

**VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU**

Nárožní 2600/9a, 158 00 Praha 5

# **DIPLOMOVÁ PRÁCE**



**MANAGEMENT FIREM**

# VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

## NÁZEV DIPLOMOVÉ PRÁCE/TITLE OF THESIS

Zavedení agilní metody Scrum v IT organizaci

## TERMÍN UKONČENÍ STUDIA A OBHAJOBA (MĚSÍC/ROK)

8/2020

## JMÉNO A PŘÍJMENÍ STUDENTA / STUDIJNÍ SKUPINA

Bc. Ludmila Buschová / MF27

## JMÉNO VEDOUCÍHO DIPLOMOVÉ PRÁCE

doc. Ing. Lucie Vnoučková, Ph.D.

## PROHLÁŠENÍ STUDENTA

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

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

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

Datum a místo: 6. 7. 2020 v Praze

## PODĚKOVÁNÍ

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

# VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Národní 2600/9a, 158 00 Praha 5

## SOUHRN

### 1. Cíl práce:

Hlavním cílem diplomové práce je navrhnout doporučení pro odstranění chyb a nedostatků, které byly zjištěny v průběhu vlastního výzkumu u implementace agilní metody Scrum na pilotním projektu v IT organizaci.

### 2. Výzkumné metody:

V teoretické části byla provedena rešerše sekundárních zdrojů, jako jsou odborné tištěné nebo elektronické monografie a články z oboru agilního a projektového managementu. Elektronické zdroje byly čerpány z knihovny Akademie věd ČR, z databáze ProQuest a z dalších internetových zdrojů. V praktické části byla provedena metoda kvalitativního výzkumu prostřednictvím strukturovaných rozhovorů ve dvou fázích. Výzkum se zaměřoval převážně na tyto oblasti: příprava na zahájení agilní transformace, role a sestavování týmu, meetingy a aktivity, praktiky a artefakty, agilní programování a agilní smlouva. Pro tyto oblasti bylo také stanoveno několik předpokladů o chybách a nedostacích, které by se měly začít projevovat během agilní transformace ve sledovaném období. Rozhovorů se účastnil Scrum Master, Product Owner a pět vývojářů. Další data byla získána při rozhovorech pomocí sebehodnocení a hodnocení: respondenti hodnotili průběh vývoje týmu (dle Tuckmanova modelu), vývojáři provedli sebehodnocení i vzájemné hodnocení, Scrum Master hodnotil jednotlivé členy týmu a Product Owner hodnotil Development tým jako celek. Tato hodnocení se týkala oblastí: týmovost, nasazení a aktivita. Výstupy z pilotního projektu, které byly zaznamenávány prostřednictvím softwarového nástroje, byly podrobeny obsahové analýze. Další obsahová analýza dokumentu byla provedena u smlouvy k pilotnímu projektu mezi firmou XY a jejím klientem. Vyhodnocení probíhalo komparací výsledků výzkumu s informacemi a doporučeními, které jsou pro dané oblasti popsány v teoretické části této diplomové práce. Následně byla formulována doporučení pro odstranění chyb a nedostatků, které byly při výzkumu zjištěny. Uvedená doporučení se opírají o teoretickou část této diplomové práce, konkrétně kapitoly: Metoda Scrum, Metoda Kanban a Implementace agilní metody Scrum.

### 3. Výsledky výzkumu/práce:

Výsledky výzkumu potvrdily některé ze stanovených předpokladů o chybách a nedostacích, se kterými se zkoumaný tým měl začít potýkat po zavedení metody Scrum:

- byl potvrzen předpoklad, že nebude věnována dostatečná pozornost přípravě členů Development týmu na agilní styl vývoje a metodu Scrum;
- u junior programátora byl prokázán předpoklad, že tým nebude vhodně sestaven z pohledu potřebných zkušeností a znalostí;
- u jednoho ze členů Development týmu byl ověřen předpoklad, že tým bude nevhodně sestaven z pohledu přístupu k agilnímu způsobu vývoje;
- potvrdil se i předpoklad, že bude docházet ke konfliktům ve formujícím se týmu a také byly prokázány problémy s adaptací na nové role ve Scrum týmu;
- na počátku výzkumu byl také potvrzen předpoklad, že schůzky týmu budou časově náročné a jejich průběh bude komplikovaný. Další fáze výzkumu prokázala určité zlepšení v této oblasti, ale zároveň potvrdila předpoklad, že tým bude opomíjet některé typy aktivit stanovené metodou Scrum;
- předpoklad, že nebude dosahováno cílů Sprintů byl potvrzen pouze částečně – během čtyř Sprintů nebylo cíle dosaženo dvakrát. Zcela se potvrdily předpoklady, že nebude jasně určena Definition of Done i to, že se v týmu objeví velká rozpracovanost úkolů;
- první fáze výzkumu také prokázala předpoklad, že nebudou implementovány agilní praktiky programování, ale druhá fáze výzkumu již zaznamenala zavedení alespoň některých vhodných praktik;
- poslední potvrzený předpoklad byl, že mezi firmou XY a jejím klientem byla uzavřena smlouva, která neodpovídá agilnímu způsobu spolupráce.

Naopak výsledky výzkum neprokázaly předpoklad, že Scrum Master a Product Owner nebudou mít dostatečnou přípravu na zahájení agilní transformace a nepotvrdil se u nich ani předpoklad, že bude docházet k problémům s adaptací na nové role. Rovněž předpoklady, že User Stories budou pro vývojáře nedostatečně popsané nebo nesrozumitelné nebyl potvrzen. A rovněž to, že tým bude používat nevhodně rozloženou Scrum tabuli, výzkum také nepotvrdil. U vývojářů se také neprokázala předpokládaná neochota účastnit se schůzek definovaných metodou Scrum.

# VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Národní 2600/9a, 158 00 Praha 5

## 4. Závěry a doporučení:

Pro předpoklady, které byly pozitivně potvrzeny během výzkumu na pilotním projektu, byla navržena níže uvedená doporučení pro odstranění zjištěných chyb a nedostatků:

- pro přípravu vývojářů na agilní transformaci a metodu Scrum byl doporučen agilní nástroj Coaching Backlog, který by měl Scrum Master využívat k vytvoření krátkodobého plánu pro školení, koučování nebo poradenství. Do Coaching Backlogu byla navržena jako první témata: školení Standup meeting, metody pro odhady časové náročnosti (T-Shirt Sizes, Planning Poker), praktiky pro retrospektivní schůzky;
- pro sestavování týmu z pohledu potřebných znalostí a zkušeností s ohledem na současného junior programátora bylo doporučeno využít agilních praktik programování Pair Programming (párové programování) a Code Review (kontrola kódu);
- pro nedostatky v oblasti sestavení týmu z pohledu přístupu k agilnímu stylu vývoje bylo doporučeno řešit příčiny disfunkce týmu pomocí pyramidového konceptu, který prověřuje 5 příčin selhávání týmů a pro jednotlivé úrovně příčin byly uvedeny oblasti, ve kterých lze hledat řešení, včetně příkladů využití u zkoumaného týmu;
- v oblasti rolí a vývoje týmu z pohledu Tuckmanova modelu byl pro Scrum Mastera doporučen přístup a zaměření se na vybrané oblasti dle vývojové fáze týmu, ve které se celý tým aktuálně nachází nebo nacházet pravděpodobně bude v budoucnosti. Pro současnou Storming fázi bylo Scrum Masterovi doporučeno několik aktivit v oblasti komunikace, spolupráce, konfliktů a odpovědnosti. Pro sledování budoucího vývoje bylo doporučeno opakovat hodnocení stavu vývoje týmu při změnách v týmu nebo po třech měsících;
- pro odstranění nedostatků u meetingů a aktivit byla navržena doporučení pro Standup a Sprint Retrospective schůzky. Pro Standup schůzky jako základní opatření bylo navrženo dodržování správného formátu této aktivity pomocí tří základních otázek. U retrospektivní schůzky bylo Scrum Masterovi navrženo, aby tuto aktivitu pravidelně zajišťoval a aby s týmem využíval jednu z pěti doporučených pěť praktik: retrospektiva s lepítky, časová osa, fishbone (rybí kost), loď nebo ESVP. Pomocí první z těchto metod byl sestaven seznam témat, kterými by se tým mohl zabývat na retrospektivní schůzce;
- pro oblast praktik a artefaktů byla navržena doporučení týkající se cíle Sprintu a Definition of Done: pro dosažení cílů sprintů byla navržena doporučení týkající se metody odhadů pracnosti, Standup schůzek a Work in Progress Limits (omezení probíhající práce) a pro chybějící Definition od Done byla doporučena její výchozí formulace obsahující čtyři základní položky;
- v rámci Agilního programování bylo zkoumanému týmu doporučeno začít využívat dvě další metody – Pair Programming (párové programování) a Code Review (kontrola kódu);
- v oblasti Agilní smlouvy bylo firmě XY doporučeno, aby za pomoci vzorové agilní smlouvy zahájila jednání a pokusila se na další období se svým klientem uzavřít agilní typ kontraktu, který by naradil současnou klasickou smlouvu.

## KLÍČOVÁ SLOVA

Agilní management, agilní metody, agilní transformace, Scrum, Kanban.

# VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

## SUMMARY

### 1. Main objective:

The main objective of the thesis is to propose recommendations to remove errors and deficiencies identified during the research of the implementation of the Scrum agile method in a pilot carried out by an IT organization.

### 2. Research methods:

The theoretical part included the research of secondary resources such as professional (printed and electronic) monographs and articles focused on leadership. Electronic resources came from the Library of the Academy of Sciences of the Czech Republic, the ProQuest database as well as other internet resources. The practical part included the method of qualitative research through structured interviews in two stages. The research primarily focused on the following areas: the preparation for the start of agile transformation, the role and building of a team, meetings and activities, practices and artifacts, agile programming, an agile contract. Participants in the Interviews included Scrum Master, Product Owner, and five developers. Some more data was obtained in interviews using self-evaluation and evaluation: the respondents evaluated the process of team development (under the Tuckman model), the developers conducted both self-evaluation and peer review, while the Scrum Master concentrated on reviewing the performance of individual team members, the Product Owner did so for the Development team as a whole. These evaluations involved areas such as teamwork, deployment, and activity. The deliverables from the pilot, recorded with a software tool, were analyzed for content. Further content analysis of the document was performed on a contract between XY and the client. The evaluation was carried out by comparing the results of the research with the information and recommendations described for each given area in the theoretical part of this thesis. Subsequently, recommendations for correcting errors and deficiencies identified in the research were formulated. These recommendations are derived from the theoretical part of this thesis.

### 3. Result of research:

The results of the research confirmed some of the assumptions concerning errors and deficiencies the team was likely to face in the aftermath of the implementation of the Scrum method. These involve the following:

- the assumption of insufficient attention paid to the preparation of the team members for the agile style and the Scrum method;
- the assumption of wrong team composition from the perspective of the required experience and knowledge (one junior programmer);
- the assumption of wrong team composition from the perspective of the approach to agile development (one of the Development team members);
- the assumption of potential conflicts when setting up the team and the adaptation to the new roles in the Scrum team;
- at the beginning of the research, the assumption of too long meetings and complicated progress was confirmed. Yet, the next phase showed some improvements, but at the same time confirmed the assumption that the team would neglect some types of activities identified by the Scrum method;
- the assumption of not meeting the Sprint Goals was confirmed partially (the goal was not achieved in two out of four Sprints). Unclear Definition of Done was confirmed as well as the fact that a lot of tasks would remain as „in progress“;
- the first phase of the research also demonstrated the assumption of failing to implement agile programming practices. Yet, the second phase saw the introduction of at least some of the methods and;
- finally, the fact that the contract signed between XY and its client would not conform with the agile way of cooperation.

On the contrary, the results of the research did not prove the assumption of insufficient preparation of the Scrum Master and the Product Owner for the opening of agile transformation. It did not confirm that there would be problems with the adaptation to the new roles. The assumptions of User Stories not being sufficiently described or unclear to developers, or that the team would use a wrong Scrum board, were not confirmed, either. This also applies to the presumed reluctance of developers to participate in meetings defined by the Scrum method.

# VYSOKÁ ŠKOLA EKONOMIE A MANAGEMENTU

Nárožní 2600/9a, 158 00 Praha 5

## 4. Conclusions and recommendation:

All assumptions positively confirmed in a pilot during the research were assigned the below recommendations to address the identified errors and deficiencies:

- to prepare developers for agile transformation and the scrum method, a tool called Agile Coaching Backlog was recommended. It should be used by the scrum master to develop a short-term training, coaching, or consulting plan,. The first topics proposed for Coaching Backlog include: Standup meeting training, methods used for estimating time consumption (T-Shirt Sizes, Planning Poker), practices for retrospective meetings;
- to form a team depending on the knowledge and experience with respect to the current junior programmer, it was recommended to use the agile practices of Pair Programming and Code Review;
- given the deficiencies in the formation of the team from the perspective of approaching the agile development style, it was recommended to address the team dysfunction by using a pyramid concept that examines the 5 causes of team failure. Each level of causes shows areas for improvement, including examples of examined team;
- in the field of team roles and development (from the Tuckman model perspective), it was recommended that the Scrum Master should tackle and focus on the selected areas depending on the current or future development phase of the team. As to the current Storming phase, it was recommended that the Scrum Master does a few activities in the area of communication, cooperation, conflict, and responsibility. To be able to monitor future developments, it was recommended that the status of the team's development be reviewed when changes in the team are made or in three-month periods;
- in order to address shortcomings in the meetings and activities, recommendations for Standup and Sprint Retrospective meetings were proposed. As to the Standup meetings, as the key measure it was proposed to follow the correct format (of this activity) using three basic questions. For the retrospective meeting, the Scrum Master was proposed to provide this activity periodically using one of the five recommended practices: a retrospective with sticks, time-box, fishbone, ship, or ESVP. Using the first of these methods, a list of topics to be dealt with by the team at its retrospective meeting was made;
- as to the practices and artifacts, recommendations regarding Sprint goal and Definition of Done were proposed: to achieve Sprint goals: recommendations on the method of effort estimation, Standup meetings, and Work-in-Progress Limits were proposed, and – regarding the missing Definition of Done – it was recommended to use the initial formulation containing four basic items;
- in the scope of Agile programming, the team was advised to start using two more methods – Pair Programming and Code Review;
- as to Agile Contract, XY was advised to make an attempt to sign an agile type of contract with the client for the next period, to replace the current traditional contract.

## KEYWORDS

Agile management, agile methods, agile transformation, Scrum, Kanban.

## JEL CLASSIFICATION

M11, M15

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Jméno a příjmení:	Ludmila Buschová
Studijní program:	Ekonomika a management (Ing.)
Studijní obor:	Management firem
Studijní skupina:	MF 27
Název DP:	Zavedení agilní metody Scrum v IT organizaci
Zásady pro vypracování (stručná osnova práce):	<ol style="list-style-type: none"><li>1. Úvod – vhléd do tématu, definice cíle a metodiky práce.</li><li>2. Teoretická část – klasické metody řízení projektů, agilní metody řízení projektů – Scrum, Kanban, Implementace agilní metody, metodika výzkumu.</li><li>3. Praktická část – představení společnosti, realizace výzkumu – rozhovory se členy týmu, vyhodnocení získaných dat, zavedení metody Scrum na pilotním projektu.</li><li>4. Závěr – shrnutí zkoumané problematiky, výsledků a doporučení.</li></ol>
Seznam literatury: (alespoň 4 zdroje)	<ul style="list-style-type: none"><li>• MYSLÍN, J. <i>Scrum: průvodce agilním vývojem softwaru</i>. Brno : Computer Press, 2016. ISBN 978-80-251-4650-7.</li><li>• PAQUETTE, P., FRANKL, M. <i>Agile Project Management for Business Transformation Success</i>. Business Expert Press, 2015. ProQuest Ebook Central, <a href="https://ebookcentral.proquest.com/lib/knav/detail.action?docID=4307174">https://ebookcentral.proquest.com/lib/knav/detail.action?docID=4307174</a>.</li><li>• ŠOCHOVÁ, Z. <i>Skvělý ScrumMaster</i>. Brno : Computer Press, 2018. ISBN 978-80-251-4927-0.</li><li>• ŠOCHOVÁ, Z., KUNCE, E. <i>Agilní metody řízení projektů</i>. 2. vydání. Brno : Computer Press, 2019. ISBN 978-80-251-4961-4.</li></ul>
Harmonogram	<ul style="list-style-type: none"><li>• Zpracování cílů a metodiky do 1. 11. 2019</li><li>• Zpracování teoretické části do 1. 2. 2020</li><li>• Zpracování výsledků do 1. 4. 2020</li><li>• Finální verze do 1. 5. 2020</li></ul>
Vedoucí práce:	doc. Ing. Lucie Vnoučková, Ph.D.

V Praze dne 7. 10. 2019

prof. Ing. Milan Žák, CSc.  
rektor

Prof. Ing.  
Milan Žák CSc.

Digitálně podepsal Prof.  
Ing. Milan Žák CSc.  
DN: cn=Prof. Ing. Milan  
Žák CSc., c=CZ, o=Vysoká  
škola ekonomie a  
managementu, a.s.,  
givenName=Milan,  
sn=Žák,  
serialNumber=ICA -  
10393535

# Obsah

1	Úvod.....	1
2	Teoreticko-metodologická část práce .....	3
2.1	Klasické metody řízení projektů.....	3
	Vodopádová metoda .....	3
	Spirálová metoda .....	4
2.2	Agilní metody řízení projektů.....	5
2.2.1	Metoda Scrum.....	6
2.2.2	Metoda Kanban.....	20
2.2.3	Scrum vs. Kanban .....	21
2.2.4	Srovnání klasických a agilních metod .....	22
2.3	Implementace agilní metody Scrum .....	23
2.3.1	Pilotní projekt .....	24
2.3.2	Sestavování týmu a jeho vývoj .....	25
2.3.3	Agilní smlouvy a formy spolupráce.....	29
2.3.4	Metriky a hodnocení .....	30
2.4	Metodika práce .....	33
3	Analytická část práce.....	37
3.1	Představení firmy XY .....	37
3.2	Vyhodnocení získaných dat.....	37
3.2.1	Vyhodnocení rozhovorů – fáze 1 .....	38
3.2.2	Vyhodnocení rozhovorů – fáze 2 .....	42
3.2.3	Hodnocení a sebehodnocení .....	47
3.2.4	Scrum tabule, Burn-down graf.....	55
3.2.5	Agilní smlouva.....	58
3.2.6	Celkové shrnutí vyhodnocení.....	59
3.2.7	Předpoklady a výsledky výzkumu .....	62
3.3	Doporučení .....	63
3.3.1	Příprava na zahájení agilní transformace a sestavování týmu.....	63
3.3.2	Role a vývoj týmu.....	65
3.3.3	Meetingy a aktivity .....	66
3.3.4	Praktiky a artefakty .....	68
3.3.5	Agilní programování.....	70
3.3.6	Agilní smlouva.....	71
4	Závěr.....	73

Literatura

Přílohy



## Seznam schémat, obrázků, tabulek a grafů

Schéma 1 Fáze Vodopádové metody .....	3
Schéma 2 Iterační cyklus.....	5
Schéma 3 Cyklický proces metody Scrum.....	7
Schéma 4 Tuckmanův model .....	26
Obrázek 1 Kanban tabule .....	20
Tabulka 1 Porovnání agilních a klasických metod.....	22
Tabulka 2 Sebehodnocení členů týmu .....	48
Tabulka 3 Vzájemné hodnocení členů týmu .....	50
Tabulka 4 Hodnocení členů týmu Scrum Masterem.....	51
Tabulka 5 Hodnocení týmu Product Ownerem.....	53
Tabulka 6 Vyhodnocení vývoje týmu dle Tuckmanova modelu .....	54
Graf 1 Burn-down graf.....	31
Graf 2 Burn-down graf – Sprint 1 .....	56
Graf 3 Burn-down graf – Sprint 2 .....	57
Graf 4 Burn-down graf – Sprint 3 .....	57
Graf 5 Burn-down graf – Sprint 4 .....	58

# 1 Úvod

Současné obchodní prostředí i obor informačních technologií jsou stále složitější a rizikovější. Změny v poptávce, rozvoj technologií i chování konkurence nutí organizace se velmi rychle přizpůsobovat a velmi rychle na tyto změny reagovat. Ale technologické projekty tradičně zaznamenaly a stále zaznamenávají omezenou úspěšnost (Paquette, Frankl, 2015, s. 6). Aby mohla být IT firma na dnešním trhu úspěšná, musí být schopna rychle a kvalitně dodávat nové produkty svým klientům, ale také přijímat zpětnou vazbu od svých klientů, a hlavně na ni umět reagovat. To všechno znamená, že se firmy odklánějí od klasického způsobu organizování a řízení své práce. Nemálo firem se vydává cestou agilních přístupů a metod, jejichž cílem je radikálně změnit způsob, jakým firma v současnosti pracuje. Agilní metody umožňují firmě, aby byla proaktivnější, dělala více práce s méně zdroji a lépe sloužila zájmům svých zákazníků. Agilní přístup řeší nové výzvy, kterým organizace čelí, a v porovnání s klasickými metodami přináší výsledky (Paquette, Frankl, 2015, s. 11). Proto se stal agilní přístup jedním z hlavních hybatelů vývoje softwaru v celém softwarovém průmyslu (Eby, 2016).

Agilní metody jsou nejčastěji využívány v IT oboru, kde pomáhají řídit jednotlivé projekty nebo celé firmy. V současnosti tento přístup využívají i mnohá další odvětví, kde jak malé firmy, tak i nadnárodní korporace chtějí pomocí agilního přístupu dosahovat vyšší úroveň kvality a konkurenceschopnosti na současném trhu. Slovo „**agilní**“ lze přeložit jako *dynamický, rychlý, interaktivní, přizpůsobivý, iterativní, zábavný, hravý, rychle reagující na změnu* (Šochová, Kunce, 2019, s. 15). Všechny tyto překlady vystihují vlastnosti tohoto přístupu, který se zásadně liší od konvenčních metod řízení projektů, mezi které například patří Vodopádová nebo Spirálová metoda. Dle Šochové (2010) není agile přístup jen soubor praktik, ale celková filozofie, kde se jednotlivé praktiky podporují a jen jejich kombinací vznikne synergický efekt. Úspěšně implementovat agilní metodu znamená významný posun paradigmatu pro všechny zúčastněné strany. Aby zavedení agilní metody mělo šanci na úspěch, vyžaduje významnou podporu a dostatek zdrojů i času. Úspěšné zavedení agilní metody revolučně mění přístup k tomu, jak firma řídí své projekty a jak řeší potřeby zákazníků. Pro malé společnosti může být agilní transformace přímočará, ale nikoliv jednoduchá cesta, proto je doporučováno začít s touto změnou pouze v jedné části organizace nebo na jednom pilotním projektu. Tento pilotní projekt je pak příležitostí ke změně a k učení se. Účelem pilotního projektu je také ukázat hodnotu agilních přístupů prostřednictvím skutečných obchodních výsledků, proto tento projekt musí být dostatečně reprezentativní.

Nejnámější agilní metodou je **Scrum**. Tento agilní framework umožňuje řešit složité adaptivní problémy a zároveň efektivně vytvářet produkty nejvyšší možné hodnoty (Schwaber, Sutherland, 2013). Dle jeho zakladatelů Kena Schwabera a Jeffa Sutherlanda lze tento rámec lehce pochopit, ale jen obtížně skutečně zvládnout. Jak autoři popisují, Scrum sám o sobě není proces nebo technika, je to rámec používající různé procesy a techniky, pomocí nichž objasňuje relativní účinnost postupů a vývoje produktů. Druhým nejpobulárnějším agilním přístupem je **Kanban**. Ale ani Kanban není procesem, je tvořen pouze třemi základními principy – omezit rozpracovanou práci, minimalizovat čas průchodu a vizualizovat progres (Šochová, 2012). Cílem je výrazně urychlit proces řešení jednotlivých úkolů, zlepšit přesnost plnění termínů a zvýšit přehlednost nad prováděnými pracemi. V praxi bývají tyto dva nejnámější agilní přístupy využívány také společně, protože se mohou vzájemně doplňovat a tím dosahovat lepšího efektu.

**Hlavním cílem diplomové práce je navrhnout doporučení pro odstranění chyb a nedostatků, které byly zjištěny v průběhu vlastního výzkumu u implementace agilní metody Scrum na pilotním projektu v IT organizaci.**

Teoreticko-metodologická část diplomové práce je založena na rešerši sekundárních zdrojů a zaměřuje se na téma agilní management, agilní metody Scrum a Kanban a implementaci agilní metody. Dále se tato část práce okrajově zabývá klasickými metodami vývoje softwaru – Vodopádová a Spirálová metoda – aby bylo možné vidět zásadní rozdíly v těchto přístupech. V závěru teoreticko-metodologické části je uvedena metodika, která byla použita pro výzkum v dané organizaci.

Praktická část diplomové práce popisuje základní znaky zkoumané IT organizace, jejíž název je z důvodu zachování anonymity změněn. Data byla získána metodou strukturovaných rozhovorů se Scrum Masterem, Product Ownerem a pěti vývojáři. Tito respondenti také hodnotili průběh vývoje týmu dle Tuckmanova modelu, vývojáři provedli sebehodnocení i vzájemné hodnocení, Scrum Master hodnotil jednotlivé členy týmu a Product Owner hodnotil Development tým jako celek. Rozhovory i hodnocení byla rozdělena na dvě fáze. První fáze rozhovorů proběhla na začátku implementace agilní metody Scrum. Tato část rozhovorů zjišťovala připravenost jednotlivých lidí, jejich očekávání, vnímání rizik a příležitostí. Druhá fáze rozhovorů proběhla dva měsíce od zahájení Scrumu na pilotním projektu a otázky byly zaměřené na hodnocení průběhu a dosaženého stavu. Výstupy z pilotního projektu, které byly zaznamenávány prostřednictvím softwarového nástroje, byly podrobeny obsahové analýze. Další obsahová analýza dokumentu byla provedena u smlouvy k pilotnímu projektu mezi firmou XY a jejím klientem. Navržená doporučení jsou rozdělena na oblasti, u kterých se na pilotním projektu dle stanovených předpokladů potvrdili určité chyby nebo nedostatky. Doporučení se opírají převážně o teoretické části této diplomové práce, konkrétně kapitoly: Metoda Scrum, Metoda Kanban a Implementace agilní metody Scrum.

V uvedené firmě XY již proběh výzkum v rámci seminární práce, který se zabýval motivací k přechodu na agilní přístup a očekáváními plynoucími z této změny. Na základě informací a dat získaných z výzkumu bylo v závěru seminární práce doporučeno, aby firma XY zahájila přechod na agilní řízení projektů (Buschová, 2019). Konkrétní doporučení vyplývající ze seminární práce jsou popsána v části 2.4 Metodika práce, a tato diplomová práce na ně navazuje.

## 2 Teoreticko-metodologická část práce

Teoretická část práce popisuje neznámější klasické metody řízení projektů – Vodopádová metoda a Spirálová metoda. Tyto dvě metody jsou zde zmíněny pro porovnání s agilními metodami Scrum a Kanban, kterým je věnována převážná část teoretické práce. Metoda Scrum je zde popsána na úrovních jednotlivých rolí, artefaktů a aktivit. Jsou zde také vysvětleny návaznosti v tomto procesu a odpovědnosti jednotlivých rolí. Následně jsou metody Scrum a Kanban porovnány a porovnání pokračuje shrnutím nejzásadnějších rozdílů mezi agilními a klasickými přístupy. Další část teoretické práce se věnuje implementaci agilní metody Scrum. Zde je popsána implementace Scrumu na pilotním projektu a doporučení pro sestavování týmu i pro jednotlivé fáze vývoje týmu. Jedna z podkapitol se okrajově věnuje také agilním smlouvám a tématem poslední podkapitoly jsou agilní metriky a hodnocení členů týmu.

