uceleného obrázku byla však ještě položena otázka, co v dotázaných obecně evokují metodiky řízení vývoje, ať už klasické, či agilní.



Obr. 16 - Co ve Vás evokují obecně metodiky řízení SW vývoje?

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Jak je vidět na obrázku 16, nejčastější odpovědí na tuto otázku bylo tvrzení: Příručka, jak dělat věci lépe. Následoval přesný postup a nepostradatelný nástroj. Omezeně se pak objevily specifické názory jednotlivců.

Předposlední otázka se týkala vlivu agilních metodik na dovednosti všech dotázaných. Koncentrovaný přehled všech odpovědí lze pozorovat v tabulce 12.

Tab. 12 – Ovlivnění dovedností zaměstnanců agilními metodikami

Scrum
Detailnější rozpad úkolů
Plánování, prioritizace
Zlepšení odhad náročnosti
Spolupráce s týmem
Informovanost napříč týmem
Ne / Nevím
Soustředění se na jednu věc
Rozšíření kompetencí
Svoboda a sebeorganizace
Flexibilita
Kreativita
Efektivnější komunikace
Empatie

Zdroj: vlastní tvorba dle výsledků dotazníkového šetření

Je vidět, že škála uvedených dovedností úzce navazuje na principy metodik. Tyto získané, či vylepšené dovednosti však dokážou jednotlivcům zlepšit celkově kvalitu života i mimo práci. Z toho lze soudit, že práce v agilní metodice je z tohoto hlediska velkým přínosem.

Poslední otázka, která se v dotazníkovém šetření objevila, se snažila zjistit, zda lidé znají i jiné agilní metodiky kromě Scrumu a Kanbanu, tedy jestli se v této

oblasti například sami více vzdělávali, či nikoliv. 12 lidí uvedlo, že žádné jiné neznají. Od ostatních 9 osob pak společně uvedlo následující metodiky: Extrémní programování, SAFe, LeSS, TDD, FDD, Lean. Lze tedy soudit, že 9 z 21 lidí se o agile zajímá nad rámec svým povinností a má tedy širší povědomí o více způsobech řízení projektů. Zkratky jednotlivých metodik jsou uvedeny v seznamu zkratk na začátku této práce.

3 Doporučení

Jelikož zaznamenané odpovědi přišly jak od standartních členů týmů, specializovaných rolí, tak i z top managementu, bylo možné vytvořit dostatečně kvalitní popis vlivu agilního prostředí na celou firmu.

Celkově z průzkumu vyplynulo, že většina oslovených osob má pozitivní vztah k práci v agilu a je si vědoma, co tato problematika obnáší, jaká vidí omezení, a co naproti tomu firmě, a lidem v ní, agilní prostředí poskytuje za výhody. Zároveň se ve firmě vyskytuje menšina, která není plně spokojena s aktuální implementací agilních metod a tvrdí, že jim spíše práci komplikuje. V rámci dalšího rozvoje v této oblasti by bylo dobré tyto lidi oslovit, společně konzultovat všechny problémy, které těmto osobám kazí celkový dojem z práce v agilu a zkusit nalézt řešení všech závad, které dotýčným vadí.

Dále bylo také díky více odpovědím lidí pracujících ve stejných týmech zjištěno, že někteří zaměstnanci nejspíše špatně rozeznávají jednotlivé metodiky, nebo se jim nedostalo v této oblasti potřebných informací. To lze ale vždy eliminovat průběžným vzděláváním a tokem informací k lidem, kteří je potřebují.

Nepřehlédnutelným faktem je i skutečnost, že velká část dotázaných zaměstnanců uvedla, že se s agilními metodami setkali poprvé právě v JCS, nebo v jiné společnosti spadající pod JABLOTRON GROUP a.s. Dalo by se říct, že co se týče vzdělávacího procesu v oblasti agilního vývoje SW, Jablotron odvádí v oblasti Liberecka chvályhodnou práci a tvoří základ kvalitních zaměstnanců. I tak bude ale potřeba stále udržovat povědomí o agilních přístupech ve svém osazenstvu a rozvíjet schopnosti lidí, aby se v agilním prostředí lépe orientovali a používali jej ke svému prospěchu. Zajímavým pokusem v tomto ohledu by například byla možnost, kdyby si jednotlivé vývojové týmy mohly vyzkoušet práci dle jiné nezažité agilní metodiky používané k řízení vývoje SW. Mohlo by to přinést nový pohled na věc, či zjištění, že již používané metodiky jsou tím nevhodnějším řešením.

