



Pedagogická  
fakulta  
Faculty  
of Education

Jihočeská univerzita  
v Českých Budějovicích  
University of South Bohemia  
in České Budějovice

**Jihočeská univerzita v Českých Budějovicích**

Pedagogická fakulta

Katedra informatiky

**Zvyšování Scrum Maturity ve vývojářském týmu**

**Raising the Scrum Maturity in developer team**

Bakalářská práce

**Vypracoval:** Michal Golas

**Vedoucí práce:** doc. Ing. Zora Říhová, CSc.

České Budějovice 2023

# JIHOČESKÁ UNIVERZITA V ČESKÝCH BUDĚJOVICÍCH

Pedagogická fakulta  
Akademický rok: 2021/2022

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Michal GOLAS**  
Osobní číslo: **P20490**  
Studijní program: **B7507 Specializace v pedagogice**  
Studijní obor: **Informační technologie a e-learning**  
Téma práce: **Zvyšování Scrum Maturity ve vývojářském týmu**  
Zadávací katedra: **Katedra informatiky**

### Zásady pro vypracování

Cílem bakalářské práce je analýza dosavadní praxe ve vybraném vývojovém týmu a na základě metodiky Scrum navrhnout sadu opatření ke zvýšení efektivity práce týmu při řešení projektů. Navržená opatření budou podrobena analýze rizik SWOT a po nasazení těchto opatření student prostřednictvím prostředí Microsoft Azure DevOps vyhodnotí účinnost těchto změn.

V teoretické části bude uvedena stručná charakteristika základních tradičních i agilních metodik a srovnání hlavních principů těchto směrů. Bude popsána metodika Scrum, na kterou je tato práce zaměřena.

Hlavním přínosem této práce budou užitečná doporučení sloužící ke zlepšení současného procesu vývoje a jejich vyhodnocení.

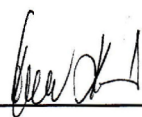
Rozsah pracovní zprávy: **40**  
Rozsah grafických prací: **-**  
Forma zpracování bakalářské práce: **tištěná**

#### Seznam doporučené literatury:

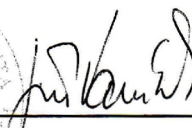
1. Scrum Guide [online]. 2020 [cit. 2022-04-05]. Dostupné z: <https://scrumguides.org/scrum-guide.html>
2. MYSLÍN, J. Scrum: průvodce agilním vývojem softwaru. Brno: Computer Press, 2016. ISBN 978-80-251-4650-7.
3. PROCHÁZKA, J. a C. KLIMES. Provozujte IT jinak: agilní a štihlý provoz, podpora a údržba informačních systémů a IT služeb. Praha: Grada, 2011. ISBN 978-80-247-4137-6.
4. Manifest Agilního vývoje software [online]. 2001 [cit. 2017-02-11]. Dostupné z: <http://agilemanifesto.org/>
5. LACEY, M. The scrum field guide: practical advice for your first year. Upper Saddle River, NJ: Addison-Wesley, c2012. ISBN 978-0-321-55415-4.

Supervizor bakalářské práce: **Mgr. Václav Šimandl, Ph.D.**  
Katedra informatiky  
Vedoucí bakalářské práce: **doc. Ing. Zora Říhová, CSc.**  
Přírodovědecká fakulta, katedra informatiky

Datum zadání bakalářské práce: 7. dubna 2022  
Termín odevzdání bakalářské práce: 30. dubna 2023



doc. RNDr. Helena Koldová, Ph.D.  
děkanka



doc. PaedDr. Jiří Vaniček, Ph.D.  
vedoucí katedry

V Českých Budějovicích dne 7. dubna 2022

## **Prohlášení**

Prohlašuji, že jsem autorem této kvalifikační práce a že jsem ji vypracoval pouze s použitím pramenů a literatury uvedených v seznamu použitých zdrojů.

V Českých Budějovicích dne 16. dubna 2023.

Michal Golas

## **Abstrakt / Anotace**

Cílem bakalářské práce je analýza dosavadní praxe ve vybraném vývojovém týmu a na základě metodiky Scrum navržení sady opatření ke zvýšení efektivity práce týmu při řešení projektů. Navržená opatření jsou podrobena analýze SWOT a po nasazení těchto opatření je prostřednictvím prostředí Microsoft Azure DevOps vyhodnocena účinnost těchto změn.

V teoretické části je uvedena stručná charakteristika základních tradičních i agilních metodik a srovnání hlavních principů těchto směrů. Je popsána metodika Scrum, na kterou je tato práce zaměřena.

Hlavním přínosem této práce jsou užitečná doporučení sloužící ke zlepšení současného procesu vývoje a jejich vyhodnocení.

## **Klíčová slova**

Agilní metodika, Scrum, Scrum Maturity, vývoj software, zvýšení efektivity

## **Abstract**

The aim of the bachelor thesis is to analyze the current practice in the selected development team and based on the Scrum methodology, propose a set of measures to increase the efficiency of the team's work in solving projects. The proposed measures are subjected to a SWOT risk analysis and after deploying these measures, the effectiveness of these changes is evaluated through the Microsoft Azure DevOps environment.

In the theoretical part, a brief characterization of the basic traditional and agile methodologies is given and a comparison of the main principles of these directions is presented. The Scrum methodology, which is the focus of this thesis is described.

The main contribution of this thesis is useful recommendations to improve the current development process and their evaluation.

## **Keywords**

Agile methodology, Scrum, Scrum Maturity, software development, raising efectivity

## Poděkování

Chtěl bych poděkovat paní docentce Zoře Říhové za důsledné vedení mé bakalářské práce a svým nadřízeným Ing. Tomáši Hurychovi a Mgr. Pavlu Urbanovi za konzultace a poskytnutý přístup k informacím nezbytným pro vznik této práce.

Dále bych chtěl poděkovat své přítelkyni za to, že mi při tvorbě této bakalářské práce byla nesmírnou oporou.

# Obsah

<b>1</b>	<b>Úvod</b>	<b>10</b>
<b>2</b>	<b>Teoretická část</b>	<b>11</b>
2.1	Proces vývoje software . . . . .	11
2.2	Model vyspělosti procesu . . . . .	12
2.2.1	Úvodní (Initial) . . . . .	12
2.2.2	Opakovatelná (Repeatable) . . . . .	12
2.2.3	Definovaná (Defined) . . . . .	13
2.2.4	Měřitelná (Measured) . . . . .	13
2.2.5	Optimalizovaná (Optimizing) . . . . .	13
2.3	Metodiky vývoje software . . . . .	14
2.4	Tradiční metodiky vývoje software . . . . .	14
2.4.1	Vodopádový model (Waterfall model) . . . . .	15
2.4.2	Ostatní iterativní metodiky . . . . .	16
2.5	Agilní metodiky vývoje software . . . . .	18
2.5.1	Manifest agilního vývoje (Agile Manifesto) . . . . .	18
2.6	Scrum . . . . .	21
2.6.1	Definice . . . . .	21
2.6.2	Role . . . . .	21
2.6.3	Projektový tým . . . . .	22
2.6.4	Product Owner . . . . .	22
2.6.5	Scrum Master . . . . .	22
2.6.6	Scrum tým . . . . .	24
2.6.7	Pojmy a artefakty . . . . .	24
2.6.8	Události . . . . .	27
2.6.9	Měření . . . . .	29
<b>3</b>	<b>Praktická část</b>	<b>31</b>
3.1	Scrum Maturity . . . . .	31



3.2	O týmu . . . . .	31
3.3	Jak tým praktikuje Scrum? . . . . .	32
3.4	Prostředí Microsoft Azure DevOps . . . . .	34
3.5	SWOT analýza současného stavu týmu . . . . .	35
3.6	Agile Maturity Matrix (Dotazník agilní vyspělosti) . . . . .	37
3.7	Pozorovací Sprint . . . . .	39
3.7.1	Vypozorované chyby a poznatky v průběhu pozorovacího Sprintu . . . . .	39
3.7.2	Výsledky Agile Maturity Matrix v pozorovacím Sprintu . . . . .	41
3.7.3	Vyvozené důsledky a návrhy na zlepšení . . . . .	45
3.8	Opravný Sprint . . . . .	50
3.8.1	Scrum Workshop . . . . .	50
3.8.2	Týmová porada ohledně práce s Backlogem . . . . .	51
3.8.3	Tvorba nové prezentace na Sprint Review . . . . .	51
3.8.4	Nový formát schůzky Sprint Retrospective . . . . .	54
3.8.5	Apel na ohleduplnost . . . . .	55
3.8.6	Barevné okódování Backlogu, doplnění komentářů . . . . .	56
3.9	Vyhodnocení účinnosti nápravných opatření . . . . .	57
<b>4</b>	<b>Závěr</b>	<b>60</b>
	<b>Seznam použité literatury a zdrojů</b>	<b>64</b>
	<b>Seznam tabulek</b>	<b>65</b>
	<b>Seznam obrázků</b>	<b>66</b>
	<b>Přílohy</b>	

## 1 Úvod

Tato bakalářská práce se zabývá zvyšováním efektivity a správnosti využívání metodiky Scrum. Zvyšování tzv. Scrum Maturity je téma, které není v českých kruzích příliš prozkoumané, tudíž bylo pro mě atraktivní k zpracování. V průběhu práce budu ověřovat účinnost opatření, které by měly vést ke zvýšení Scrum Maturity. Finálním výstupem mé práce bude vyhodnocení účinnosti daných opatření a užitečnosti při využívání metodiky.

Teoretická část práce poskytuje nezbytná východiska, která budou dále potřebná v praktické části pro zhodnocení současného stavu ve vývojářském týmu. Tato bakalářská práce se bude zaměřovat výhradně na metodiku Scrum, ostatní metodiky budou z důvodu rozsahu a zachování věcnosti stručně zmíněny a představeny.

## 2 Teoretická část

Již od počátku samotného vývoje software se používaly takzvané těžké, rigorózní metodiky. Takové metodiky jsou typickými zástupci vodopádového modelu vývoje, od kterého se moderní vývojářské týmy pokouší v dnešní době upouštět. Pro těžkou metodiku je typické, že je třeba již od začátku projektu mít detailní návrh, práci omezuje přesně nadefinovaná spolupráce a v projektu je velmi těžké reagovat na změny.[1]

Naprostým opakem jsou metodiky odlehčené. Na počátku 21. století se pro ně ujalo dnes již známé označení agilní metodiky. Ty se podle slov svých tvůrců navrací k původním praktikám ze samotných počátků vývoje software. Upřednostňují uspokojení zákazníka průběžnými a rychlými dodávkami kvalitního software, maximalizaci hodnoty vyvíjeného produktu, transparentní komunikaci mezi lidmi z byznysu a vývojáři. Díky schopnosti pružně reagovat na změny poskytují zákazníkům konkurenční výhody.[1]

### 2.1 Proces vývoje software

Samotný proces vývoje je definován jako po částech uspořádaná posloupnost aktivit vedoucí od počátečního stavu k danému cíli. Vývoj software má ovšem mnoho specifik. Na rozdíl od „řemeslné“ výroby má specifikovaný cíl zákazníkem, ale zároveň nemá daný postup. Každé řešení může být zcela unikátní a na jeho tvorbu neexistuje univerzální návod. Pochopitelně můžeme recyklovat předchozí zkušenosti, komponenty, či konstrukce, ale vždy bychom měli brát v potaz, zda je to v souladu se zájmy zákazníka.[1]

## 2.2 Model vyspělosti procesu

Kvalitu procesu vývoje je třeba měřit, některé z možností jsou jednodušší, některé jsou spíše ve formě auditu. Model měření vyspělosti procesu se nazývá Capability Maturity Model (Stupňovitý model zralosti), spíše známý pod zkratkou SW-CMM. Rozlišuje pět úrovní vyspělosti procesu na základě toho, jak proces na dané úrovni vypadá a co je nutno zlepšit, aby se mohl posunout na vyšší úroveň.[1]

### 2.2.1 Úvodní (Initial)

Na první úrovni se společnost octne tím, že započne svou existenci. Tato úroveň je velmi nízká a společnost při ní nedokáže produkovat kvalitní software, proces není definován. Vyvíjí se intuitivně, problémy jsou řešeny ad hoc. U projektů vedených tímto způsobem dojde dříve, nebo později k potížím v nákladech, času, nebo kvalitě. Je zde nemožné jakékoliv řízení lidských zdrojů a komunikace se zákazníky je omezena na řešení nastalých problémů. Při zachování procesu této úrovně po celou dobu projektu je projekt předurčen k neúspěchu. Je velmi důležité se z této úrovně co nejrychleji dostat na vyšší.[1]

### 2.2.2 Opakovatelná (Repeatable)

Přechod z první na druhou úroveň se zpravidla děje samovolně. Pokud projekt začíná a na všechny nastalé situace se reaguje ad hoc, postupem času přirozeně zjistíme, že se některé opakují. Druhá úroveň je tedy obohacena o takzvané "best practices", neboli praktiky, které fungují a osvědčily se. Takové postupy je možné následně používat i v jiných projektech. Nicméně takové popsání části procesu nejsou ucelené, není popsán proces jako takový. Jedná se pouze o získání zkušenosti, že úkol lze splnit konkrétním způsobem.[1]

### 2.2.3 Definovaná (Defined)

Třetí úroveň je možné již považovat za systematický vývoj softwaru. Popsán je celý proces vývoje software, nikoliv pouze jeho části. Jedná se o ucelený, předem stanovený postup, který je rozumný a optimální. V případě, že by se firma ucházela o certifikát jakosti ISO 9001 – její úroveň zralosti procesu musí být minimálně na úrovni 3.[1]

### 2.2.4 Měřitelná (Measured)

Čtvrtá úroveň vyspělosti je první dostačující úrovní pro vývoj moderního softwaru. Při vývoji se postupuje systematicky a metodicky. Obsahuje měření a vyhodnocení efektivnosti postupu. Podmínkou pro postup na čtvrtou úroveň je existence „tvrdých dat“ – sady ukazatelů, které mají vlastní měřicí stupnice, které následně vyhodnocujeme. Může to být cokoliv, co je objektivně měřitelné – například procento implementovaných případů užití, úspěšných testů apod.[1]

### 2.2.5 Optimalizovaná (Optimizing)

Nejvyšší úrovně zralosti procesu lze dosáhnout tehdy, když na základě provedeného měření dokáže tým implementovat nápravy odchylek reálného stavu a přivést je do stavu očekávaného. Úspěšným zvládnutím takových náprav se tým postupně učí ze svých chyb a zlepšuje se.[1]

### 2.3 Metodiky vývoje software

Pokud je sada procesů vyspělá alespoň na úroveň 4, nebo 5, společnost zvládá hodnotit proces samotný a vylepšovat ho, dá mluvit o tzv. metodice vývoje software. Metodika říká, jaká bude posloupnost jednotlivých činností, jak a kým bude projekt řízen, či jak bude probíhat plánování projektu aj. V procesu vývoje software směřuje k tomu co dělat, ale neříká jak.