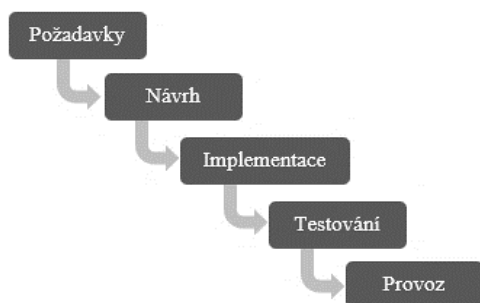
### 2.1 Klasické metody řízení projektů

Cílem této podkapitoly je základní představení dvou klasických metod, jimiž jsou Vodopádová a Spirálová metoda. Tyto dvě metody patří mezi neznámější klasické metody řízení projektů a jsou i v současnosti stále využívány v praxi. Klasické metody se vyznačují tradičním přístupem k vývoji – maximální možné svázání procesu, precizní dokumentace, minimalizace vágnosti, určení jednotlivých požadavků a termínů (Myslín, 2016, s. 23). Jak autor dále uvádí, také jednotlivé role jsou poměrně přesně stanoveny dle specializace jednotlivých pracovníků (programátor, tester, projektový manažer atd.). Zastupitelnost jednotlivých rolí je velmi nízká nebo nulová.

#### Vodopádová metoda

Nejstarší a neznámější klasickou metodou řízení projektů je Vodopádová metoda – Waterfall Model (Alshamrani, Bahattab, 2015). Tato metoda, která vznikla v sedmdesátých letech minulého století, je pro současné projekty již nedostatečná, ale pro svou jednoduchost je i nadále využívána (Myslín, 2016, s. 24). Zvláštností tohoto modelu jsou jeho postupné kroky, které se nepřekrývají, což znamená, že každá fáze končí před zahájením fáze další (Alshamrani, Bahattab, 2015). Tento lineární průběh patří mezi charakteristické znaky této metody (Myslín, 2016, s. 25). Dle autora je tato metoda jednoduchá, proto ji lze snadno řídit a plánovat. Následující Schéma 1 Fáze Vodopádové metody zobrazuje lineární návaznost fází.

Schéma 1 Fáze Vodopádové metody



Zdroj: Myslín (2016, s. 25), upraveno.

Jak je z výše uvedeného schématu patrné, jednotlivé fáze navazují postupně jedna na druhou, proces je lineární a neopakuje se. Následující kroky stručně popisují obsah těchto jednotlivých fází (Alshamrani, Bahattab, 2015):

- **požadavky:** tato fáze obsahuje specifikaci funkcí (obvykle se jedná o informace poskytované klienty), požadavky jsou shromažďovány, analyzovány a poté je připravena dokumentace, která dále pomáhá při vývoji;
- **návrh:** v této fázi se vyhodnocují shromážděné informace z předchozí fáze, je formulována správná implementace, jedná se o proces plánování a řešení problémů softwarového návrhu;
- **implementace:** v této fázi jsou všechny požadavky implementovány;
- **testování:** tato část se zabývá testováním a kontrolou toho, co bylo vyvinuto, dále probíhají opravy nalezených chyb a vylepšení systému;
- **provoz:** po dokončení realizace a testování následuje fáze, která zajišťuje potřebnou údržbu produktu, provádění úprav a oprav.

Vodopádová metoda pracuje s předpokladem, že po stanovení počátečních požadavků a odstranění nejasností existuje bezbariérová cesta, po které může vývojový tým postupovat k dokončení projektu (Andrei at al., 2019). Jak ale autoři upozorňují, ve většině případů to není pravda, protože zákazníci mohou své požadavky změnit a v některých případech je pak potřeba přehodnotit určité fáze projektu nebo i celý projekt. Toto označují autoři jako nejzjevnější chybu této metody. Protože dochází k navyšování nákladů, prodlužuje se také doba strávená na jednotlivých částech projektu, což dle těchto autorů může vést k nespokojenosti klienta. Andrei at al. (2019) ale zároveň připouštějí, že existují projekty, pro které je tato metoda vhodná – například malé projekty s pevnými termíny, rozpočtem a rozsahem. Svozilová (2016, s. 386) také souhlasí s využitím této metody u krátkodobých projektů, kde není předpoklad změnových požadavků. Naopak u velkých projektů má tato metoda spíše nevýhody (Myslín, 2016, s. 25–26). Dle autora jsou například velmi obtížně odhalovány chyby, zákazník není zapojen do vývoje a vidí až konečný produkt, kdy je pak problematické řešit připomínky a požadavky na změny. Svozilová (2016, s. 387) doplňuje, že náklady na odstraňování těchto chyb a řešení nových požadavků rostou v průběhu projektu exponenciálně.

Přestože plánování času lze u této metody optimálně rozdělit, protože každé fázi je přiděleno pevné časové období, tak další nevýhodou je, že při zpoždění jednoho úkolu dojde automaticky ke zpoždění u všech následujících (Andrei at al., 2019). Vzhledem k tomu, že kontrolní milníky jsou v podstatě až na konci jednotlivých fází projektu, při zdržení jedné části projektu nebo při výskytu chyb dochází ke zdržení dokončení projektu a k nárůstu nákladů (Svozilová, 2016, s. 385). Také dle jiných autorů neposkytuje tato metoda očekávané výsledky, protože na začátku projektu nelze všem požadavkům kompletně porozumět a následný proces evidence změnových požadavků je pak příliš komplikovaný a nevyhovující pro měnící se podmínky během fáze realizace (Paquette, Frankl, 2015, s. 2).

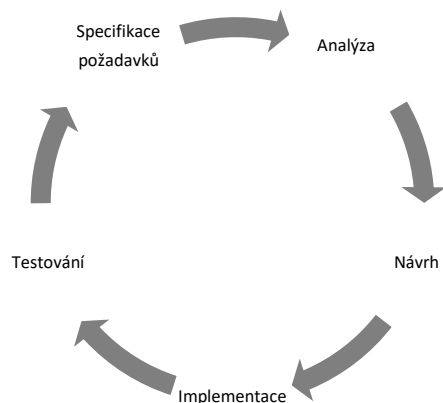
### **Spirálová metoda**

Hlavní nevýhodou výše uvedené metody Vodopád je, že ověření a kontrola výsledku probíhá až v závěrečných krocích projektu (Svozilová, 2016, s. 387). Tento problém řeší iterativní metody řízení projektů, z nichž nejznámější je Spirála. Mezi výhody iterativních metod patří hlavně rychlý a nákladově efektivní vývoj, snadný monitoring, eliminace rizik, důraz na shodu s klientem a možnost reakce na změny (Keinitz, 2017). Jako nevýhody autor naopak označuje nároky na odborné znalosti v oblasti rizik, závislost úspěchu projektu na analýze rizik, složité určení konce projektu, možný růst nákladů, pokud cyklus není v čas ukončen. Keinitz (2017) doporučuje tyto metody pouze pro rozsáhlejší projekty.

Spirálová metoda je proces, který v jednotlivých etapách kombinuje prvky návrhu s prvky vývoje, a kromě toho se zaměřuje na hodnocení rizik i minimalizaci rizika projektu

(Alshamrani, Bahattab, 2015). Jak autoři popisují, tento model kombinuje prvky vodopádového modelu iteračním způsobem – vývojový tým začíná s malou sadou požadavků a s nimi prochází každou vývojovou fází. Poté dostává tým další požadavky, které opět zapracuje, a to do té doby, než je projekt finálně dokončen. Jak autoři vysvětlují, v každé iteraci dodává tým určitý přírůstek, čímž pokaždé zvyšuje funkčnost daného produktu. Rozdělení předmětu projektu na jednotlivé fáze je prováděno dle priority zákazníka, možnosti fungování a předvedení daných částí (Svozilová, 2016, s. 387). Tento iterační cyklus níže znázorňuje Schéma 2 Iterační cyklus.

**Schéma 2 Iterační cyklus**



Zdroj: Myslín (2016, s. 27), upraveno.

Výše znázorněný iterační cyklus je základem iterativních metod. Jednotlivé fáze na sebe navazují a cyklus probíhá, dokud není projekt dokončen. Pro tyto metody je charakteristické zejména to, že je možné v každé iteraci vidět dílčí výsledky, testovat danou část, opravovat chyby a upravovat případné požadavky (Myslín, 2016, s. 26).

## **Shrnutí kapitoly**

Tato kapitola popsala základní vlastnosti dvou nejznámějších klasických metod řízení projektů, jimiž jsou metody Vodopád a Spirála. Vysvětleny byly jejich procesy i hlavní výhody a nevýhody. Pro potřeby této diplomové práce není nutné se těmito metodami podrobněji zabývat. Jsou zde uvedeny pro srovnání s agilními metodami, kterými se již podrobněji zabývá následující kapitola.

## **2.2 Agilní metody řízení projektů**

Agilní metody jsou relativně novým přístupem, který přináší systémovou integraci do podnikového prostředí a zároveň zpochybňuje současný stav (Paquette, Frankl, 2015, s. 3–4). Dle autorů zlepšují agilní techniky výkonnost technologických projektů díky lepší zpětné vazbě od zainteresovaných stran a zvyšují konkurenční výhodu organizace díky lepšímu řízení organizačních změn, komunikaci a týmové práci. Dle Eby (2016) platí v agilních metodách názor, že změny vždy vylepšují projekt a poskytují přidanou hodnotu. Agilní metody nejsou striktním procesem nebo modelem, ale mají svá pravidla a definují hranice, v jejichž rámci si týmy mohou stanovovat svá vlastní pravidla tak, aby byly co nejvíce efektivní a dodávaly kvalitní produkty v nejkratším možném čase (Šochová, Kuncce, 2019,

s. 15). I další autoři potvrzují, že dle agilních přístupů je nutné na každý projekt nahlížet individuálně a je potřeba si metodu přizpůsobit tak, aby co nejlépe vyhovovala požadavkům na daný projekt (Baseer at al., 2015). Šochová a Kunce (2019, s. 15) uvádí, že se agilní týmy zaměřují na hodnotu pro zákazníka (business value), což znamená, že se vývoj zaměřuje na funkcionalitu, která zákazníkovi přinese to, co skutečně potřebuje a má pro něj užitek. Také dle Eby (2016) je záměrem agilního přístupu sladění vývoje s obchodními potřebami, proto agilní projekty podporují účast zákazníků. Agilní týmy také zlepšují komunikaci vertikálně i horizontálně v celé organizaci (Paquette, Frankl, 2015, s. 4–5). Tyto metody dle autorů podporují organizační sladění, optimalizaci zdrojů a přidělené odpovědnosti mají pozitivní dopad na inovace i motivaci pracovníků. Základem agilního přístupu je **Agilní manifest**, který definuje čtyři základní hodnoty (Manifesto for Agile Software Development, 2001):

- 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.

Výše uvedené hodnoty zdůrazňují to, na co je potřeba se zaměřit a co upřednostnit před dogmatickým dodržováním procesů a plánů nebo sestavováním podrobné dokumentace a rozsáhlé smlouvy se zákazníkem. Agilní manifest se mimo čtyř základních hodnot dále skládá z 12 principů. Všechny agilní metody používají tyto hodnoty a principy různými způsoby, ale vždy se na ně spoléhají při vývoji vysoce kvalitního produktu (Eby, 2016). Následující dvě podkapitoly se věnují nejznámějším a nejoblíbenějším agilním metodám Scrum a Kanban. Scrum je jedním z rámců používaných v Agilním přístupu, a protože byl v praxi úspěšně použit a zdokumentován, stal se nakonec základem agilního programování (Redka, 2018).

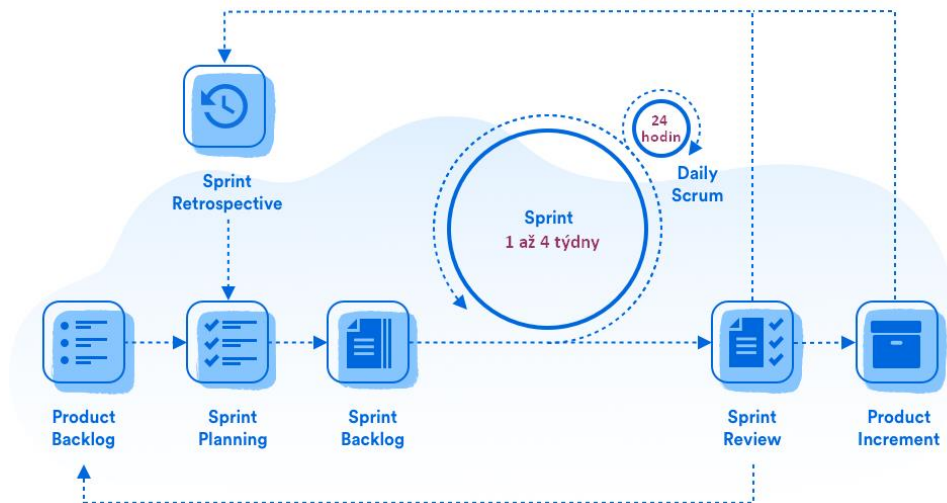
### 2.2.1 Metoda Scrum

Zakladatelé Hirotaka Takeuchi a Ikujiro Nonaka popsali Scrum jako přístup, který by měl zvýšit rychlost a flexibilitu procesu vývoje, ale byl poprvé plně implementován až o deset let později, Jeffem Sutherlandem a Kenem Schwabem (Redka, 2018). Termín Scrum pochází původně z ragby a označuje tým, který pracuje společně na dosažení společného cíle (Schwaber, Sutherland, 2013). Dle autorů poskytuje Scrum vhodný rámec pro řízení jak malých, tak i složitějších projektů prostřednictvím produktových požadavků, každodenních schůzek a retrospektiv. Scrum je založen na třech pilířích: transparentnost, inspekce a adaptace (Redka, 2018). Jak autorka popisuje, transparentnost zajišťuje, že všechny strany odpovědné za výsledek shodně vidí a chápou klíčové aspekty procesu, inspekce znamená kontrolu postupu a ujištění, že nic nebrání týmu k dosažení stanoveného cíle. Adaptace je dle autorky odpovědná za přizpůsobení procesu na základě výsledků inspekce, což následně vede ke zlepšování.

Scrum se skládá z týmů a jejich přidružených rolí, událostí, artefaktů a pravidel (Schwaber, Sutherland, 2013). Dle autorů každá součást Scrumu slouží specifickému účelu a je nezbytná pro úspěch a používání této metody, protože pravidla Scrumu jsou propojena událostmi, rolmi a artefakty, společně řídí vztahy a vzájemné působení mezi nimi. Scrum využívá samoorganizující týmy, iterativní dodávky produktu, překrývající se fáze, dynamickou hodnotu multidisciplinárních týmů a organizační učení (Paquette, Frankl, 2015, s. 57–58). Dle autorů patří mezi klíčové faktory úspěchu této metody práce na určitém množství úkolů v prioritním pořadí dle potřeb zákazníka. Nové týmy okamžitě zaznamenají, že Scrum vyžaduje mnohem více plánování – je potřeba plánovat iterace, odhadovat úkoly a určit položky, které jsou nejdůležitější při dané rychlosti vývoje (Hebb, 2019).

Cyklická povaha Scrumu umožňuje zjistit, jaké funkce by měl produkt poskytovat, a zároveň usnadňuje experimentování a učení (Pichler, 2012). Jak autor popisuje, nový produkt se vyvíjí v tomto cyklu na základě zpětné vazby uživatelů, zákazníků a dalších zainteresovaných stran. Následující **Schéma 3 Cyklický proces metody Scrum** popisuje klíčové kroky této metody.

**Schéma 3** Cyklický proces metody Scrum



Zdroj: Redka (2018), upraveno

Výše uvedené schéma znázorňuje cyklický proces metody Scrum, kdy nejdříve je naplněn **Product Backlog** (seznam požadavků na produkt v pořadí podle priority) a tým provádí odhady náročnosti těchto požadavků. Následuje schůzka **Sprint Planning**, na které je definován cíl daného Sprintu a do **Sprint Backlogu** jsou vybrány položky, které tento cíl naplní. Pak lze spustit 1 až 4týdenní **Sprint**, během kterého probíhá vývoj položek ze Sprint Backlogu a na denní bázi se pořádají krátká setkání týmu – **Daily Scrum Meeting** (neboli Standup Meeting). Po dokončení každého Sprintu probíhá **Sprint Review Meeting**, na kterém se prezentuje dokončená práce, díky které dochází po každém Sprintu k navýšení funkcionality produktu – **Product Increment**. Na konci Sprintu se také koná schůzka **Sprint Retrospective**, během které tým vyhodnocuje pozitivní i negativní stránky procesu a hledá možná řešení. Nová iterace cyklu pak začíná novým Sprint Planningem. Dále již cyklus opět pokračuje výše popsanými kroky, dokud jsou v Product Backlogu požadavky k realizaci. Následující text se podrobněji věnuje popisu těchto jednotlivých elementů Scrumu, které jsou základními stavebními kameny této metody a lze je rozdělit do tří kategorií: **role, artefakty a aktivity**.

### Role ve Scrumu

Scrum je založen na self-organized týmu, transparentní komunikaci, otevřené kultuře, sdílení informací a spolupráci. Aby tento systém fungoval, jsou ve Scrumu specifické role, které v klasických metodách managementu neexistují (Šochová, Kunce, 2019, s. 43). Těmito rolami jsou Scrum Master, Product Owner, Scrum tým a Development tým.

### **Scrum Master**

Scrum Master je průvodcem agilní transformace, proto má být vždy o jeden krok před týmem a organizací, aby mohl lidi odpoutat od jejich stereotypů a běžných praktik (Šochová, 2018a, s. 37–41). Scrum Master je koučem, facilitátorem a servant leaderem, jehož hlavním cílem je



vytvořit samostatný efektivní a spokojený self-organized tým a díky této roli by se mělo týmu dobře pracovat (Šochová, Kuncce, 2019, s. 43). Skvělý Scrum Master musí využívat soft-skills, umět dobře naslouchat, být expertem na agilní přístup a Scrum (Šochová, 2018a, s. 29). Dále musí mít dobré rozhodovací schopnosti, aby zvládl flexibilitu v agilním prostředí (Srivastava, Jain, 2017). Scrum Master je zodpovědný za to, že Scrum bude pochopen a přijat (Schwaber, Sutherland, 2013). Také koordinuje veškeré projektové aktivity a propojuje zákazníky a různé týmy, které jsou součástí projektu (Choudary, 2019). Scrum Master pomáhá lidem mimo Scrum tým pochopit, které z jejich interakcí se Scrum týmem jsou užitečné a které nikoli (Schwaber, Sutherland, 2013). Scrum Master je součástí týmu, má být týmu kdykoliv k dispozici, a proto by měl sedět s týmem v jedné místnosti (Šochová, Kuncce, 2019, s. 43). Všechny překážky, které tým nemůže sám vyřešit, přicházejí k řešení na Scrum Mastera, který tyto překážky prioritizuje na základě jejich dopadu na produktivitu týmu a aktivně je řeší (Choudary, 2019). Product Ownerovi pomáhá s porozuměním plánování produktů v empirickém prostředí, s pochopením agility, se zvládáním aktivit Scrumu a s technikami pro efektivní správu Product Backlogu tak, aby maximalizoval hodnotu produktu (Schwaber, Sutherland, 2013).

Scrum Master má také na starosti efektivitu Scrum procesu a zároveň dohlíží na jeho fungování, v případě potřeby iniciuje změnu (Šochová, Kuncce, 2019, s. 43). Jak autoři popisují, měl by upřednostňovat principy koučování, podporovat tým i jednotlivce v rozvoji, být komunikativní, vnímavý a utlumovat případné konflikty v rámci týmu. Autoři zdůrazňují, že by neměl být direktivní. Je také zodpovědný za kultivaci vhodného pracovního prostředí pro tým – od fyzického pracovního prostoru až po komunikační nástroje (Choudary, 2019). Například pokud je potřeba nechat opravit rozbité počítače, jinak rozmístit stoly, aby lidé v týmu mohli lépe komunikovat (Rehkopf, 2018). Mezi povinnosti Scrum Mastera patří také podpora týmu k lepšímu dosahování cílů, facilitací pomáhá odstraňovat vzniklé problémy, koučuje tým a stará se o jeho rozvoj (Šochová, Kuncce, 2019, s. 43). Měl by také oceňovat úsilí členů týmu, aby je motivoval v náročném prostředí (Srivastava, Jain, 2017). Scrum Master může rozhodnout, zda se bude setkávat se členy týmu také samostatně, protože individuální interakce jsou zásadní pro rozvoj týmu, vzájemné poznávání a řešení problémů (Rehkopf, 2018).

Role Scrum Mastera je pro úspěch produktu klíčová, ale po určitém čase se může zdát, že tým již tuto roli nepotřebuje (Šochová, Kuncce, 2019, s. 44). Jak autoři uvádějí, Scrum Master bývá pak přesunut na jinou práci, ale po čase se obvykle začnou v týmu množit problémy a tým přestává fungovat (Šochová, Kuncce, 2019, s. 44). Je to také jedna z nejpodceňovanějších rolí ve Scrumu a většina nových týmů se domnívá, že zaměstnat Scrum Mastera na plný úvazek je zbytečné a snaží se tuto roli spojit s rolí vývojáře nebo testera (Šochová, 2018a, s. 23). Jak autorka dále popisuje, velmi často končí Scrum Master jako „asistentka“ týmu, protože okolí této roli vůbec nerozumí a neví, co od ní má očekávat. Také není vhodné kombinovat tuto roli s rolí Product Ownera, protože z definice této role vyplývají často konfliktní cíle (Šochová, Kuncce, 2019, s. 43–44). Autoři dále vysvětlí, že není vhodná kombinace s jakoukoliv vývojářskou nebo manažerskou rolí, protože lze očekávat, že vývojář bude upřednostňovat svojí technickou roli před rolí Scrum Mastera a manažerský přístup k týmu je výrazně odlišný a mohl by narušit budování self-organized týmu. Pokud je nutné tuto roli kombinovat, autoři doporučují, aby vždy role Scrum Mastera byla prioritní.

Scrum Master ale není zdroj dodatečných nákladů, které nepřinášejí žádný užitek, naopak je zodpovědný za formování vysoce výkonného týmu a je zosobněním agilního myšlení, protože je přesvědčen o tom, že agilní metody a Scrum jsou jedinou správnou cestou, která povede k úspěchu (Šochová, 2018a, s. 23). Tím, že dobře fungující Scrum Master rozhybává celý tým

a zajišťuje efektivitu, kvalitu a motivaci, ušetří firmě více nákladů, než tato role jako nákladová položka firmu stojí (Šochová, Kunce, 2019, s. 43).

Scrum Master je ochráncem týmu, chrání ho před vnějšími vlivy, které se často pokoušejí ovládat výstupy Scrum týmu nebo diktovat strategie technického designu a vytvářejí tak zbytečná omezení (Choudary, 2019). Scrum Master nemá být direktivní, ale nemůže být ani nevýrazný a nechat vše plynout, jeho přístup se v průběhu mění v závislosti na zkušenostech a schopnostech týmu (Šochová, Kunce, 2019, s. 44). Jak autoři připomínají, cílem Scrum Mastera je vybudovat self-organized tým, proto z počátku více radí, organizuje a určuje, co se bude dít, ale postupně nechává týmu prostor, aby se mohl sám organizovat, růst a řešit své problémy.

## **Product Owner**

Product Owner je vlastníkem produktu, definuje produktovou vizi a komunikuje tuto vizi směrem k týmu, zákazníkům a firmě (Šochová, Kunce, 2019, s. 47). Jak autoři vysvětlují, tato role určuje priority a rozhoduje o tom, na čem a kdy tým pracuje, proto tato role vyžaduje komunikativního člověka se silnými znalostmi o produktu. Product Owner má v projektu plnou moc a je to on, kdo musí být spokojen s kvalitou konečného produktu (Unhelkar, 2012, s. 210). Product Owner je vždy jedna osoba, která může zastupovat určitou skupinu osob a jejich požadavky pro Product Backlog, ale tato skupina se vždy musí obracet pouze na Product Ownera (Schwaber, Sutherland, 2013). Role Product Ownera je zásadní pro agilní úspěch a zajišťuje, že se tým zaměřuje na obchodní hodnotu produktu (Paquette, Frankl, 2015, s. 94–95). Cílem této role je úspěšný produkt, z titulu své funkce ale také zodpovídá za návratnost investice (Šochová, Kunce, 2019, s. 47), za maximalizaci hodnoty produktu i práci vývojového týmu, a je jedinou osobou odpovědnou za správu Product Backlogu (Schwaber, Sutherland, 2013). I když je za Product Backlog vždy zodpovědný, s jeho správou mu pomáhá celý Development tým, zástupci zákazníka, uživatelé a další specialisté (Šochová, Kunce, 2019, s. 47). Tato správa Product Backlogu zahrnuje například jasné popsání jednotlivých položek, optimalizaci hodnoty práce, zajištění viditelnosti, transparentnosti a srozumitelnosti jednotlivých položek na potřebnou úroveň (Schwaber, Sutherland, 2013).

Product Owner by měl zapojit tým do identifikace obchodních i projektových rizik a měl by s týmem diskutovat o prioritách v souvislosti se zjištěnými riziky (Unhelkar, 2012, s. 290). Product Owner je týmu k dispozici pravidelně dle potřeby, ale nemusí s týmem sdílet společný prostor, jako je tomu u Scrum Mastera (Šochová, Kunce, 2019, s. 47). Jak autoři dále popisují, Product Owner tráví svůj čas se zákazníky, aby pochopil skutečnou hodnotu produktu a mohl pak správně rozhodovat. Dle autorů je obvykle jeho čas rozdělován na 80 % času u zákazníka a 20 % času s týmem, ale záleží hlavně na typu produktu. Product Owner má podpůrnou roli během Sprintu i při jeho plánování a poskytuje zpětnou vazbu k výsledkům během Sprint Review (Unhelkar, 2012, s. 168). Product Owner neřídí členy týmu, nemá oprávnění přikazovat, co má tým dokončit, ale pouze stanovuje, co má tým dělat a v jakém pořadí (Šochová, Kunce, 2019, s. 47). Stanovení těchto priorit může být iteračním procesem, kdy po přibližném stanovení priorit Product Ownerem následuje společný workshop klíčových zainteresovaných stran a všech členů týmu, při němž jsou priority a funkce upřesňovány (Unhelkar, 2012, s. 291). Celá organizace musí respektovat rozhodnutí Product Ownera, která jsou zanesena v obsahu Product Backlogu (Schwaber, Sutherland, 2013). Jak autoři upozorňují, nikdo jiný nesmí pověřit Development tým prací, která není obsažena v Product Backlogu.