4 Přínosy a ekonomické zhodnocení práce

I přes fakt, že je použití agilních metodik ve vývoji SW desktopových i mobilních aplikací často skloňovaným tématem, najít kvalitní literaturu v českém jazyce zabývající se touto problematikou je velice náročné. Jedinou možností, jak načerpat informace, je použití zahraničních zdrojů. Tato závěrečná práce proto ve své teoretické části přinesla přehledné shrnutí všech důležitých aspektů těch nejčastěji používaných agilních metodik, se kterými se můžeme v rámci mnoha domácích i zahraničních firem zabývajících se vývojem SW setkat. Tvoří prvotní náhled do každé z metodik a snaží se, aby měl čtenář možnost pochopit, jak metodika ve svém základu funguje, jaké benefity nabízí, a na co si dát případně při jejím používání pozor.

Pro lepší pochopení, proč jsou agilní metodiky řízení tak populární v moderním vývoji, práce zahrnula do své teorie i popis klasických metodik řízení vývoje, které také přehledně popisuje a dává možnost čtenáři si udělat objektivní názor na oba směry, kterými se řízení SW vývoje snaží, či snažilo ubírat.

Praktická část naopak přinesla cenný pohled do ryze české firmy, která se díky agilním metodikám stále rozvíjí a tvoří kvalitní produkty od lidí pro lidi. Po dotazníkovém šetření bylo jasné, že popsané principy agilních metodik opravdu fungují, a že zmíněná omezení daných metod se opravdu čas od času projevují a tvoří jistou bariéru. I přes to pozitivní vliv značně převyšuje ten negativní.

Celkově by tato závěrečná práce mohla být označena jako jednoduchý průvodce základními principy agilních metodik řízení vývoje SW s praktickou ukázkou, a proč je dobré se v dnešní době vydat právě tímto směrem.

5 Závěr

Metodiky řízení vývoje SW jsou stále, v kontrastu k jiným lidským objevům, poměrně novým způsobem myšlení, jelikož se začaly vytvářet až přibližně v polovině 20. století s příchodem počítačů. Postupná potřeba vývoje SW pro nové stroje vedla k zavedení určitých postupů a metod k zefektivnění práce. V 70. letech Winston W. Royce představuje světu první moderní dedikovanou metodiku pro vývoj softwaru zvanou Waterfall. Jedná se o průlomový objev v této oblasti, který odstartoval éru takzvaného klasického, či tradičního přístupu k vývoji SW.

Postupem času, ale vznikla potřeba vymyslet nové metody, protože stávající, díky zvyšování náročnosti projektů v oblasti IT, přestávaly být vhodné a ztrácely na relevanci. V 90. letech tedy světlo světa spatřují nové agilní metodiky. V jádru stále přechovávají zbytky svých předků, zbytek však tvoří úplně jiný přístup k problematice vývoje. Do dnešní doby existují desítky agilních metod, které sice od sebe často dědí různé principy, ale každá se zaměřuje na vylepšení jiné části vývojového prostředí. Není tedy divu, že ve stejné společnosti může být implementováno více agilních metodik.

Začátek teoretické části této závěrečné práce se zaměřil na objasnění obecné problematiky metodik řízení SW vývoje. Byla přijata fakta od autorů různých publikací, o která se dále opírala zbylá část teorie a následně i praktická část. Jedním z nejdůležitějších stanovisek, která byla zmíněna říká, že nutnost použít metodiku k řízení vývoje SW je nezbytným krokem, aby byl zaručen úspěch projektu. To jestli je už metodika vhodná, či není, zde řešeno není. Dále bylo přijmut důležitý fakt, že jakákoliv metodika řízení není v žádném případě přesným návodem, jak k vývoji přistupovat, protože SW se v zásadě nevyrábí, ale vyvíjí. Okolnosti tedy tvoří velké množství proměnných, které je těžké předpovídat.