### 2.4 Tradiční metodiky vývoje software

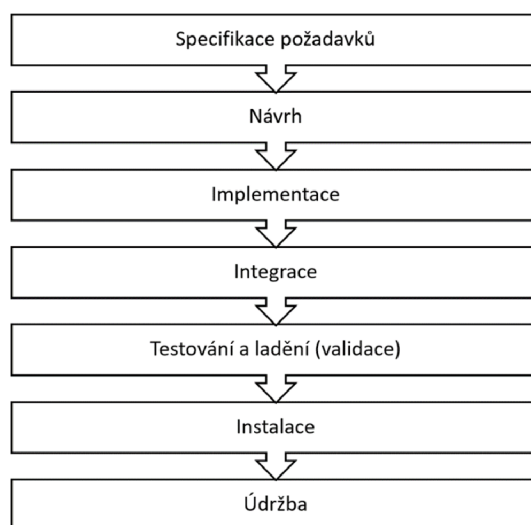
Tradiční neboli rigorózní metodiky vývoje software spojuje snaha o co nejmenší vágnost, snaha o určení jednotlivých termínů, požadavků. Role jsou přesně dány – lidé mají přesně určenou svou specializaci a neúčastní se jiných fází a činností vývoje softwaru, které jim nepřísluší. Z jedné strany je v takové metodice výhodné, že jsou členové týmů specializováni v konkrétní činnosti, kterou dokonale ovládají a dovedávají se v ní. Nicméně z druhé strany je pro splnění úkolu nutná spolupráce a komunikace mnoha lidí, což proces prodlužuje a zvyšuje množství režie. V tradičních metodikách se také klade velký důraz na to, aby bylo vše dokumentováno. Implementace začíná až v momentu, kdy je dostupná kompletní analýza a návrh výsledného systému. Pro zákazníky může být nevýhodou to, že velmi dlouhou dobu od zadání zakázky nevidí žádné, ani dílčí výsledky. Implementace sebemenších změn je v případě tradičních metodik zdoluhavá a těžkopádná, často může trvat i měsíce, než se vyřídí veškerá byrokracie. Vhodné využití pro tradiční metodiky je ve větších projektech, na kterých zpravidla pracuje více vývojářských týmů a využívá se v nich více různých technologií a systémů. Pokud je třeba dodržet termíny, je přesně daný rozpočet a je nutné dodržet veškeré funkcionality.[1]

### 2.4.1 Vodopádový model (Waterfall model)

Jedná se o nejznámější ze všech tradičních metodik. Byla představena Winstonelem Roycem v sedmdesátých letech a je založena na myšlence, že se softwarový proces skládá z několika fází. Aby bylo možné takový proces řídit, je nutné, aby fáze následovaly po sobě.[1]

Mezi hlavní charakteristiky vodopádového modelu patří:

- **Lineární průběh** - proces je jednosměrný a nelze se v něm vracet
- **Jednoznačnost** – je vždy jednoznačné, ve které fázi se projekt nachází
- **Úplné zadání** – do další fáze se dá vstoupit pouze tehdy, jsou-li předchozí náležitosti splněny[1]



Obrázek 1: Vodopádový model vývoje software, Zdroj: autor, vlastní zpracování obecného schéma vodopádového modelu vývoje software

Sám autor metodiky časem objevil problém, který vývoj komplikuje. V případě, že je v rámci testování odhalena chyba, k tomu, aby započaly opravy je třeba mnoho schvalování, které vede ke zdržení při vývoji. Šlo o nejčastější problém při využívání modelu, model byl později upraven tak, že z každého

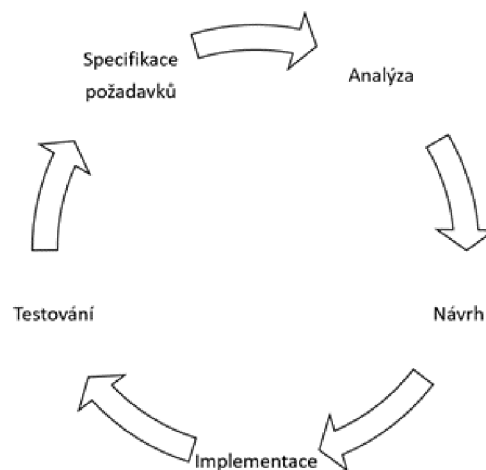
kroku procesu je možné se vrátit zpět a potřebnou fázi opakovat.[2]

V dnešní době již není vodopádový model příliš užívaný, zejména na základě uvedené časté chyby. Vhodné využití modelu je u menších projektů.

#### 2.4.2 Ostatní iterativní metodiky

Problémy jako pomalé reakce na změny, přebytečná byrokracie, nebo to, že zákazník v průběhu nevidí ani dílčí části práce pochopitelně vedly k bádání po něčem, co takové nevýhody mít nebude. Tady přichází na scénu iterativní metodiky. Jejich vznik byl přirozenou reakcí na nedostatky vodopádového modelu.[1]

Základem všech iterativních metodik je jednoduchý iterační cyklus.[3]



Obrázek 2: Jednoduchý iterační cyklus, Zdroj: autor, vlastní zpracování obecného schéma iteračního cyklu

Mezi zástupce iterativních metodik patří například:

- Rational Unified Process (RUP)
- Spirálový model
- Unified Software Development Process (UP)[3]



Společným jmenovatelem iteračních metodik je:

- Spustitelný kód na konci každé iterace – na konci iterace může zákazník ozkoušet funkční výsledek (tzv. demo) a může dávat zpětnou vazbu
- Testování v průběhu každé iterace – na konci iterace jsou objeveny všechny chyby skrze testování a v další mohou být opraveny
- Změny jsou součástí samotného procesu – na základě vyzkoušení funkčního výsledku může zákazník vznášet zpětnou vazbu, která následně může být zapracována do dalších iterací[3]

## 2.5 Agilní metodiky vývoje software

Z důvodu rychle rozvíjejícího se trhu s informačními technologiemi nastala potřeba po něčem efektivnějším, než jsou tradiční metodiky vývoje softwaru. Oproti tradičním metodikám jsou agilní metodiky flexibilnější a počítají s vývojem nových nástrojů. Eliminují přebytečnou byrokracii, dokumentování nevýznamných událostí a zjednodušují proces změny.[1]

Základním principem a hodnotou agilních metodik je snaha o dodání kvalitního softwaru. Dokumentace se nedělá z povinnosti zanesené ve směrnici, ale aby se vývojář mohl v kódu orientovat. Změnové řízení je značně zjednodušené tak, aby se udržel pořádek v projektu.[1]

### 2.5.1 Manifest agilního vývoje (Agile Manifesto)

Manifest agilního vývoje je prohlášení, které vytvořila skupina vývojářů, věnujících se vývoji softwaru, která nabyla přesvědčení, že vývoj lze provádět lépe než dosud. Znění agilního manifestu je následující:

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

**Jednotlivci a interakce** před procesy a nástroji

**Fungující software** před vyčerpávající dokumentací

**Spolupráce se zákazníkem** před vyjednáváním o smlouvě

**Reagování na změny** před dodržováním plánu

Jakkoliv jsou body napravo hodnotnější, bodů nalevo si ceníme více.[4]

Z agilního manifestu následně vychází 12 principů:

1. Uspokojení zákazníka skrze brzké a rychlé dodávky software
2. Uspokojení požadavků na změny skrze proces vývoje
3. Časté dodávky funkčního software
4. Spolupráce mezi zákazníky a vývojáři na denní bázi
5. Podporujte, důvěřujte a motivujte lidi zapojené ve vývoji
6. Nejúčinnější metodou pro komunikaci je komunikace tváří v tvář
7. Funkční software je hlavním měřítkem postupu ve vývoji
8. Agilní procesy podporují udržitelné tempo vývoje
9. Věnujte pozornost technickým detailům a dokonalosti
10. Základem je jednoduchost – vyvíňte takové úsilí, které je nejmenší k tomu, aby byl projekt hotový
11. Sebeorganizující tým vede k nejlepším architekturám, požadavkům a návrhům
12. V pravidelných intervalech vyhodnocujte svou práci a na základě zpětné vazby upravujte své postupy, aby byly co nejefektivnější.[5]

Myslím tyto principy komentuje, a usuzuje, že agilní manifest a principy z něj vycházející vychází ze zdravého rozumu. Vychází z toho, co je osvědčené a lidé se s tím dokáží snadno ztotožnit.[1]

Některé z agilních metodik, které vychází z hodnot uvedených v Manifestu agilního vývoje jsou:

- **Scrum**
- Extrémní programování
- Vývoj řízený vlastnostmi (Feature Driven Development)
- Vývoj řízený testy (Test Driven Development)
- Lean Development[1]

## 2.6 Scrum

### 2.6.1 Definice

Scrum je agilní metodika, která pomáhá lidem, týmům a organizacím vytvářet hodnotný software, Využívá agilních řešení na komplikované problémy, které mohou nastat v projektu.[6]

Metodika je založena na principu empirismu, který tvrdí, že znalosti vycházejí z minulých zkušeností a pozorování. Další podstatnou částí metodiky je tzv. Líné myšlení, které se zaměřuje pouze na podstatné náležitosti vývoje a snižuje plýtvání zdroji. Jako většina agilních metodik využívá spojení iterativního a inkrementálního vývoje.[6]

Lidé využívající metodiku Scrum následně využívají různé nástroje, pojmy a praktiky, které budou popsány v následujících kapitolách.

### 2.6.2 Role

V metodice Scrum je rolí méně a bývají univerzálnější než v tradičních metodikách. Základní dělení rolí v metodice Scrum je na manažerské a výkonné, nebo jak sama metodika Scrum říká, na "Prasata a kuřata"(Pigs and chickens). Metodika tuto metaforu vysvětluje následovně:

Produktem jsou **vajíčka se slaninou**. Zatímco **kuře** se na tvorbě produktu **pouze podílí**, ale **není mu nijak uškozeno**, **prase** se musí pro toto jídlo **"obětovat"**, **je jeho součástí**. [7]

Ekvivalentem "prasiat" jsou v praxi členové vývojářského týmu, kteří rozhodují, jak se bude vývoj provádět a jakým tempem ho lze dosáhnout. "Kuřata" jsou v týmu zastupována Product Ownerem. Posledním členem je Scrum Master, který je zodpovědný za organizaci Scrum událostí.