## Scrum tým

Týmový model Scrumu je navržen tak, aby optimalizoval flexibilitu, kreativitu i produktivitu a skládá se z rolí Product Ownera, Scrum Mastera a Development týmu (Schwaber, Sutherland, 2013). Mezi těmito třemi rolemi není žádná hierarchie (Šochová, Kunce, 2019, s. 53). To znamená, že každý člen týmu by měl mít stejné právo vyjádřit svůj názor (Hoek, 2017). Nejúspěšnější agilní týmy s vysokým výkonem jsou malé, jejich optimální velikost se pohybuje od pěti do deseti lidí v závislosti na projektu, typech úkolů a požadavcích projektu (Paquette, Frankl, 2015, s. 95). Jak autoři vysvětlují, dynamika malého týmu podporuje konsenzus, soudržnost, respekt, vzájemnou závislost i kreativitu, zvyšuje výkonnost a kvalitu, a současně se vyhýbá skupinovému myšlení. Scrum týmy jsou multifunkční a organizují se samy, proto si také samy vyberou, jak nejlépe vykonají svou práci, ke které mají veškeré potřebné kompetence (Schwaber, Sutherland, 2013). Proto by tým měl mít veškeré znalosti potřebné k dodání funkčního produktu (Hoek, 2017). Jak autor vysvětluje, nespadají sem obchodní znalosti a ani to neznámá, že každý člen týmu by měl být dokonalým vývojářem a měl by mít všechny tyto znalosti. Nutnou podmínkou pro založení self-organized týmu je společný cíl, dále tým musí rozumět zákazníkovi, chápat dané prostředí a vědět, jak bude daný produkt používán (Šochová, Kunce, 2019, s. 47). Dle autorů je také důležitým prvkem důvěra, a to nejen v týmu, ale také směrem k zákazníkovi a k ostatním částem firmy. Důvěru vzájemně posiluje snížený dohled nad týmem, protože tým se snaží nezklamat důvěru, kterou dostal od manažerů (Unhelkar, 2012, s. 39). Naopak by v týmech neměly být prováděny časté personální změny, protože nový Scrum tým potřebuje čas, než se naučí spolupracovat (Hoek, 2017). Změny v týmu mají (dočasný) negativní dopad na produktivitu a stabilitu, spolehlivost i předvídatelnost týmu (Levison, 2017), a pokud pak selže jeden člen týmu, znamená to, že selhal celý Scrum tým (Šochová, Kunce, 2019, s. 47).

Scrum je založen na úzké spolupráci, takže v ideálním případě by celý tým měl pracovat ve stejné místnosti, aby neexistovaly překážky komunikace, kvůli kterým pak lidé odkládají svou interakci (Hoek, 2017). Scrum týmy dodávají produkty iterativně a maximalizují tak příležitosti pro zpětnou vazbu (Schwaber, Sutherland, 2013), zároveň musí dostat možnost rozhodovat, měnit způsoby své práce, podílet se na tvorbě Product Backlogu (Šochová, Kunce, 2019, s. 47). Jak autoři dále vysvětlují, tým naopak nemá oprávnění rozhodovat o tom, kdo bude členem týmu a zda se bude nebo nebude pracovat podle Scrum procesů a agilních přístupů, protože tyto záležitosti obvykle spadají do kompetence dané organizace. Všichni členové týmu by měli být přiřazeni do projektu na plný úvazek, protože přecházení mezi úkoly nebo rozdělování pozornosti mezi dva projekty není efektivní (Hoek, 2017). Jak autor doplňuje, plná alokace na jednom projektu také posiluje odpovědnost jednotlivých členů.

## Development tým

Development tým by měl usilovat o dosažení vize stanovené Product Ownerem (Chapman, 2019). V tomto týmu již neexistují žádné další role, pracovníci jsou členy tohoto týmu a v rámci Sprintu pracují na realizaci inkrementu produktu (Šochová, Kunce, 2019, s. 53). Scrum neuznává žádné jiné názvy rolí pro členy Development týmu než označení „vývojář“ (Developer), a to bez ohledu na práci, kterou zde daný člověk provádí (Schwaber, Sutherland, 2013). Dle Nijland (2018) to činí tým kolektivně odpovědný, napomáhá přizpůsobení se a rozvoji schopností jednotlivců. Na druhou stranu je toto označení někdy bráno příliš doslovně a mylně vede k domněnce, že Development tým zahrnuje pouze programátory a nezahrnuje testery, obchodní analytiku, odborníky na použitelnost atd. (Levison, 2017). Autor vysvětluje, že Development tým zahrnuje každého, kdo přispívá k tomu, aby byla daná část produktu skutečně dokončena. Takovéto vývojové týmy jsou multifunkční a pracovníci jsou schopni se do určité míry zastupovat (Šochová, Kunce, 2019, s. 51). K tomu jim dle

autorů pomáhá sdílení alespoň 10 až 20 % znalostí mezi členy týmu, který je pak efektivnější a flexibilnější. Jednotliví členové týmu mohou mít specializované dovednosti a oblasti zaměření, ale odpovědnost vždy náleží Development týmu jako celku (Schwaber, Sutherland, 2013). Současně se lidé v tomto týmu mají navzájem respektovat jako nezávislé osoby, které jsou schopné jednotlivě přispět k projektu (Chapman, 2019). Pro Development tým je charakteristické, že jsou self-organized a nikdo (ani Scrum Master) jim neříká, jak mají požadavky z Product Backlogu realizovat (Schwaber, Sutherland, 2013). I z těchto důvodů musí mít tento tým odvahu čelit problémům, jako jsou konflikty vzniklé v důsledku nesprávné komunikace nebo neočekávané požadavky klienta (Chapman, 2019).

Development tým by měl mít takovou velikost, aby zůstal svižný, ale zároveň byl schopen dokončit významnou práci v rámci Sprintu (Schwaber, Sutherland, 2013). Autoři zdůrazňují, že pokud je tým tvořen méně než třemi lidmi, snižuje se interakce, což vede k nižší produktivitě a také mohou chybět potřebné dovednosti. Na druhou stranu autoři upozorňují, že více než devět členů vyžaduje přílišnou koordinaci a složitý proces řízení (role Product Ownera a Scrum Mastera nejsou zahrnuty do těchto počtů, pokud zároveň nepracují jako vývojáři). Sutherland (2014, s. 58) doplňuje, že důsledkem je zpomalení rychlosti práce, proto také doporučuje 5 až 9 lidí v jednom týmu. Development tým se také musí neustále zaměřovat na nejbližší úkoly, vyvarovat se rozptýlení a snažit se dosáhnout cíle aktuálního Sprintu (Chapman, 2019). Každý člen týmu musí vědět, jakou práci dělají ostatní, a práce každého člena musí být pro ostatní členy transparentní (Sutherland, 2014, s. 61).

### **Role projektového manažera**

Scrum týmy se organizují samy, proto na řízení z vyšších pozic nereagují dobře (Rehkopf, 2018). Na druhou stranu koncept samoorganizace nemusí vyhovovat těm, kteří dlouhodobě působili v roli projektového manažera nebo na vedoucí pozici (Levison, 2017). Tradiční projektový manažer i Scrum Master jsou zodpovědní za pomoc svým týmům, zaměřují se na to, jak pracovat a jak řešit problémy prostřednictvím procesu, ale jejich přístupy jsou naprosto odlišné (Rehkopf, 2018). Role projektového manažera v agilním světě není potřebná, protože ve Scrumu přebírá jeho odpovědnost rovným dílem Product Owner, Scrum Master a Development tým (Šochová, Kunce, 2019, s. 58). Scrum Master pomáhá týmu vylepšit a zefektivnit procesy, kterými dosahuje svých cílů, ale dělá to jako člen týmu, ne jako jejich manažer (Rehkopf, 2018). V agilním prostředí tak projektoví manažeři často přecházejí do role Product Ownera, Scrum Mastera nebo stakeholdera (Šochová, Kunce, 2019, s. 58).

### **Praktiky a artefakty ve Scrumu**

Artefakty definované Scrumem jsou speciálně navrženy tak, aby maximalizovaly transparentnost klíčových informací tak, aby všichni rozuměli danému artefaktu shodně (Schwaber, Sutherland, 2013). Mezi tyto praktiky a artefakty patří hlavně: Sprint, Sprint Goal, Product Backlog, Sprint Backlog, User Story, Epic a Témata, Definition of Done.

### **Sprint**

Jádrem Scrumu je Sprint – časově ohraničený rámec, během kterého je vytvořen použitelný produktový přírůstek (Schwaber, Sutherland, 2013). Každý Sprint začíná plánovací schůzkou – Sprint Planning – během, které se Product Owner a Development tým dohodnou na tom, co přesně bude během sprintu realizováno (Rous, 2019). Celý Sprint se skládá z plánování Sprintu, denních schůzek, vývojové práce, Sprint Review a retrospektivy Sprintu (Schwaber, Sutherland, 2013). Dle autorů mají Sprints ideálně konzistentní trvání po celou dobu vývoje.

Délka Sprintu má odrážet dynamiku businessu, zároveň musí být tým schopen v jedné iteraci dokončit určitou funkcionalitu a k tomu platí pravidlo „čím kratší Sprint, tím rychlejší zpětná vazba a adaptace na případné změny“ (Šochová, Kuncce, 2019, s. 69). Schwaber a Sutherland (2013) omezují délku Sprintu maximálně na jeden kalendářní měsíc, ale Šochová a Kuncce (2019, s. 69) doporučují délku jednoho Sprintu maximálně dva týdny, protože zpětná vazba pak přichází pozdě. Během sprintu tým pořádá každodenní Standup schůzky, aby diskutoval o svém pokroku a řešil nastalé problémy (Rous, 2019).

Cílem Sprintu není dokončit všechny položky ze Sprint Backlogu, ale dosáhnout stanoveného cíle Sprintu – **Sprint Goal** (Šochová, Kuncce, 2019, s. 69). Jak autoři vysvětlují, během Sprintu by tento cíl neměl být měněn, ale je možné měnit Sprint Backlog tak, aby bylo možné cíle dosáhnout. Také nemají být provedeny žádné změny, které by Sprint Goal ohrozily (Schwaber, Sutherland, 2013). Autoři dodávají, že se nesnižují kvalitativní cíle, ale rozsah Sprintu může být vyjasněn a znovu dohodnut mezi Product Ownerem a Development týmem. Development tým rozhoduje o množství práce, které je možné během jedné iterace realizovat. Product Owner má konečné slovo o tom, jaká kritéria musí být splněna, aby byla tato práce schválena a přijata (Rous, 2019). Finální výsledky Sprintu tým prezentuje na Sprint Review (Šochová, Kuncce, 2019, s. 70) a nová iterace začíná ihned po uzavření té předchozí (Schwaber, Sutherland, 2013). Product Owner může na konci Sprintu provedenou práci buď přijmout, nebo odmítnout na základě kritérií, která stanovil na Sprint Planningu (Rous, 2019). Sprint lze také zrušit před uplynutím časového limitu, toto oprávnění má pouze Product Owner, ačkoli může tak učinit pod vlivem zúčastněných stran, Development týmu nebo Scrum Mastera (Schwaber, Sutherland, 2013). Jak autoři vysvětlují, k rušení Sprintu dochází ve Scrumu zcela výjimečně, a to v případě, když by již realizace daného cíle neměla význam, protože se změnila okolnosti. Autoři upozorňují, že rušení Sprintů je pro tým velmi náročné a díky potřebě nového plánování spotřebovává zdroje.

## **Sprint Goal**

Sprint Goal, tedy cíl Sprintu, je jednou z nejpodstatnějších částí Scrumu, kterou lze označit jako malou vizi pro jeden Sprint (Šochová, Kuncce, 2019, s. 61). Je to sada cílů, která je vytvářena během schůzky Sprint Planning a lze jí dosáhnout implementací Product Backlogu (Schwaber, Sutherland, 2013). Autoři doplňují, že je to zároveň návod pro Development tým, aby pochopil, proč pracuje na daném přírůstku produktu. Správně formulovaný cíl je zaměřen na hodnotu produktu, nikoliv na funkcionalitu, a odráží potřeby zákazníků (Šochová, Kuncce, 2019, s. 61). Sprint Goal také slouží ke kontrole pokroku, k řešení rizik, pomáhá stanovit priority, podporuje týmovou práci a umožňuje efektivní rozhodování (Overeem, 2016). Bez jasného a společného cíle může být během plánování Sprintu vybráno příliš velké množství položek, členové týmu mohou spíše vnímat svoji vlastní práci než práci týmu jako celku, může být složité zvolit osoby, kterým má být výsledek prezentován na Sprint Review a celkově lze těžko hodnotit úspěšnost Sprintu (Verwijs, 2019). Pro tým je také důležitá úroveň náročnosti stanovených cílů. Cíle, které jsou náročné, jsou také motivující, pokud je ale nemožné cíle dosáhnout, je to pro tým deprimující (Sutherland, 2014, s. 109).

## **Product Backlog**

Product Backlog je uspořádaný seznam všeho, co by mohlo být potřeba realizovat, a je také jediným zdrojem požadavků na jakékoli změny (Schwaber, Sutherland, 2013). Product Backlog má na starosti Product Owner, ten zodpovídá za priority i celkovou smysluplnost, ale zároveň má být tento seznam přístupný všem (Šochová, Kuncce, 2019, s. 73). Product Backlog

není nikdy kompletní, vyvíjí se a mění s vývojem produktu a prostředí (Schwaber, Sutherland, 2013). Položka Backlogu (Product Backlog Item) je nezávislá funkcionalita, která přináší zákazníkovi hodnotu a díky tomu je možné na ni získat zpětnou vazbu (Šochová, Kunc, 2019, s. 67). Jak autoři vysvětlují, nejčastěji jsou tyto položky popsány formou **User Story**, to znamená, že jsou popsány z pohledu zákazníka, nikoliv z pohledu technologie nebo systému. V každém okamžiku lze také spočítat zbývající práci k dosažení cíle a za odhady náročnosti položek Product Backlogu odpovídá Development tým (Schwaber, Sutherland, 2013). Product Backlog ale obvykle neobsahuje jednotlivé tasky (úkoly), protože je to zbytečně velká úroveň detailu, která patří až na **Scrum tabuli** (Šochová, Kunc, 2019, s. 74–75).

Product Backlog má být seřazen dle priorit a na jeho vrcholu jsou nejdůležitější User Stories, které jsou definovány tak, aby je tým mohl akceptovat do Sprint Backlogu a realizovat v rámci Sprintu (Šochová, Kunc, 2019, s. 73). Dle autorů má stále být takto detailně připravena práce na 2 až 3 Sprinty dopředu, s nižší mírou podrobnosti by měla být nadefinována práce na následujících 5 až 10 sprintů a zbývající část může zůstat na úrovni velkých **Epiců** – velkých položek, které je ještě potřeba rozdělit na User Stories.

### **Sprint Backlog**

Sprint Backlog je malá součást Product Backlogu, která obsahuje funkcionality plánované pro jeden Sprint (Šochová, Kunc, 2019, s. 75). K vytvoření Sprint Backlogu dochází během Sprint Planningu, kde Product Owner představuje seznam prioritních položek z Product Backlogu a Development tým provádí analýzu seznamu. Zjišťuje, kolik práce může udělat, a připraví plán realizace (Mastropasqua, 2018). Díky Sprint Backlogu je veškerá práce, kterou tým označil jako potřebnou ke splnění cíle Sprintu, dobře viditelná (Schwaber, Sutherland, 2013). Jak autoři doplňují, pouze Development tým může měnit obsah Sprint Backlogu během Sprintu. Scrum nevyžaduje, aby tým vytvářel jednotlivé úkoly, ale většina týmů to i tak dělá (Mastropasqua, 2018). Dle Šochové a Kunc (2019, s. 75) rozpadnutí na jednotlivé úkoly zvyšuje přehled o rizicích a aktuálním stavu práce. Jak ale doplňují, tyto úkoly nejsou součástí Sprint Backlogu a jsou pouze interními praktikami daného týmu. Development tým také sleduje zbývající práci ve Sprint Backlogu na denních schůzkách – **Daily Scrum** – kontrolují tak šanci dosažení cíle Sprintu (Schwaber, Sutherland, 2013).

### **User Story**

Agilní přístupy často používají k popisu budoucí funkcionality termín User Story, základní podoba je: „Jako uživatel chci Funkcionalitu, abych dostal Business Value“ (Šochová, Kunc, 2019, s. 77). Jak autoři vysvětlují, každá User Story má být popsatelná, má vytvářet určitý obraz, být nezávislá, přinášet hodnotu a má být malá – pro toto označení se používá zkratka **INVEST** (Independent, Negotiable, Valuable, Estimable, Small a Testable). User Story není specifikace funkcionality, je to nástroj pro spolupráci (Pichler, 2016). Autor upozorňuje, že User Stories by nikdy neměly být pouze předány Development týmu, naopak Product Owner a tým by o nich měli společně diskutovat.

### **Epic a Témata**

Epic je v agilní metodologii velký objem práce, který nelze dokončit najednou, a lze jej dále rozdělit na menší User Stories (Sinha, 2019a). Pomocí Epicu lze načrtnout funkčnost produktu, aniž by se muselo zajít k jednotlivým podrobnostem (Pichler, 2016). Dle autora je

tento postup vhodný u popisu nových produktů i nových funkcionalit. Epic obvykle nebývá psán ve formátu User Story, protože je moc velký, obsahuje velkou míru nejistoty a nelze udělat ani reálný odhad náročnosti (Šochová, Kunce, 2019, s. 83). Epic může obsahovat stručný popis toho, co si klient přeje a co mu přináší hodnotu (Sinha, 2019a). Jak autor dále uvádí, tento popis vyžaduje následnou diskusi nebo brainstorming, aby byl možný bylo možné rozložení na menší části. Větší celky, které spolu nějakým způsobem souvisí, bývají spojovány do Témat (Šochová, Kunce, 2019, s. 83–84). Jak autoři vysvětlují, každý Epic je nezávislou položkou Product Backlogu a Témata jdou napříč funkcionalitami a sdružují jednotlivé položky dle logiky businessu.

### **Definition of Done**

Definition of Done definují globální podmínky, dle kterých lze určit, zda je položka kompletně zpracována (Šochová, Kunce, 2019, s. 84). Tato definice je zásadní pro dosažení transparentnosti v agilní praxi (Mitchell, 2017) a každý produkt nebo systém by měl mít popsanou svou vlastní Definition of Done jako standard pro veškerou svou práci (Schwaber, Sutherland, 2013). Jak autoři doplňují, očekává se, že tato definice se rozšíří a bude zahrnovat přísnější kritéria pro vyšší kvalitu práce společně s tím, jak se daný tým stává zkušenějším. Také Mitchell (2017) doporučuje začít nejdříve s jednodušší definicí, se kterou tým bude souhlasit, zvládne ji dodržovat, a postupně tento standard vylepšovat. Nejběžnější použití Definition of Done je na úrovni User Story a Development týmu, kdy lze určit například tato kritéria: byl zkontrolován kód, byla splněna kritéria akceptace, funkcionální, nefunkcionální a unit testy prošly, Product Owner akceptoval danou User Story (Huether, 2017). Tato definice je také nápomocná pro Development tým, aby byl schopen během plánování sprintu zvolit z Product Backlogu vhodné množství položek (Schwaber, Sutherland, 2013), protože práce, kterou tým nestihne dokončit, zůstává stále v procesu a generuje náklady bez možnosti dodat zákazníkovi určitou hodnotu (Sutherland, 2014, s. 109). Huether (2017) upozorňuje, že Scrum tým by vůbec neměl začínat pracovat, dokud se společně nedohodne na své Definition of Done.

### **Ohodnocení a jejich techniky**

Odhad je často vnímán jako závazek a vždy se na něj někdo ze zainteresovaných stran spoléhá, proto se Product Owner nemůže zavázat k dodání funkce k určitému datu, dokud danou věc nejdříve neodhadne Development tým (Moskalenko, 2017). Agilní metody používají nejčastěji k ohodnocení náročnosti User Stories relativní jednotky, kde odhady nejsou o čase, ale hodnotí se náročnost a komplexita, a to relativně vůči jednotlivým User Stories (Šochová, Kunce, 2019, s. 84). Scrum ale nevyžaduje, aby bylo odhadováno v relativních jednotkách a ani tým nezavazuje k používání určité techniky odhadů (Moskalenko, 2017). Jak autor vysvětluje, techniky, jako je Planning Poker, Story Points, Focus Factor, Dirty Hours, nejsou součástí Scrumu, ten pouze stanovuje určitá pravidla odhadů a tým si může vybrat techniku odhadů, která odpovídá jeho potřebám.

Ohodnocení provádí vždy celý tým, aby bylo do odhadu započteno skutečně vše a snížila se pravděpodobnost, že se na něco zapomene (Šochová, Kunce, 2019, s. 84). Techniky odhadů jsou také navrženy tak, aby byly rychlé (v porovnání s tradičními technikami) a aby nebylo možné obvinít někoho z nesprávného odhadu (Berteig, 2019). Dle Moskalenko (2017) mnoho Scrum týmů používá pro odhady techniku T-Shirt Sizes nebo relativní Story Points a dle Berteig (2019) a Šochové, Kunce (2019, s. 99) je velmi často využívána technika Planning

Poker, ale existují i další možnosti. Níže je uveden stručný popis nejpoužívanějších technik pro odhady v agilních metodách – Planning Poker a T-Shirt Sizes.

**Planning Poker** je neuvěřitelně jednoduchá metoda, jak se vyhnout ukotvení a zkreslení vnímání pomocí Halo efektu nebo efektu rozjetého vlaku, a zároveň umožňuje celému týmu sdílet znalosti na konkrétním úkolu (Sutherland, 2014, s. 131). Planning Poker obsahuje karty s čísly, obvykle je to stupnice: 0, ½, 1, 3, 5, 8, 13, 20, 40, 100, ?, nekonečno a kartu s obrázkem kávy (Šochová, Kunce, 2019, s. 99). Jak autoři vysvětlují, karta „nekonečno“ je použita, pokud je User Story příliš komplexní nebo ji neumí Product Owner popsat, kartu „otazník“ použije člen týmu v případě, že ještě nemá dostačující informace a chce se na něco zeptat Product Ownera. Karta „káva“ je určena pro dlouhé estimation meetingy, kde se odhaduje i několik hodin celý Backlog a je potřeba udělat si přestávku. Postup při Planning Pokeru je takový, že Product Owner představí User Story a každý člen Development týmu si zvolí kartu pro svůj odhad, umístí ji lícem dolů před sebe a na požádání všichni kartičky otočí současně (Alles, 2017). Jak autor upozorňuje, tomuto odhadu by neměla předcházet žádná diskuse mezi týmem. Pokud dle autora dojde k situaci, že jsou všechny odhady stejné, stačí číslo zaznamenat a lze přejít k odhadu další položky. Pokud se čísla liší, tak se Scrum Master ptá člena týmu s nejvyšším a nejnižším hodnocením na důvody jejich odhadu. Ti vysvětlí svá rozhodnutí a následuje druhé kolo hodnocení, kde mohou všichni své odhady přehodnotit (Šochová, Kunce, 2019, s. 99). Jak autoři doplňují, cílem této hry a cílem týmu je domluvit se, ale ne ve smyslu kompromisu, jedná se spíše o shodu celého týmu po společné diskusi. Autoři také upozorňují, že by neměly být počítány průměry hodnocení a o velikosti User Story by neměl rozhodovat manažer ani Scrum Master, protože tým nebude za ohodnocení cítit vlastní zodpovědnost a nebude motivovaný ke včasnému dokončení práce. Pokud se v dalším kole všichni členové týmu ústně dohodnou na odhadu, stačí odhad zaznamenat a je možné pokračovat na další položku (Alles, 2017). Alles ještě doplňuje, že v případě, že debata dále pokračuje více než pár minut, je potřeba tým požádat o nové hlasování. Dle autora by také členové týmu měli vždy volit vyšší číslo ze dvou, pokud se mezi nimi nemohou rozhodnout. Tato technika je hlavně v počátcích pro nový tým časově náročná, ale postupně, jak se celý Scrum tým zdokonaluje, stává se efektivní (Šochová, Kunce, 2019, s. 99). Dle Berteig (2019) je to dobrá technika pro odhad velmi malého počtu položek (2 až 10 položek).

U techniky **T-Shirt Sizes** Product Owner vysvětlí odhadovanou User Story a Development tým se ptá, pokud má nějaké nejasnosti. Pak každý vývojář ohodnotí User Story pomocí kartičky s velikostí triček S, M, L nebo XL a celý Development tým zvedne své karty současně (Mistry, 2017). Jak autor uvádí dále, následuje diskuse o rozdílných odhadech a tým se vrátí zpět k hlasování, dokud se všichni neshodnou na jedné velikosti. Stupnice T-Shirt Sizes se obvykle používá pro úvodní odhady, kdy jsou k dispozici pouze velké položky jako je Epic, a je potřeba mít představu o rámcových nákladech a čase (Šochová, Kunce, 2019, s. 84). Jak uvádí Berteig (2019), jedná se o velmi neformální techniku, kterou lze rychle použít i s velkým počtem položek. Mistry (2017) považuje tuto populární techniku za vhodnou pro seznámení s relativními odhady, výhodu vidí také v tom, že tato technika umožňuje rychlý postup při odhadování.



## **Meetingy a aktivity ve Scrumu**

Pro kontrolu a adaptaci procesu předepisuje metoda Scrum čtyři formální aktivity, kterými jsou Sprint Planning, Daily Scrum Meeting (Standup), Sprint Review a Sprint Retrospective (Schwaber, Sutherland, 2013).

### **Daily Scrum (Standup)**

K nejnámějším praktikám Scrumu patří Daily Scrum, kterému se také říká Standup nebo denní Scrum (Šochová, Kunce, 2019, s. 107). Jak autoři popisují, jedná se o pravidelnou každodenní schůzku, která probíhá obvykle ráno u Scrum tabule nebo na jiném místě tak, aby členové týmu neseděli na svých pracovních místech, ale naopak stáli. Standup se koná vždy ve stejnou dobu na stejném místě a každý člen Development týmu hovoří o tom, co udělal během předchozího dne, jakou práci bude dělat v aktuální den a případně o překážkách, které mu brání v práci (Schwaber, Sutherland, 2013). Standup není status meeting, protože účelem této schůzky je kontrola pokroku za předchozí den, plánování práce na aktuální den a hledání způsobu, jak díky spolupráci dosáhnout v daný den co největšího pokroku (Patton, 2019). Celá tato schůzka by měla trvat maximálně 15 minut, a pokud diskuse nebo řešení jednoho problému trvá déle než minutu, má být k dané věci naplánována samostatná schůzka, ale řešení nemá probíhat během Standupu (Patton, 2019). Dle Šochové a Kunce (2019, s. 107) lze pak v praxi stihnout schůzku sedmičlenného týmu i za 5 minut, pokud se nezabíhá do technických detailů. Development tým nebo někteří jeho členové se z těchto důvodů scházejí bezprostředně po Standupu, aby o problémech podrobně diskutovali nebo aby upravili naplánovanou práci pro zbývající část Sprintu (Schwaber, Sutherland, 2013). Scrum Master je zde pouze v roli moderátora nebo facilitátora, a pokud se schůzky účastní i někdo mimo Scrum tým, tak pouze v roli pozorovatele a do průběhu nijak nezasahuje (Šochová, Kunce, 2019, s. 107). Scrum Master také jen zajišťuje, že tuto schůzku pořádá vývojový tým, ale Development tým je zodpovědný za její průběh (Schwaber, Sutherland, 2013).

Jak Patton (2019) upozorňuje, tato schůzka je pro některé týmy nejvíce nepříjemná část dne, což je ale náznak toho, že by někteří členové měli změnit způsob, jakým pracují. Na to upozorňuje také Lum (2016) a doporučuje dodržovat několik zásad, díky kterým Standup pomáhá efektivněji řídit tým. Dle autora mají členové týmu během schůzky skutečně zůstat stát, držet se striktně tří základních otázek (co udělali včera, na čem pracují dnes a co jim brání v práci) a mít na viditelném místě nástroj pro správu projektů – fyzická nebo elektronická Scrum tabule. Dle autora mají schůzky vždy začít včas, aniž by se čekalo na pozdě přichodící členy. Také není potřeba zabíhat do podrobností, co vše bylo předchozí den uděláno, stačí stručná informace (Šochová, Kunce, 2019, s. 107). Správně realizované Standup schůzky zlepšují komunikaci, eliminují další setkání, identifikují překážky, podporují rychlé rozhodování a zlepšují úroveň znalostí Development týmu (Schwaber, Sutherland, 2013).