Zbylá část první poloviny teoretické části byla věnována celkovému popisu klasických metodik a dvou vybraných zástupců z této oblasti, jmenovitě metodě Waterfall a Spirálovému modelu. Shrnutí klasických metod tedy stanovilo, že jejich přístup je založen na jednosměrném toku informací, které jsou na začátku celého procesu pevně dané a detailně popsány. Dodatečná implementace zde není téměř v žádném případě možná. Byl vytvořeno koncentrovaná shrnutí, které charakterizovalo vybrané klasické metody, popsalo jejich silné stránky a slabiny a vydalo doporučení, pro jaké projekty a za jakých okolností je dobré tyto metody řízení využít.

V druhé polovině teorie padlo zaměření na agilní metodiky, které bylo strukturované identicky ke klasickým metodám, tedy obecné uvedení do problematiky, kdy se ustanovila fakta, že agilní metodika je založená primárně na menších iteracích, během kterých vývoj aktivně komunikuje se zákazníkem, který dostává produkt postupně po menších celcích, čímž dostává možnost projekt průběžně hodnotit a případně zadat změny, které bude nutné zpracovat, aby výsledek co nejvíce odpovídal daným požadavkům. Agile má však i určitá pravidla, které je dobré ctít, ale je možné si je upravit podle potřeby.

Jako příklad agilních metodik byl uveden Scrum a Kanban, jelikož jsou nejvíce skloňovanými a používanými nástroji v této oblasti. Bylo uvedeno jejich jádro, principy a potenciální plusy a mínusy. V neposlední řadě pak doporučení o použití v SW vývoji. Zároveň bylo avizováno, že tyto metody se dokážou vzájemně doplnit, čím vzniká kombinace zvaná Scrumban.

Praktická část této práce se věnovala reálnému využívání metod Scrum a Kanban. Agilní teorie sice slibuje světlé zítřky, praxe bývá často ale krutou realitou. Bylo tedy rozhodnuto, že se pomocí dotazníkového šetření provede analýza v rámci firmy, která se zabývá vývojem SW desktopových aplikací a aktivně využívá agilní metodiky Scrum a Kanban. Osloveným subjektem se stala společnost JABLOTRON CLOUD Services s. r. o. Osloveno bylo celkem 67 zaměstnanců zastávajících různé pracovní pozice a firemní role, z čehož celkem 21 se do šetření aktivně zapojilo.

Zaznamenané odpovědi byly rozděleny do jednotlivých skupin a pečlivě analyzovány s pomocí grafů a tabulek zachycujících individuální názory. S takovým to přístupem si pak bylo možné vytvořit kompletní obrázek o vlivu agilního prostředí na celou firmu. Závěrem lze říci, že většina oslovených osob má pozitivní vztah k práci v agilu a je si vědoma, co tato problematika obnáší, jaká má omezení, a co naproti tomu firmě, a lidem v ní, poskytuje za výhody. Povědomí stávajících zaměstnanců o agilním řízení bylo však z velké části zapříčiněno právě společností JCS, kdy svým aktivním vzděláváním a motivováním lidí k práci v agilním prostředí vytváří velmi výkonný firemní organismus, který pracuje efektivně a tvoří kvalitní softwarové produkty, které následně zlepšují život lidem po celém světě.

Seznam použité literatury

- MYSLÍN, Josef, 2016. Scrum: průvodce agilním vývojem softwaru. Brno: Computer Press. ISBN 978-80-251-4650-7.
- MAVURU, Indusree, 2018. Traditional vs. Agile Software Development Methodologies. KpiPartners [online]. KPI Partners News Team [cit. 2023-02-08]. Dostupné z: <https://www.kpipartners.com/blog/traditional-vs-agile-software-development-methodologies>
- WHYATT, Tom, 2023. The Pros and Cons of Waterfall Software Development. One Beyond [online]. [cit. 2023-01-27]. Dostupné z: <https://www.one-beyond.com/pros-cons-waterfall-software-development/>
- WALKER ROYCE, Winston, 1970. MANAGING THE DEVELOPMENT OF LARGE SOFTWARE SYSTEMS. The institute of Electrical and Electronics Engineers. TRW, 1-9. Dostupné z: [doi:https://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf](https://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf)
- PRATT, John, 2021. Rozdíl mezi metodikou Waterfall a Agile. Strephonsays [online]. [cit. 2023-01-05]. Dostupné z: <https://cs.strephonsays.com/waterfall-methodology-and-vs-agile-5581>
- KUMAR PAL, Sayan, 2022. Software Engineering | Classical Waterfall Model [online]. Geeksforgeeks [cit. 2023-01-05]. Dostupné z: <https://www.geeksforgeeks.org/software-engineering-classical-waterfall-model/>
- ŠIMŮNEK, David, 2018. Jaký je rozdíl mezi Waterfall a Agile přístupem. David Šimůnek [online]. Praha [cit. 2023-01-09]. Dostupné z: <https://www.davidsimunek.com/post/jaky-je-rozdil-mezi-waterfall-a-agile>
- SMARTIQHUB, 2021. What is Waterfall Model? Modified Waterfall Model. SMART IQ HUB [online]. [cit. 2023-01-09]. Dostupné z: <https://www.smartiqhub.com/what-is-waterfall-model-modified-waterfall-model/>
- YAHYA, Norzariyah a Siti Sarah MAIDIN, 2022. The Waterfall Model with Agile Scrum as the Hybrid Agile Model for the Software Engineering Team [online]. Piscataway [cit. 2023-01-11]. ISSN 2770-159X. Dostupné z:

doi:[https://doi.org/10.1109/
CITSM56380.2022.9936036](https://doi.org/10.1109/CITSM56380.2022.9936036)

ART OF TESTING, 2022. Waterfall Model in Software Engineering. Artoftesting [online]. [cit. 2023-01-10]. Dostupné z: <https://artoftesting.com/waterfall-model>

NEHA T, 2020. Waterfall Process Model. Binary Terms [online]. [cit. 2023-01-09]. Dostupné z: <https://binaryterms.com/waterfall-process-model.html>

TUTORIALS POINT, 2013. SDLC - Waterfall Model. Tutorials Point [online]. [cit. 2023-01-11]. Dostupné z: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm

EBY, Kate Eby, 2018. Which Management Style Is Right for You: Top-Down or Bottom-Up Approach?. Smartsheet [online]. [cit. 2023-01-16]. Dostupné z: <https://www.smartsheet.com/top-down-bottom-up-approach>

ALSHAMRANI, Adel a Abdullah BAHATTAB, 2015. A Comparison Between Three SDLC Models Waterfall Model, Spiral Model, and Incremental/Iterative Model. International Journal of Computer Science Issues (IJCSI); Mahebourg. 15(1), 106-111. ISSN 1694-0814. Dostupné z: [doi:https://www.proquest.com/scholarly-journals/comparison-between-three-sdlc-models-waterfall/docview/1660801422/se-2?accountid=17116](https://www.proquest.com/scholarly-journals/comparison-between-three-sdlc-models-waterfall/docview/1660801422/se-2?accountid=17116)

NOTEPUB, 2021. SDLC – Iterative Waterfall Model. In: Notepub [online]. [cit. 2023-04-04]. Dostupné z: <https://notepub.io/notes/software-engineering/software-development-life-cycle/sdlc-iterative-waterfall-model/>

C. REDDY, G., 2021. Spiral Model: Spiral Model – Software Development. Software Testing [online]. [cit. 2023-01-17]. Dostupné z: <https://www.gcreddy.com/2021/06/spiral-model.html>

TECH TARGET, 2019. Spiral model: Definition. Tech Target | Software quality [online]. [cit. 2023-01-18]. Dostupné z: <https://www.techtarget.com/searchsoftwarequality/definition/spiral-model>

NARANG, Mohita, 2022. Spiral Model in Software Engineering. Naukri Learning [online]. [cit. 2023-01-23]. Dostupné z:

<https://www.naukri.com/learning/articles/spiral-model-in-software-engineering/>

BANKS, Frances, 2016. Spiral Model: Software Development For Critical Projects. In: Airbrake [online]. [cit. 2023-04-04]. Dostupné z: <https://blog.airbrake.io/blog/sdlc/spiral-model>

WILLIAM BOEHM, Barry, 1986. A spiral model of software development and enhancement. ACM SIGSOFT Software Engineering Notes. 11, 12-24. Dostupné z: [doi:https://doi.org/10.1145/12944.12948](https://doi.org/10.1145/12944.12948)

KASHYAP, Sandeep, 2020. Traditional vs Agile Project Management Method: Which One is Right for Your Project?. ProofHub [online]. [cit. 2023-02-06]. Dostupné z: <https://www.proofhub.com/articles/traditional-vs-agile-project-management>

DOLEŽAL, Jan, Pavel MÁČHAL a Branislav LACKO, 2012. Projektový management podle IPMA. 2., aktualiz. a dopl. vyd. Praha: Grada. ISBN 978-80-247-4275-5.