Ekvivalentem kuřat jsou v praxi zákazníci (koncoví uživatelé), či zástupci vedení ve společnosti zákazníka. [7]

### 2.6.3 Projektový tým

Oproti jiným metodikám (např. RUP) se Scrum snaží počet rolí minimalizovat, nalezneme zde proto pouze tři základní. Scrum tým se skládá z vývojového týmu, Product Ownera a Scrum Mastera.[1]

Jedním z cílů metodiky Scrum je dosáhnout maximální univerzálnosti členů vývojového týmu, očekává se rozvoj pracovníků s ohledem na jejich přirozenou specializaci. S ohledem na to je nutné poskládat tým tak, aby se vývojáři zkušenostně vyvažovali.[1]

### 2.6.4 Product Owner

Product Owner má za cíl maximalizovat prodejní hodnotu vyvíjeného produktu. Před vývojovým týmem zastupuje zákazníka a jeho zájmy. V týmu se nachází pouze jeden.[6]

Další z jeho povinností je správa a udržování Product Backlogu, která zahrnuje:

- Komunikaci týkající se Product Goal, ať už se zákazníky, nebo s vývojovým týmem,
- Vytváření a prioritizace položek Product Backlogu,
- Zajištění transparentnosti a srozumitelnosti Product Backlogu[6]

Je možné, že Product Owner některou ze svých povinností deleguje na některého z členů vývojářského týmu. V tom případě ale i tak má Product Owner plnou zodpovědnost za plnění těchto povinností a doručování inkrementu (použitelného přírůstku).[8] Nesmí se stát, že je Product Owner zároveň Scrum Masterem. V ten moment dochází ke střetu zájmů. Zatímco Product Owner zastupuje zájmy zákazníků, Scrum Master zájmy vývojářů.[9]

### 2.6.5 Scrum Master

Scrum Master je přímo odpovědný za efektivitu projektového týmu. Podle Scrum Guide odpovídá za založení a dodržování Scrum kultury ve projektovém

týmu. Jeho cílem by mělo být každému členovi týmu co nejlépe porozumět a dodržovat teoretickou i praktickou část metodiky.[6]

Mezi základní povinnosti Scrum Mastera patří:

- Pomáhat projektovému týmu dosahovat hodnotného inkrementu
- Odstraňování překážek a problémů Scrum týmu
- Zajištění toho, aby se odehrávaly všechny Scrum události
- Koučování členů Scrum týmu v autonomii a všestrannosti[6]

Jednou z nejdůležitějších povinností Scrum Mastera je řešení konfliktů, potíží v komunikaci, nebo sporů mezi členy týmu. Je velmi důležité, aby dokázal odhalit konflikt v počátku a nedovolil mu přerůst v problém devastující celý tým a práci na projektu.[1]

Scrum Master také dále vede a řídí jednotlivé Scrum události, usměrňuje diskusi a zařizuje podpůrné záležitosti, jako třeba místnost, čas a technické prostředky. Dále se stará o "ochranu týmu před vnějším světem", aby se mohl plně soustředit na vývoj. Například se stará o to, aby tým uprostřed vývoje neobtěžovali rozčilení zákazníci.[1]

Myslín tvrdí, že: "Product Owner má úkoly definované spíš "tvrdým způsobem", Scrum Master musí disponovat zejména měkkými dovednostmi. Product Owner definuje, vybírá, stanovuje, ruší. Scrum Master spíše podporuje, motivuje, chrání." [1]

Role Scrum Mastera bývá často nepřiliš oceňovaná, členové vývojářských týmů mívají tendence Scrum Mastery podceňovat, či si o nich myslet, že jsou zbyteční. Často tak dochází ke spojování role Scrum Mastera s jinými rolemi (tester, vývojář, business analytik), jedná se tak o zásadní nepochopení základních principů metodiky Scrum.[10]

### 2.6.6 Scrum tým

Členové Scrum týmu jsou ti, kteří se zavázali vytvářet jakýkoliv inkrement v každém Sprintu.[6]

Mezi základní povinnosti členů Scrum týmu patří:

- Vytváření Sprint Backlogu (neboli plánu pro konkrétní Sprint)
- Dodržování Definition of Done a z toho vycházející plnění kvalitativních kritérií
- Profesionální odpovědnost[6]

### 2.6.7 Pojmy a artefakty

V metodice Scrum se nachází mnoho pojmů, často nepřekládaných do českého jazyka. Zde je jejich stručné vysvětlení:

#### 1. Product Backlog

Jedná se o seřazený, neustále se vyvíjející seznam všech aktivit, které jsou třeba vykonat pro vylepšení produktu.[6] Zjednodušeně řečeno se jedná o seznam veškeré práce, která je před vývojovým týmem a je nutno ji vykonat.

O tvorbu položek Product Backlogu a jeho údržbu se stará Product Owner.[6]

Častým způsobem vzniku Product Backlogu je setkání všech členů Scrum týmu, při kterém se začne diskutovat o všech potřebných funkcích, které projekt musí mít. Funkce se následně sepíší a jejich množství je zpravidla dostačující pro první Sprint. Takto vzniklý Product Backlog je následně rozšiřován v dalších Sprintech na základě nových požadavků a zpětné vazby od zákazníků.[11]

Product Backlog zpravidla obsahuje položky z kategorií nových funkcí, bugs (chyb), administrativních záležitostí, nebo získávání znalostí.[11]



## 2. Product Goal

Product Goal (produktový cíl) je závazek vycházející z Product Backlogu. Jedná se o dlouhodobý cíl, kterého se snaží Scrum tým dosáhnout. Produktovým cílem může být například kompletní dokončení vývoje aplikace.[6]

Správný produktový cíl je měřitelný, adaptivní a založený na faktech. Vhodné otázky pro ustanovení cíle jsou:

- Jak nám tento cíl může pomoci?
- Jak víme, jestli je tento cíl přesný?
- Jak tento cíl můžeme měřit? Jak zjistíme, že jsme ho dosáhli?[12]

## 3. Sprint Backlog

Sprint Backlog je plán vývojového týmu, jedná se o množství práce, které plánují vývojáři provést během Sprintu.[6]

Před zahájením každého Sprintu se členové vývojového týmu při konzultaci s Product Ownerem domluví, které položky si z Product Backlogu vyberou a následně z nich sestaví Sprint Backlog. Sprint Backlog by měl být dostatečně detailně zdokumentován natolik, aby se jím denně dalo procházet při schůzce Daily Scrum.[13]

## 4. Sprint Goal

Sprint Goal je cílem Sprintu. Dosáhne se ho vyhotovením všech položek, které se nacházejí v Sprint Backlogu. Je výsledkem vyjednávání Product Ownera a vývojářského týmu. Měl by být konkrétní a měřitelný.[14]

Pokud se vývojový tým v průběhu Sprintu dozví, že nebude schopen Sprint Goalu dosáhnout, je povinen informovat Product Ownera, se kterým následně vyjednají změnu počtu položek ve Sprint Backlogu.[6]

## 5. Inkrement

Inkrement je konkrétním odrazovým můstkem k dosažení Product Goal.

Jedná se o přírůstek hodnoty k vyvíjenému produktu. V průběhu jednoho Sprintu může docházet i k více přírůstkům inkrementu.[6]

Souhrn inkrementu dosaženého při Sprintu se následně prezentuje zákazníkům na události Sprint Review.[15]

## 6. Definition of Done

Definition of Done je Scrum týmem stanovená definice, která určuje, kdy je možné označit položku ze Sprint Backlogu za hotovou. Okamžikem, kdy je položka hotová, vzniká inkrement.[16]

Definition of Done je jinak konkrétně definovaná pro každou společnost, nebo Scrum tým a jeho produkt. Vývojáři ji jsou povinni dodržovat. Pokud položka ve Sprint Backlogu nespĺňuje Definition of Done, nemůž být prezentována při Sprint Review. Namísto toho se vrací zpět do Backlogu.[6]

## 7. User Story

User Story je uživatelským popisem toho, co by měl systém vykonávat. Neobsahuje technické detaily. Zpravidla bývá formulován:

"Jako **role** chci **cíl** tak, aby **užitek**"[1]

Například "Jako správce e-shopu chci tlačítko, které přidá nový produkt", nebo "Jako zaměstnanec firmy chci formulář, kterým mohu kontaktovat technickou podporu".[1] Takto vytvořené User Story mají následně přiřazenou prioritu a jsou součástí Product Backlogu.

Některé vývojářské týmy místo User Story používají Product Backlog Item (PBI). PBI je obecný název pro položku v Product Backlogu, která je podobná, ale zároveň nemusí splňovat náležitosti User Story.[17]

### 2.6.8 Události

#### 1. Sprint

Sprint je speciální označení pro iteraci v metodice Scrum. Jedná se o časový úsek trvající maximálně měsíc, který se pravidelně opakuje až do splnění Product Goal. Nový Sprint začíná okamžikem ukončení předchozího. Každý Sprint se dá považovat za malý projekt.[6]

#### 2. Plánovací schůzka (Sprint Planning)

Při Sprint Planning vývojový tým ve spolupráci s Product Ownerem stanovuje Sprint Goal. Tým vybírá položky z Product Backlogu, ze kterých je následně sestaven Sprint Backlog.

Maximální délka plánovací schůzky je dle Scrum Guide 8 hodin na měsíc dlouhý Sprint.[6] V praxi je ale zpravidla mnohem kratší.

#### 3. Denní schůzka (Daily Scrum)

Daily Scrum (zastarale Daily Standup — odvozeno od toho, že se u něj v kanceláři stálo[18]) je každodenní schůzka sloužící ke kontrole pokroku směrem ke Sprint Goalu.

Scrum počítá se skutečností, že členové Scrum týmu jsou nezávislí. Může dojít k tomu, že kolegové mezi sebou navzájem neví, na čem pracují, nebo v jakém stádiu práce jsou. Druhou věcí, ke které Daily Scrum slouží je včasné zachycení problémů, jejich identifikace a sjednání nápravy. Scrum je velmi užitečný v eliminaci "zametání problémů pod stůl".[1]

V průběhu schůzky by každý člen vývojového týmu měl odpovědět na následující otázky:

- Co jsi dělal včera?
- Co budeš dělat dnes?
- Jsou nějaké problémy, které ti brání v práci?[19]

Product Owner je v tomto meetingu pouhým pozorovatelem, jeho účast je stejně jako u Scrum Mastera nepovinná. Cílem Scrum Mastera je vést

tým k autonomii tak, aby si Daily Scrum zvládl uspořádat sám.

Maximální délka schůzky Daily Scrum je 15 minut.[6] Nejčastější potíže a výzvy týkající se schůzky Daily Scrum jsou:

- Délka schůzky přesahuje 15 minut
- Členové týmu se nedělí o vzniklé problémy
- Dochází k dlouhé diskusi jen mezi pár členy týmu
- Dochází k odklonu od tématu schůzky[20]

#### 4. Sprint Review

Cílem Sprint Review je kontrola, zda došlo ke splnění Sprint Goal. Scrum tým prezentuje zákazníkům výsledky své práce v průběhu Sprintu a diskutuje o dalších možných změnách či požadavcích.[6]

Sprint Review schůzka by měla sloužit jako oslava dosažených úspěchů v uplynulém Sprintu. Product Owner se Scrum Masterem při ní prezentují výsledky práce a vývojový tým se pouze účastní, případně odpovídá na otázky technického rázu.[1]

Maximální délka schůzky je dle Scrum Guide 4 hodiny pro měsíční Sprint.[6]

#### 5. Sprint Retrospective

Narozdíl od Sprint Review, kde se zákazníkům prezentuje to, co se týmu podařilo, se při Sprint Retrospective dívá Scrum tým na uplynulý Sprint kriticky.[1]

V průběhu Sprint Retrospective se probírá na základě vstupů od členů Scrum týmu, co se podařilo, co se nepodařilo a nápady na zlepšení. Probírá se, na jaké problémy se v průběhu Sprintu narazilo a jak se je povedlo, či nepovedlo vyřešit. Může také dojít k tomu, že některé výstupy a návrhy na vylepšení budou zaneseny do dalšího Sprint Backlogu.[6]

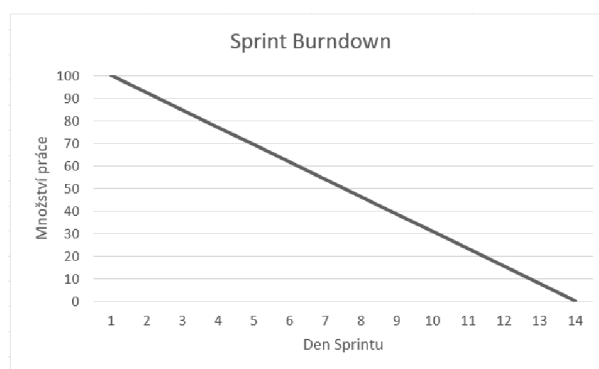
Sprint Retrospective schůzkou se Sprint plně uzavírá. Maximální délka jsou 3 hodiny při měsíčním Sprintu.[6]

### 2.6.9 Měření

Stejně jako tradiční (rigorózní) metodiky s minimální úrovní vyspělosti 4 se Scrum snaží měřit efektivitu práce. Následně z ní pak vyvozuje důsledky a opatření. Nejčastěji používaným nástrojem v metodice Scrum je tzv. Burndown Chart.

#### 1. Burndown Chart

Burndown Chart je grafickou reprezentací toho, kolik zbývá vykonat práce proti času, který zbývá. Osou Y se značí množství zbývajících práce, zatímco na ose X jsou značeny jednotlivé zbývajících dny Sprintu.



Obrázek 3: Burndown Chart, Zdroj: autor

Diagram se konstruuje tak, že se každý den zanesou, jak velká část Backlogu je zvládnuta. Ideální průběh grafu je takový, že každý den tým stihne stejné množství práce, jde tedy o tzv. konstantní tempo.[1]

#### 2. Hodnocení celkového objemu prací

V metodice Scrum se v praxi využívají dva možné přístupy k ohodnocení množství práce ve Sprintu.

- **Člověkohodiny/člověkodny**

Ohodnocení práce probíhá takovým způsobem, že se odhadne, kolik

času zabere pracovníkovi splnění úkolu. Zásadní nevýhodou je nepřesnost takového odhadu, jednomu pracovníkovi může úkol zabrat 30 hodin a druhému 10.[1]

- **Ohodnocení interní metodikou**

Každý úkol se ohodnotí určitou hodnotou, následně se při plánování Sprint Backlogu tým snaží, aby výsledný inkrement tvořený ze všech pracovních úkolů byl stejný, jako v předchozích Sprintech.[1]

## 3 Praktická část

Praktická část se bude zabývat analýzou dosavadní praxe ve Scrum týmu a následnou tvorbou opravných opatření, týkajících se zjištěných nedostatků. Členové Scrum týmu vyplní na začátku pozorovacího Sprintu Agile Maturity matici, kde budou moci vypovídat jak o poměrech v týmu, tak i o způsobu práce a jejich celkové spokojenosti. Následně proběhne Sprint opravný, ve kterém budou praktikována všechna opatření vyvozená z chyb vypořádaných při Sprintu pozorovacím. Na konci opravného Sprintu tuto matici vyplní členové týmu ještě jednou a bude možné pozorovat, zda došlo ke zlepšení efektivity, vztahů a motivace v týmu.