### **Sprint Planning**

Práce, které mají být provedeny v rámci jednoho Sprintu, jsou plánovány na schůzce Sprint Planning, kde plánování provádí celý Scrum tým (Schwaber, Sutherland, 2013). Schůzka začíná tím, že Product Owner představí týmu Sprint Goal, ale pro tým by tento cíl neměl být úplnou novinkou, protože tým již má mít znalosti o produktovém Backlogu (Šochová, Kunce, 2019, s. 121). Jak autoři dále vysvětlují, Scrum Master schůzku moderuje, a hlavně u začínajících týmů musí být aktivní a pomáhat týmu pochopit, k čemu se zavazuje, aby byl

jejich závazek reálný. Během schůzky Development tým vybere z Product Backlogu položky, u kterých se chce zavázat k jejich dokončení do konce Sprintu (Gratis, 2017). Jak autorka vysvětluje, Product Owner pouze určuje prioritu, ale množství provedené práce určuje pouze Development tým. Rozpad na jednotlivé úkoly je také součástí Sprint Planningu, i když některé týmy tuto část nechávají na první den Sprintu (Šochová, Kunce, 2019, s. 121). Jak autoři vysvětlují, úkoly by měly být zhruba jednodenní, maximálně dvoudenní. Na konci schůzky by měl každý člen týmu vyjádřit verbálně svůj souhlas se závazkem daného Sprintu a rovněž je vhodné se ujistit, že každý člen týmu má alespoň jeden úkol, na němž může začít pracovat (Radigan, 2019). Když Development tým dojde k závazku – vybere určitý počet User Stories pro nadcházející Sprint – měl by mít ještě možnost určitý čas nad svým rozhodnutím diskutovat bez Scrum Mastera a Product Ownera, a teprve pak svůj závazek potvrdit nebo změnit (Šochová, Kunce, 2019, s. 121).

V ideálním případě by měl Sprint Planning probíhat na začátku týdne, aby nebyl narušován kontext a práce týmu víkendem (Radigan, 2019). Plánování Sprintu je časově omezeno na maximálně osm hodin pro Sprint trvající jeden měsíc a u kratších Sprintů je schůzka obvykle kratší (Schwaber, Sutherland, 2013). Dle Šochové a Kunce (2019, s. 121) trvá zkušenějšímu týmu plánování dvoutýdenního Sprintu přibližně 30 minut, pokud tým zná Product Backlog a připravuje se průběžně na plánování. Jednotliví členové Development týmu také musí odhadnout, kolik čistého času mají pro samotnou práci na daném Sprintu, protože se také účastní schůzek, opravují chyby nebo mají na starost údržbu systému (Gratis, 2017). Dle autorky má každý člen týmu realisticky 4 až 6 hodin čisté práce denně na realizaci Sprintu, proto odhadnutý čas k dokončení všech položek vybraných pro Sprint nesmí přesáhnout dobu, kterou mají členové týmu k dispozici.

Se Sprint Planningem souvisí také **Backlog Grooming**, což je schůzka pro lepší porozumění obsahu Product Backlogu, při které Product Owner prochází s Development týmem jednotlivé User Stories, vysvětluje jejich význam a tým ohodnocuje nové User Stories (Šochová, Kunce, 2019, s. 119). Backlog Grooming má trvat 30 minut a tým by zde měl řešit problémy, jako jsou nedostatek podrobností nebo chybějící kontextové informace, které bude potřebovat v následujících dvou sprintech (Radigan, 2019). Dle autora pak tyto záležitosti nejsou blokující a nezdržují samotný Sprint Planning. Backlog Grooming lze naplánovat do poloviny Sprintu nebo i několikrát během něj (Šochová, Kunce, 2019, s. 119).

## **Sprint Review**

Sprint Review je pořádáno po ukončení aktuálního a zároveň před zahájením nového Sprintu (Azanha at al., 2017). Jak autoři popisují, při této schůzce Development tým demonstruje výsledek své práce. U Sprintů, které trvají jeden měsíc, je Sprint Review zhruba čtyřhodinová schůzka, u kratších Sprintů je událost obvykle kratší (Schwaber, Sutherland, 2013). Schůzky se účastní Scrum Master, Product Owner, Development tým, ale také například zákazníci, vývojáři z jiných projektů (Conçalves, 2019). Zákazníkem je zde kdokoliv, kdo má na produktu nějaký zájem (Šochová, Kunce, 2019, s. 125). Jak autoři doplňují, tyto zákazníky obvykle nezajímají jednotlivé User Stories, ale celkový inkrement produktu, proto je vhodné co nejvíce zákazníky zapojit a umožnit jim vyzkoušet si novou funkcionalitu. Ani toto setkání není „status meeting“, je to neformální schůzka, která má vyvolat zpětnou vazbu a podpořit spolupráci (Schwaber, Sutherland, 2013). To potvrzuje i Gonçalves (2019) a doplňuje, že tyto schůzky jsou neformální záměrně, aby nebyly velkou zátěží nebo rozptýlením pro tým, proto je také stanoven limit na přípravu (maximálně 2 hodiny) a je zakázáno používat nástroje jako PowerPoint. Wolpers (2019) vysvětluje, že prezentace pomocí těchto typů nástrojů je zakázána z toho důvodu, že účastníci by se mohli nudit. Jak autor vysvětluje, o novém přírůstku by se nemělo pouze hovořit, ale měl by být reálně předveden a zákazníci by měli mít

možnost si funkcionalitu vyzkoušet. Samotná prezentace má být provedena přímo Development týmem, protože členové týmu nejlépe vědí, jak daný produkt funguje a co mají zákazníkovi ukázat (Šochová, Kuncce, 2019, s. 125). Jak autoři doplňují, tento přístup také zvyšuje motivaci jednotlivých členů týmu i pochopení požadavků zákazníka.

Na Sprint Review patří jen zcela hotové User Stories (používá se výraz „done-done“), jsou to User Stories, které splňují Definition of Done, a jsou již akceptované Product Ownerem v průběhu Sprintu (Šochová, Kuncce, 2019, s. 125). Je také důležité, aby byl prezentovaný produkt posuzován z hlediska cíle Sprintu a mohlo tak být potvrzeno, že tým cíle dosáhl (Conçalves, 2019). Mimo to na schůzce diskutuje celá skupina o nápadech, které mohou pomoci při definování obsahu pro další Sprint (Azanha at al., 2017). Development tým také hovoří o tom, co se během Sprintu povedlo, ale i o problémech, které musel řešit (Schwaber, Sutherland, 2013). Product Owner by se ale o těchto problémech neměl dozvědět až na Sprint Review, tým by ho měl informovat průběžně (Wolpers, 2019). Jak autor uvádí, chybou by také bylo, pokud by v průběhu schůzky Product Owner nepřijímal zpětnou vazbu od zúčastněných stran nebo vývojového týmu. Product Owner během schůzky obvykle také hovoří o stavu Product Backlogu a o termínech dokončení na základě dosavadního pokroku (Schwaber, Sutherland, 2013). Jak autoři doplňují, úkolem Scrum Mastera je zajištění této schůzky, udržení časového rámce a informování všech stran o smyslu a cílech setkání.

## **Sprint Retrospective**

Po Sprint Review, ale ještě před Sprint Planningem by mělo proběhnout přezkoumání dokončeného Sprintu – Sprint Retrospective (Azanha at al., 2017). Dle autorů by na tomto setkání měl Scrum Master pomoci týmu zamyslet se nad získanými zkušenostmi, protože hlavním cílem tohoto setkání je zlepšování skrze získané zkušenosti. Sprint Retrospective poskytuje formální příležitost k zaměření se na kontrolu a adaptaci na úrovni jednotlivých lidí, vztahů, procesů a nástrojů (Schwaber, Sutherland, 2013). Sprint Retrospective může pomoci motivovat členy Development týmu tím, že jim poskytne příležitost hovořit, sdílet své nápady a zároveň být vyslechnuti (Ribeiro, 2019). Ale retrospektiva bez konkrétního seznamu akcí, které povedou ke zlepšení, by neměla smysl (Šochová, Kuncce, 2019, s. 111). Dle Ribeiro (2019) mnoho agilních lídrů ze začátku nerozumí skutečné hodnotě této schůzky a nepořádají tato setkání vůbec, protože je chybně považují za ztrátu času. Autor naopak vysvětluje, že tato setkání jsou skvělou příležitostí jak pro ocenění přínosu všech členů týmu, tak i pro odhalení časových a finančních rizik. Během retrospektivy Scrum tým také plánuje, jak zvýšit kvalitu produktu úpravou Definition od Done (Schwaber, Sutherland, 2013). Dle Andrei at al. (2019) je právě Sprint Retrospective důkazem toho, že Scrum je adaptabilní rámec, který umožňuje implementaci potřebných změn.

Sprint Retrospective má standardně několik fází: úvod, sběr dat, hlubší porozumění informacím, brainstorming možností a shrnutí konkrétních akcí (Šochová, Kuncce, 2019, s. 111). Jak autoři vysvětlují, schůzka může mít různé formáty, ale nejčastějším způsobem sběru dat je takzvané „kolečko“, kdy každý člen týmu postupně odpovídá na tři otázky: co se mu líbilo a v čem chce pokračovat, co se mu nelíbilo a chtěl by to změnit, co by chtěl zavést nového. Jak dále autoři popisují, v další fázi Scrum Master shrne vše, co členové týmu říkali, a nechá tým diskutovat (brainstorming), případně požádá o vysvětlení a následně tým zvolí konkrétní akce, které chce v příštím Sprintu realizovat. Dle (Ribeiro, 2019) je klíčem úspěchu hlasování týmu o aktivitách, které by měly být následně realizovány. Úkolem Scrum Mastera je opět zajištění schůzky, dodržení stanovené délky (3 hodiny pro měsíční Sprint) i toho, že tým rozumí smyslu schůzky (Schwaber, Sutherland, 2013).

## Agilní programování

Scrum je rámec pro vytvoření dobré spolupráce v týmu, zapojení zákazníka, zvýšení flexibility, efektivity a motivace, ale bez implementace některých praktik agilního programování není kompletní (Šochová, Kunce, 2019, s. 127). Jak autoři popisují, mezi tyto metody patří například Pair Programming, Review, Coding Standard, Continuous Integration a Code Refactoring.

**Pair Programming** je technika, kdy dva programátoři pracují společně na jednom pracovním místě a střídají se v psaní kódu – jeden z nich kód píše, druhý ho kontroluje a společně diskutují (Bathia, 2019, s. 39). Dle Šochové a Kunce (2019, s. 127) je tato technika rychlejší, než když dva programátoři pracují samostatně, urychluje zaškolení pracovníků a také se výrazně snižuje procento chyb. Tato metoda také souvisí s metodou **Review**, protože zde je kód kontrolován přímo při jeho vytváření (Bathia, 2019, s. 39). Při Review má být zkontrolován kód, dokumentace, testy, design i analýza další osobou, aby byly případné chyby odhaleny co nejdříve (Šochová, Kunce, 2019, s. 128). Nalezené chyby mají být opraveny ihned po jejich zjištění, protože odkládání na později mnohonásobně zvyšuje čas potřebný na jejich opravu (Sutherland, 2014, s. 109). Lepší čitelnost kódu a snadnější opravy chyb zajišťuje **Code Refactoring**, při kterém je stávající kód upravován, aniž by se změnilo vnější chování kódu, protože dochází pouze k „pročištění“ a ke zjednodušení kódu (Bathia, 2019, s. 42). Jak popisuje autor, výhodou této metody je následná lepší čitelnost kódu a snadnější opravy chyb. Jak doplňuje Šochová a Kunce (2019, s. 134), tým by měl refaktorovat průběžně a aktivně hledat, které části by měly být refaktorovány.

Další sadou pravidel a postupů jsou **Coding Standards** (standarty psaní kódu), které zajišťují, že kód je snadno čitelný, udržovatelný, rozšiřitelný, a také snižují riziko vzniku chyb (Powers, 2014). Autor uvádí jako běžně používané standardy psaní kódu například konzistentní velikost písmen v názvech metod a vlastností (dle doporučení pro daný programovací jazyk, popisné názvy metod a tříd, udržování metod v přiměřené délce. Standard psaní kódu by měl být akceptovaný a dodržovaný celým týmem za všech okolností (Šochová, Kunce, 2019, s. 129–131). Také by dle autorů měla probíhat automatická kontrola pomocí vhodného nástroje, aby bylo vždy ověřeno, že nový kód v systému **Continuous Integration** neporušuje dohodnuté standardy. Jak autoři dále vysvětlují, prostředí Continuous Integration umožňuje integraci kódu v reálném čase a z pohledu procesu je také jednou z prvních věcí, nad kterými se musí Scrum tým zamyslet (Dileep, 2019). A jak autor dále uvádí, klíčem je automatizace, protože manuální testování často vytváří překážky. Pro rychlý nárůst automatických testů je vhodné zahrnout je do Definition of Done, pokrytí kódu automatickými testy se pak poměrně rychle zvětšuje, a naopak klesá množství chyb nalezených zákazníkem (Šochová, Kunce, 2019, s. 132).

## **Shrnutí kapitoly**

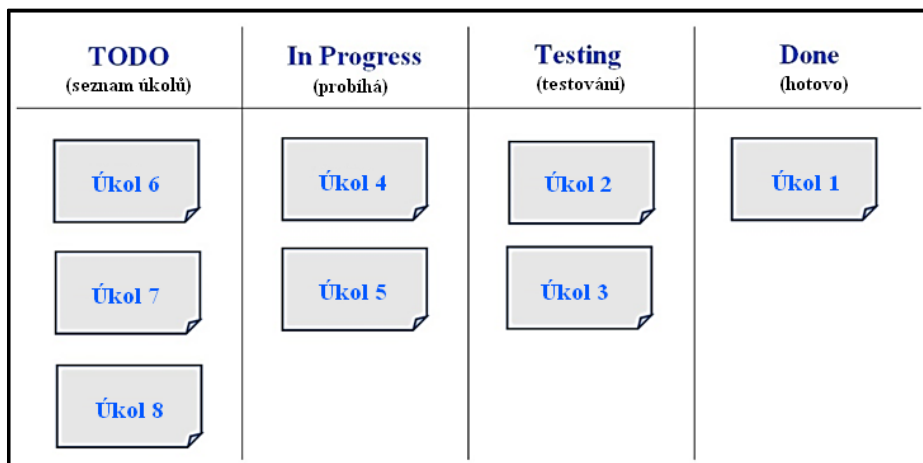
Tato kapitola popsala nejdůležitější praktiky a aktivity této agilní metody. Vysvětlena byla jejich náplň, cíle a vzájemná provázanost v tomto procesu. Uvedeny byly také všechny důležité role a jejich odpovědnost v rámci jednotlivých aktivit. V závěru kapitoly byly zmíněny doporučené techniky agilního programování, které je vhodné do Scrumu doplnit. Role, artefakty, události a pravidla Scrumu jsou neměnné, a ačkoli je možná implementace pouze částí Scrumu, výsledkem pak není Scrum, protože Scrum existuje pouze jako celek (Schwaber, Sutherland, 2013). Metoda Scrum bývá v IT oboru využívána samostatně, ale v praxi se také osvědčily její různé kombinace s metodou Kanban, protože se mohou vzájemně doplňovat a celý proces tak ještě více zefektivnit.

## 2.2.2 Metoda Kanban

Kanban původně pochází z Japonska a řídil se jím počet lidí, kteří byli vpuštěni do chrámu, aby nebyla přeplněna jeho kapacita (Šochová, Kunce, 2019, s. 151). Slovo Kanban v japonštině znamená „štítek“ nebo „lístek“ (Redka, 2018). Metoda Kanban byla poprvé představena jako metoda plánování v továrnách Toyota, kde byla vyvinuta za účelem zlepšení pracovního postupu a udržení vysoké úrovně výroby montážních linek (Andrei at al., 2019). Kanban je běžně používán týmy, které potřebují být schopny měnit prioritu práce během dne a jejichž práce je založena na nepřetržitém toku činností (Hebb, 2019). Dle Šochové a Kunce (2019, s. 151) Kanban není sám o sobě proces, ale procesem se stává až při samotné implementaci. Autoři také doporučují implementovat Kanban společně s prvky Scrum metody, aby se zvýšila šance na úspěch, anebo naopak, jak je často v praxi děje, doplnit principy Kanbanu do Scrum procesu.

Aplikování procesů metody Kanban při vývoji softwarových produktů znamená v podstatě vytvoření přehledné tabule s minimálně třemi sloupci s názvem „**Backlog**“, „**In progress**“ a „**Done**“, přípravu lístečků s úkoly a nastavení limitu, kolik úkolů může být v jednotlivých sloupcích (Šochová, Kunce, 2019, s. 151). Jak autoři dále vysvětlují, když kvůli limitu nelze přidat další úkol, tak musí člen týmu nejdříve dokončit některý z již rozpracovaných. Každý úkol je znázorněn na kartě a těmi se pohybuje napříč tabulí, ale před přesunem do dalšího sloupce musí být dokončena aktuální fáze úkolu (Brown, 2019). Kanban metoda tímto přístupem snižuje množství právě vykonávané práce (Work In Progress) a zároveň vše vizualizuje prostřednictvím klasické nebo elektronické tabule, což je běžně používaná metoda vizualizace v agilním prostředí (Paquette, Frankl, 2015, s. 60). Možnosti nastavení Kanban tabule jsou nekonečné, avšak princip vždy zůstává v tom, že řada pracovních úkolů prochází chronologicky zleva doprava (Roseke, 2019). Obrázek 1 Kanban tabule níže znázorňuje jednoduchou variantu se 4 sloupci.

Obrázek 1 Kanban tabule



Zdroj: Roseke (2019), upraveno

Obrázek 1 Kanban tabule uvedený výše je variantou se čtyřmi sloupci – TODO, In Progress, Testing a Done. V prvním sloupci TODO, který je někdy označen také jako Backlog, jsou nashromážděny nové úkoly (Roseke, 2019). Sloupec In Progress obsahuje pouze úkoly, na kterých se aktuálně pracuje. Pro oddělení úkolů, které jsou určeny k testování, je zde přidán třetí sloupec Testing. Čtvrtý sloupec Done obsahuje již kompletně dokončené položky. Každý úkol je umístěn na samostatnou kartu (nalepovací poznámka na fyzické tabuli nebo položka na elektronické tabuli ve vhodném softwarovém nástroji) a v agilní metodice

projektového řízení jsou User Stories umístěny na tabuli společně s jejich dílčími úkoly (Roseke, 2019).

U metody Kanban není nastavení rolí v týmu podmínkou, neexistují časově ohraničené iterace vývoje, Kanban tabule může být průběžně aktualizována a odhady úkolů nejsou potřeba (Andrei at al., 2019). Kanban je systém postavený na agilních principech, je transparentní, dobře vyvážený a významně přispívá k organizaci práce (Redka, 2018). Dle autorky lze díky flexibilitě tuto metodu snadno přizpůsobit potřebám týmu, zejména pokud se rychle změni priority. Tuto větší volnost považuje naopak Šochová a Kunce (2019, s. 26) za hlavní slabinu Kanbanu, který si tak změnu řízení sám nevynucuje. Jak autoři popisují, je potřeba dodržet pouze tři základní principy: **omezit rozpracovanou práci, minimalizovat čas průchodu a vizualizovat progres** (Šochová, Kunce, 2019, s. 151). U Kanban týmů lze také očekávat, že vytvoří špatné návyky a neefektivní procesy, protože chybí časový rámec a existuje mnohem méně požadovaných aktivit, které pomáhají neustálému zlepšování (Hebb, 2019). Dle autora tento stav vyžaduje více týmové disciplíny a autor vidí možné řešení v zavedení týmových rolí. Naopak přínosem metody Kanban je podpora otevřené komunikace, transparentnosti, spolupráce a neustálého zlepšování tím, že pomáhá řešit úzká místa, překonávat překážky a maximalizovat produktivitu (Paquette, Frankl, 2015, s. 60).

## Shrnutí kapitoly

Tato kapitola popsala základní principy metody Kanban a zaměřila se na její výhody i možná rizika plynoucí z toho, pokud je tato metoda využívána samostatně nebo je aplikována na nevhodný typ projektu. Popsány byly také způsob vizualizace pomocí Kanban tabule a správa jednotlivých úkolů. V následující části je metoda Kanban porovnána s metodou Scrum, jsou zde uvedeny jejich společné rysy i odlišnosti.

### 2.2.3 Scrum vs. Kanban

Agilní přístupy, jako je Kanban a Scrum, lze také považovat za nástroje, které jsou navrženy tak, aby pomáhaly regulovat a organizovat práci na projektu (Redka, 2018). Stejně jako se Scrum a Kanban od sebe liší, sdílejí i mnoho podobných vlastností – jsou to metody empirické, lean (štíhlé) a agilní, zaměřují se na organizování týmů i zlepšování procesů (Hebb, 2019). I když jsou obě metody agilního typu, liší se z pohledu toho, jak přísně má být projekt řízen (Andrei at al., 2019). Jak autor popisuje, Kanban umožňuje týmům a jejich členům svobodnější organizaci. Na druhou stranu autor uvádí i řadu podobností jako například to, že obě metody jsou zaměřené na rychlé dodávání funkcionalit a v obou případech je častá komunikace se zákazníkem, aby bylo možné získávat zpětnou vazbu co nejdříve. Mezi společné rysy těchto dvou metod patří také flexibilita a samostatně organizované týmy, které pracují na vývoji produktu (Redka, 2018). Jak autorka dále vysvětluje, obě metody také používají k vizualizaci tabule (fyzické nebo elektronické). Scrum tabule je ale po dokončení iterace (Sprint) vždy smazána, zatímco Kanban tabule zůstává zachována po celou dobu vývoje (Redka, 2018). Obě metody také používají plánování, ale pro Kanban je plánování relativně přímé, protože neexistují žádné Sprints, ale existuje pouze jeden seznam nevyřízených požadavků, ze kterého členové týmu čerpají dle určených priorit (Hebb, 2019). Protože je Kanban založen na nepřetržitém pracovním postupu, může tým dodávat přírůstky produktu tak často, jak je potřeba (Redka, 2018). Z tohoto důvodu není práce odhadována (Hebb, 2019). Oproti tomu Scrum požaduje, aby Development tým dodal produktový přírůstek na konci každého Sprintu (Redka, 2018). Jak autorka dále popisuje, tyto metody se zásadně liší v definování rolí a schůzek – Scrum má role i schůzky striktně určené a Kanban naopak nechává definici rolí i schůzek zcela volně na jeho uživatelích.

Jak uvádí Hebb (2019), Scrum je efektivnější pro řízení vývoje produktů a softwarových projektů a Kanban se zdá být efektivnější pro projekty typu Help Desk nebo Service Desk. Brown (2019) také potvrzuje, že pro vývoj softwaru obvykle týmy používají metodu Scrum, protože je užitečná v procesu životního cyklu softwaru. Redka (2018) doporučuje Kanban pro již probíhající projekty, kde se nové úkoly řadí dle priorit, a Scrum pro velké projekty a malé startupy. Dle Šochové (2018, s. 156) neexistuje otázka, zda zvolit Scrum nebo Kanban jako vhodnou agilní metodu. Dle autorky je Kanban integrální součástí Scrumu, do kterého doplňuje potřebné principy. Také dle Brown (2019) je v některých případech vhodné současné používání obou těchto metod.

## 2.2.4 Srovnání klasických a agilních metod

Předcházející kapitoly se věnovaly klasickým metodám řízení projektů – Vodopádová metoda a Spirálová metoda a následně popsaly přístupy dvou agilních metod – Metoda Scrum a Metoda Kanban. Protože se klasické a agilní metody výrazně liší, tato kapitola shrnuje jejich nejpodstatnější rozdíly.

Všechny metody řízení vývoje softwaru lze také rozdělit na adaptivní nebo prediktivní (Anurina, 2019). Prediktivní metody, do kterých patří metoda Vodopád a Spirála, zcela závisí na analýze požadavků a pečlivém plánování na začátku cyklu, každá změna prochází přísnou správou řízení změn a stanovením priorit (Stoica at al., 2013). Jak autoři dále vysvětlují, agilní metody naopak používají adaptivní přístup všude tam, kde neexistuje žádné podrobné plánování a tým se přizpůsobuje dynamickým změnám v požadavcích na vyvíjený produkt. Níže uvedená **Tabulka 1 Porovnání agilních a klasických metod** stručně popisuje hlavní odlišnosti těchto dvou přístupů.

**Tabulka 1 Porovnání agilních a klasických metod**

	Agilní metody	Klasické metody
Požadavky	nejsou zcela jasné, mohou se během procesu měnit	jsou jasné a nemění se během vývojové fáze
Velikost projektu a týmu	malý až střední projekt i tým	velký projekt i tým
Plánování	silně závisí na zpětné vazbě	je zahrnuto v dokumentaci
Refaktorování kódu	není drahé a lze jej snadno provádět	je vždy drahé a náročné
Architektura projektu	je během celého procesu flexibilní	je vytvářena podle aktuálních a budoucích požadavků
Rizika	rizika se mohou objevit v kterékoli fázi a mohou mít výrazný dopad na projekt	jsou dobře známa již ve fázi přípravy a projekt je na ně připraven

Zdroj: Stoica at al. (2013), upraveno.

Výše uvedená shrnuje základní rozdíly těchto dvou přístupů, mezi které patří zřetelnost a neměnnost požadavků i návrhu architektury u klasických metod, a naopak flexibilita a určitá míra vágnosti u metod agilních. To se pak také odráží ve znalosti nebo neznalosti rizik projektu. U agilních metod se flexibilita odráží i v plánování, které je závislé na zpětné vazbě. A naopak klasické metody se více soustředí na proces plánování jako takový a jeho podrobnou dokumentaci. Klasické metody řízení jsou doporučovány spíše pro velké projekty, kde se na vývoji podílí také velké týmy. Agilní metody jsou vhodnější pro malé až střední projekty, jimž svou velikostí odpovídají vývojové týmy. Podstatně se tyto dva přístupy liší

i ve schopnosti refaktorování již napsaného kódu, kdy u agilní metody bývá tato činnost prováděna poměrně snadno bez vysokých nákladů. U klasických metod je tomu přesně naopak.

Největší výhodou agilního přístupu je jeho flexibilita (Anurina, 2019). Jak autorka popisuje, tento přístup je navržen tak, aby vedl k nepřetržitým dodávkám, neustálému zlepšování a k rychlému vývoji. U agilních metod jsou velké úkoly vždy rozděleny na menší, které lze relativně rychle vyvinout (Redka, 2018), a díky tomu lze změnit priority z jedné iterace na druhou (Eby, 2016). Agilní procesy zvyšují obchodní hodnotu produktu pomocí User Stories, které popisují nezbytné funkce očekávané týmem i zákazníkem (Paquette, Frankl, 2015, s. 9). Silnou stránkou agilní metodiky je interakce s klienty (Stoica at al., 2013). Pro tento přístup je dle autorů také typická otevřená komunikace, minimální dokumentace, úzká spolupráce týmů.