ŠOCHOVÁ, Zuzana a Eduard KUNCE, 2019. Agilní metody řízení projektů. 2. vydání. Brno: Computer Press. ISBN 978-80-251-4961-4.

SCRUM.ORG, 2012. What is Scrum?: A better way to work together and get work done. Scrum.org: The Home of Scrum [online]. [cit. 2023-02-25]. Dostupné z: <https://www.scrum.org/resources/what-is-scrum>

SCHWABER, Ken a Jeff SUTHERLAND, 2020. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. Scrum.org.

RAD, Nader K. a Frank TURLEY, 2013. The Scrum Master Training Manual: A Guide to Passing the Professional Scrum Master (PSM) Exam. Management Plaza. ISBN 978-90-821147-1-3.

CHANDANA, 2022. Scrum Project Management: Advantages and Disadvantages. Simply Learn [online]. [cit. 2023-03-01]. Dostupné z: <https://www.simplilearn.com/scrum-project-management-article>

KANBAN TOOL, 2014. Kanban Software Development: Kanban boards for software development teams. Kanban tool [online]. [cit. 2023-03-01]. Dostupné z: <https://kanbantool.com/kanban-software-development>

ATLASSIAN, 2014. Kanban: How the kanban methodology applies to software development. Atlassian [online]. [cit. 2023-03-01]. Dostupné z: <https://www.atlassian.com/agile/kanban>

BENNETT, Jason a Jennifer BOWEN, 2018. Lean: 8-in-1 book ultimate collection - lean startup, lean analytics, lean enterprise, kaizen, six sigma, agile project management, kanban, scrum. England: Independently published. ISBN 9781720216292.

ATLASSIAN, 2021. Continuous delivery starts on the Jira kanban board. In: Atlassian [online]. [cit. 2023-03-05]. Dostupné z: <https://www.atlassian.com/software/jira/features/kanban-boards>

TSONEV, Nikolay, 2019. Kanban Backlog: The Least Explored Corner of Kanban. Kanbanize [online]. [cit. 2023-03-05]. Dostupné z: <https://kanbanize.com/blog/kanban-backlog/>

NOVKOV, Alex, 2019. The Kanban Roles You've Probably Never Heard Of. Kanbanize [online]. [cit. 2023-03-05]. Dostupné z: <https://kanbanize.com/blog/kanban-roles/>

JADEALM, 2021. Comparison of Different Agile Methodologies: Pros and Cons. Jadealm [online]. [cit. 2023-03-06]. Dostupné z: <https://jadealm.com/blog/comparison-of-different-agile-methodologies-pros-and-cons/>

KAUR, Arshpreet, 2022. Kanban vs Scrum: Choose the Right Agile Method for Your Product. Net solutions [online]. [cit. 2023-03-06]. Dostupné z: <https://www.netsolutions.com/insights/kanban-vs-scrum/>

INSIGHT SPOTTER, 2020. Does Kanban have a Product Owner?. Insight Spotter [online]. [cit. 2023-03-06]. Dostupné z: <https://insightspotter.com/does-kanban-have-a-product-owner/>

Seznam příloh

Příloha A: Dotazníkové šetření

Příloha A: Dotazníkové šetření

Dotazníkové šetření k závěrečné práci

Toto dotazníkové šetření slouží k analýze používání agilních metodik v reálném prostředí. Odpovědi respondentů jsou anonymní, z důvodu zachování pravidel GDPR, a budou použity jen a pouze k vypracování praktické části mé závěrečné práce bakalářského studia.

Všem budoucím respondentům v tomto šetření předem velice děkuji.

Jan Otmar - Ekonomická fakulta Technické univerzity v Liberci (Informační management)

* Označuje povinnou otázku

1. Popis Vaší pracovní pozice *

Stačí zjednodušeně

2. Vaše primární firemní role *

Označte jen jednu elipsu.