### 3.1 Scrum Maturity

Scrum Maturity je pojem, který nemá přesnou definici. Velmi obecnou definicí Scrum Maturity je, že se jedná o stupeň schopnosti týmu správně a efektivně využívat metodiku Scrum k dodání inkrementu a dosáhnutí agility.[21]

### 3.2 O týmu

Vývojářský tým byl založen 1. 1. 2014 s 8 členy, sestaven z poloviny z business analytiků a poloviny softwarových vývojářů. Jedná se o BizDevOps multidisciplinární tým pracující s IT odborníky, finančními experty a pojistnými matematicky.

Vyvíjeným produktem je aplikace sloužící k internímu výpočtu rizika na základně regulatorních požadavků pojistného standardu Solvency II a IFRS17. V současné době je řešení poskytováno sedmi mezinárodním jednotkám.

V týmu se momentálně nachází 14 zaměstnanců, z toho šest v Business části (jeden manažer a zároveň Product Owner, dva Business analytici, dva Junior Testeři a jeden Informatica Administrátor) a osm v IT části (jeden manažer, dva Senior Python vývojáři, dva .NET vývojáři, dva Informatica Administrá-

toři a jeden Scrum Master).

### 3.3 Jak tým praktikuje Scrum?

V týmu je Scrum praktikován téměř standardně, jsou zde tradiční Scrum události - Daily Scrum, Backlog refinement, Sprint Review, Sprint Retrospective a Sprint Planning. Délka Sprintu je 14 dní.

Události Daily Scrum a Sprint Retrospective pořádám já, jakožto trainee na roli Scrum Master. Ostatní Scrum události má prozatím ve své kompetenci Product Owner. Tým využívá webového prostředí Microsoft Azure DevOps pro výkon metodiky Scrum. K ohodnocení množství práce se využívá odhadu člověkohodin. Tým k zapisování práce do Product Backlogu nevyužívá User Story, ale pouze Product Backlog Item (PBI).

Délky Scrum událostí jsou časově ohraničeny následovně:

- Daily Scrum - 15 minut + 5 minut rezerva v případě nutnosti řešení problémů/bugů/incidentů
- Backlog Refinement - 1 hodina
- Sprint Review - 1 hodina
- Sprint Retrospective - 45 minut
- Sprint Planning - 1 hodina 15 minut

V následujícím výčtu je uvedeno, jak tyto Scrum události probíhají:

#### 1. Daily Scrum

Daily Scrum probíhá každý den v 9:35. V pozici Scrum Mastera využívám funkci sdílené obrazovky v aplikaci Microsoft Teams. Sprint Backlog je seřazený dle priority a postupně se prochází každá položka. Průběh se může změnit v případě problémů/bugů/incidentů. Zaměřuji se na stav položek



v oddělení "In Progress" (v průběhu) a v "Blocked" (zablokovaný). Dotazovaný kolega se stručně vyjádří k průběhu práce za poslední den či zda ho něco blokuje. V tom případě se pokusím zařídit pomoc či nápravu.

## 2. Backlog Refinement

Backlog Refinement probíhá předposlední den Sprintu. Product Owner prochází Product Backlogem v prostředí Azure DevOps a na základě zjištěných informací v průběhu Sprintu je konzultuje s týmem, doplňuje nově zjištěné podúkoly, opravuje časové ohodnocení, či případně odstraňuje již nerelevantní úkoly.

## 3. Sprint Review

Sprint Review probíhá v poslední den Sprintu. Product Owner prezentuje zákazníkům prezentaci, kterou připravili členové týmu společně se Scrum Masterem. Prezentace obsahuje informace o tom, zda byl splněný Sprint Goal, jaké byly dosaženy úspěchy a splněny úkoly, zda tým narazil na nějaké problémy, jak je vyřešil a výhled do toho, které úkoly bude obsahovat příští Sprint. Je zde také prostor na dotazy z řad zákazníků, na které Product Owner odpovídá. Pokud jsou otázky techničtějšího rázu, tak může odpovídat i někdo z vývojářů.

## 4. Sprint Retrospective

Sprint Retrospective probíhá v poslední den Sprintu. Retrospektiva probíhá skrze předpřipravenou šablonu v Azure DevOps, která obsahuje sloupce Appreciate, Continue, Start to do, Improve a Stop (ocenit, pokračovat, začít dělat, zlepšit, přestat). Jedná se o "virtuální tabuli", na kterou může kdokoliv ze členů Scrum týmu přilepit "lísteček" do jakéhokoliv z těchto sloupců. Následně jako Scrum Master procházím skrze jednotlivé lístky, a dávám možnost se každému vyjádřit a moderuji následně vzniklou diskusi. Pro efektivní využití retrospektivy je nutné, aby lidé přispívali se svými vstupy a podněty.

## 5. Sprint Planning

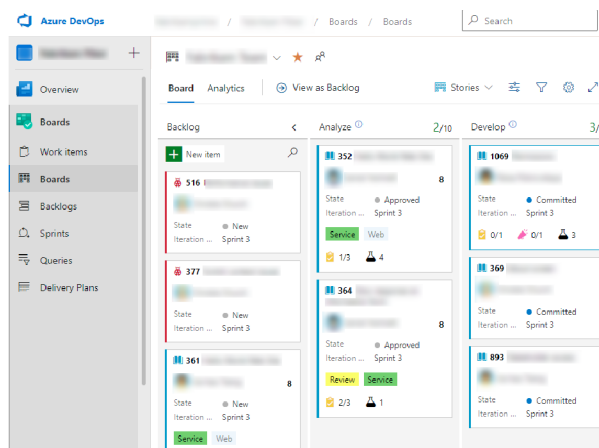
Sprint Planning probíhá v poslední den Sprintu. Je v kompetenci Product Ownera, utvoří se Sprint Goal a následně se podle něj vybírají pracovní položky z Product Backlogu, ze kterých je sestaven Sprint Backlog. Následně pracovníci, kterých se těchto úkoly týkají, ohodnotí podle množství času, které jim dle jejich uvážení zaberou. Práce se rozděluje převážně demokraticky, spoléhá se na zodpovědnost a kompetence jednotlivých členů týmu. U některých úkolů je tak možné, že se členové týmu mohou domluvit, kdo je splní. U úkolů, které vyžadují přesné zaměření, to možné není.

## 3.4 Prostředí Microsoft Azure DevOps

Azure DevOps je cloudový produkt společnosti Microsoft obsahující služby umožňující činnosti, jako je například projektové řízení, vývoj aplikací, testování, nebo automatizaci vývojových procesů.[22]

Pro práci v týmu se využívá zejména funkce Azure Boards, obsahující řadu podpůrných nástrojů pro agilní metodiky. Azure Boards umožňuje týmu provozovat Product Backlog, Sprint Backlog, Retrospective Board a další. Tyto nástroje jsou nesmírně důležité pro každodenní řízení vývojářského týmu používajícího agilní metodiku. Scrum Master, Product Owner a vývojáři mohou každý přistupovat ke svým zadaným pracovním úkolům, upravovat jejich pracovní status, vytvářet nové na základě zadání, nebo je mazat.

Azure Boards také nabízí možnost široké škály analytických nástrojů, jako například sledování Burdown Chart, Cycle Time, či Velocity Graph. V současné době se v týmu tyto analytické nástroje aktivně nevyužívají, a proto v rozsahu této bakalářské práce budou zmíněny pouze okrajově.



Obrázek 4: Ukázka Sprint Backlogu v prostředí Azure DevOps, Zdroj: autor

### 3.5 SWOT analýza současného stavu týmu

V rámci výzkumu jsem se pro počáteční zhodnocení situace rozhodl využít SWOT analýzu. SWOT analýza umožňuje posoudit úspěšnost firmy, nebo projektu zjištěním slabých a silných stránek, příležitostí a hrozeb. Pro kvalitní a objektivní SWOT analýzu je třeba vzít v potaz všechny aspekty týmu tak, aby nedošlo ke zkreslení zkoumaných údajů.[23]

<p><b>SILNÉ STRÁNKY</b></p> <ol style="list-style-type: none"> <li>1. Vysoká úroveň odbornosti a zkušeností s vývojem napříč týmem</li> <li>2. Dlouhodobá spokojenost zákazníků se Scrum týmem</li> <li>3. Skvělé mezilidské vztahy napříč týmem</li> </ol>	<p><b>SLABÉ STRÁNKY</b></p> <ol style="list-style-type: none"> <li>1. Nevýhody spojené s hybridním způsobem práce (nepozornost, pozdní příchody)</li> <li>2. Nedisciplinovanost při práci s Backlogem</li> <li>3. Zakořeněná stereotypní nedůvěra seniorních vývojářů k agilním metodikám</li> </ol>
<p><b>PŘÍLEŽITOSTI</b></p> <ol style="list-style-type: none"> <li>1. Nástup dedikovaného Scrum Mastera a pozvednutí správnosti a efektivity vykonávání metodiky Scrum</li> <li>2. Zlepšení dovedností a znalostí členů týmů dalším vzděláváním</li> <li>3. Možnost nezaniknutí týmu na základě nového projektu kalkulace rizika pro pojišťovací framework Solvency II</li> </ol>	<p><b>HROZBY</b></p> <ol style="list-style-type: none"> <li>1. Možná demotivace zkoušet něco nového/zlepšovat vzhledem k plánovanému rozpuštění týmu ve 2. čtvrtletí roku 2024</li> <li>2. Možná ztráta pracovní morálky po oznámení odchodu Product Ownera</li> </ol>

Tabulka 1: SWOT analýza týmu, Zdroj: autor

Jak je již uvedeno v cílech bakalářské práce, cílem je pozvednout úroveň vykonávání metodiky Scrum v tomto vývojářském týmu. Tým je vysoce produktivní, multidisciplinární, efektivní a je složen z velké části ze seniorních

zaměstnanců, ať už se jedná o programátory, nebo business analytiky. Pravidelně má vysoká hodnocení při všech žádostech o zpětnou vazbu u svých zákazníků.

### **3.6 Agile Maturity Matrix (Dotazník agilní vyspělosti)**

Ke zhodnocení poměrů v týmu jsem využil nástroje Agile Maturity Matrix, neboli dotazník (matici) agilní vyspělosti. V mém případě to byl dotazník obsahující 39 otázek z 5 různých kategorií (týmová dynamika, týmové prostředí, produkt, mechaniky agilního procesu a praktiky agilního vývoje). Nemohl jsem využít standardizovaný dotazník kvůli specifičnosti svého výzkumu. Proto jsem vyhledal dotazník, který nejlépe odpovídal mým potřebám.

Area	Current Level (0-4)	Target Level	Impeded (0)	In Transition (1)	Sustainable (2)	Agile (3)	Ideal (4)
Being Agile		4	Not yet doing or being Agile.	Doing the mechanics of a specific methodology that supports Agile such as Scrum, Kanban, SAFe, Enterprise Agility, XP, etc.	80% of the team can explain the workings and benefits of Agile and a specific methodology and believe in them. The team is making improvements on a regular basis	Working in an Agile manner	Actively pursuing new ways of working in an Agile manner
Morale		4	Regular occurrence of behaviours such as blame game, finger pointing, denial, anger, shouting, backstabbing, passive aggressiveness, and/or scapegoating. Active resistance to change. There is churn or people are frequently making references to quitting or how much they dislike their work or work environment.	There are still elements of the previous state, but there is steady progress away from those behaviors, problems are being actively addressed, and there is a general feeling that morale is improving	For the most part people are getting along and happy at work.	The team is generally happy, engaged, productive, and genuinely enjoy working together.	Most team members feel like this is one of the best teams they have ever worked on, they are excited to come in to work and are looking forward to the next day when they leave.
Teamwork		4	Non-existent	Teamwork is improving	At least 70% of the ideal column	At least 80% of the ideal column	Between each individual and every other individual on the team, each believes that the other has the necessary skills, has high integrity, wants and helps them to succeed, and is dependable.
Tuckman Stage (Time stages, which every team has to go through while forming to work effectively) Basically our team alignment and coordination, you have to gain it every time you lose or get a new team member		4	Forming. New team or a team member just left or was added.	Storming. The team is just starting to figure out how to work together and there is an unusual amount of conflict.	Norming. The team has mostly figured out how to work together as a team and is well on their way to high performance.	Have been performing consistently for at least 8 weeks	Have been performing consistently for the past 6 months

Obrázek 5: Ukázka formuláře Sprint Maturity Matrix, Zdroj: Formulář Agile Maturity Matrix

Každá z otázek dotazníku je škálovaná, od 0 do 4. Význam hodnot je následující (od nejhoršího k nejlepšímu):

0. Impeded (Zabrzděný)
1. In Transition (V přechodné fázi)
2. Sustainable (Udržitelný)
3. Agile (Agilní)

#### 4. Ideal (Ideální)

Dotazník byl využit jako nástroj sloužící k subjektivnímu posouzení situace v týmu z hlediska využívání agilní metodiky Scrum. Respondentů z řad členů týmů bylo 9 (2 .NET vývojáři, 2 Python vývojáři, 3 Informatica administrátoři, 2 testeři). Z odpovědí respondentů v dotazníku byl zhotoven aritmetický průměr. Aritmetický průměr je v případě hodnocení metodiky Scrum vhodným řešením, protože Scrum spoléhá na rovnocennost a autonomii členů týmu. Plná podoba dotazníku se nachází v příloze.