U klasických metod je nutné provést podrobný přehled všech potřeb a navrhnout software tak, aby splňoval všechny předpokládané požadavky (Anurina, 2019). Tyto metody jsou zaměřeny na úkoly a zdůrazňují spíše nadefinované náklady, smlouvy a dokumentaci než obchodní hodnotu (Paquette, Frankl, 2015, s. 7). V klasických metodách pracují týmy s podrobným plánem, mají kompletní seznam úkolů, které musí být dokončeny v příštích několika měsících nebo během celého životního cyklu produktu (Stoica at al., 2013). Díky tomuto postupu je klasické řízení procesu vývoje časově náročné (Anurina, 2019).

Agilní a klasické metody jsou zcela odlišné přístupy k řízení projektů, ale mají jeden společný cíl, a tím je dodat zákazníkovi vysoce kvalitní produkt (Andrei at al., 2019). Konečný výběr metody pro řízení projektu by měl být týmovým rozhodnutím na základě konkrétních potřeb (Andrei at al., 2019). V současné době stále roste popularita metody Scrum i metody Kanban a společně zaujímají místo dříve populárního Vodopádového modelu (Redka, 2018). Dle Autorů se obě metody osvědčily jako nástroje zvyšující výkon, navíc výzkum dokazuje, že metoda Scrum je více než dvakrát účinnější v porovnání s klasickými metodami (Paquette, Frankl, 2015, s. 58).

### **2.3 Implementace agilní metody Scrum**

Být agilní by samo o sobě nemělo být cílem firmy, protože je to pouze cesta a způsob, jak dosahovat strategických cílů (Šochová, Kunce, 2019, s. 29). Agilní transformace je v podstatě o formování týmu, pravidelném přírůstku produktu, testování softwaru a vytváření Backlogů (Zack, 2018). Jak autor dále popisuje, v širším pohledu je tato změna také o propojování týmů, koordinování závislostí, dělání kompromisů a rychlém dodávání nových produktů na trh. Agilní transformace je náročná změna, protože je to zásadní změna myšlení a přístupu (Šochová, Kunce, 2019, s. 29). Jak autoři vysvětlují, není to jen změna procesů a nástrojů, ale naopak cesta, která nikdy nekončí a vede k neustálému zlepšování. Proto agilní transformace vyžaduje změny ve všech oblastech podnikání a nutná je také podpora vedení (Zack, 2018). Tak jako při implementaci jakéhokoli typu procesního modelu existuje také u metody Scrum více možností implementace (Haslinger, Wintersteiger, 2017). Jak autoři popisují, existuje buď možnost „velkého třesku“, kdy celá organizace přejde na metodu Scrum a všechny týmy zahájí svůj první sprint ve stejný den, anebo jednoduchý „Roll-Out“ model s jedním pilotním projektem, na kterém se bude firma učit.

Následující podkapitoly se zabývají variantou agilní transformace pomocí pilotního projektu a věnují se oblastem, které s touto změnou souvisejí. Mezi tato témata patří – sestavení týmu a jeho následný vývoj, agilní smlouva, agilní metriky a metody hodnocení týmu.



### 2.3.1 Pilotní projekt

Zahájit pilotní projekt není snadné, protože tento projekt bude čelit organizační gravitaci (Pavlichenko, 2017). Proto je dle autora potřeba získat podporu u vedení i řadových pracovníků. Obsahem pilotního projektu je vytvoření skutečné aplikace, která bude nasazena do produkčního prostředí. Cílem pilotního projektu je prokázat životaschopnost agilní metodiky v dané organizaci (Marcias, 2015). Proto je dle autora potřeba najít pro pilotní projekt vhodného kandidáta. Ideálně to může být aplikace s nízkým rizikem, která však bude pro danou organizaci přínosem (Marcias, 2015). Pilotní projekt by také neměl být příliš malý nebo nedůležitý (Haslinger, Wintersteiger, 2017). Tým také může na pilotním projektu vyzkoušet přístup Minimum Viable Product (MVP), jehož klíčovou myšlenkou je dodání životaschopného produktu s minimem funkcí (Waldock, 2015, s. 75). Jak autorka dále vysvětluje, MVP je koncept, který pomáhá identifikovat minimální životaschopnou funkcionalitu produktu, která přináší adekvátní hodnotu.

Dle Haslinger, Wintersteiger (2017) je pro pilotní projekt ideální velikost Development týmu sedm lidí plus Scrum Master a Product Owner. Takový projekt může poskytnout cenné poznatky o schopnosti přijmout agilní přístup v rámci celé organizace, také může odhalit skryté organizační hodnoty a priority nebo upozornit na nedostatky, kterým se dříve nevěnovala pozornost (Oshida, 2019). Pilotní projekt obvykle mnohem více zviditelní i současné problémy, jako jsou nedostatek technických dovedností vývojářů, nedostatek kvality, složitost, nepochopitelný a neudržitelný kód, chybná opatření a metriky, žádné automatizované testy (Haslinger, Wintersteiger, 2017). Kromě technických problémů může pilotní projekt dle autorů pomoci odhalit i nevhodné organizační struktury a procesy.

Metoda Scrum provází agilní transformaci jako ukazatel směru, ale konkrétní cestu si musí najít a vyzkoušet každá firma až v praxi (Šochová, Kunc, 2019, s. 141). To ale dle autorů neznamená vybrat si k implementaci pouze některé části Scrumu, ale naopak implementovat Scrum jako celek. Haslinger a Wintersteiger (2017) nedoporučují vytvářet dlouhodobý plán implementace metody Scrum po vzoru vodopádových projektů, ale doporučují zaměřit se na odhalení realistických, měřitelných výsledků v krátkodobém horizontu. Dle autorů by plánování mělo být individuální a krátkodobé, protože každá organizace je jiná a nachází se v jiné situaci. Také Sutherland (2014, s. 143) nedoporučuje vše plánovat dlouhodobě dopředu, ale pouze tak, aby bylo možné udržet tým plně zaměstnaný. V malé organizaci doporučuje Šochová a Kunc (2019, s. 177) začít s implementací agilních procesů co nejdříve, a to na úrovni týmu, který si sám na sobě jednotlivé praktiky přímo vyzkouší. V malém projektu je úspěch do značné míry definován dovednostmi a kompetencemi členů týmu, úrovní důvěry mezi nimi, jejich snahou, nadšením a dobrými postupy (Pavlichenko, 2017). Přesto je dle autora potřeba před spuštěním projektu připravit vhodnou strukturu a podmínky pro tým, například tým a Scrum Master budou pracovat v jedné místnosti, ideálně u jednoho velkého stolu, a Scrum Master bude pracovat s jedním nebo maximálně se třemi týmy a na plný úvazek.

Před spuštěním prvního Sprintu je třeba, aby Product Owner předložil týmu vizi produktu – jasně komunikovatelný a zastřešující cíl. Tato vize má být známa všem v týmu, má být srozumitelná, komplexní, ale přesto konkrétní, snadno sdílená a poutavá (Haslinger, Wintersteiger, 2017). Marcias (2015) doporučuje také uspořádat úvodní workshop pro User Stories, na kterém se sejdou všechny zúčastněné strany a je jim vysvětlen kontext, probíhá brainstorming nad každou User Story a následně jejich prioritizace. Velmi důležitým krokem při implementaci Scrumu je vytvoření Product Backlogu (Sinha, 2019b). Je potřeba sestavit první verzi Product Backlogu, kde již budou mít položky stanové priority, budou dostatečně podrobně popsány včetně odhadu (alespoň ty na nejvyšších místech), následně lze spustit první Sprint (Haslinger, Wintersteiger, 2017). Začátek práce týmu by měl být pozvolný.

Pokud se očekává plná rychlost 30 bodů, tak na první Sprint by měla být plánována práce jen 5 bodů, druhý Sprint 12 bodů, třetí Sprint 25 bodů a na čtvrtý Sprint je možné naplánovat plnou hodnotu (Šochová, Kunce, 2019, s. 141). Po třech až pěti sprintech se již obvykle objevuje mnoho poznatků, jaké procesy a standardy způsobují problémy a jak obtížné je jejich odstranění (Haslinger, Wintersteiger, 2017). Šochová a Kunce (2019, s. 141) nedoporučují začínat nultým Sprintem, ve kterém tým nedodá žádnou funkcionalitu, ale pouze se připravuje na společnou práci. Autoři naopak doporučují začít v prvním Sprintu s dodáním funkcionality, i když jen s malou hodnotou, protože bude možné rovnou také získat první zpětnou vazbu.

### **2.3.2 Sestavování týmu a jeho vývoj**

Jednou z kompetencí Scrum Mastera by měla být schopnost sestavovat skvělé týmy (Šochová, 2018a, s. 91). I v případě, že je Development tým obvykle v dané firmě již k dispozici, je vhodné se zamyslet nad tím, jak by měl být sestaven, a následně definovat jeho konkrétní složení (Haslinger, Wintersteiger, 2017). Jak autoři připomínají, dle teorie metody Scrum má být tento tým malý a multifunkční. Agilní procesy také úzce souvisí s faktorem důvěry a spoléhají se na kontextovou komunikaci se zákazníky a týmovou práci, proto úspěšná implementace agilního přístupu závisí na pozitivním a zdravém pracovním prostředí, ve kterém se členové týmu cítí bezpečně a sebevědomě (Paquette, Frankl, 2015, s. 94). Snahou je vytvořit multifunkční a self-organized tým, to ale trvá určitý čas a obvykle je potřeba i hodně úsilí (Sinha, 2019b), proto je nutné zahrnout do týmu pracovníky se všemi potřebnými znalostmi (Oshida, 2019). Mimo to agilní přístupy vyžadují dovednosti a osobnostní rysy členů týmu, jako je sebekázeň, sebeorganizace, sebevědomí a schopnost dobře spolupracovat s ostatními (Paquette, Frankl, 2015, s. 93). Tým má také mít přístup k nezbytnému školení, aby mohl získávat další kompetence (Pavlichenko, 2017). Oshida (2019) ještě upozorňuje na to, že je důležité nevybírat členy týmu na částečný pracovní úvazek. U pilotního projektu je také potřeba proškolit všechny zúčastněné strany o řádných agilních postupech (Marcias, 2015). Přestože někteří techničtí pracovníci mohou být dobře obeznámeni s agilním přístupem, je dle autora potřeba zajistit, aby vývojový tým měl homogenní porozumění ohledně procesů, které mají být použity. Haslinger a Wintersteiger (2017) doporučují agilní nástroj „Coaching Backlog“, který se skládá z malých pracovních položek, jako jsou školení, koučování nebo konzultace.

## Tuckmanův model vývoje týmů

Týmy obvykle procházejí čtyřmi fázemi vývoje: **Forming**, **Storming**, **Norming** a **Performing** (Tuckman, 1965 in Oshida, 2019). Jak uvádí Šochová a Kunce (2019, s. 155), přestože je Tuckmanův model poměrně starý a existují dnes novější teorie, je tento model považován za základ. Fázemi uvedenými v tomto modelu musí projít každý tým, aby se stal efektivním a dobře fungoval. Schéma 4 Tuckmanův model níže znázorňuje průchod mezi jednotlivými fázemi vývoje týmu.

Schéma 4 Tuckmanův model



Zdroj: Ferguson (2019), upraveno

Výše uvedené Schéma 4 Tuckmanův model nezobrazuje autor jako lineární křivku, ale jako stále opakující se cyklus, protože měnící se podmínky a okolnosti, jako je například přijímání nových členů týmu, mohou dovést tým k návratu do dřívější fáze (Hyde, 2017). Jak autor uvádí, i zkušené týmy mohou procházet těmito fázemi opakovaně, přičemž díky zkušenostem a znalostem je tento další průchod v jednotlivých fázích rychlejší. Dle Šochové a Kunce (2019, s. 155) je i výměna jednoho člena týmu změnou, na kterou si musí celý tým zvyknout, protože se mění osobní vazby, a díky jiným schopnostem i vlastnostem nového člena se mění také rovnováha týmu. Dle autorů musí na tuto změnu tým zareagovat a postupně ji absorbovat. Následující část popisuje čtyři fáze tohoto cyklu, jejich typické znaky a doporučený přístup Scrum Mastera dle toho, v jaké fázi se tým nachází.

### Fáze Forming

Pokud firma zahájí agilní transformaci a přechází na metodu Scrum, tak dle Tuckmanova modelu se tým nachází ve fázi Forming (Šochová, 2018a, s. 91). V agilním kontextu je Forming fáze klíčovou součástí přechodu od tradičních způsobů práce (Ferguson, 2019). Hlavním cílem této fáze je, aby se členové týmu navzájem poznali a věděli, jak kdo pracuje a jak jsou jednotliví členové týmu prospěšní (Šochová, Kunce, 2019, s. 155). Jak popisuje Šochová (2018, s. 91), v této fázi spolu lidé příliš nekomunikují, nespolupracují, stále převládají jejich staré zvyky, fungují jako specializovaní jednotlivci a obvykle si stěžují, že Scrum je nevyhovující. Také Hyde (2017) popisuje tým v této fázi spíše jako skupinu lidí, kteří pracují na úkolech nezávisle, vyhýbají se konfliktům a zaměřují na shromažďování informací a plánování. I když se v menších firmách jednotliví členové již znali, tak vzhledem k jinému stylu práce panuje v týmu nedůvěra, tým tak není příliš efektivní ani spokojený

(Šochová, Kunce, 2019, s. 155). Dle Ferguson (2019) bývají členové týmu v tomto období ale i nadšení, zvědaví, lehce nervózní, panuje mírný zmatek kvůli nejistotě, jakou práci dělají ostatní v týmu nebo jak správně rozdělit velkou část (Epic) na menší komponenty.

Hlavním úkolem Scrum Mastera ve Forming fázi je primárně vysvětlování základních principů metody Scrum a zahájení transformace nejen oficiálně, ale také v myslích jednotlivých členů týmu, protože je potřeba, aby změnili svůj styl práce a uvažování (Šochová, 2018a, s. 91). Obvyklou náplní jeho práce je nyní organizace meetingů, připomínání agilních principů a řešení problémů (Šochová, Kunce, 2019, s. 155). Řešení problémů by mělo probíhat tváří v tvář, nikoliv přes e-mail nebo chat, aby bylo možné získat lepší představu o tom, jak se k sobě lidé v týmu chovají (Ferguson, 2019). Scrum Master pracuje v této fázi na podpoře pozitivního chování týmu i tím, že vytváří správné fyzické prostředí pro spolupráci týmu, povzbuzuje tým a spolupracuje se zainteresovanými stranami (Hyde, 2017). Scrum Master by jako kouč měl členům týmu pokládat otázky i když si myslí, že zná jejich odpovědi (Ferguson, 2019).

### **Fáze Storming**

Protože metoda Scrum posiluje spolupráci, posouvá komunikaci v týmu za hranici komfortní zóny, tak fáze Storming přichází poměrně rychle (Šochová, 2018a, s. 91). Jak dále popisuje autorka, v této fázi dochází ke sporům v týmu i k negativnímu postoji vůči okolí. Členové týmu si začínají budovat vlastní názory a střety osobností i odlišné argumenty zvyšují napětí ve vztazích (Hyde, 2017). V týmu je mnoho zmatku i frustrace a vše se jeví mnohem složitější, než se očekávalo (Ferguson, 2019). Průchod fází Storming je dle Oshida (2019) nejdůležitější, protože pokud nedojde k vybudování důvěry a neobjeví se v týmu pocit bezpečí, tak tým se bude defenzivně normalizovat na základě kultury vyhýbání se, což následně omezí výkon týmu. I když jsou v tomto období rozhodnutí obtížná kvůli vysoké míře nejistoty (Hyde, 2017), klíčem k posunu v před je činit věci co nejjednoduššími (Ferguson, 2019). Pomoci může také vhodná kombinace efektivního koučování, školení a facilitace (Oshida, 2019).

Úkolem Scrum Mastera ve fázi Storming je povzbudit členy týmu ke komunikaci a pomoci jim dohodnout se na spolupráci (Šochová, 2018a, s. 91). Jak autorka vysvětluje, k tomu má pomoci facilitace, pomocí níž lze odstranit překážky a problémy v komunikaci a přejít do fáze Normingu. Proto by se Scrum Master měl snažit přimět tým, aby čelil svým konfliktům a hledal pro ně řešení (Hyde, 2017). Také v této fázi je role Scrum Mastera hodně určující, ale zároveň organizace týmu a část rozhodnutí již přechází na samotný tým (Šochová, Kunce, 2019, s. 156). Jak autoři dále uvádějí, Scrum Master se i nadále stará o agilní proces, ale určité úkoly již deleguje na členy týmu a tím umožňuje, aby se tým sám organizoval tak, aby byl co nejefektivnější. V této fázi by již tým jako celek měl chápat svůj účel a je tedy potřeba se ujistit, že i jednotliví členové rozumí dílčím milníkům, které stojí na cestě k cíli, a chápou svou úlohu i to, jak mohou pomoci týmu dosáhnout cíle (Ferguson, 2019). Jak autor vysvětluje, toto porozumění dodává týmu potřebný pocit důvěry.

### **Fáze Norming**

Ve fázi Norming tým začíná fungovat jednotně, přijímá společný cíl, je schopen společně plánovat, dohodnout se na spolupráci, přebírat odpovědnost a vytvářet vlastní procesy (Šochová, Kunce, 2019, s. 156). Také jednotlivé role jsou pochopeny a tým, že se jednotliví členové stávají odpovědnými, začínají si také navzájem rozumět (Hyde, 2017). Po fázi Stormingu přichází odpočinek od stresu (Šochová, 2018a, s. 92), tým nabývá na síle a začíná se objevovat pocit hrdosti (Ferguson, 2019). V tom, že tým má pocit, že nyní je vše již

v pořádku, tkví riziko ztráty motivace k dalším změnám (Šochová, 2018a, s. 92). Jak dále autorka upozorňuje, tento stav by nikdy neměl být cílem a je úkolem Scrum Mastera, aby vedl tým k dalšímu zlepšení. Výzvou pro tým v této fázi je zrychlit svou práci, ale zároveň udržovat vysokou kvalitu (Ferguson, 2019).

Ve fázi Norming je Scrum Master převážně v roli facilitátora, který je nápomocný s formováním společného cíle a definováním procesu (Šochová, Kunce, 2019, s. 156). V této fázi je nejlepším přístupem koučování, kdy Scrum Master povzbuzuje členy týmu k převzetí odpovědnosti a hledá cesty ke zlepšení (Šochová, 2018a, s. 92). Jak autorka vysvětluje, bez dobrého koučování může tým uvíznout v této fázi, protože je to pohodlné, tým vykazuje dobré výsledky a urgentní potřeba zlepšení se tak začíná vytrácet. Dle Ferguson (2019) je důležité jednak udržet dynamiku týmu na vysoké úrovni, ale i rozpoznat a zlepšit slabé stránky týmu. Scrum Master v této fázi také pomáhá s workshopy pro zákazníky, s organizací vnějších aktivit pro zvýšení socializace, zaměřuje se na inovace a pomáhá širší organizaci při podpoře týmu (Hyde, 2017).

### **Fáze Performing**

Hlavním cílem je dosažení fáze Performing, protože až zde se z týmu stává skutečný Scrum tým (Šochová, 2018a, s. 92). Jak autorka dále popisuje, tuto fázi lze rozpoznat dle toho, že tým je sebevědomý, otevřený, transparentní a stále hledá lepší způsoby své práce. Členové týmu si dostatečně důvěřují a mohou si tak dovolit být více kreativní i inovativní, a přitom dokončit včas kvalitní práci (Ferguson, 2019). Tým funguje bez zbytečných konfliktů, má společný cíl a vnímá svou odpovědnost za výsledek (Šochová, Kunce, 2019, s. 156). Jak autoři dále popisují, jednotliví členové jsou motivovaní, sdílejí své znalosti a zkušenosti s ostatními, podílejí se na rozhodování o organizaci práce a většinu problémů jsou schopni vyřešit bez potřeby další supervize. Tým se nyní může soustředit na dosažení společného cíle a má velkou šanci uspět (Hyde, 2017).

Cílem Scrum Mastera je nechat vzniknout tento self-organized tým a teprve nyní se jeho role stává výrazně méně určující, protože tým již činí většinu rozhodnutí sám (Šochová, Kunce, 2019, s. 156). Jak autoři popisují, kromě role facilitátora a kouče se Scrum Master dále stará o to, aby měl tým podmínky, za kterých může dobře fungovat. Dále také dohlíží, aby se tým z nějakého důvodu nevrátil zpět do některé předchozí fáze (Šochová, 2018a, s. 92). Scrum Master nyní podporuje vysokou míru autonomie týmu, jeho růst na úrovni jednotlivců i ve vztahu k cíli (Hyde, 2017) a napomáhá budování osobních vztahů mezi členy týmu (Ferguson, 2019). Jak uvádí Ferguson, v této fázi, kdy tým má již zavedené postupy, je vhodné sdílet své zkušenosti s ostatními týmy v organizaci. Šochová a Kunce (2019, s. 156) ještě upozorňují, že pokud má tým slabého Scrum Mastera, tak této fáze nikdy nedosáhne a zůstává v některé předchozí fázi svého vývoje.

Tým může těmito jednotlivými fázemi vývoje procházet po dobu několika dnů až týdnů v závislosti na složení týmu, na typu firemní kultury a prostředí (Šochová, Kunce, 2019, s. 156). Jak autoři dále vysvětlují, každá změna složení týmu vrací tým na začátek vývojového cyklu a také není garantováno, že tým dojde až do poslední fáze. Z důvodu potřeby identifikace těchto změn a jejich vlivu na tým je nutné, aby i skvěle fungující self-organized tým měl stále svého Scrum Mastera (Šochová, 2018a, s. 93). Jak autorka upozorňuje, bez Scrum Mastera se může stát, že se tým nevyrovná se změnami a vrátí se zpět do fáze Normingu nebo Stormingu.

### 2.3.3 Agilní smlouvy a formy spolupráce

Správa smluv mezi dodavatelem a zákazníkem je citlivou oblastí, která potřebuje pro dlouhodobý úspěch soulad (Dharmalingam, 2018). Tradiční smlouvy jsou považovány za nástroje, jejichž záměrem je, aby zainteresované strany uznaly závazky jako pevné, s jasně stanovenými pravidly ohledně ceny, rozsahu a doby trvání projektu (Naughter, 2018). To ale vyžaduje podrobný popis produktu již na samém počátku projektu (Dharmalingam, 2018). Jak autor dále popisuje, agilní metodologie je založena na iterativních dodávkách produktů a plnění požadavků zákazníka, a tak nelze definovat konečné výsledky, proto agilní smlouvy přijímají změny během celého životního cyklu projektu. Přestože agilní přístup nevyžaduje žádnou specifickou smlouvu nebo konkrétní formu spolupráce (Šochová, Kunce, 2019, s. 163), metodika agilních kontraktů řeší mnohé problémy, při kterých tradiční metodologie selhává (Dharmalingam, 2018). Jak Šochová a Kunce připomínají, dle Agilního manifestu tento přístup upřednostňuje spolupráci se zákazníkem před vyjednáváním o smlouvě. Následující text popisuje hlavní důvody, proč není vhodné používat klasické smlouvy při agilním způsobu vývoje, a dále vysvětluje základní vlastnosti agilních smluv.

#### Tradiční smlouvy

Relativně častou formou spolupráce je Fixed Time (Fixed Price) smlouva, protože zákazník chce zafixovat datum nasazení produktu a jeho cenu (Šochová, Kunce, 2019, s. 163). Pro klienta představuje smlouva s pevnou cenou minimalizaci rizika, že náklady překročí stanovený rozpočet (Lehmann-Benz, 2019). Jak ale autorka uvádí, tento typ smlouvy není vhodný pro agilní projekty, protože agilní vývoj se silně zaměřuje na potřeby klienta a je velmi obtížné dosáhnout dohody o pevné ceně, když ještě celková funkčnost nebyla definována. Takto nastavené projekty se také snaží o dodržení termínu, ceny, kvality i plné funkcionality, a to se v praxi daří pouze v 15 % případů (Šochová, Kunce, 2019, s. 163). Navíc doba trvání projektu a cena nesouvisí s výsledky, ale spíše se sdílenou vizí a očekáváním (Alfonso, 2018). Pokud dle smlouvy musí dodavatel také převzít všechna rizika, bývají předpokládané cílové náklady výrazně vyšší, navíc z obavy penalizace za překročení nákladů může dojít ke snížení kvalitativního cíle (Naughter, 2018).

#### Agilní smlouvy

V současném tržním prostředí, kdy se technologie a požadavky často mění, funguje agilní typ smlouvy lépe než tradiční smluvní model (Dharmalingam, 2018). Jak autor dále doplňuje, samotný agilní vývoj je svou povahou nepředvídatelný, proto agilní smlouvy poskytují flexibilní rozsah vývoje. I v agilní smlouvě existují cílové náklady, ale je zde ponechána potřebná flexibilita (Naughter, 2018). Vhodnou formou kontraktu při agilním vývoji je Effort-based spolupráce neboli agilní smlouva (Šochová, Kunce, 2019, s. 164). Jak autoři popisují, zákazník platí tým za každou iteraci (Sprint) a před dalším Sprintem se rozhoduje, zda chce ještě ve vývoji pokračovat, nebo zda je funkcionality dostatečné. Je tedy účtována paušální částka za zdroje, které po stanovenou dobu pracují na projektu (Alfonso, 2018). Autor doporučuje započítat práci pro klienta 35 hodin týdně za jednoho člena týmu a zbývající čas ponechat každému členovi na interní projekty a režii. Pro malé firmy je obvykle dostatečná smlouva se stručným popisem, jak se bude pracovat, co a zhruba kdy a za jakou částku má být hotovo (Šochová, Kunce, 2019, s. 165). Smlouva tak může obecně ustoupit do pozadí, což prospívá celkově vzájemnému vztahu a tým se může zaměřit na samotný produkt (Alfonso, 2018). Ačkoli agilní smlouvy nedefinují žádné přesně popsání výstupu, musí vždy obsahovat ustanovení o odpovědnosti dodavatele (Dharmalingam, 2018). Jak autor dále uvádí, smlouva

by měla obsahovat popis pracovních zásad projektu, včetně rolí a odpovědností, jak spravovat položky v Backlogu (priority, odhady), Definition of Done a záruku.

Změnou rozsahu projektu nebo produktu může dojít k navýšení rozpočtu oproti původním odhadům, proto žádný projekt není stanoven fixní cenou (Dharmalingam, 2018). Ale jak autor dále vysvětluje, pokud jsou správně stanoveny priority, tak i fixní rozpočet může být spravován prostřednictvím agilní smlouvy, protože vývoj se zaměřuje na položky s vysokou prioritou a v případě potřeby jsou položky s nízkou prioritou odebrány. Díky agilní smlouvě je tak méně režijních nákladů na dodatečné změny (Alfonso, 2018). Agilní smlouva může také obsahovat vzájemnou dohodu, že k předčasnému ukončení projektu může dojít pouze za určitých podmínek – například oznámením tři sprinty předem (Šochová, Kunce, 2019, s. 165). V tomto typu smlouvy by ale měly být zdůrazněny obchodní hodnoty a měl by být kladen důraz spíše na vizi a proces než na časovou osu projektu (Dharmalingam, 2018). Jak popisuje Šochová a Kunce (2019, s. 165) agilní smlouva je založena na rovnocenném partnerství, z toho důvodu musí být vyvážená a nesmí zvýhodňovat jednu ze stran, protože rovnocenní partneři mají sdílet svůj úspěch i případná rizika.