- Člen týmu *Přeskočte na otázku 3*
- Product Owner *Přeskočte na otázku 15*
- Team Lead *Přeskočte na otázku 26*
- Jiná role *Přeskočte na otázku 37*

Člen týmu

3. Uveďte název týmu, ve kterém pracujete *

Reálné názvy týmů nebudou v BP uvedeny. Finální označení bude ctít formát "Tým [1-X]".
Do této dopovědi prosím neuvádějte týmy typu POs a Testers.

4. Jaká je hlavní pracovní náplň Vašeho týmu? *

Označte jen jednu elipsu.

- Vývoj
- Údržba
- Podpora
- Jiné: _____

5. Napište první věc, která Vás napadne, když se řekne: "Agilní metodika" *

6. Jakou agilní metodiku Váš tým aktuálně používá? *

Označte jen jednu elipsu.

- Scrum
- Kanban
- Kombinace (Scrumban)
- Jiné: _____

7. Dodržujete v týmu striktně zásady vybrané agilní metodiky? *

Označte jen jednu elipsu.

- Ano
- Ne

8. Vyhovuje Vám osobně metodika, kterou tým aktuálně používá? *

Označte jen jednu elipsu.

- Ano
 Spíše ano
 Ne
 Spíše ne

9. Usnadňuje Vám řízení dle agilních metodik práci?

Označte jen jednu elipsu.

- Ano
 Spíše ano
 Ne
 Spíše ne

10. Myslíte si, že je zvolená agilní metodika pro Váš tým přínosná? *

Označte jen jednu elipsu.

- Ano
 Spíše ano
 Ne
 Spíše ne

11. Uveďte jednu výhodu týmem aktuálně používané metodiky. *

12. Uved'te jednu nevýhodu týmem aktuálně používané metodiky. *

13. Jaký nástroj používá Váš tým primárně pro sledování pracovního postupu? *

Označte jen jednu elipsu.

JIRA

Phabricator

Jiné: _____

14. Jaký hlavní komunikační nástroj používá Váš tým ? *

Označte jen jednu elipsu.

Slack

MS Teams

E-mail

Jiné: _____

Přeskočte na otázku 46

Product Owner

15. Máte na starosti více než 1 tým? *

Označte jen jednu elipsu.

Ano

Ne

16. Uvedte tým/y, se kterým/i se podílíte na řízení projektu/ů. *

Reálné názvy týmů nebudou v BP uvedeny. Finální označení bude ctít formát "Tým [1-X]".
Do této dopovědi prosím neuvádějte týmy typu POs a Testers.

17. Napište první věc, která Vás napadne, když se řekne: "Agilní metodika" *

18. Jakou agilní metodiku nejraději používáte k řízení projektů? *

Označte jen jednu elipsu.

- Scrum
- Kanban
- Kombinace (Scrumban)
- Jiné: _____

19. Dodržujete s týmem/y striktně zásady vybrané agilní metodiky? *

Označte jen jednu elipsu.

- Ano
- Ne

20. Myslíte si, že je řízení dle agilních metodik pro tým/y přínosem? *

Označte jen jednu elipsu.

- Ano
- Spíše ano
- Ne
- Spíše ne

21. Usnadňuje Vám řízení dle agilních metodik práci?

Označte jen jednu elipsu.

- Ano
- Spíše ano
- Ne
- Spíše ne

22. Jakou výhodu vidíte v používání agilní metodiky Scrum a Kanban? *

23. Jakou nevýhodu naopak vidíte v používání agilní metodiky Scrum a Kanban? *

24. Jaký nástroj používáte pro sledování pracovního postupu týmu/ů? *

Označte jen jednu elipsu.

- JIRA
- Phabricator
- Obě možnosti jsou nezbytné
- Jiné: _____

25. Jaký hlavní komunikační nástroj používáte pro komunikaci s týmem/y? *

Označte jen jednu elipsu.

- Slack
- MS Teams
- E-mail
- Jiné: _____

Přeskočte na otázku 46

Team Lead

26. Máte na starosti více než 1 tým? *

Označte jen jednu elipsu.

- Ano
- Ne

27. Uveďte tým/y, se kterým/i se podílíte na řízení projektu/ů. *

Reálné názvy týmů nebudou v BP uvedeny. Finální označení bude ctít formát "Tým [1-X]".
Do této odpovědi prosím neuvádějte týmy typu POs a Testers.