### 3.7 Pozorovací Sprint

Na počátku pozorovacího Sprintu jsem uspořádal pro vývojáře schůzku, při které jsem je informoval se záměrem mého výzkumu. Reakce byla pozitivní vzhledem k možnému přínosu mého výzkumu pro tým a možného zlepšení výkonu metodiky Scrum. V týmu jsou dobré mezilidské vztahy. V oblasti metodiky se zde ale vyskytují chyby, které následně představím.

#### 3.7.1 Vypozorované chyby a poznatky v průběhu pozorovacího Sprintu

Zde jsou uvedené chyby a poznatky, které se v deseti pracovních dnech pozorovacího Sprintu vyskytly. Pro lepší přehlednost jsou rozepsány do dnů, ve kterých jsem je vypozoroval.

##### 1. Den Sprintu

- Product Owner nezorganizoval Sprint Backlog při jeho vytváření a následně vznikly zmatky při organizaci.
- Vývojář neaktualizoval většinou část svých pracovních úkolů, došlo ke zkreslení vnímání provedené práce.
- Product Owner oznámil, že k 31. 4. končí na své pozici.

##### 4. Den Sprintu

- Část vývojářů se nepřipojila na Daily Scrum, i přes předem avizovaná témata, u kterých byla jejich přítomnost nutná.
- Daily Scrum začal se zpožděním.
- Příliš podrobné vysvětlování pracovních povinností ze strany vývojáře, které bylo nutné přerušit z důvodu plynulého chodu schůzky.
- Vývojáři aktualizovali své pracovní úkoly v Backlogu až průběhu meetingu.

### 7. Den Sprintu

- Pozorování, že členové týmu nedávají pozor při Daily Scrum. Poté co popíše obsah vlastní práce, nevěnují pozornost popisu ostatních členů týmu. Následně se tím zbytečně prodlužuje doba Scrum událostí.

### 10. Den Sprintu

- Backlog nemá jednotné formátování, střídají se velká/malá písmena, překlepy v anglickém jazyce, bylo by vhodné jednotlivé předměty barevně odlišit.

### Nezařazené poznatky:

- Znepokojení manažera, že Sprint Review prezentace dostatečně neinformuje zákazníky o stavu projektu.
- Retrospektiva je některými členy týmu označovaná jako monotónní, opakující se, bez originality.
- Ve Sprint Backlogu chybí komentáře u jednotlivých pracovních úkolů, působí to nejasnosti a nedorozumění.

Ze zběžného pohledu na seznam zjištěných pochybení můžeme usoudit, že situace ve vývojářském týmu není tak kritická. Tým opravdu pracuje dobře,

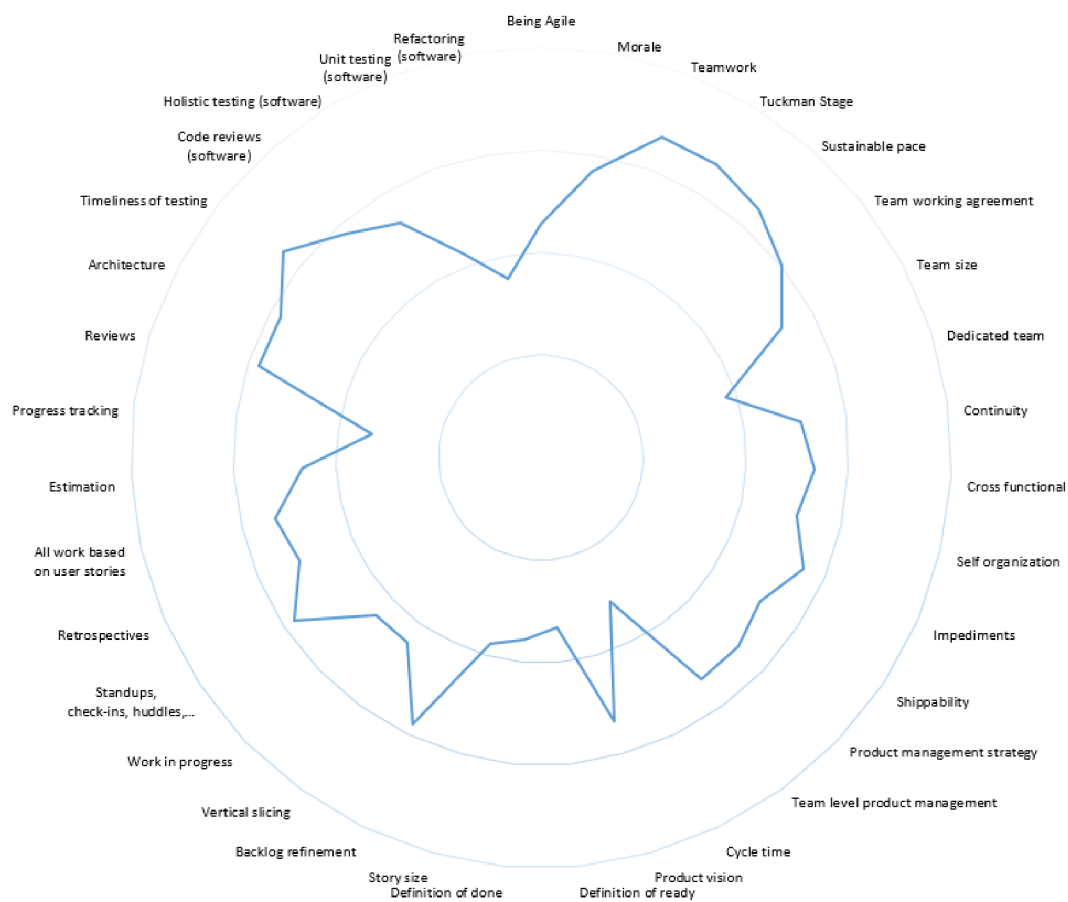


dodává skvělé výsledky a panují v něm dobré vztahy. Nicméně i tak je zde z hlediska agilní vyspělosti prostor pro zlepšení. Pro efektivitu týmu a správnost výkonu metodiky Scrum je nutná dobrá morálka a atmosféra v týmu. Zjištěná pochybení a podněty ke zlepšení lze zařadit do tří kategorií:

- Zapomínání na své povinnosti / Nedisciplinovanost
- Nezájem o dění v týmu / Nuda
- Položky s prostorem pro zlepšení

### **3.7.2 Výsledky Agile Maturity Matrix v pozorovacím Sprintu**

Výsledky dotazníku agilní vyspělosti jsem obdržel od 9 kolegů. Průzkum jsem vykonal v prvních dnech pozorovacího Sprintu, abych stihl pozorovat vnímání situace očima kolegů. Grafický model průzkumu vypadá následovně:



Obrázek 6: Grafický model vnímání Agile Maturity na počátku pozorování,  
Zdroj: autor

Dotazník agilní vyspělosti zabírá i oblasti, které Scrum Master nemá v silách ovlivnit. Záležitosti jako způsob, jakým se testuje, nebo jak funguje management společnosti jsou zcela mimo jeho pravomoce. Pro účely výzkumu jsou tedy nejpodstatnějšími částmi severovýchodní a jihozápadní části modelu. Zde jsou uvedeny zkoumané záležitosti z kategorie týmové dynamiky a mechanik agilního procesu. Informace jsou rozděleny do dvou tabulek podle kategorií. Pozornost je nutno věnovat odpovědím, které jsou na nižší úrovni než 3. Odpovědi, u kterých je aritmetický průměr větší než 3 jsou v pořádku a proto není třeba v současnosti činit nápravná opatření, ani je podrobněji komentovat. Výsledky subjektivního posouzení členy týmu z jsou následující:

#### Sekce "Týmová dynamika"

Otázka	Aritmetický průměr odpovědí
Vnímání celkové týmové agility	2,278
Morálka v týmu	2,833
Úroveň týmové práce	3,333
Tuckmanova fáze vývoje skupiny	3,333
Udržitelnost tempa vývoje	3,222
Úmluvy týkající se práce v týmu	3,000

Tabulka 2: Výsledky dotazníku ze sekce Týmová dynamika, Zdroj: autor

#### Vnímání týmové agility

Samotné vnímání celkové týmové agility není dle členů vývojářského týmu zcela ideální. Taková skutečnost může být spojena s mnoha faktory. Například tým nemá dosud dedikovaného Scrum Mastera, jeho povinnosti z části pokrývám já a Product Owner. Tým nikdy nebyl formálně seznámen se základy metodiky Scrum, nepodstoupil žádné školení, dalo by se tak říci, že jejich znalostní báze je velmi kolísavá. Bylo by vhodné do budoucna, aby tým podstoupil školení. Zaučování (onboarding) nových členů týmu není příliš standardizované, pro-

bíhá spíše způsobem ad hoc. Dá se tedy říci, že když do týmu nastoupí někdo nový, tak prakticky "stojí ve dveřích". Mojí povinností je seznamovat nové příchozí členy týmu s prostředím Microsoft Azure DevOps a ukázat jim základy práce s Backlogem, ať už Product, nebo Sprint. Jedním ze způsobů, kterým by bylo možné tento problém eliminovat, je zestandardizovat proces školení nových zaměstnanců, vytvořit na něj návod, nebo směrnici, podle které by bylo nutné se následně řídit.

### Morálka v týmu

Důvodů, proč byla hodnota morálky týmu nižší, než tři může být několik. Aplikace, kterou tým vyvíjí a udržuje bude ve 2. čtvrtletí roku 2024 nahrazena novější, vyvíjenou jiným týmem. Do nové aplikace budou integrovány veškeré funkce z této původní. Tým bude následně rozpuštěn a jeho členové vstoupí do týmů vyvíjejících jiný software. V době průzkumu také Product Owner oznámil, že ve své funkci bude končit k 31. 4. 2023.

### Sekce "Mechaniky agilního procesu"

Otázka	Aritmetický průměr odpovědí
Úroveň konání schůzky Daily Scrum	2,889
Úroveň konání Sprint Retrospective	2,556
Úroveň dokumentování veškeré práce do User Stories	2,667
Úroveň odhadování pracnosti	2,333
Úroveň sledování postupu vývoje projektu	1,667
Úroveň konání Sprint Review	2,889

Tabulka 3: Výsledky části dotazníku Mechaniky agilního procesu, Zdroj: autor

#### Úroveň konání Daily Scrum

Úroveň konání schůzky Daily Scrum je pod ideální hodnotou, v týmu již dlouhodobě panuje mírná nespokojenost se způsobem, kterým je tato událost vedena.

#### Úroveň konání Sprint Retrospective

Spokojenost s konáním Sprint Retrospective je na nízké úrovni. V nalezených

nedostatků při pozorovacím Sprintu je na tuto skutečnost několikrát upozorňováno. Členové týmu považují událost za neoriginální a monotónní.

### **Úroveň dokumentování veškeré práce do User Stories**

Další chybou, která se v dotazníku shoduje s pozorováním je, že úroveň dokumentování pracovních položek v Backlogu je na nízké úrovni. Tím může docházet k nedorozuměním při komunikaci a spolupráci s ostatními členy týmu, kteří nemusí nutně poznat, čeho se daná položka Backlogu týká. Objevující se problematikou jsou také nesprávně psané názvy položek či používání velkého množství odborných zkratk.

### **Úroveň odhadování pracnosti**

Odhadování pracnosti také není na uspokojivé úrovni. Špatné odhadování pracnosti může vést k následným problémům při vývoji, jakožto nárůst množství práce uprostřed iterace, může docházet k problémům při nastavování tempa práce, nebo také "nepříjemnému překvapení" před koncem Sprintu, kdy tým zjistí, že nestihl velký kus práce.

### **Úroveň sledování postupu vývoje projektu**

Úroveň sledování postupu vývoje projektu je na velmi nízké úrovni, je zde velký prostor pro zlepšení, ať už v ohledu vizualizace postupu, tvorbě směrnice/pravidel, nebo edukaci týmu, aby své pracovní položky věrohodně oceňoval a aktualizoval svůj postup.

### **Úroveň konání Sprint Review**

Úroveň konání Sprint Review je také pod ideální hodnotou. Mnoho členů týmu se již dlouhodobě vyslovuje pro změnu formátu schůzky a prezentace.

## **3.7.3 Vyvozené důsledky a návrhy na zlepšení**

Na základě pozorování 14denního Sprintu, prozkoumání odborných zdrojů a konzultaci kolegů, kteří jsou v oboru agilních metodik seniornější, jsem došel k následujícím opatřením, které se pokusím v opravném Sprintu dodržovat. Následně budu pozorovat, zda se některé vypořezované skutečnosti zlepší a bude docházet k většímu vnímání agilní vyspělosti členy vývojářského týmu.