#### **2.3.4 Metriky a hodnocení**

Agilní metriky se liší od běžně známých ukazatelů používaných v klasických metodách, jako je Waterfall (Paquette, Frankl, 2015, s. 95–96). Dle autorů jsou agilní metriky pro měření výkonu významnými faktory pro posouzení efektivnosti týmu v průběhu času a zlepšují komunikaci a transparentnost. K předpovědi progresu jsou používány různé zobrazovací metody, jako je Burn-down graf, Burn-up graf nebo kumulativní vývojový diagram (Schwaber, Sutherland, 2013). Dle autorů jsou to užitečné nástroje, ale v komplikovaném prostředí nenahrazují empirický přístup (Schwaber, Sutherland, 2013). Dle Paquette a Frankl (2015, s. 96) jsou tyto automatizované metriky, které zaznamenávají data a popisují kritické události, velmi užitečné, a pokud jsou vedlejším produktem softwarových nástrojů, tak nevyžadují žádné další úsilí a data z nich lze sbírat konzistentně. V každém okamžiku tak lze zjistit objem zbývajících práce k dosažení cíle, tyto informace jsou transparentní pro všechny zainteresované strany (Schwaber, Sutherland, 2013). Jak autoři uvádějí tato informace má být sledována Product Ownerem minimálně po každém Sprint Review, aby mohl být vyhodnocen pokrok. Tým tak může používat tyto údaje k řízení a předpovídání pravděpodobnosti, že dosáhne cíle Sprintu (Carroll, Morris, 2017, s. 58).

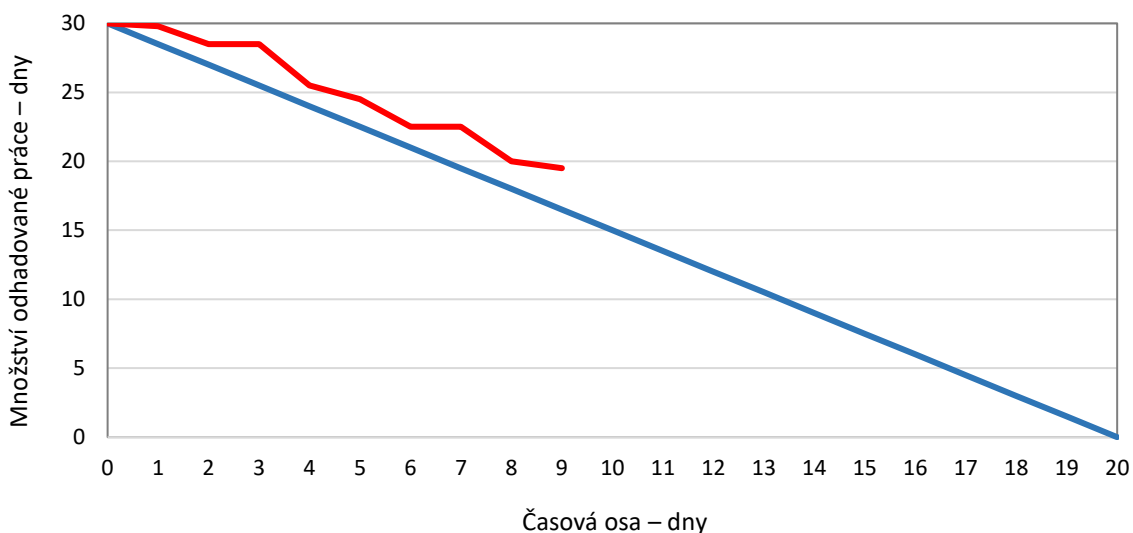
Šochová a Kunce (2019, s. 201) upozorňují, že je vhodné vyvarovat se měření pomocí tvrdých metrik, jako je například počet řádků kódu nebo počet chyb, protože lidé se pak soustředí na pokyny metriky a opomíjejí skutečný cíl, což přímo narušuje agilní přístup. Dle autorů obecně není vhodné používat individuální metriky, proto se nikdy neměří rychlost za jednotlivé členy, ale za celý tým a Sprint. Nežli se zaměřovat na metriky samotné, je dle Paquette a Frankl (2015, s. 96) doporučeným přístupem identifikovat užitečné metriky, které poskytují skutečnou hodnotu a zaměřují se na důležité prvky a na řízení týmu. Z grafů a výstupů měření musí být týmu okamžitě jasné, jak se jim daří, ale také pro informované návštěvníky by měly být tyto výstupy snadno čitelné a srozumitelné (Carroll, Morris, 2017, s. 59). I když, jak autoři doplňují, tyto grafy jsou pouze orientační a pro vyřešení problémů je potřeba dalšího šetření. Proto také tvůrci musí dobře rozumět těmto metrikám a ukazatelům (Paquette, Frankl, 2015, s. 96).

## Monitorování pokroku

V tradičně řízených projektech vypracuje projektový manažer podrobný plán projektu, obvykle pomocí Ganttova diagramu, ale v agilním projektu většinou probíhá mnoho aktivit paralelně, a proto jsou vyžadovány různé přístupy ke sledování pokroku (Carroll, Morris, 2017, s. 32). Mezi tyto metriky patří nejčastěji Burn-down graf a měření rychlosti týmu (Velocity).

**Burn-down graf** je grafické znázornění množství práce, kterou zbývá realizovat ve stanoveném čase (Carroll, Morris, 2017, s. 58). Členové týmu aktualizují svůj pokrok přesunutím příslušných úkolů na Scrum tabuli a toto se dále promítá také v Burn-Down grafu (Morris, 2017, s. 108). Jak autor dále popisuje, Burn-Down graf je obvykle aktualizován ručně jako součást fyzické Scrum tabule, nebo automaticky, pokud tým používá softwarový nástroj pro správu agilního vývoje. Práce ze Sprint Backlogu je obvykle znázorněna na svislé ose a čas na ose vodorovné. Níže je zobrazen typický Burn-down graf.

Graf 1 Burn-down graf



Zdroj: Carroll, Morris (2017, s. 58), upraveno.

Výše uvedený **Graf 1 Burn-down graf** znázorňuje na vodorovné ose 20 pracovních dní a na svislé ose množství naplánované a odhadované práce. Modrá křivka naznačuje plánovanou realizaci položek z Backlogu a červená křivka ukazuje skutečnost, jak jsou položky postupně zpracovávány. Dle Carroll a Morris (2017, s. 58) může tento uvedený příklad naznačovat to, že práce byla zpočátku podceňována, nebo fakt, že se týmu nepodařilo dostat na potřebnou rychlost. Jak autoři uvádějí, tyto informace může tým použít při rozhodování, zda se podaří dosáhnout cíle Sprintu nebo zda je nutné odstranit některé položky ze Sprint Backlogu. Modrá přímka tedy představuje ideální trend, kdy tým realizuje každý den shodné průměrné množství práce, takže by vždy veškerou svou práci měl dokončit do konce Sprintu (Morris, 2017, s. 108–109). Jak autor dále vysvětluje, když tým postupuje dobře, tak se jejich křivka pohybuje na nebo pod linií trendu, ale pokud se jejich křivka dostává nad tuto linii příliš dlouho, znamená to, že pravděpodobně nestihnou vše dokončit včas. Tento typ grafu existuje ve dvou variantách – Sprint Burn-down graf pro práci v jednotlivých iteracích a Product Burn-down graf pro znázornění zbývajících práce na celém projektu (Blackburn, 2019). Jak autorka dále popisuje, mezi výhody tohoto grafu patří jeho jednoduchost a možnost zobrazení aktuálních i historických dat. Dle Blackburn má však i jistá



omezení, např. neodhalí všechny problémy, neukáže změny v rozsahu práce a je závislý na kvalitě původních odhadů času pro práci.

**Rychlost (Velocity)** je jedním z indikátorů „zdravosti“ agilního týmu (Šochová, Kunce, 2019, s. 201). Je to měřítko skutečné kapacity pro realizaci funkcionalit Development týmem v daném časovém rámci (Carroll, Morris, 2017, s. 58). Jak autoři dále vysvětlují, jednotky jsou stejné jako u odhadů, ale na začátku projektu se předpokládaná rychlost pouze odhaduje a poté se na konci každého Sprintu měří skutečná rychlost. Stanovená rychlost není žádné dogma a to, že ji tým neplní, má svůj důvod – opravuje chyby, pracuje na technickém dluhu, učí se nebo má větší režii, než se předpokládalo (Šochová, Kunce, 2019, s. 201). Jak autoři vysvětlují, tato rychlost ukazuje realitu, proto by se z ní neměla stát tvrdá metrika, protože tým se přizpůsobí například tím, že bude dělat vyšší odhady následující práce, a tento přístup pak ztratí svůj smysl. Již po několika sprintech by mělo být zřejmé, zda má vývojový tým dostatečnou kapacitu pro poskytování funkcí a dosažení cíle (Carroll, Morris, 2017, s. 59). Proto při aktualizování úkolů s odhadem zbývajících hodin by tým neměl od původního odhadu jen odečíst hodiny, které doposud odpracoval, ale měl by přehodnotit práci na základě toho, o kolik jednodušší nebo složitější se daná práce stala (Morris, 2017, s. 107). Jak autor dále upozorňuje, někdy může být potřeba méně času, než se původně předpokládalo, ale bývá pravděpodobnější, že práce bude trvat déle. V tom případě Morris upozorňuje, že je potřeba mít tuto informaci co nejdříve, protože může být v ohrožení cíl Sprintu.

### **Hodnocení členů týmu**

Agilní organizace zcela upouští od „top-down“ hodnocení zaměstnanců a posouvají se k hodnocení zaměřenému na rozvoj (Šochová, 2018b). Při agilním přístupu je potřeba pomocí metrik podpořit týmovost, schopnost se učit nové věci a přebírat odpovědnost za výsledek (Šochová, Kunce, 2019, s. 201). Jak autoři upozorňují, tyto atributy nejsou měřitelné na denní bázi, ale je viditelný až jejich konečný dopad. Jak upozorňuje Myslín (2016, s. 117), hodnocení jednotlivců není smyslem retrospektivy, i když k tomu někdy dochází, ale jedna z možností, jak hodnotit jednotlivé členy týmu, je coachingová stupnice 1 až 10, pomocí které Scrum Master hodnotí týmovost, nasazení a aktivitu (Šochová, Kunce, 2019, s. 201). Jak autoři vysvětlují, je vhodné nejprve nechat člena týmu, aby ohodnotil sám sebe, pak mu sdělit hodnocení Scrum Mastera a společně se pak zaměřit na způsob, jak se posunout déle. Nutnou podmínkou tohoto sebehodnocení je pochopit, že se nejedná o hodnocení v pravém slova smyslu, ale o nástroj na hledání potenciálu a správného směru rozvoje (Šochová, 2018b). Jak autorka dále vysvětluje, neočekává se, že hodnoty v čase porostou, ale naopak, neboť si lidé uvědomí svoje vlastní fungování v širším kontextu až poté, co jsou v dané roli delší čas. Dalším hodnotitelem může být Product Owner (také stupnice 1 až 10), ale zde je hodnocen tým jako celek (Šochová, Kunce, 2019, s. 201).

Dle Šochové (2018) je základem dobrého použití těchto metod hodnocení opět coaching, naopak je nelze používat direktivně, protože pak nepřinášejí žádný užitek a mají spíše negativní dopady. Další variantou hodnocení je nechat členy týmu, aby se ohodnotili sami navzájem, pomocí rozdělení dané fiktivní částky peněz mezi jednotlivé členy týmu (Šochová, Kunce, 2019, s. 201). Jak autoři popisují, každý člen týmu dostane otázku, jak by rozdělil danou částku mezi členy týmu kromě sebe, což umožňuje zjistit, jak se jednotliví lidé vnímají v rámci týmu.

## Shrnutí kapitoly

Cílem agilní transformace je změnit způsob, jakým firma nebo tým v současnosti pracuje (Zack, 2018). Úspěch pilotního projektu je klíčový, protože slouží jako základ pro představení agilní metody ostatním částem organizace, vytváří důvěru obchodních partnerů a může pomoci získat důvěryhodnost u lidí, kteří nejsou agilnímu přístupu nakloněni (Marcias, 2015). Úspěšná agilní transformace má přinést mnohem lepší a příjemnější procesy pro všechny zúčastněné, ale cesta ani zdaleka není jednoduchá a může trvat až několik let, v závislosti na velikosti organizace (Haslinger, Wintersteiger, 2017). Také je ale potřeba počítat s tím, že po přechodu na agilní přístup se stávají všechny dosavadní problémy okamžitě viditelné a často se tyto problémy jeví i větší než dříve (Šochová, Kunce, 2019, s. 140).

Tato kapitola se věnovala implementaci agilní metody Scrum pomocí pilotního projektu a popsala doplňující informace o sestavování týmu (základní informace o rolích a týmu jsou uvedeny v kapitole **Role ve Scrumu**) a pomocí Tuckmanova modelu vysvětlila vývojové fáze týmu a doporučení pro Scrum Mastera v jednotlivých fázích. Dále byly zmíněny základní informace o agilních smlouvách, které by měly být používány při tomto typu vývoje, protože přinášejí výhody pro všechny zúčastněné strany. Poslední část této kapitoly popsala nejčastěji využívané agilní metriky pro měření úspěšnosti práce na produktu a hodnocení členů týmu.

## 2.4 Metodika práce

V teoretické části byla provedena rešerše sekundárních zdrojů, jako jsou odborné tištěné nebo elektronické monografie a články převážně z oboru agilního a projektového managementu. Elektronické zdroje byly čerpány z knihovny Akademie věd ČR, z databáze ProQuest a z dalších internetových zdrojů. Teoretická část práce popsala nejběžnější klasické metody řízení projektů, kterými jsou Vodopádová a Spirálová metoda. Práce se dále podrobněji zabývala agilními metodami Scrum a Kanban. Další část rešerše se věnovala implementaci metody Scrum na pilotním projektu a oblastem, které souvisejí s touto implementací – sestavování týmu, agilní smlouva, metriky a hodnocení při agilním přístupu. V praktické části byla provedena metoda kvalitativního výzkumu prostřednictvím strukturovaných rozhovorů ve dvou fázích. Rozhovorů se účastnil Scrum Master, Product Owner a pět vývojářů. Tito respondenti také hodnotili průběh vývoje týmu dle Tuckmanova modelu, vývojáři provedli sebehodnocení i vzájemné hodnocení, Scrum Master hodnotil jednotlivé členy týmu a Product Owner hodnotil Development tým jako celek. Výstupy z pilotního projektu, které byly zaznamenávány prostřednictvím softwarového nástroje, byly podrobeny obsahové analýze. Další obsahová analýza dokumentu byla provedena také u smlouvy firmy XY s klientem.

## Výzkum v rámci seminární práce

Ve firmě XY proběhl výzkum v rámci seminární práce, který se zabýval motivací k přechodu na agilní přístup a očekáváními plynoucími z této změny (Buschová, 2019). V tomto výzkumu se autorka zaměřila na segmenty: kvalita, efektivita, spokojenost týmu, reakce na změnu, předvídatelnost, spolupráce, komunikace a spokojenost zákazníka. Respondenti hodnotili současný stav v dané firmě a své očekávání po jednom roce od implementace agilního přístupu. Ze získaných dat bylo zjištěno, že nejproblematictějšími oblastmi jsou: kvalita, reakce na změnu a předvídatelnost. Z výzkumu také vyplynulo, že v těchto třech segmentech mají respondenti nejvyšší očekávání pozitivní změny, kterou by jim měl přinést přechod na agilní přístup. Na základě těchto výsledků byly tyto tři segmenty – kvalita, reakce na změnu a předvídatelnost – stanoveny hlavními důvody pro změnu řízení projektů v dané firmě. Na základě informací a dat získaných z výzkumu bylo v závěru seminární práce

(Buschová, 2019) doporučeno, aby firma XY zahájila přechod na agilní řízení projektů. Také bylo doporučeno implementovat konkrétně metodu Scrum a pro zefektivnění procesu následně doplnit principy metody Kanban. Dále byl vybrán pilotní projekt, na kterém by se měla transformace zahájit. Vzhledem k tomu, že firma XY chce realizovat agilní transformaci v rámci vnitřních zdrojů, byl jako Scrum Master doporučen současný projektový manažer. Dále bylo navrženo sestavit 5 až 7členný tým vývojářů, kteří s touto metodou mají zkušenost.

### **Předpoklady a zaměření výzkumu**

Porozumět základním principům a postupům metody Scrum není příliš náročné, ale zavedení této metody do praxe bývá doprovázeno mnoha komplikacemi a problémy, se kterými se musí vyrovnat jednotliví členové i tým jako celek (Schwaber, Sutherland, 2013) a (Šochová, Kunce, 2019). Lze předpokládat, že i tento zkoumaný tým se může potýkat s níže uvedenými problémy nebo chybami v těchto oblastech:

- **příprava na zahájení agilní transformace a sestavování týmu:** nedostatečná příprava Scrum Mastera, Product Ownera nebo členů Development týmu, nevhodně sestavený tým z pohledu potřebných zkušeností, znalostí nebo přístupu k agilnímu způsobu vývoje;
- **role a vývoj týmu:** problémy s adaptací na nové role, konflikty ve formujícím se týmu;
- **meetingy a aktivity:** časově náročné schůzky, komplikovaný průběh schůzek, neochota vývojářů účastnit se schůzek, vynechávání některých typů aktivit;
- **praktiky a artefakty:** není dosahováno cílů Sprintu, nedostatečně popsané nebo nesrozumitelné User Stories, není jasně určena Definition of Done, nevhodné rozložení Scrum tabule, velká rozpracovanost úkolů;
- **agilní programování:** s agilním přístupem nejsou zavedeny praktiky agilního vývoje;
- **agilní smlouva:** typ kontraktu neodpovídá agilnímu způsobu spolupráce.

Praktická část této práce se na výše uvedené oblasti zaměřuje a vyhodnocuje, jak se v dané oblasti týmu podařilo realizovat doporučené postupy a plnit stanovené cíle, a kde jsou naopak rezervy pro zlepšení. Vyhodnocení probíhalo komparací výsledků výzkumu s informacemi a doporučeními, které jsou pro dané oblasti popsány v teoretické části této diplomové práce v kapitolách **Metoda Scrum, Metoda Kanban a Implementace agilní metody Scrum**.

### **Strukturované rozhovory s otevřenými otázkami**

Informace byly získány metodou strukturovaných rozhovorů s otevřenými otázkami (Hendl, 2016, s. 177). Otázky pro rozhovory jsou uvedeny v Příloze 1 a v Příloze 2. Odpovědi byly zaznamenány metodou doslovné transkripce (Hendl, 2016, s. 212). Text byl pouze očištěn od dialektu a chyb. Rozhovorů se účastnili jednotlivě Scrum Master, Product Owner a 5 vývojářů. Rozhovory byly rozděleny na dvě fáze – fáze 1 proběhla po dokončení prvního Sprintu a fáze 2 proběhla dva měsíce od zahájení Scrumu na daném projektu. Všichni respondenti byli předem informováni o účelech rozhovorů a byly individuálně domluveny čas a místo pro jednotlivé rozhovory. Ve fázi 1 byly rozhovory zaměřeny převážně na zahájení Scrumu, připravenost jednotlivých členů Scrum týmu a na průběh prvního Sprintu. Nejdříve proběhl rozhovor se Scrum Masterem, a to v zasedací místnosti ve firmě XY. Rozhovor trval přibližně 45 minut. Následně proběhl rozhovor s Product Ownerem v sídle společnosti klienta. Rozhovor trval 50 minut. Druhý den proběhly postupně rozhovory s pěti členy Development týmu. Jednotlivé rozhovory trvaly přibližně 60 minut a byly realizovány v zasedací místnosti firmy XY. Záznamy rozhovorů fáze 1 jsou v Příloze 3. Rozhovory ve fázi 2 byly retrospektivního charakteru – otázky byly zaměřené na hodnocení stavu po

přechodu na agilní přístup – co se zlepšilo, a naopak co nedaří dle očekávání. Opět jako první proběhl rozhovor se Scrum Masterem. Rozhovor trval přibližně 30 minut. Následně proběhl rozhovor s Product Ownerem, který trval rovněž 30 minut. Poté byly realizovány cca 40minutové rozhovory se členy týmu. Všechny rozhovory ve fázi 2 proběhly v zasedací místnosti firmy XY. Záznamy rozhovorů jsou v Příloze 4. Vyhodnocení dat z rozhovorů probíhalo metodou srovnávání a sledování rozdílů v odpovědích u všech respondentů (Hendl, 2016, s. 237). Výklad relevantních výsledků byl následně slovně interpretován a byly shrnuty vlastní závěry (Reichel, 2009, s. 169). Tato interpretace je v praktické části rozdělena do dvou podkapitol dle fází rozhovorů. V první fázi byl hodnocen stav na začátku implementace agilní metody. V druhé fázi vývoj a změny po dvouměsíční agilní spolupráci.

## **Hodnocení a sebehodnocení**

Během pilotního projektu bylo realizováno vzájemné hodnocení a sebehodnocení členů Development týmu. Jednotliví členové byli hodnoceni Scrum Masterem a tým jako celek byl hodnocen Product Ownerem. Všechna hodnocení i sebehodnocení byla realizována po ukončení rozhovorů, a to na začátku pilotního projektu (po prvním Sprintu) i po dvouměsíční agilní spolupráci, aby bylo možné provést srovnání a vyhodnotit dosavadní vývoj.

**Vzájemné hodnocení členů Development týmu** bylo realizováno pomocí rozdělení fiktivní částky peněz (Šochová a Kunce, 2019, s. 201). Každý člen týmu měl za úkol rozdělit fiktivní odměnu 20 000 Kč mezi své čtyři kolegy vývojáře dle vlastního uvážení. Vývojáři nehodnotili Scrum Mastera ani Product Ownera. Výsledky tohoto hodnocení od jednotlivých vývojářů byly zaznamenány do tabulek v Příloze 5. Druhou metodou hodnocení bylo **sebehodnocení členů Development týmu** (Šochová a Kunce, 2019, s. 201). K hodnocení byla použita stupnice 1 až 10 a hodnoceny byly tři oblasti – týmovost, nasazení, aktivita. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Výsledky sebehodnocení jsou uvedeny v Tabulka 2 Sebehodnocení členů týmu.

Scrum Master provedl v rámci rozhovorů **hodnocení členů Development týmu** (Šochová a Kunce, 2019, s. 201). Toto hodnocení se zaměřilo na stejné oblasti jako sebehodnocení vývojářů (týmovost, nasazení, aktivita) a byla použita stejná stupnice 1 až 10. Výsledky hodnocení Scrum Mastera jsou Tabulka 4 Hodnocení členů týmu Scrum Masterem. **Product Owner provedl hodnocení Development týmu jako celku** – nehodnotil jednotlivé vývojáře jako Scrum Master (Šochová a Kunce, 2019, s. 201). Hodnoceny byly ale stejné oblasti (týmovost, nasazení, aktivita) a opět byla použita hodnotící stupnice 1 až 10. Product Owner hodnocení také provedl v rámci obou rozhovorů a získaná data jsou v Tabulka 5 Hodnocení týmu Product Ownerem.

Data všech hodnocení a sebehodnocení respondentů byla pro účely prezentace a formulace závěrů převedena do výše uvedených tabulek (Hendl, 2016, s. 219–220). Vyhodnocení výsledků na stupnici 1 až 10 bylo zaměřeno na změny číselných hodnot v první a druhé fázi, které mohly zachytit určitý vývoj v dané oblasti nebo u konkrétního respondenta. Snížení bodového hodnocení v tomto případě nemusí být vždy vnímáno jako negativní výsledek nebo chyba, ale naopak jako možný prostor pro zdokonalení díky lepší sebereflexi respondenta a uvědomění si širšího kontextu (Šochová, 2018b). Pomocí rozdělení fiktivních odměn v Development týmu bylo vyhodnoceno, jak se vzájemně vnímají jednotliví členové týmu, koho si nejvíce váží (nejvyšší odměny), a u kterých členů přínos naopak nevidí (nejnižší odměny). Výsledky hodnocení a sebehodnocení byly slovně interpretovány a shrnuty do vlastních závěrů. Interpretace výsledků hodnocení probíhala vždy v kontextu se získanými informacemi z obou fází rozhovorů.

## **Hodnocení vývoje týmu dle Tuckmanova modelu**

Během výzkumu byli členové Scrum týmu požádáni o vyhodnocení, v jaké fázi se nachází jejich tým z hlediska Tuckmanova modelu (Tuckman, 1965 in Oshida, 2019). Aby členové Scrum týmu byli schopni tímto způsobem identifikovat fázi vývoje svého týmu, byl jim vysvětlen princip Tuckmanova modelu a jeho jednotlivé fáze tak, jak je popisuje teoretická část této práce v kapitole **Sestavování týmu a jeho vývoj**. První hodnocení provedli členové týmu týden po spuštění pilotního projektu. Každý člen Scrum týmu přiřadil na vytištěné schéma Tuckmanova modelu bod (značku) k jedné vývojové fázi, ve které se, dle jeho názoru, tým aktuálně nachází (Ferguson, 2019). Body u jednotlivých fází byly zaznamenány do Tabulka 6 Vyhodnocení vývoje týmu dle Tuckmanova modelu. Druhé hodnocení provedl tým stejným způsobem po dvouměsíční spolupráci na pilotním projektu. Data z druhého hodnocení byla přidána do výše uvedené tabulky. Následně byla provedena komparace dat z obou hodnotících fází a dle počtu přiřazených bodů bylo vyhodnoceno, v jaké fázi dle Tuckmanova modelu se tým nacházel na začátku pilotního projektu a k jaké změně došlo po dvou měsících spolupráce.

## **Obsahová analýza interních dokumentů – agilní smlouva**

Pomocí obsahové analýzy (Reichel, 2009, s. 126–129) interních dokumentů firmy XY byl vyhodnocen současný typ obchodního kontraktu mezi firmou XY a jejím klientem. Obsahová analýza daného dokumentu byla zaměřena na části smlouvy, které se týkají definice rozsahu díla, termínů plnění a rozvržení plateb. Cílem obsahové analýzy nebylo zjistit, zda uvedené ceny a termíny odpovídají náročnosti projektu. Obsahová analýza se zabývala pouze tím, zda současná smlouva odpovídá nebo neodpovídá agilním smlouvám, které jsou doporučovány pro agilní formu spolupráce (Šochová, Kunce, 2019, s. 164–165) (Dharmalingam, 2018). Obsahová analýza agilní smlouvy je popsána v kapitole 3.2.5 Agilní smlouva.

## **Obsahová analýza elektronických dokumentů – Scrum tabule, Burn-down graf**

Průběh pilotního projektu byl zaznamenáván prostřednictvím softwarového nástroje JIRA. Tento nástroj umožňuje zobrazení elektronické Scrum tabule a automatické generování Burn-down grafů. Obsahová analýza (Reichel, 2009, s. 126–129) Scrum tabule byla zaměřena na to, zda je tato tabule vhodně rozdělena na jednotlivé části (Roseke, 2019), a na vyhodnocení rozpracovanosti jednotlivých pracovníků během sprintu dle doporučení vycházejících z metody Kanban (Šochová, Kunce, 2019, s. 151), (Brown, 2019), (Paquette, Frankl, 2015, s. 60). U Burn-down grafů bylo vyhodnocováno, zda se týmu podařilo dosáhnout cílů jednotlivých sprintů (Carroll, Morris, 2017, s. 58), (Morris, 2017, s. 108–109). Obsahová analýza těchto dokumentů je popsána v praktické části práce v kapitole Scrum tabule, Burn-down graf. Interpretace výsledků obsahové analýzy byla porovnána a doplněna informacemi získanými z obou fází rozhovorů.