28. Napište první věc, která Vás napadne, když se řekne: "Agilní metodika" *

29. Jakou agilní metodiku nejraději používáte k řízení projektů? *

Označte jen jednu elipsu.

- Scrum
- Kanban
- Kombinace (Scrumban)
- Jiné: _____

30. Dodržujete s týmem/y striktně zásady vybrané agilní metodiky? *

Označte jen jednu elipsu.

- Ano
- Ne

31. Myslíte si, že je řízení dle agilních metodik pro tým/y přínosem? *

Označte jen jednu elipsu.

- Ano
- Spíše ano
- Ne
- Spíše ne

32. Usnadňuje Vám řízení dle agilních metodik práci?

Označte jen jednu elipsu.

- Ano
- Spíše ano
- Ne
- Spíše ne

33. Jakou výhodu vidíte v používání agilní metodiky Scrum a Kanban? *

34. Jakou nevýhodu naopak vidíte v používání agilní metodiky Scrum a Kanban? *

35. Jaký nástroj používáte pro sledování pracovního postupu týmu/ů? *

Označte jen jednu elipsu.

- JIRA
- Phabricator
- Obě možnosti jsou nezbytné
- Jiné: _____

36. Jaký hlavní komunikační nástroj používáte pro komunikaci s týmem/y? *

Označte jen jednu elipsu.

- Slack
- MS Teams
- E-mail
- Jiné: _____

Přeskočte na otázku 46

Jiná role

Vrcholový management, Backoffice a další podobně specializované role

37. Napište první věc, která Vás napadne, když se řekne: "Agilní metodika" *

38. Jaká agilní metodika je z Vašeho pohledu nejvhodnější k řízení projektů? *

Označte jen jednu elipsu.

- Scrum
- Kanban
- Kombinace (Scrumban)
- Jiné: _____

39. Myslíte si, že je důležité dodržovat striktně zásady agilních metodik? *

Označte jen jednu elipsu.

- Ano
- Ne

40. Myslíte si, že je řízení dle agilních metodik pro týmy přínosem? *

Označte jen jednu elipsu.

- Ano
- Spíše ano
- Ne
- Spíše ne

41. Usnadňuje podle Vás agilní metodiky práci?

Označte jen jednu elipsu.

- Ano
- Spíše ano
- Ne
- Spíše ne

42. Jakou výhodu vidíte v používání agilní metodiky Scrum a Kanban? *

43. Jakou nevýhodu naopak vidíte v používání agilní metodiky Scrum a Kanban? *

44. Jaký nástroj Vám pomáhá sledovat pracovní postup týmu/ů, pokud je to potřeba? *

Označte jen jednu elipsu.

- JIRA
- Phabricator
- Obě možnosti jsou nezbytné
- Jiné: _____

45. Jaký hlavní komunikační nástroj používáte pro komunikaci s týmem/y a jejich členy? *

Označte jen jednu elipsu.

- Slack
- MS Teams
- E-mail
- Jiné: _____

Přeskočte na otázku 46

Doplňující otázky

46. Kde jste poprvé na vlastní kůži zažili práci v agilním prostředí? *

Označte jen jednu elipsu.

- V JCS
- V jiné firmě spadající pod JABLOTRON GROUP
- Mimo JABLOTRON GROUP

47. Co ve Vás evokují obecně metodiky řízení SW vývoje? *

Můžete zvolit více odpovědí

Zaškrtněte všechny platné možnosti.

- Přesný postup
- Příručku, jak dělat věci lépe
- Zbytečnou zátěž
- Nepostradatelný nástroj
- Jiné: _____

48. Došlo u Vás díky práci v agilním prostředí ke zlepšení některých dovedností? *
Pokud ano, prosím specifikujte.

Pokud si myslíte, že k žádnému zlepšení nedošlo, stačí vyplnit "Ne"

49. Znáte i jinou agilní metodiku kromě Scrumu, Kanbanu a jejich kombinace (Scrumbanu)? *
Pokud nevíte, stačí odpovědět "Ne".

Odeslání dotazníku

Tímto Vám velmi děkuji za absolvování tohoto dotazníkového šetření k mé bakalářské práci. Moc si vážím Vaší podpory, která je v tomto ohledu neocenitelná :)

Obsah není vytvořen ani schválen Googlem.

Google Formuláře