1. **Uspořádání workshopu k zopakování úplných základů agilní metodiky Scrum**, sloužícího k sjednocení znalostní báze vývojářského týmu. V týmu jsou silné nedostatky a odlišná úroveň zkušenosti a znalosti metodiky Scrum. Každý z vývojářů nastoupil do projektu jindy a úroveň zaškolení, kterou podstoupili je u každého jiná. S podobným pořádáním workshopů pro kolegy z jiných týmů mám již zkušenosti v rámci svého zaměstnání.
2. **Uspořádání týmové porady sloužící k minimalizaci nedisciplinované práce s Backlogem**, kolegové nedostatečně oceňují své pracovní úkoly, neaktualizují úkoly, na kterých pracují, nebo které již mají hotové. Následně dochází k problémům se sledováním stavu projektu a organizaci práce v týmu. Porada bude zakomponována jako část workshopu týkajícího se základů metodiky Scrum.
3. **Tvorba nové prezentace na Scrum událost Sprint Review**, vývojáři i zákazníci shledávají současnou prezentaci jako nepříliš přehlednou a špatně reprezentující současný stav postupu projektu.
4. **Tvorba nového formátu Scrum události Sprint Retrospective**, členové týmu si stěžují, že je monotónní a neoriginální, stále opakující se dle stejného formátu. Je třeba vytvořit nový formát a snažit se jej pojmout poutavěji.
5. **Apel na věcnost diskuse, účast na Scrum události včas, povzbudit vývojářský tým k zapínání si webkamery**, jak je již zmíněno, spousta z problémů je způsobena obyčejnou nedisciplinovaností, nudou, nezájmem na lepší práci v týmu. Tým je třeba pozitivně motivovat a jejich monotónní práci nějak ozvláštnit. Je třeba pojmout společné schůzky originálně, s vtipem, více angažovat jednotlivé členy, aby se účastnili diskuse.
6. **Barevné okódování položek ve Sprint Backlogu, korektura chyb**,

**doplnění dokumentace**, aby bylo všem kolegům jasné, čeho se daná položka v Backlogu týká. Pozitivním důsledkem tohoto opatření bude i snazší tvorba prezentace pro Sprint Review, protože tvůrce prezentace bude vědět, čeho se daná věc týká.

Výše zmíněná nápravná opatření nyní podrobím SWOT analýze z důvodu zhodnocení jejich možného dopadu. Analýza nám umožní identifikovat silné a slabé stránky těchto opatření a následně můžeme lépe porozumět a usoudit, jaké bude mít jejich nasazení vliv.

SILNÉ STRÁNKY	SLABÉ STRÁNKY
<ol style="list-style-type: none"> <li>1. Workshopy pomohou sjednotit znalosti týmu a tím zlepšit výkon metodiky</li> <li>2. Změna formátu Sprint Review prezentace pomůže vylepšit zpětnou vazbu a transparentnost projektu</li> <li>3. Zvýšení originality konání události Daily Scrum může pomoci snížit monotónnost a zlepšit náladu v týmu</li> </ol>	<ol style="list-style-type: none"> <li>1. Workshop o základech metodiky může být pro seniornější členy týmu zbytečný</li> <li>2. Konání Sprint Retrospective mimo prostředí Azure DevOps může způsobit dodatečné finanční náklady</li> </ol>
PŘÍLEŽITOSTI	HROZBY
<ol style="list-style-type: none"> <li>1. Zlepšení produktivity týmu</li> <li>2. Zvýšení úrovně konání jednotlivých Scrum událostí</li> <li>3. Zvýšení morálky a nálady v týmu</li> </ol>	<ol style="list-style-type: none"> <li>1. Možnost odmítnutí opatření týmem</li> <li>2. Možnost nedosažení očekávaných výsledků</li> </ol>

Tabulka 4: SWOT analýza nápravných opatření, Zdroj: autor

Z výsledné analýzy můžeme usoudit, že potenciální zisk ze zavedení daných opatření je vyšší než případná ztráta. Záležitosti jako workshop sloužící ke sjednocení znalostní báze týmu a nová Review prezentace budou vítanou ná-



pravou současného stavu. Možné negativní důsledky zavedených opatření jsou možné dodatečné finanční náklady na provoz Sprint Retrospective. Opatření mohou být příležitostí, jak pozitivně ovlivnit náladu, morálku a samotné vnímání agility týmu. Velkým rizikem je možnost odmítnutí zavedení opatření týmem, jelikož v metodice Scrum je velký důraz na autonomii a rovnocennost jednotlivých členů týmu.

## 3.8 Opravný Sprint

Na počátku opravného Sprintu jsem uspořádal schůzku, kde jsem informoval vývojáře o nápravných opatřeních, která budou zavedena a následně si vyslechl jejich reakci a zpětnou vazbu. Reakce byla pozitivní a postupně jsem začal zavádět jednotlivá opatření.

### 3.8.1 Scrum Workshop

Workshop jsem naplánoval, jakožto schůzku na telekomunikační platformě Microsoft Teams. Osnova prezentace, kterou jsem plánoval přednášet a diskutovat o ní s týmem, je velmi podobná teoretické části této bakalářské práce. Mezi hlavní zdroje pro přípravu prezentace posloužila verze Scrum Guide pro rok 2020[6] a kniha Scrum od Josefa Myslína[1], která je jednou z nejlepších na tuto problematiku. Prošli jsme si definici metodiky Scrum, její základní hodnoty, role v metodice a jejich povinnosti a rozsah činnosti. Zvláštní pozornost jsem věnoval Scrum artefaktům a událostem, neboť ty sledávám při zkoumání problémů v týmu jako zvláště podstatné. Konference proběhla úspěšně a členové týmu mi poskytli pozitivní zpětnou vazbu i pár užitečných připomínek.

## What is Scrum?

Scrum is a **lightweight framework that helps people, teams and organizations generate value through adaptive solutions for complex problems.**

In a nutshell, Scrum requires a Scrum Master to foster an environment where:

1. A **Product Owner orders the work for a complex problem** into a Product Backlog.
2. The **Scrum Team turns a selection of the work into an Increment of value** during a Sprint.
3. The **Scrum Team and its stakeholders inspect the results and adjust for the next Sprint.**
4. **Repeat**



## Scrum Values

### Transparency

Important process and work must be visible to those performing the work as well as those receiving the work

### Inspection

Scrum artifacts and the progress toward agreed goals must be inspected frequently and diligently, to detect potentially undesirable variances or problems

### Adaption

If any aspects of a process create undue acceptable limits or if the resulting product is unacceptable, the process being applied or the materials being produced must be adjusted

## Who is accountable for what?

### Developers

- Creating a plan for the Sprint, the Sprint Backlog
- Instilling quality by adhering to a Definition of Done
- Adapting their plan each day toward the Sprint Goal
- Holding each other accountable as professionals.

### Product Owner

- Developing and explicitly communicating the Product Goal
- Creating and clearly communicating Product Backlog Items
- Ordering Product Backlog Items
- Ensuring that the Product Backlog is transparent, visible and understood.



## Scrum Master duties

- The Scrum Master serves the Scrum Team in several ways, including:

- Coaching the team members in self-management and cross-functionality
- Helping the Scrum Team focus on creating high-value increment to meet the Definition of Done
- Ensuring the removal of impediments to the Scrum Team's progress
- Ensuring that all Scrum events take place and are positive, productive, and kept within the timebox.

- The Scrum Master serves the Product Owner in several ways, including:

- Helping find techniques for effective Product Goal definition and Product Backlog management
- Helping the Scrum Team understand the need for clear and concise Product Backlog items
- Helping establish empirical product planning for a complex environment
- Facilitating stakeholder collaboration as requested or needed.

- The Scrum Master serves the organization in several ways, including:

- Leading, training, and coaching the organization in its Scrum adoption
- Planning and advising Scrum implementations within the organization
- Helping employees and stakeholders understand and enact an empirical approach for complex work
- Removing barriers between stakeholders and Scrum Teams.



Obrázek 7: Ukázka prezentace z workshopu, Zdroj: autor

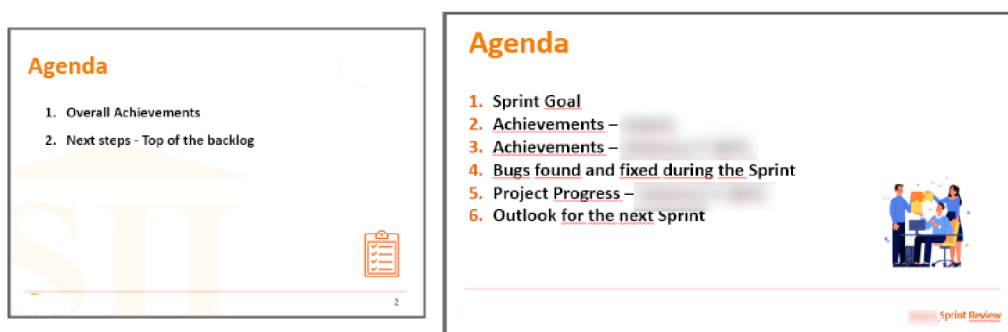
### 3.8.2 Týmová porada ohledně práce s Backlogem

Následně proběhla týmová porada, která se týkala tématu nedostatečně disciplinované práce s Backlogem, ať již Sprintovým do kterého zasahují přímo vývojáři, tak nízkou úroveň úpravy produktového Backlogu o který se stará Product Owner. Vzhledem k hybridní povaze práce se porada opět konala přes platformu Microsoft Teams, představil jsem kolegům několik bodů, které shledávám jako práci ztěžující. Mezi problémy patřily již zmíněné záležitosti, jako včasné (nebo vůbec) neaktualizování pracovních položek v Backlogu, špatné odhadování pracnosti a nejasné popisky týkající se jednotlivých "tasků". Domnívám se, že diskuse mohla pro některé z účastníků vést k objasnění důvodů, proč je nutné tyto věci pečlivě dodržovat.

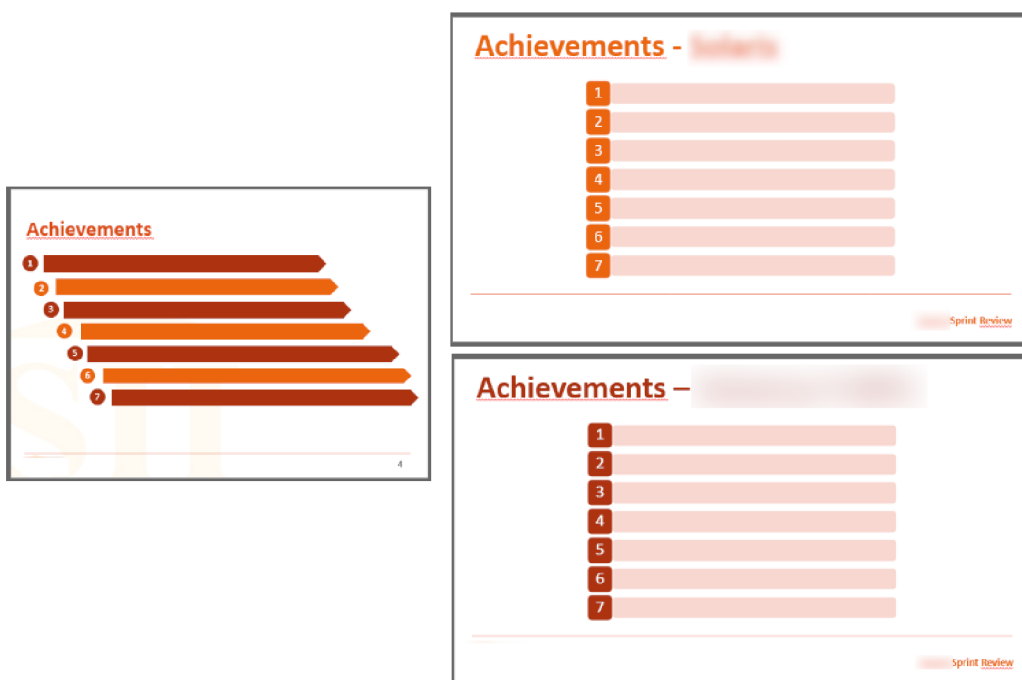
### 3.8.3 Tvorba nové prezentace na Sprint Review

Původní prezentace je používána již mnoho let a nevyhovujícím aspektem je, že dostatečně nereprezentuje postup projektem, vypadá zastarale. Bylo třeba

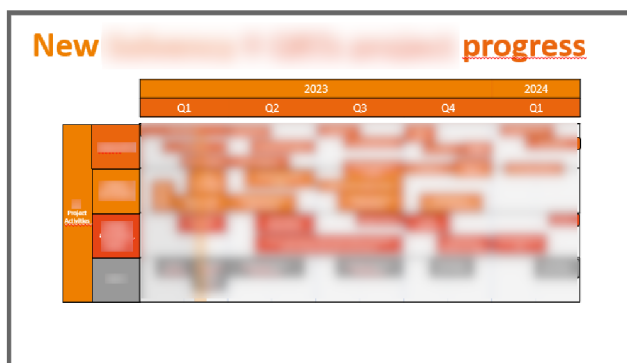
ji trochu zmodernizovat a upravit její strukturu. Prezentace je standardně v programu Microsoft PowerPoint, z původní verze jsem ponechal jen její osnovu a dle ní vytvořil prezentaci znovu. Nová verze prezentace pro Sprint Review začíná agendou schůzky, následně informuje, zda byl splněn Sprint Goal. Dále je rozdělena do dvou částí, kde se prezentují úspěchy na současném projektovém řešení a následně na novém produktovém řešení, které se vyvíjí odděleně. V původní prezentaci byly dosažené úspěchy prezentovány dohromady, což mohlo zákazníky mást. Následně se informuje o bugs, které byly nalezeny a opraveny v průběhu Sprintu. V poslední části se nachází nový diagram, který ukazuje současný procentuální stav dokončenosti projektu ve formě mapy, kde jsou vidět všechny záležitosti potřebné k dokončení projektu. Na závěr je zde nastíněn obsah příštího Sprintu. Původní prezentace byla také obtížná pro editaci, vzhledem k tomu, že veškeré grafické prvky byly ve formě tvarů a textových polí, takže mívala tendenci se "rozsypat". Nová verze prezentace funguje jako SmartArt, jedná se o výčet převedený do grafické podoby přímo PowerPoin-tem. Úprava je tak mnohem snazší. Dále byla převedena z dnes již zastaralého poměru stran 4:3 do poměru 16:9.