## **Doporučení**

Po vyhodnocení dat získaných během výzkumu byla následně formulována doporučení pro odstranění chyb a nedostatků, které byly při výzkumu zjištěny. Uvedená doporučení se opírají o teoretickou část této diplomové práce, konkrétně kapitoly: **Metoda Scrum, Metoda Kanban a Implementace agilní metody Scrum**.

### 3 Analytická část práce

V praktické části této práce je představena firma, ve které byl výzkum realizován. Z důvodu zachování anonymity není v této práci uveden skutečný název firmy, název pilotního projektu ani popisky jednotlivých úkolů na Scrum tabulích v přílohách. Data pro výzkum byla získána prostřednictvím rozhovorů se Scrum Masterem, Product Ownerem a členy Development týmu. Další data byla získána pomocí hodnocení, které provedli výše uvedení respondenti v rámci rozhovorů. Data a informace o průběhu implementace byla získána prostřednictvím softwarového nástroje, který byl využíván pro monitoring pilotního projektu. V další části je provedeno vyhodnocení získaných dat a na základě výsledků jsou uvedena doporučení.

#### 3.1 Představení firmy XY

Jak je uvedeno na webových stránkách (XY, 2019), firma XY spol. s r.o. byla založena v roce 1998 se sídlem v Praze a působí na poli vývoje IT systémů, mobilních aplikací, webových stránek a podnikových portálů. Od svého vzniku již realizovala přes 400 projektů v oblasti informačních technologií. Mezi klienty společnosti XY patří jak velké mezinárodní společnosti, tak domácí i zahraniční malé a střední firmy. Společnost XY dlouhodobě spolupracuje také s výzkumnou a vývojovou sférou, a to zejména při poskytování informačních technologií pro mezinárodní vědeckou spolupráci. Firma XY se také podílí na mnoha nekomerčních aktivitách a sponzoruje webové stránky pro školy a dětské domovy. Dle poskytnutých údajů pracuje ve firmě XY (2020) celkem 17 zaměstnanců – dvanáct programátorů, tři projektoví manažeři, jeden obchodní manažer a office manažerka. Pro hostingové služby, grafické návrhy, účetnictví a další oblasti nemá firma XY vlastní zaměstnance, ale využívá tyto služby formou outsourcingu – svěřuje tyto služby jiným společnostem, které jsou na tuto činnost specializovány.

Dle interních dokumentů firmy XY (2020) pracují aktuálně vývojáři na čtyřech hlavních projektech většího rozsahu, několik dalších dokončených projektů je v režimu podpory a průběžně se vyskytují i malé projekty s dobou trvání v jednotkách týdnů. Dle interních dokumentů firmy XY (2020) je každý projekt řízen projektovým manažerem, který zároveň koordinuje spolupráci s externími dodavateli a klienty. Všichni tři současní projektoví manažeři pracují na více projektech zároveň a ke každému projektu má manažer k dispozici obvykle 2 až 6 vývojářů. Většina vývojářů pracuje také na více projektech, pouze několik málo vývojářů je alokováno na jeden konkrétní projekt. V rámci seminární práce (Buschová, 2019) byl doporučen konkrétní projekt, který by měl být využit jako pilotní projekt pro zahájení agilní transformace. Spuštění projektu bylo původně plánováno na říjen 2019, ale kvůli strategickým změnám na straně klienta firmy XY byl start projektu odložen na leden 2020. Firma XY zvažovala zahájení agilní transformace v plánovaném termínu s jinými projekty, ale nakonec se rozhodla, že počká až na start zvoleného projektu. Tento odklad tak umožnil Scrum Masterovi, Product Ownerovi i vývojářům mít více času na přípravu pro plánovanou změnu.

#### 3.2 Vyhodnocení získaných dat

Rozhovory byly realizovány se Scrum Masterem, Product Ownerem a se všemi pěti vývojáři samostatně a proběhly ve dvou fázích. První fáze rozhovorů proběhla po ukončení prvního Sprintu a druhá fáze rozhovorů proběhla dva měsíce od zahájení Scrumu. Součástí obou fází rozhovorů byla také vzájemná hodnocení a sebehodnocení členů týmu, hodnocení jednotlivých členů týmu Scrum Masterem a hodnocení Development týmu jako celku Product Ownerem. Vyhodnocení rozhovorů a výsledky jednotlivých metod hodnocení jsou popsány

v následujících podkapitolách. Další část vyhodnocení obsahuje výstupy ze sledování a měření pokroku pomocí softwarových nástrojů.

### **3.2.1 Vyhodnocení rozhovorů – fáze 1**

Tato podkapitola se zabývá vyhodnocením první fáze rozhovorů, které proběhly po dokončení prvního Sprintu. Vyhodnocení je rozděleno na několik oblastí: příprava na zahájení agilní transformace, role a sestavování týmu, meetingy a aktivity, praktiky a artefakty, agilní programování a agilní smlouva. Otázky pro tyto rozhovory jsou v Příloze 1 a přepisy samotných rozhovorů jsou v Příloze 3.

#### **Příprava na zahájení agilní transformace**

Z rozhovorů vyplynulo, že Scrum Master i Product Owner se na zahájení agilní transformace a převzetí těchto (pro ně) nových rolí připravovali předem. Scrum Master jako zdroj informací využíval odbornou literaturu a webové portály. Své vzdělávání zaměřoval převážně na oblasti týkající se náplně role Scrum Mastera a na vhodné přístupy k týmu. Dále také navštívil dvoudenní kurz, jehož obsahem byla implementace agilních metod. Scrum Master uvedl, že má zájem se i nadále v této oblasti vzdělávat a pro získání dalších znalostí by se chtěl účastnit certifikačního kurzu, který je určen přímo Scrum Masterům. Product Owner během své přípravy na agilní řízení využíval znalostí Scrum Mastera, který mu pomáhal se v dané problematice lépe zorientovat. Dále se pak také sám zabýval studiem jednotlivých ceremonií, které jsou nedílnou součástí metody Scrum. Product Owner se ve své přípravě zaměřil převážně na získání znalostí o přípravě User Stories a o správě Backlogu. Sám hodnotil své znalosti jako základní a je si vědom potřeby dalšího rozvoje a učení. Znalosti a zkušenosti samotných vývojářů – členů Development týmu – se výrazně liší. Tři vývojáři z pěti uvedli, že nemají vlastní praktickou zkušenost s touto metodou. Metodu sice teoreticky znají, mají obecné informace, ale ve většině případů spoléhají na vedení od Scrum Mastera a zkušenějších kolegů. Dále také potvrdili, že žádnou přípravu v souvislosti s transformací na agilní přístup neměli. Další dva vývojáři se zmínili, že mají s metodou Scrum praktické zkušenosti z dřívějšího zaměstnání. Oba v rozhovoru uvedli, že necítí potřebu se nyní na tuto změnu připravovat. Vývojář 1 v rozhovoru doplnil, že jeho praktické zkušenosti jsou využívány na konzultacích se Scrum Masterem.

#### **Role a sestavování týmu**

Scrum Master v rozhovoru definoval svou roli jako roli ochránce, kouče a lídra, který vede celý tým, vytváří pro něj vhodné podmínky, motivuje vývojáře, odstraňuje překážky a řeší spory v týmu. Dle Scrum Mastera lze tuto roli také definovat jako „průvodce Scrumu“, který předává své znalosti o této metodě dále na členy týmu nebo další zainteresované osoby. Dále také zmínil, že vnímá úzký vztah této role směrem k Product Ownerovi, kdy svou roli vidí jako „pravou ruku“ Product Ownera. Jak dále uvedl, Scrum Master by měl umět všechny tyto role a přístupy využívat dle aktuální potřeby a zároveň by nikdy neměl situace řešit jako nadřízený nebo dávat členům týmu příkazy. Za svou odpovědnost označil Scrum Master také zajištění a organizaci všech aktivit a meetingů, dohled nad jejich časovým harmonogramem i věcným obsahem. Scrum Master je s týmem v kontaktu během celého pracovního dne, sdílají jeden společný prostor (Open Space) a je tak týmu stále k dispozici.

Product Owner popsal svou roli ve Scrumu jako roli odpovědnou za nastavení priorit a určení toho, co má být realizováno Development týmem. Také dále uvedl, že člověk v této roli musí

danému produktu rozumět, musí vědět, co od nového produktu očekávají zákazníci, a musí být tyto informace schopen předat vývojářům i dalším zainteresovaným stranám. Jako své povinnosti Product Owner uvedl přípravu Backlogu a User Stories, součinnost s Development týmem a účast na Review a Planningu. Product Owner se během prvního Sprintu účastnil osobně Planningu i Review a stále je dostupný online nebo telefonicky. Dále také uvedl, že v případě potřeby je ochoten za týmem přijet v průběhu Sprintu.

Jak v rozhovoru uvedl Scrum Master, členové Development týmu byli vybráni pro pilotní projekt dle jejich odborných znalostí a zaměření tak, aby byl tým schopen realizovat potřebné typy činností, které jsou pro vývoj daného produktu potřebné. V potaz byla vzata také seniorita vývojářů tak, aby bylo možné využívat zkušenosti a znalosti senior vývojářů pro méně zkušené kolegy. Dalším kritériem pro výběr do týmu byla zkušenost s agilním vývojem alespoň u jednoho z vývojářů. Jak Scrum Master uvedl, tento tým sestavili z pěti stávajících pracovníků, kteří se znají, a v rámci 2 až 3 měsíců uvažují o možném rozšíření o 1 až 2 nové vývojáře.

Vývojáři se většinou shodli, že jejich role se mění převážně z toho důvodu, že mají být v přímém kontaktu se zákazníkem, a to prostřednictvím Product Ownera a případně i s dalšími jeho kolegy. Další změnu vnímají dva vývojáři ve způsobu spolupráce členů týmu, kde vidí rozdíl ve větší míře spolupráce mezi členy a zapojením do řešení a návrhů produktu. Jeden z vývojářů popsal své negativní zkušenosti a svou nespokojenost z bývalého zaměstnání, kde zavedení Scrumu mělo negativní dopad a bylo spojeno s nárůstem tlaku na vývojáře, a tím následně i se zvýšením stresu a snížením kvality produktu. Dle jeho názoru vývojáři spíše upřednostňují klasické přístupy řízení, kde nedochází k častým změnám a vývojáři se neúčastní takového množství schůzek.

## Meetingy a aktivity

Při rozhovoru Scrum Master uvedl, že první **Planning** trval přibližně 2,5 hodiny, ale součástí schůzky byl také výběr metody odhadů. Během Planningu si každý člen týmu připravil 3 až 4 úkoly, ke kterým přiřadil své odhady v hodinách. Product Owner uvedl, že pro první Planning byla vybrána User Story, jejíž vývoj byl snadný, ale přesto pro něj byla plánovací schůzka náročná. Časovou i obsahovou náročnost potvrdili při rozhovorech také všichni vývojáři. Někteří vývojáři uvedli, že je pro ně nezvyklé řešit plánování a odhady před klientem, nebo dokonce jakkoliv odhadovat svou práci. Jeden vývojář uvedl, že schůzka byla dlouhá a chaotická. Dle jeho názoru, pokud vývojáři nemají dostatek času pro odhady, tak se může stát, že je raději nadhodnocují, nebo naopak daný úkol podcení a přiřadí nedostačující odhad.

Scrum Master uvedl, že **Standupy** se při prvním Sprintu konaly v zasedací místnosti od 10:00 a trvaly přibližně 20 až 30 minut. Jak Scrum Master popsal, vývojáři postupně hovoří o tom, co udělali předchozí den, jakou práci mají v plánu daný den a následně tým řeší dotazy, upřesnění informací nebo případné problémy. Během schůzky vývojáři také aktualizují stav úkolů na Scrum Boardu. Product Owner se účastnil pouze dvou Standupů, a to vzdáleně prostřednictvím online aplikace. Scrum Master označil tyto schůzky za poměrně dlouhé a chaotické, což potvrdili i sami vývojáři. Dle dvou vývojářů je na Standupu řešen příliš podrobně popis práce jednotlivých členů a je zabíháno do velkých detailů. Déle také uvedli, že jsou na Standupu řešeny problémy týkající se pouze části týmu. Dle rozhovorů mají někteří vývojáři problém s dochvilností. Na Standupy se dostávají se zpožděním a ostatní členové týmu na ně čekají. Dva vývojáři ale již nyní vnímají přínos těchto schůzek, protože jim umožňují mít přehled o práci v celém týmu, předcházet možným problémům a vytvářet si reálný obraz o aktuálním stavu.



Prvního **Review** se účastnil Scrum Master, všichni vývojáři a Product Owner. Jak uvedl Scrum Master, samotná prezentace trvala několik minut a následovala diskuse o drobných úpravách. Dle Scrum Mastera probíhala prezentace dobře, přestože pro vývojáře to nebyla zcela komfortní situace, protože nejsou na prezentování zvyklí. Nervozitu vývojářů potvrdil také Product Owner, ale zároveň dodal, že byl s prezentací jejich práce spokojen. Sami vývojáři uvedli, že většinu prezentace přenechali kolegovi, který má s prezentacemi největší zkušenosti, a pouze ho doplňovali. Potvrdili také bezproblémový průběh a spokojenost Product Owenera.

Jak uvedl Scrum Master i vývojáři, hlavním tématem první **Retrospektivy** byly problémy, které souvisely se Standupy – délka schůzek, pozdní příchody členů týmu nebo přístup vývojářů k těmto schůzkám. Jako možné řešení pozdních příchodů byl navržen posun Standupu na pozdější čas. S tím ale nesouhlasili všichni vývojáři, proto i nadále zůstal čas této schůzky nastaven na desátou hodinu dopoledne. Další vyhodnocení Standupů má být provedeno po dalším Sprintu. Během Retrospektivy dále Scrum Master obecně hovořil s týmem o dalších ceremoniích ve Scrumu.

### **Praktiky a artefakty**

Scrum Master i Product Owner uvedli, že pro první **Sprint** nechali pro každého vývojáře jednodenní rezervu, protože předpokládali, že se vývoj novým přístupem může zpomalit. Proto také začali s jednodušší User Story, aby snížili pravděpodobnost neúspěchu na samém začátku pilotního projektu. Scrum Master při rozhovoru potvrdil, že první Sprint byl nastaven pouze na jeden týden, i to, že cíl tohoto Sprintu byl splněn. Samotný vývoj se dle Scrum Mastera obešel bez větších problémů, což potvrdili i sami vývojáři. Vývojáři se v podstatě shodli, že během prvního Sprintu jim největší problémy činily schůzky, které byly náročné časově i obsahem. Zároveň byli ale spokojeni s tím, že dokončili naplánovanou práci a úspěšně prezentovali výsledek klientovi. Dle Scrum Mastera měli vývojáři také problém s aktualizováním stavu úkolů na **Scrum Boardu**, na které často zapomínali, stav tabule tak neodpovídal realitě. Jak všichni respondenti uvedli, zatím se nebyli schopni shodnout na volbě nové metody odhadů, proto jako dočasné řešení ponechali odhady v hodinách a dnech. Jak uvedl Scrum Master, doporučované metody odhadů připadaly některým členům týmu příliš abstraktní. Vývojáři shodně potvrdili, že první **User Story**, se kterou pracovali, pro ně byla srozumitelná, i to, že Product Owner byl schopen vysvětlit potřebné detaily. Product Owner při rozhovoru popsal základní myšlenku toho, jak má být napsána User Story, a uvedl, že připravoval User Stories sám nebo s pomocí Scrum Mastera. Při rozhovoru Scrum Master také zmínil fakt, že se dosud nezabýval formulací **Definition of Done**. Sám si je vědom, že se jedná o chybu, ale z důvodu náročnosti příprav a změn, se mu formulaci této definice nepodařilo realizovat. Jako náhradu Definition of Done použili postup, že za hotovou práci považují to, co je otestované. Tyto otestované funkcionality pak předávají Product Ownerovi k akceptaci.

### **Agilní programování**

Všichni vývojáři při rozhovorech potvrdili, že se zahájením Scrumu nezačali používat žádné metody agilního programování a ani s nimi Scrum Master nehovořil o jejich zavádění. Sami vývojáři by nejvíce uvítali zavedení automatických testů a Continuous Integration. Dále také navrhovali Code Refactoring nebo Pair Programming, a to hlavně kvůli méně zkušeným kolegům, jejichž kód zatím není dostatečně kontrolován.

## **Agilní smlouva**

Product Owner při rozhovoru potvrdil, že dodavatel projevil zájem o úpravu kontraktu na agilní typ smlouvy. Ale jeho firma trvá na zachování klasické formy kontraktu, kde je uvedena záloha, celková částka a termín dokončení projektu. Jak dále uvedl, tato smlouva pokrývá 5měsíční vývoj na pilotním projektu, po kterém by měl následovat další rozvoj produktu. Jak bude formulována tato další spolupráce, není v současné době řešeno na straně dodavatele, ani na straně klienta.

### **Shrnutí rozhovorů – fáze 1**

Z výsledků rozhovorů lze vyhodnotit, že Scrum Master přistupoval ke své přípravě na tuto novou roli aktivně, vzdělával se, navštívil kurz, využíval praktické zkušenosti jednoho z vývojářů a byl také ochoten pomoci s přípravou Product Ownerovi. Product Owner se také před zahájením věnoval získávání potřebných informací a znalostí ohledně náplně své role a základních elementů metody Scrum. Oba mají zájem se v této oblasti nadále aktivně rozvíjet a vzdělávat. Scrum Master i Product Owner se ale při své přípravě zaměřili převážně na své vlastní role, jejich náplň a povinnosti. Scrum Master zcela opomenul přípravu samotného Development týmu, a to zvláště vývojářů, kteří nemají praktickou zkušenost s touto metodou.

Scrum Master si je vědom rozdílnosti přístupu manažera a své nové role Scrum Mastera. Správně vnímá svoji roli jako roli lídra, kouče a člověka odpovědného za předávání informací a znalostí o metodě Scrum všem zainteresovaným osobám. Vnímá také náplň své práce na operativní úrovni, kdy je potřeba zajišťovat konání a průběh všech potřebných aktivit. Product Owner definoval podstatu své role také správně, protože si je vědom své odpovědnosti za produkt, jeho obsah, řízení priorit i potřebné schopnosti komunikace v týmu i mimo něj. Dále také adekvátně popsal své základní povinnosti a úkoly. Je velmi pozitivní, že Scrum Master je pro Development tým stále fyzicky k dispozici a Product Owner se osobně účastní všech důležitých schůzek a je ochoten za týmem přijet v případě potřeby.

Jedním z důležitých kritérií výběru vývojářů byly odborné znalosti a zkušenosti. Alespoň u jednoho vývojáře v týmu byla požadována zkušenost s agilním vývojem. Počet vývojářů byl pro tento typ vývoje zvolen vhodně. Z těchto informací vyplynulo, že tým byl vhodně sestaven tak, aby mohl být multifukční a byl schopen realizovat potřebné typy činností. Postoje a názory vývojářů k agilnímu způsobu vývoje nebyly uvedeny mezi kritérii výběru členů týmu. To může být příčinou toho, že byl do pilotního projektu vybrán i jeden vývojář, který má poměrně silný negativní postoj k agilnímu způsobu řízení a špatné zkušenosti s metodou Scrum, proto upřednostňuje klasický přístup řízení. Otázkou zatím zůstává, jak se bude projevovat jeho negativní zkušenost s metodou Scrum a jeho postoje v následujícím období. Zda bude tento přístup působit v týmu komplikace, nebo zda bude možné využít negativní zkušenosti tohoto vývojáře naopak ve prospěch pilotního projektu a vyvarovat se některých chyb. Vzhledem k tomu, že v agilním přístupu je důvěra, spolupráce a komunikace velmi důležitá, Scrum Master měl při výběru členů týmu brát v potaz i jejich názory a postoje k metodě Scrum i k agilnímu přístupu jako takovému. Následně pak mohl zvážit a případně i přehodnotit původní výběr.

Během prvního Planningu řešil tým, jakou metodu bude používat pro odhadování své práce. Toto rozhodování mělo proběhnout separátně ještě před samotným Planningem. Nemusel by se toho tak účastnit i Product Owner. Samotný Planning mohl být pro členy týmu kratší a méně náročný. Členové týmu si správně rozdělili práci na několik úkolů. Pro první Sprint byla také vhodně vybrána User Story, která nebyla pro tým náročná na vývoj, a každému vývojáři byla ponechána jednodenní časová rezerva. Odhadování v hodinách, na kterém se

nakonec tým prozatím dohodl, není samo o sobě problém. Scrum pouze určité metody odhadu doporučuje, ale rozhodnutí je na konkrétním týmu. Přesto je pozitivní, že tým souhlasí, že se o změnu v budoucnu pokusí. Při Standup schůzkách se během prvního Sprintu vyskytlo několik problémů, které jsou typické pro týmy začínající s agilním vývojem. Problémy se týkaly převážně doby trvání schůzky (20 až 30 minut), zabíhání do přílišných detailů, řešení obsáhlejších problémů již během této schůzky a pozdních příchodů vývojářů. Tyto problémy byly také hlavním tématem diskuse na první Retrospektivě. Scrum Master se s týmem zatím nedohodl na žádných konkrétních opatřeních, která by mohla vést ke zlepšení situace ohledně průběhu Standupů. Před Retrospektivou proběhlo ještě v rámci ukončení Sprintu první Review, které bylo kvůli nutnosti prezentace pro některé vývojáře mimo jejich komfortní zónu. I přesto byly výsledky i samotná prezentace všemi účastníky hodnoceny nakonec pozitivně.

Na závěr vyhodnocení první fáze rozhovorů, které proběhly po prvním Sprintu, lze ještě stručně shrnout, že Scrum tým se při prvním Sprintu potýkal převážně s problémy, které souvisely s meetingy a aktivitami. Tyto ceremonie byly pro tým časově i obsahově náročné. Byly také zjištěny dílčí nedostatky při výběru vývojářů pro pilotní projekt a absence přípravy vývojářů na agilní transformaci a metodu Scrum. Zcela zatím také chybí alespoň rámcové představy k Definition of Done a o zavádění agilních metod programování. Klient zatím nesouhlasí se změnou kontraktu na agilní typ smlouvy. I přes tyto počáteční problémy byl tým schopen dosáhnout cíle prvního Sprintu a výsledky práce prezentovat.

### **3.2.2 Vyhodnocení rozhovorů – fáze 2**

Tato podkapitola se zabývá vyhodnocením druhé fáze rozhovorů, které proběhly dva měsíce od zahájení Scrumu. Vyhodnocení je rozděleno na několik oblastí: průběh po zahájení agilní transformace, role a fungování týmu, meetingy a aktivity, praktiky a artefakty, agilní programování a agilní smlouva. Otázky pro tyto rozhovory jsou v Příloze 2 a přepis samotných rozhovorů jsou v Příloze 4.

#### **Průběh po zahájení agilní transformace**

Scrum Master v úvodu rozhovoru uvedl, že uplynulé dva měsíce byly z jeho pohledu náročnější, než očekával, a to i přesto, že se dle jeho slov stav zlepšoval poměrně rychle. Dále také uvedl, že během uplynulého období získal mnoho nových zkušeností, ale také větší potřebu se v této oblasti i nadále rozvíjet. Přestože základní procesy a přístupy metody Scrum na pilotním projektu již fungují a situace je stabilnější, vnímá tento stav Scrum Master pouze jako začátek. Náročný až chaotický začátek při rozhovorech zmínili také dva vývojáři. Zároveň oba potvrdili, že již došlo ke zlepšení, jsou lépe organizovaní a začali si zvykat na nová pravidla. Dle jejich názoru již Scrum aktivity nedělají členům týmu takové problémy jako na samém začátku. Pro jednoho z vývojářů je samotný vývoj na pilotním projektu bez zásadních problémů a pokračuje správným směrem, i když si je vědom, že stále je a bude co zlepšovat. Jistý prostor pro zlepšení zmínil v rozhovoru také další vývojář, a to v oblastech vývoje nebo ve spolupráci. I tak ale vnímá dosavadní výsledky celkem pozitivně. Dle jeho názoru je jejich práce úspěšná a tým udělal za dané období pokrok i přes jisté nezdary, jako bylo například nesplnění cíle sprintu. Dále také uvedl, že by rád tímto stylem vývoje pokračoval i v budoucnu. I přes náročnou změnu a jisté obavy z ohrožení projektu i další jeho kolega uvedl, že by tímto stylem práce rád pokračoval i nadále. Pouze pro jednoho z vývojářů představuje tento přístup velké množství schůzek, přidanou administrativu a zároveň tlak na dokončení cíle sprintu pro potřebu prezentace klientovi. Junior programátor v rozhovoru uvedl, že tento styl práce je pro něj, kvůli jeho menším zkušenostem, poměrně náročný.

Zmínil své obavy z toho, že jeho nezkušenosti mohou způsobit problémy celému týmu, a to zejména v případech, kdy svou práci nestihne dokončit a musí ho nahradit někdo z kolegů. Přes tyto problémy ale rovněž vnímá pozitiva agilního vývoje, protože při tomto způsobu práce může poměrně rychle získat nové zkušenosti a znalosti. Zároveň také doplnil, že by uvítal více podpory od svých zkušenějších kolegů, protože v současnosti je tento styl práce pro něj více stresující, než býval při předchozím klasickém vývoji. I přes počáteční problémy, které zmínil Scrum Master i vývojáři, hodnotil Product Owner kvalitu práce vývojového týmu i nového produktu velmi dobře. Je spokojen i s rychlostí, jakou vývojáři odstraňují případné chyby a zjištěné nedostatky.

Na otázku, jaké je očekávání na následující měsíce, Scrum Master při rozhovoru odpověděl, že by se chtěli zlepšit v úspěšnosti dosahování cílů Sprintu i v tom, aby vždy dosahovali daného cíle včetně otestování. Také si je vědom toho, že bude potřeba řešit personální obsazení v týmu vývojářů vzhledem k současné ne zcela vyhovující situaci. Na závěr doplnil, že by se chtěl s týmem věnovat agilním technikám programování. Na stejnou otázku ohledně očekávání Product Owner uvedl, že věří i nadále v dobrou spolupráci a zlepšující se výsledky. Zmínil, že by rád rozšířil tým o dalšího vývojáře z důvodu nárůstu požadavků na produkt i jeho rychlejší vývoj.

### **Role a fungování týmu**

Scrum Master při rozhovoru hodnotil dvouměsíční spolupráci tří vývojářů z pětičlenného týmu jako bezproblémovou. Dále uvedl, že problematické situace se někdy vyskytují u junior programátora, zejména v souvislosti s novým stylem práce a náročností vývoje na daném projektu. Také zmínil, že za dva měsíce nedošlo k žádné změně u jednoho vývojáře, který má od začátku negativní přístup k agilnímu stylu vývoje a k metodě Scrum. Scrum Master také upozornil, že tento negativní přístup jednoho člena již působí problémy ostatním členům týmu. Product Owner hodnotil dvouměsíční spolupráci na projektu pozitivně, i přestože věřili, že v tuto chvíli budou dále v samotném vývoji i zavedených procesech, ale zároveň vidí za týmem velký kus odvedené práce. Uvedl také, že s týmem i se Scrum Masterem se mu celkově spolupracuje dobře a že se celý tým posouvá směrem dopředu a získává cenné zkušenosti. Je si zároveň vědom toho, že všichni mají ještě rezervy, protože se tento způsob práce učí pouze krátkou dobu. Dle jeho názoru bude i nadále většina lidí v týmu s tímto způsobem práce spokojenější a bude podávat lepší výsledky než dříve. Spokojenost s agilním stylem vývoje a s metodou Scrum potvrdili při rozhovoru také tři vývojáři z týmu. Vývojáři uváděli, že je práce baví, spolupracuje se jim dobře, a to i přes občasnou vyšší náročnost. Přestože se tito vývojáři znají již z předešlých projektů, jeden z vývojářů uvedl, že vnímá změnu v jejich spolupráci, protože řeší situace, které dříve nenastávaly, a každý z nich si teprve svým vlastním tempem zvyká na nová pravidla a postupy. Dle tohoto vývojáře jsou také všichni členové týmu ve větším kontaktu než dříve. Vývojáři při rozhovorech také zmiňovali podstatný rozdíl v přístupu jejich kolegy, kterého se nepodařilo přesvědčit o tom, že tento způsob práce je efektivnější. Tento vývojář si je sám vědom, že jeho přístup působí v týmu problémy, ale stále je přesvědčený o tom, že by jeho práce byla efektivnější při klasickém vývoji. Dle jednoho z vývojářů je také občas junior programátorovi přidělena práce, která přesahuje jeho zkušenosti a znalosti. Junior programátor při rozhovoru uvedl, že by uvítal nad svou prací větší kontrolu od svých kolegů i více času pro společnou práci s kolegy, protože jsou schopni mu poradit a může se od nich učit.