Obrázek 8: Porovnání obou prezentací, Zdroj: autor



Obrázek 9: Tvorba separátního slidu pro oba projekty, Zdroj: autor

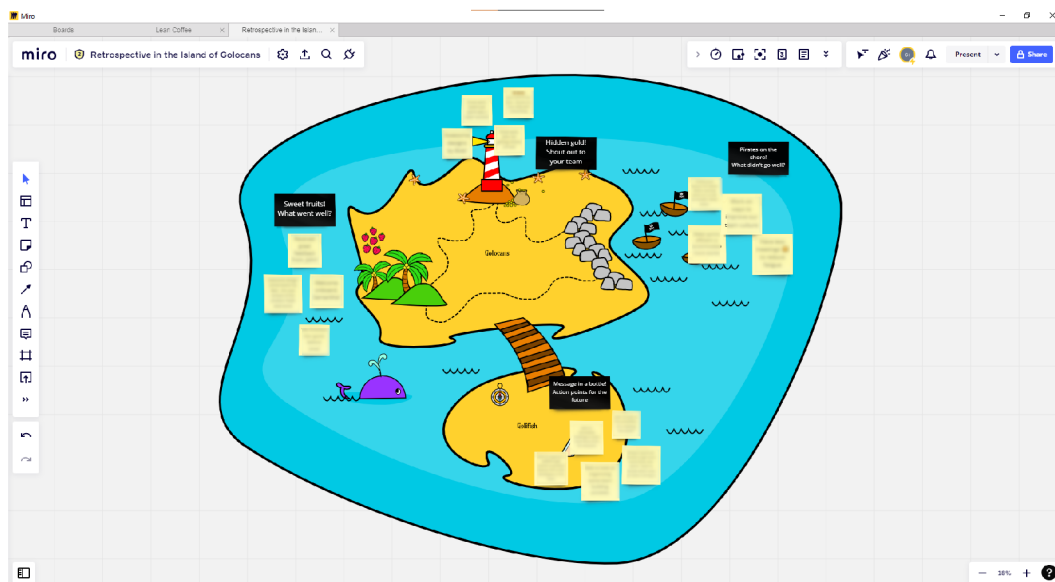


Obrázek 10: Nový snímek vizualizující postup projektem, Zdroj: autor

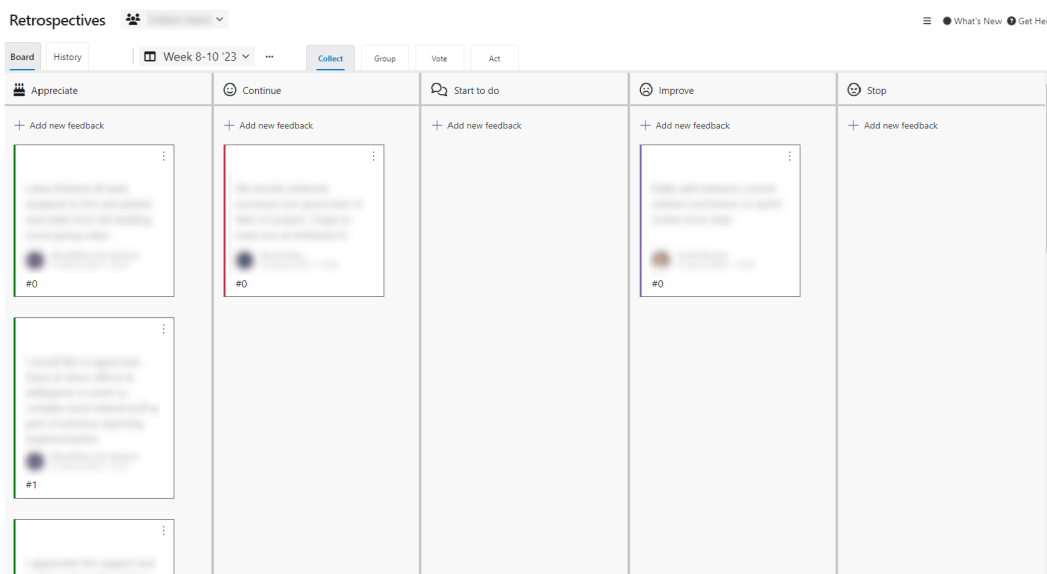
Při Sprint Review jsem s Product Ownerem požádal zákazníky o zpětnou vazbu a byla velmi pozitivní. Zákazníci byli především spokojeni s větší jasností, kde se produkt nachází a také s novou vizuální podobou prezentace.

### 3.8.4 Nový formát schůzky Sprint Retrospective

Vzhledem ke stížnostem na monotónnost schůzky Sprint Retrospective bylo jasné, že je třeba změnit formát, jakým schůzka probíhá. Naše společnost má k dispozici licenci softwaru Miro, kterým se dají tvořit různé virtuální tabule do kterých může přispívat více uživatelů najednou. V současné době probíhá Sprint Retrospective přes Microsoft Azure DevOps, které nabízí pouze možnost nazvat si sloupce v retrospektivě podle svého uvážení, jinak zde není příliš prostoru pro invenci. Proto jsem pro tuto zkušební retrospektivu zvolil program Miro, konkrétně jeho šablonu pro retrospektivy s názvem "Retrospective in the Island of Golocans"s pirátskou tematikou.



Obrázek 11: Pirátská retrospektiva v programu Miro, Zdroj: autor



Obrázek 12: Původní podoba retrospektivy v prostředí Azure DevOps, Zdroj: autor

Reakce na pirátskou retrospektivu byly velmi pozitivní. Vzhledem k tomu, že si novou aplikaci chtěli kolegové vyzkoušet a seznámit se s nabízenými funkcemi, bylo zde mnohem více podnětů k diskusi (ve formě lístků), než obvykle. Členové týmu působili více zaujatě a diskuse byla mnohem živější než obvykle, kdy je retrospektiva spíše formální záležitostí sloužící k ukončení Sprintu.

### 3.8.5 Apel na ohleduplnost

Toto opatření nebylo ani tak opatřením, jako spíše snahou o to být pozitivním vzorem. S týmem jsem probral tyto problémové záležitosti a na každou z nich upozornil. Na každou schůzku jsem si zapínal webkameru, aby na mě členové týmu viděli, když s nimi něco řeším. Aktualizoval jsem včas své pracovní položky v Backlogu a vzhledem k tomu, že většinu Scrum událostí pořádám, tak jsem na nich byl vždy včas. Někteří kolegové v týmu si následně webkameru začali v průběhu Sprintu zapínat, takže diskuse byly více osobní a v přátelském duchu. Vzhledem k tomu, že dále docházelo k diskusím, které nebyly věcné,

využil jsem pro tyto případy jednu z technik v metodice Scrum zvanou Parking Lot. V případě, že se diskuse oddálí od tématu, může Scrum Master navrhnout problém, který se probírá tzv. "zaparkovat". Takto zaparkované problémy se řeší pak pouze s lidmi, kterých se týkají, až po konci Scrum události. Využitím techniky se nikdo z členů týmu nemůže následně cítit, jako že mu není věnována pozornost, nebo že nedostal příležitost mít slovo. Problém se vyřeší až po konci schůzky s relevantními lidmi, kdy už diskuse může trvat, jak dlouho chtějí a neomezuje to ostatní členy týmu. K pozdním příchodům v průběhu Sprintu vůbec nedošlo.

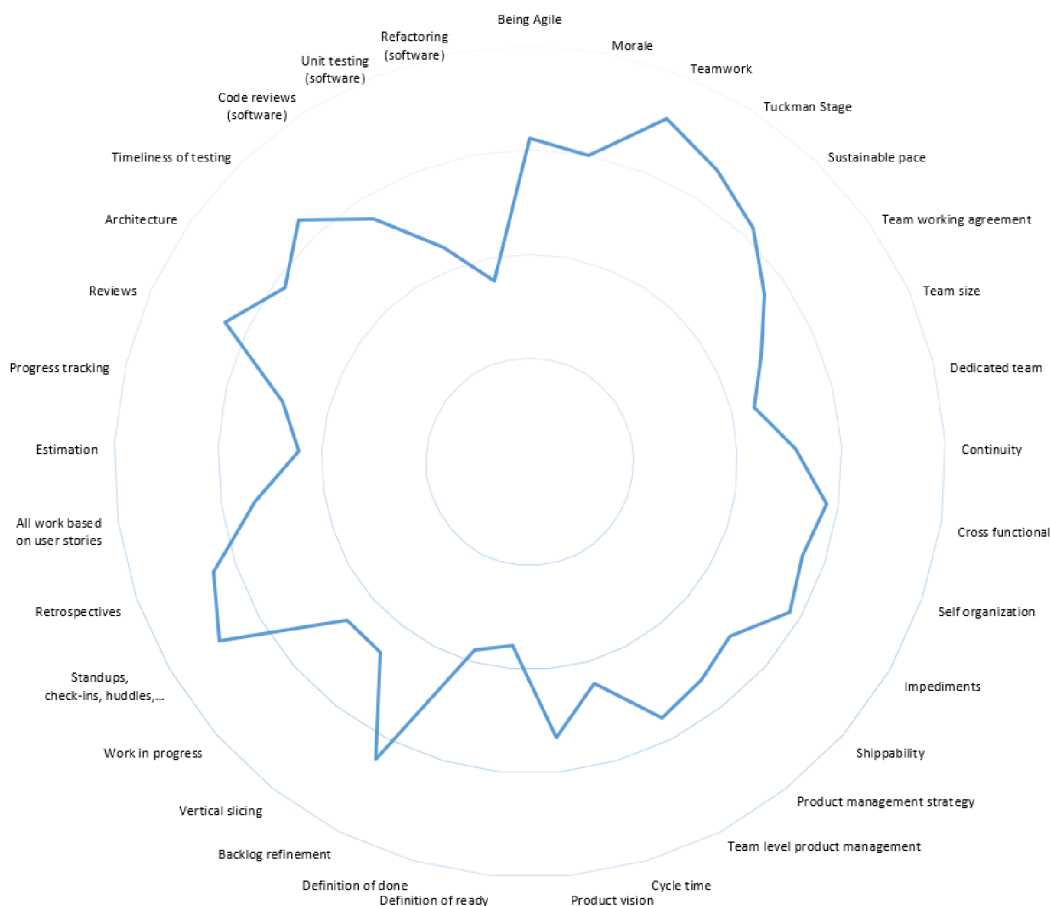
### **3.8.6 Barevné okódování Backlogu, doplnění komentářů**

Po zavedení tohoto opatření se původně prázdná místa v Backlogu začala plnit stručnými popisky a komentáři, čeho se vlastně dané Product Backlog Items týkají. Dále začal být daleko větší důraz na formální úpravu a pravopisné chyby. V týmu se používá mnoho zkratk, proto jsem využil funkce Wiki v Azure DevOps a vytvořil jsem zde jednoduchý slovník. Byl vyplněn pojmy z technického, business a právního odvětví. Toto opatření bylo zvláště oceňované novými kolegy, kteří nebyli se všemi zkratkami seznámení a jejich používání pro ně bylo matoucí. Takto jsou všechny užitečné informace k nalezení na jednom místě. Samotné obarvení jednotlivých PBI v Backlogu podle tématických celků už byl pouze příjemný bonus, který značně zvýšil přehlednost Backlogu.



### 3.9 Vyhodnocení účinnosti nápravných opatření

Po závěrečné Sprint Retrospektivě jsem opět předložil respondentům dotazník, tentokrát ovšem po zavedení a dodržování všech nápravných opatření. Respondenti odpovídali na stejnou sadu otázek jako na začátku pozorovacího Sprintu. Z výsledku dotazníkového šetření jsem opět vyhotovil grafický model.



Obrázek 13: Grafický model vnímání Agile Maturity po konci opravného Sprintu, Zdroj: autor

Jak můžeme pozorovat, v severovýchodní a jihozápadní části modelu došlo ke zlepšení. Došlo ke zlepšení z hlediska vnímání celkové týmové agility, což se dá přisuzovat mé aktivní snaze o zlepšení situace týkající se výkonu metodiky v týmu. Následně také stoupla morálka, což je možným důsledkem aktivnějšího

pořádání Scrum schůzek, snahy o větší originalitu jednotlivých událostí a zkušební retrospektivy. Při rozhovoru s kolegy vyplynulo, že se cítí lépe, protože "se někdo snaží řešit věci, které se už dlouho dělají špatně". Zavedená opatření bohužel neměla vliv na otázky z kategorie Tuckmanovy fáze vývoje skupiny. Dále výsledky ze sekce Udržitelnost tempa vývoje a Úmluvy týkající se práce v týmu byly lehce horší, než na počátku zkoumání. Dá se to odůvodnit tím, že zavedená opatření měla vliv na dosavadní způsob práce, a bude trvat ještě nějaký čas, než se na ně členové týmu dostatečně adaptují.

### Závěrečné výsledky ze sekce "Týmová dynamika"

Otázka	Aritmetický průměr odpovědí	Rozdíl
Vnímání celkové týmové agility	3,111	+0,833
Morálka v týmu	3,000	+0,166
Úroveň týmové práce	3,556	+0,222
Tuckmanova fáze vývoje skupiny	3,333	+0,000
Udržitelnost tempa vývoje	3,111	-0,111
Úmluvy týkající se práce v týmu	2,779	-0,222

Tabulka 5: Výsledky závěrečného dotazníku ze sekce Týmová dynamika, Zdroj: autor

V části týkající se mechanik agilního procesu, můžeme pozorovat výrazné zvýšení úrovně konání schůzky Daily Scrum, což se dá také přisoudit větší snaze o originalitu konání události, větší přehlednosti Backlogu a také některým základním pravidlům vysvětleným na Scrum Workshopu. Členové týmu také ocenili zkušební retrospektivu v aplikaci Miro, je vidět, že originální způsob retrospektivy je něčím, s čím je vhodné pokračovat. Opatření bohužel neměla vliv na úroveň dokumentování veškeré práce do User Stories a také měla lehce negativní vliv na odhadování pracnosti. Je možné, že část Scrum Workshopu týkající se odhadování pracnosti nebyla dostatečně vysvětlena. S novou prezentací a vizualizací postupu projektem na Sprint Review také stouply hodnoty

u otázek týkajících se úrovně sledování postupu vývoje produktu a úrovně konání Sprint Review. S novým formátem Sprint Review byli také spokojeni samotní zákazníci, nejen členové týmu.

### Závěrečné výsledky ze sekce "Mechaniky agilního procesu"

Otázka	Aritmetický průměr odpovědí	Rozdíl
Úroveň konání schůzky Daily Scrum	3,444	+0,556
Úroveň konání Sprint Retrospective	3,222	+0,667
Úroveň dokumentování veškeré práce do User Stories	2,667	+0,000
Úroveň odhadování pracnosti	2,222	-0,111
Úroveň sledování postupu vývoje projektu	2,444	+0,779
Úroveň konání Sprint Review	3,222	+0,333

Tabulka 6: Výsledky části závěrečného dotazníku Mechaniky agilního procesu,

Zdroj: autor

## 4 Závěr

Hlavním cílem této bakalářské práce byla tvorba užitečných opatření, sloužících ke zvýšení správnosti a efektivity výkonu metodiky Scrum ve Scrum týmu. V teoretické části jsem popsal proces vývoje software, model vyspělosti procesu a následně tradiční a agilní metodiky v čele s metodikou Scrum, kterou se tato práce zabývá. Tato část byla užitečným úvodem do světa agilních metodik.

V první polovině praktické části jsem se zabýval pozorováním současných poměrů týkajících se výkonu metodiky a následným zapisováním si chyb, které by byly třeba opravit, nebo záležitostí u kterých byl prostor na zlepšení. Došel jsem ke 12 chybám, které jsem zakategorizoval a následně vyhotovil 6 opravných opatření určených k nápravě. Dále jsem vykonal dotazníkové šetření, sloužící k zjištění pohledu členů Scrum týmu na současnou situaci.

V druhé polovině praktické části jsem se soustředil na výkon a nasazení daných nápravných opatření, postupně jsem je implementoval. Některá hned (apel na slušnost, týmová porada), další z jejich podstaty později (nová prezentace na Sprint Review, nová Sprint Retrospective, Workshop). Implementace daných opatření nebyla místy jednoduchá, ale v závěrečném dotazníkovém šetření došlo k významnému zlepšení u většiny sledovaných oblastí. Zajímavým poznatkem je, že k nejvýraznějšímu zlepšení došlo u vnímání celkové týmové agility, úrovně konání schůzky Daily Scrum a Sprint Retrospective.

Hlavním přínosem bakalářské práce je 6 nápravných opatření sloužících ke zlepšení výkonu metodiky, které může implementovat libovolný Scrum Master ve svém vývojářském týmu. Tým je více koordinovaný, postup projektem je daleko věrněji zobrazen. V Backlogu jsou okomentované pracovní položky, lidé chodí včas na schůzky, více diskutují, angažují se. Schůzky jsou vedeny originálně. Přínos nápravných opatření je zde znázorněn tabulkou:

Pořadí	Opatření	Výsledek
1.	Uspořádání workshopu k zopakování úplných základů agilní metodiky Scrum, sloužícího ke sjednocení znalostní báze vývojářského týmu	Zvýšení celkové týmové agility, úrovně Daily Scrum, sjednocení znalostní báze týmu
2.	Uspořádání týmové porady sloužící k minimalizaci nedisciplinované práce s Backlogem	Věrohodné zobrazování aktuálního stavu prací na projektu
3.	Tvorba nové prezentace na Sprint Review	Zvýšení úrovně sledování postupu vývoje projektu a konání události Sprint Review, spokojenost členů týmu a zákazníků s novou formou prezentace
4.	Nový formát Scrum události Sprint Retrospective	Zvýšení úrovně události Sprint Retrospective, větší angažovanost členů týmu při diskusi
5.	Apel na věcnost diskuse, účast na Scrum událostech včas, povzbuzení vývojářského týmu k zapínání si webkamery	Zvýšení úrovně Daily Scrum, vnímání celkové týmové agility a morálky, větší koordinovanost členů týmu, omezení pozdních příchodů
6.	Barevné okódování položek ve Sprint Backlogu, korektura chyb	Větší přehlednost Sprint a Product Backlogu, Backlog doplněn o dokumentaci, snazší průběh náboru nových členů týmu

Tabulka 7: Hlavní přínos bakalářské práce v tabulce, Zdroj: autor

Zvyšování Scrum Maturity je během na dlouhou trať, ale na cestě k samoorganizujícímu a agilně na změny reagujícímu týmu byl tento výzkum dobrým začátkem.

## Seznam použité literatury a zdrojů

- [1] MYSLÍN, Josef. *Scrum: průvodce agilním vývojem softwaru*. Brno: Computer Press, 2016. ISBN 978-80-251-4650-7.
- [2] ROYCE W. *Proceedings: 9th International Conference on Software Engineering*, 30 April 1987, Monterey, California, USA. Washington, D.C.: IEEE Computer Society Press.
- [3] KRÁTKÝ, Tomáš a Bohumír ZOUBEK. Softwarový proces. In: *Webové stránky fakulty elektrotechnické ČVUT* [online]. Praha [cit. 2023-03-27]. Dostupné z: [https://cw.fel.cvut.cz/old/\\_media/courses/a4m33sep/-prednasky/12\\_softwareprocess.pdf](https://cw.fel.cvut.cz/old/_media/courses/a4m33sep/-prednasky/12_softwareprocess.pdf)
- [4] *Manifest Agilního vývoje software* [online]. USA: Ward Cunningham, 2001 [cit. 2022-08-12]. Dostupné z: [agilemanifesto.org/iso/cs/manifesto.html](http://agilemanifesto.org/iso/cs/manifesto.html)
- [5] *Comprehensive Guide to the Agile Manifesto* [online]. USA: Smartsheet, 2016 [cit. 2022-08-12]. Dostupné z: <https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto>
- [6] *The 2020 Scrum Guide* [online]. USA: Ken Schwaber & Jeff Sutherland, 2020 [cit. 2022-12-02]. Dostupné z: <https://scrumguides.org/scrum-guide.html>
- [7] *What is Pig and Chicken in Scrum?* [online]. USA: visual-paradigm.com, 2022 [cit. 2023-01-27]. Dostupné z: <https://www.visual-paradigm.com/scrum/scrum-pig-and-chicken/>
- [8] What is a Product Owner?. In: *Scrum.org* [online]. [cit. 2023-03-27]. Dostupné z: <https://www.scrum.org/resources/what-is-a-product-owner>
- [9] RAHMAN, Jay. Can the Product Owner also be the Scrum Master?. In: *Scrum.org* [online]. Velká Británie, 2021 [cit. 2023-03-27].

- Dostupné z: <https://www.scrum.org/resources/blog/can-product-owner-also-be-scrum-master>
- [10] ŠOCHOVÁ, Zuzana. *Skvělý ScrumMaster*. Brno: Computer Press, 2018. ISBN 978-80-251-4927-0.
- [11] What is a Product Backlog?. In: *Productboard.com* [online]. [cit. 2023-03-27]. Dostupné z: <https://www.productboard.com/glossary/product-Backlog/>
- [12] FIRLIT, Magdalena. The Product Goal explained. In: *Scrum.org* [online]. Polsko, 2022 [cit. 2023-03-27]. Dostupné z: <https://www.scrum.org/resources/blog/product-goal-explained>
- [13] What is a Sprint Backlog?. In: *Scrum.org* [online]. [cit. 2023-03-27]. Dostupné z: <https://www.scrum.org/resources/what-is-a-Sprint-Backlog>
- [14] OVEREEM, Barry. The 11 Advantages of Using a Sprint Goal. In: *Scrum.org* [online]. Nizozemsko, 2016 [cit. 2023-03-27]. Dostupné z: <https://www.scrum.org/resources/blog/11-advantages-using-Sprint-goal>
- [15] What is an Increment?. In: *Scrum.org* [online]. [cit. 2023-03-27]. Dostupné z: <https://www.scrum.org/resources/what-is-an-increment>
- [16] What is a Definition of Done?. In: *Scrum.org* [online]. [cit. 2023-03-27]. Dostupné z: <https://www.scrum.org/resources/what-definition-done>
- [17] PBI (Product Backlog Item) vs. User Story: When and How to Use them in Software Project Management. In: *Unosquare.com* [online]. [cit. 2023-03-27]. Dostupné z: <https://www.unosquare.com/blog/pbi-product-Backlog-item-vs-user-story-when-and-how-to-use-them-in-software-project-management/>
- [18] SUTHERLAND, Jeff. The Origin of The Daily Stand-up. In: *LinkedIn* [online]. USA: LinkedIn, 2014 [cit. 2023-02-08]. Dostupné z:

[www.linkedin.com/pulse/20140926150354-136414-the-origin-of-the-daily-stand-up/](https://www.linkedin.com/pulse/20140926150354-136414-the-origin-of-the-daily-stand-up/)

- [19] What are the 3 Important Questions in Daily Scrum?. In: *Visual Paradigm* [online]. USA: Visual Paradigm, 2022 [cit. 2023-02-08]. Dostupné z: <https://www.visual-paradigm.com/scrum/3-daily-scrum-questions/>
- [20] MELNIKOVA, Alexandra. Daily Scrum Challenges & Solutions. In: *Timboretro.com* [online]. 2021 [cit. 2023-03-27]. Dostupné z: <https://timboretro.com/blog/daily-scrum-challenges-and-solutions.html>
- [21] WIKSTRAND, Greger. The Scrum Maturity Model. In: *Gregerwikstrand.com* [online]. 2012 [cit. 2023-03-27]. Dostupné z: <https://www.gregerwikstrand.com/the-scrum-maturity-models/>
- [22] What is Azure DevOps?. In: *Microsoft Learn* [online]. USA: Microsoft, 2022 [cit. 2023-02-21]. Dostupné z: <https://learn.microsoft.com/en-us/azure/devops/user-guide/what-is-azure-devops?view=azure-devops>
- [23] DOMANSKÁ, Lucie. Rizika a příležitosti v podnikání pomůže odhalit SWOT analýza. In: *podnikatel.cz* [online]. Praha: Internet info, 2008 [cit. 2023-02-28]. Dostupné z: <https://www.podnikatel.cz/clanky/rizika-a-prilezitosti-odhali-swot-analyza/>



## Seznam tabulek

1	SWOT analýza týmu, Zdroj: autor . . . . .	36
2	Výsledky dotazníku ze sekce Týmová dynamika, Zdroj: autor . .	43
3	Výsledky části dotazníku Mechaniky agilního procesu, Zdroj: autor . . . . .	44
4	SWOT analýza nápravných opatření, Zdroj: autor . . . . .	48
5	Výsledky závěrečného dotazníku ze sekce Týmová dynamika, Zdroj: autor . . . . .	58
6	Výsledky části závěrečného dotazníku Mechaniky agilního pro- cesu, Zdroj: autor . . . . .	59
7	Hlavní přínos bakalářské práce v tabulce, Zdroj: autor . . . . .	61

## Seznam obrázků

1	Vodopádový model vývoje software, Zdroj: autor, vlastní zpracování obecného schéma vodopádového modelu vývoje software . . . . .	15
2	Jednoduchý iterační cyklus, Zdroj: autor, vlastní zpracování obecného schéma iteračního cyklu . . . . .	16
3	Burndown Chart, Zdroj: autor . . . . .	29
4	Ukázka Sprint Backlogu v prostředí Azure DevOps, Zdroj: autor . . . . .	35
5	Ukázka formuláře Sprint Maturity Matrix, Zdroj: Formulář Agile Maturity Matrix . . . . .	38
6	Grafický model vnímání Agile Maturity na počátku pozorování, Zdroj: autor . . . . .	42
7	Ukázka prezentace z workshopu, Zdroj: autor . . . . .	51
8	Porovnání obou prezentací, Zdroj: autor . . . . .	52
9	Tvorba separátního slidu pro oba projekty, Zdroj: autor . . . . .	53
10	Nový snímek vizualizující postup projektem, Zdroj: autor . . . . .	53
11	Pirátská retrospektiva v programu Miro, Zdroj: autor . . . . .	54
12	Původní podoba retrospektivy v prostředí Azure DevOps, Zdroj: autor . . . . .	55
13	Grafický model vnímání Agile Maturity po konci opravného Sprintu, Zdroj: autor . . . . .	57

## Přílohy

CD obsahující:

- Plné znění bakalářské práce ve formátu PDF
- Dotazník "Agile Maturity Matrix" ve formátu XLS