Téměř všichni vývojáři se při rozhovorech v podstatě shodli na tom, že jejich Scrum Master dělá svou práci dobře. Uváděli například, že se o svůj tým stará, je stále k dispozici, snaží se pomoci a poradit, zároveň vede tým k samostatnosti a vyslechne si jejich názory. Dle jednoho

vývojáře jejich Scrum Mastera práce baví a přesunutí do této role z role projektového manažera považuje za úspěšné. Dle dalšího vývojáře musí Scrum Master také řešit složité situace, protože ne vše se týmu daří dle očekávání. Zároveň ale vývojáři uváděli při rozhovorech také to, že Scrum Masterovi chybí praktické zkušenosti a ne všechny praktiky metody Scrum se mu podařilo úspěšně zavést. Jako příklad jeden z vývojářů uvedl, že Scrum Master opomenul nastavení Definition of Done nebo že došlo k vynechání retrospektivních schůzek. Práci Product Ownera hodnotili vývojáři při rozhovorech velmi pozitivně. Zmínili například, že velmi dobře rozumí danému produktu, dokáže srozumitelně vysvětlit své očekávání a představy nebo doplnit potřebné informace k zadaným User Stories. Jeden z vývojářů uvedl, že Product Owner je také ochotný diskutovat o návrzích řešení a o možnostech realizace. Také dále doplnil, že Product Owner je na schůzky vždy pečlivě připraven. Vývojáři vnímají pozitivně jeho práci i v tom, že získávají rychlou zpětnou vazbu a mohou produkt přizpůsobit požadavkům a konkrétním představám. Jejich kolega také uvedl, že zatím nedošlo k situaci, kdy by Product Owner během Sprintu změnil své požadavky tak, aby ohrozil dokončení vývoje v dané iteraci. Junior programátor také neměl k práci Product Ownera výhrady a doplnil, že díky jeho práci mají informace jak o současných požadavcích, tak o práci na další Sprinty.

## Meetingy a aktivity

Při rozhovoru Scrum Master uvedl, že u Standup schůzek již mírně zkrátili dobu trvání, a to na 15 až maximálně 25 minut. Zkrácení doby potřebné pro Standup schůzky na přibližně čtvrt hodiny potvrdili také všichni vývojáři. Dle dvou vývojářů je v některých případech důvodem prodloužení těchto schůzek delší řešení problémů nebo získávání potřebných informací od kolegů. Zároveň také dva vývojáři uvedli, že by tyto schůzky měly být ještě kratší, než je tomuto v současnosti. Scrum Master dále také uvedl, že Standup schůzky jsou již lépe organizované, než tomu bylo na počátku, a také potvrdil, že vývojáři si na tento typ schůzky zvykli. Dle jeho slov jsou stále v řešení pozdní příchody vývojářů, ale v porovnání s počátečním stavem již došlo ke zlepšení. Tyto již pouze občasné problémy s pozdními příchody na Standup schůzky potvrdili také dva vývojáři.

Scrum Master také uvedl, že ke zlepšení došlo u Planning schůzek, které jsou lépe organizované a kratší. Vysvětlil také, že společně s Product Ownerem seznamují vývojáře průběžně s novými funkcionalitami, které budou implementovány, aby je průběžně připravovali na plánování. Tento postup potvrdili při rozhovorech i vývojáři a všichni se také shodli na tom, že se jim v současné době plánuje lépe a vnímají zde poměrně výrazné zlepšení. Scrum Master potvrdil, že tým zůstal u odhadování v hodinách a žádnou z jiných metod zatím nevyužili. To potvrdili sami vývojáři a dva z nich při rozhovoru uvedli, že by rádi zkusili také jiné metody odhadu. Jeden z vývojářů zmínil při rozhovoru svou pochybnost nad přesností odhadů. Product Owner při rozhovoru uváděl, že se stále účastní všech Planning a Review schůzek. Celkově hodnotil, že je u schůzek viditelný progres a schůzky nejsou tak chaotické a vyčerpávající jako na začátku. Dle jeho názoru i většina vývojářů již s těmito schůzkami nemá takový problém a například při Review jsou si u prezentace produktu jistější. Scrum Master při rozhovoru potvrdil, že Review schůzek se účastní on, všichni vývojáři a Product Owner. Jak dále uvedl, pouze na jedné této schůzce byli přítomni další zástupci klienta. Tuto účast dalších zástupců klienta potvrdil i Product Owner a doplnil, že se jednalo o účast jeho nadřízeného a projektového manažera. Jak uvedl, jeho kolegové byli s prezentací produktu spokojeni, ale dalších schůzek se již neúčastnili z časových důvodů. Také dle Scrum Mastera probíhají prezentace bez problémů a vývojáři si již na tyto schůzky zvykli. Dále se ale v rozhovoru zmínil, že neproběhly některé retrospektivní schůzky, a to z časových důvodů, protože se tým věnoval dokončování prací z předešlého Sprintu. Na retrospektivních

schůzkách se tým věnoval převážně problémům spolupráce mezi jednotlivci nebo problémům při odhadování časové náročnosti.

## **Praktiky a artefakty**

Scrum Master při rozhovoru potvrdil, že mají nastaveny dvoutýdenní sprinty, které jim dávají více prostoru na vývoj. Kapacitu na jednoho vývojáře počítá Scrum Master 36 hodin na týden a čtyři hodiny času zbývají na režii, opravy chyb a potřebné aktualizace. Z uvedených 36 hodin dále odečítá ještě schůzky a ceremonie Scrumu, aby zjistil čistý čas pro samotný vývoj. Tento čistý čas pro vývoj na týden pak vynásobí dvěma a získá čistý čas pro jeden Sprint. Což, jak uvedl, je 67 hodin času na jednoho vývojáře. Scrum Master také při rozhovoru upozornil, že ne vždy dosáhli cíle Sprintu. Jak dále vysvětlil, cíle Sprintu nebylo dosaženo dvakrát, což potvrdil při rozhovoru i Product Owner a následně všichni vývojáři. Dle Scrum Mastera a vývojářů byl v prvním případě vývoj pro tým náročnější, než odhadovali. Podle jednoho z vývojářů měli jako tým věnovat odhadování více času a vše důkladněji promyslet. Dle tohoto vývojáře byl také junior programátorovi zadán příliš složitý úkol, který tento méně zkušený vývojář nemohl zvládnout sám, a jeho kolegové neměli čas mu během své vlastní práce pomoci. Jak nakonec ještě doplnil, tato práce junior programátora musela být nakonec dokončena jedním z kolegů. Tuto náročnou situaci potvrdil při rozhovoru i junior programátor, který uvedl, že si s daným úkolem nevěděl sám zcela rady a potřeboval pomoc od kolegů. V druhém případě, kdy nebyl splněn cíl Sprintu, se tým potýkal s větším množstvím chyb, což potvrdili všichni respondenti. A jak uvedl jeden z vývojářů, bylo potřeba nejprve chyby opravit, aby bylo možné práci prezentovat klientovi. Dle Scrum Mastera a Product Ownera v dalších případech cíle dosaženo bylo. Ale jak upřesnil Scrum Master, nastala také situace, že některé části nebyly v rámci sprintu otestovány a toto testování probíhalo až v následujícím sprintu. Scrum Master v rozhovoru také uvedl, že v Definition of Done zatím nedošlo k žádné změně.

## **Agilní programování**

Při rozhovoru uvedl jeden z vývojářů, že již pokročili v řešeních týkajících se Continuous Integration. Vysvětlil, že používají pro sdílení kódu verzovací systém Git, a každý update kódu automaticky spouští kompilování na serveru a následné automatické unit testy, jejichž výsledky jsou pak doručeny vývojářům. Jak programátor uvedl, psaní unit testů zavedli nově, takže zdrojové kódy nejsou ještě těmito testy kompletně pokryty. Kromě automatických testů by se dle jeho slov chtěli také věnovat integračním testům a případně dalším potřebným analýzám. Nově zavedené psaní automatických testů potvrzují také ostatní vývojáři, tuto činnost vnímají pozitivně a mají zájem, aby těchto testů bylo co možná největší množství. Junior programátor při rozhovoru uvedl, že zatím bohužel nebyla vyzkoušena metoda párového programování, o kterou měl zájem. Dále se zmínil také o tom, že jeho kolegové nemají bohužel čas kontrolovat jeho kód, aby mu mohli poradit, jak se zlepšovat, nebo mu alespoň poskytl zpětnou vazbu ohledně chyb a nedostatků v jeho kódu.

## **Agilní smlouva**

Product Owner v rozhovoru opět potvrdil, že dosud nebyla řešena otázka agilní smlouvy, přestože dodavatel o tento typ kontraktu projevil zájem. Dle Product Ownera stále zůstává otázkou, jestli se tento typ spolupráce promítne v následujících měsících také do nového typu smlouvy.

## Shrnutí rozhovorů – fáze 2

Z informací získaných z rozhovorů v druhé fázi lze vyhodnotit, že po poměrně náročném až chaotickém začátku docházelo na pilotním projektu během dvou měsíců ke stabilizaci situace. Celková situace, vývoj na projektu i kvalita je, i přes dílčí neúspěchy, vnímána pozitivně Scrum Masterem, Product Ownerem i většinou vývojářů. Členové týmu v tomto období začali přijímat nová pravidla a aktivity, které jim metoda Scrum přinesla. Všichni také získali nové zkušenosti a pociťují poměrně značný progres, zároveň jsou si však vědomi prostoru pro zlepšení. Pouze jeden z vývojářů vnímá tento agilní přístup stále negativně a i nadále dává přednost klasickému řízení projektů. Přístup tohoto vývojáře má již negativní dopady na spolupráci v týmu. Potřebu řešit tuto situaci si uvědomují vývojáři i Scrum Master. Také samotný vývojář si je vědom, že jeho přístup působí komplikace, proto by Scrum Master neměl řešení této situace již dále odkládat, protože spolupráce ostatních členů týmu je bezproblémová a funkční. Ostatní vývojáře práce baví a rádi by v tomto stylu práce pokračovali i nadále. S problémy se začal potýkat také junior programátor, protože náročnost práce je, vzhledem k jeho menším znalostem a zkušenostem, neúměrně vysoká a bohužel také nemá potřebnou pomoc od svých seniorních kolegů. U tohoto programátora bohužel také chybí potřebná kontrola jeho práce, protože kolegové nemají dostatek času. U junior programátora je ale velmi pozitivní, že i přes tyto komplikace dokáže vnímat výhody, které tento styl práce přináší, protože má zájem se rozvíjet a získávat potřebné znalosti.

Nastavení Sprintů na dvoutýdenní iterace v případě, že vyhovují vývoji daného produktu, je z teoretického hlediska v pořádku i včetně toho, že délka Sprintu je konstantní a v průběhu uplynulé doby nebyla měněna. Výjimkou byl pouze první Sprint, který trval jeden týden. Scrum Master také vhodně počítá kapacitu vývojářů pro jednotlivé Sprints, protože ponechává čas na potřebnou režii a schůzky, kterých se vývojáři účastní během sprintu. Během sledovaného období tým 2x nedosáhl cíle Sprintu. V jednom případě byl důvodem nesplnění cíle primárně chybný odhad náročnosti práce týmu a dále také neúměrná náročnost práce pro junior programátora, která neodpovídá jeho znalostem a zkušenostem. V druhém případě, kdy cíl nebyl splněn, se tým potýkal s větším množstvím chyb, kvůli kterým nebylo možné práci prezentovat, tým tak potřeboval více času na opravu chyb. Za sledované období tým také udělal poměrně značný pokrok v řešeních týkajících se Continuous Integration, které na počátku nebylo zavedeno a nebylo zřejmé, kdy se těmto záležitostem bude tým a Scrum Master věnovat. Téměř všichni vývojáři se také aktivně podílejí na psaní unit testů. Nejen vzhledem k tomu, že v týmu je také junior programátor, ale celkově chybí zatím využití dalších agilních metod, jako jsou například Code Review nebo párové programování. Tyto metody by týmu pomohly předcházet chybám a pomohly by daný vývoj také zefektivnit.

Práce Scrum Mastera je členy týmu celkově vnímána pozitivně, přestože má v této roli málo praktických zkušeností. Scrum Master zvládl zatím implementovat většinu základních technik metody Scrum, být stále týmu k dispozici a pomáhat členům týmu. Některé praktiky metody Scrum se mu ale správně zavést nepodařilo. Jedná se hlavně o Definition of Done, která stále nebyla nastavena, což lze označit za zásadní pochybení z jeho strany. Další problematickou aktivitou se zdají být retrospektivní schůzky, jejichž konání nebylo pravidelně zajištěno a po některých Srintech byly tyto schůzky vynechány. Kvůli tomu nebylo možné, aby tým diskutoval o problémech nebo komplikacích v uplynulém sprintu a snažil se najít řešení. Prospektivní schůzky po ukončení sprintu neproběhly z časových důvodů, protože tým dokončoval práci, kterou v rámci Sprintu nestihl realizovat. U Standup schůzek došlo během uplynulých dvou měsíců k určitému zlepšení v organizaci, ke zkrácení doby trvání na 15 až 25 minut, částečně se zlepšila také dochvilnost vývojářů, ale ani tento problém nebyl zcela odstraněn. Přesto se stává, že některé Standup schůzky jsou delší, než by měly být, a jsou na nich řešeny otázky, které již do obsahu těchto schůzek nepatří. Byl zaznamenán také progres

u Planning schůzek, a to opět jak v organizování obsahu schůzky, tak ve zkrácení potřebného času pro tyto schůzky. Scrum Master a Product Owner správně zavedli průběžné informování vývojářů o nových funkcionalitách, které budou v budoucích sprintech realizovány, tzv. Backlog Grooming. Během uplynulých dvou měsíců zůstalo odhadování náročnosti práce stále v jednotkách hodin a nebyla zatím využita žádná z metod, které doporučuje Scrum pro odhady náročnosti. Tým se pouze při retrospektivních schůzkách zabýval problémy, které měli při odhadech náročnosti, ale žádná změna v této oblasti realizována nebyla. Během uplynulého období si vývojáři také více zvykli na předvádění produktu při Review schůzkách a v prezentaci jsou si již jistější. Až na jednu výjimku se těchto schůzek účastní ze strany klienta stále jen Product Owner, ale nikdo z jeho kolegů nebo nadřízených se těchto schůzek pravidelně neúčastní. Chybí zde tak přijímá zpětná vazba od dalších osob, které by přímo vývojářům mohly sdělit svůj názor. Vývojáři ale velmi pozitivně vnímají práci svého Product Ownera. Z uvedených informací vyplynulo, že svou roli zvládá velmi dobře a je podporou pro tým. Jeho práce je pro vývoj přínosná, protože velmi dobře rozumí novému produktu. O vývoj má zájem a s týmem se setkává na pravidelných schůzkách, na které je vždy připraven, aby dokázal odpovídat na otázky vývojářů. Stále neřešenou oblastí zatím zůstává forma kontraktu mezi firmou XY a jejím klientem, i když již firma XY projevila zájem o změnu kontraktu, který by odpovídal současnému typu spolupráce. Klient ale zatím ohledně změny smlouvy nezaujal jasné stanovisko.

Na závěr vyhodnocení druhé fáze rozhovorů lze shrnout, že během zkoumaného období došlo k celkovému zlepšení fungování Scrum týmu na pilotním projektu. Také adaptace jednotlivých lidí na nové role probíhala celkově pozitivně a Scrum Master, Product Owner i jednotliví vývojáři postupně přivykali na aktivity a pravidla metody Scrum. Výjimku tvoří pouze jeden z vývojářů, jehož odmítavý přístup k agilnímu stylu vývoje má negativní dopady na spolupráci v týmu. Dále se také podařilo alespoň částečně odstranit problémy, se kterými se tým na začátku potýkal při téměř všech aktivitách a schůzkách. Tyto schůzky se již podařilo časově zkrátit a lépe řídit jejich obsah, i když v této oblasti zůstává pro tým stále určitý prostor ke zlepšení. Mezi hlavní zjištěné nedostatky, které se týkají aktivit a praktik metody Scrum, lze zařadit chybějící Definition of Done, nedosažení cílů dvou Sprintů a vynechávání retrospektivních schůzek. Pozitivně se naopak projevily progres v implementaci potřebných technologií souvisejících s Continuous Integration a zavedení unit testů pro kontrolu kódu. I když jsou to zatím pouze tyto agilní metody, které tým zvládl realizovat, jedná se o důležitý posun. Development tým by však neměl zůstat jen u těchto aktivit a měl by využít další doporučované techniky. Zejména je vhodné se zaměřit na techniky, které budou nápomocné pro kontrolu kódu a včasné odstranění chyb nebo také pro podporu junior programátora v týmu, která zatím není dostatečná. Neměly by být opomíjeny ani doporučované techniky pro odhady náročnosti, které by mohly týmu tuto činnost usnadnit. I když agilní manifest upřednostňuje spolupráci před smlouvou, měl by typ kontraktu odpovídat způsobu spolupráce mezi dodavatelem a klientem, což se do této chvíle u tohoto projektu zatím nepodařilo.

### **3.2.3 Hodnocení a sebehodnocení**

Tato podkapitola se zabývá výsledky vzájemného hodnocení a sebehodnocení členů týmu, hodnocením jednotlivých členů týmu Scrum Masterem a hodnocením Development týmu Product Ownerem. Tato hodnocení byla provedena během první i druhé fáze rozhovorů, aby bylo možné vyhodnotit vývoj a změny během agilní transformace.



## Sebehodnocení členů týmu

Všichni vývojáři provedli v rámci rozhovorů sebehodnocení ve třech oblastech: **aktivita, nasazení a týmovost**. Sebehodnocení probíhalo pomocí stupnice 1 až 10 pro jednotlivé oblasti. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup daného vývojáře. Hodnota 10 znamená nejvyšší možnou aktivitu, nasazení a nejlepší týmový přístup daného vývojáře. Tabulka níže zobrazuje sebehodnocení jednotlivých členů Development týmu.

**Tabulka 2 Sebehodnocení členů týmu**

	Aktivita		Nasazení		Týmovost	
	Fáze 1	Fáze 2	Fáze 1	Fáze 2	Fáze 1	Fáze 2
Vývojář 1	7	7	8	6	7	6
Vývojář 2	5	7	8	6	6	6
Vývojář 3	4	8	7	7	8	9
Vývojář 4	3	4	4	4	5	4
Vývojář 5	6	5	7	8	7	4

Zdroj: Vlastní zpracování

První řádek výše uvedené **Tabulky 2 Sebehodnocení členů týmu** obsahuje tři oblasti, ve kterých vývojáři prováděli sebehodnocení. Sloupce Fáze 1 obsahují hodnoty z prvního sebehodnocení, které proběhlo po prvním Sprintu. Sloupce Fáze 2 obsahují hodnoty z druhého sebehodnocení, které proběhlo dva měsíce po zahájení agilního vývoje metodou Scrum.

U **vývojáře 1** nedošlo k výrazným změnám v sebehodnocení mezi první a druhou fází. U sebehodnocení aktivity v obou případech uvedl hodnotu sedm bodů. Své nasazení v první fázi hodnotil osmi body a v druhé fázi šesti body. Zde tedy vývojář hodnotil své nasazení menším počtem bodů v druhé fázi. Týmovost hodnotil vývojář v první fázi sedmi body a v druhé fázi uvedl u této oblasti šest bodů. U tohoto vývojáře nedošlo k výrazným změnám v sebehodnocení v uvedených oblastech. Tento vývojář měl již zkušenosti s metodou Scrum z minulosti, což může být důvodem, proč bylo jeho sebehodnocení poměrně stabilní v obou fázích. Na druhou stranu za mírným snížením bodů u sebehodnocení může stát například jen to, že již mírně opadlo počáteční nadšení. Příčinou mohou být také problémy v týmu, o kterých hovořil tento vývojář v rozhovoru a které týkaly se spolupráce s jedním jejich kolegou.

**Vývojář 2** také výrazně nezměnil své sebehodnocení za dané období. V oblasti aktivity nastalo zvýšení hodnocení z pěti bodů v první fázi na sedm bodů ve fázi druhé. Naopak v oblasti nasazení hodnotil v první fázi tuto oblast osmi body a v druhé fázi pouze šesti body. Týmovost má u tohoto vývojáře shodné hodnocení obou fází – šest bodů. Tento vývojář neměl praktické zkušenosti s metodou Scrum z minulosti a jak uváděl při rozhovoru, spoléhal na začátku převážně na Scrum Mastera. Zvýšení bodů u aktivity může znamenat, že je vývojář již schopen se sám více zapojit do aktivit a praktik metody Scrum. U nasazení se jedná o stejné bodové snížení jako u vývojáře 1. Zde se také mohl projevit vliv problémů spolupráce v týmu nebo některé neúspěchy při nedosažení cílů. Důvodem může být ale také to, že díky zkušenosti vnímá tuto oblast ze širšího pohledu a je si vědom rezerv i prostoru pro zlepšení, o kterém hovořil také při rozhovoru v druhé fázi.

**Vývojář 3** měl největší rozdíl mezi první a druhou fází sebehodnocení v oblasti aktivity. V první fázi hodnotil svou aktivitu čtyřmi body a v druhé fázi osmi body. Naopak žádný rozdíl neměl v sebehodnocení oblasti nasazení, zde v obou fázích uvedl sedm bodů. Hodnocení týmovosti se zvýšilo sebehodnocením v druhé fázi o jeden bod z původních osmi na devět bodů. U tohoto vývojáře došlo také k navýšení hodnocení aktivity. Zde bylo ale výrazně vyšší, což opět může znamenat větší zapojení do Scrum aktivit a procesů. Ostatní rozdíly v sebehodnocení nejsou nijak výrazné. V porovnání s ostatními členy týmu hodnotil tento vývojář svou týmovost velmi vysokým počtem bodů. I přes problémy, které v týmu mezi jeho členy existují, hodnotil svou týmovost téměř nejvyšší hodnotou. Při rozhovoru v druhé fázi tyto problémy sice jednou zmínil, ale hlavně uvedl, že je za tento tým rád, že se mu spolupracuje dobře a práce s kolegy ho baví. Takže je možné, že se ho tyto problémy tolik nedotýkají a neovlivňují ho jako ostatní členy týmu.

U **vývojáře 4** se hodnocení uvedených oblastí při porovnání s jeho kolegy bodově výrazně liší. Zároveň tento vývojář své hodnocení za dané období změnil pouze nepatrně. Svou aktivitu hodnotil v první fázi třemi body, v druhé fázi navýšil hodnocení na čtyři body. Své nasazení hodnotil v obou fázích čtyřmi body a týmovost ohodnotil v první fázi pěti body a v druhé fázi snížil hodnocení na čtyři body. U tohoto vývojáře je bodové hodnocení všech tří oblastí výrazně nižší, než je tomu u jeho kolegů. Lze uvést, že nižší sebehodnocení v daných oblastech potvrzuje vývojářův negativní přístup k agilnímu vývoji a metodě Scrum, o kterém hovořil jak on sám při obou rozhovorech, tak tento jeho přístup potvrzovali i jeho kolegové.

**Vývojář 5** zaznamenal největší změny při sebehodnocení v oblasti své týmovosti a pouze drobné změny v oblasti aktivity a nasazení. Svou aktivitou hodnotil v první fázi šesti body a v druhé fázi snížil toto hodnocení na pět bodů. Své nasazení hodnotil v první fázi sedmi body a v druhé fázi zvýšil hodnocení na osm bodů. Oblast týmovosti hodnotil v první fázi sedmi body, v druhé fázi hodnocení výrazně snížil, a to na čtyři body. Výraznější snížení vlastního sebehodnocení v této oblasti může být ovlivněno komplikovanější spoluprací s kolegy, od kterých, jak uváděl při rozhovoru v druhé fázi, nemá dostatek podpory, pomoci a kontroly při složitých úkolech, kde mu chybí jeho vlastní zkušenosti. Ze sebehodnocení aktivity a nasazení vyplývá, že tyto problémy zatím nijak výrazně v těchto oblastech vývojáře negativně neovlivnily. Může to být vysvětleno tím, jak vývojář uváděl při rozhovoru v druhé fázi – tuto práci vnímá jako dobrou příležitost učit se nové věci a získávat zkušenosti.

## **Shrnutí sebehodnocení**

Slovní interpretace sebehodnocení má v rámci této diplomové práce jistá omezení. Po tomto sebehodnocení by měl dále následovat rozhovor Scrum Mastera s jednotlivými vývojáři a hodnoty by měly být vyhodnoceny společně. Teprve na základě informací získaných při těchto rozhovorech lze výsledky dále upřesnit, protože u nižšího bodového hodnocení se nemusí vždy jednat o chybu nebo problém daného vývojáře. Naopak daný vývojář může být ovlivněn situací v celém týmu anebo se ještě do dané role a pravidel Scrumu zapracovává. Tím, jak získává jistý nadhled a schopnost vidět problémy v širším kontextu, tak některé své oblasti může hodnotit s odstupem času méně body, než tomu bylo na počátku. Zároveň vysoké zvýšení bodového hodnocení může znamenat, že se vývojáři povedlo do nových aktivit poměrně snadno zapracovat, což pociťuje jako svůj úspěch. Dále lze potvrdit, že uvedené sebehodnocení, minimálně v oblastech, které byly při rozhovorech zjištěny jako problematické, odpovídá nízkému sebehodnocení u vývojáře 4 a snížení sebehodnocení v druhé fázi u vývojáře 5.

## Vzájemné hodnocení členů týmu

Všichni vývojáři provedli v rámci obou fází rozhovorů také hodnocení ostatních vývojářů. Vývojáři nehodnotili Scrum Mastera ani Product Ownera. Toto hodnocení probíhalo pomocí fiktivní odměny 20 000 Kč, kterou měl k dispozici každý vývojář a měl ji rozdělit mezi členy Development týmu tak, jak uznal za vhodné. Hodnocení jednotlivými vývojáři v obou fázích je kompletně uvedeno v Příloze 5, tabulka níže zobrazuje pouze celkové výsledky všech hodnocení.

Tabulka 3 Vzájemné hodnocení členů týmu

Hodnocený vývojář	Fiktivní odměna celkem	
	Fáze 1	Fáze 2
Vývojář 1	25 000 Kč	26 000 Kč
Vývojář 2	23 000 Kč	25 000 Kč
Vývojář 3	22 000 Kč	24 000 Kč
Vývojář 4	13 000 Kč	4 000 Kč
Vývojář 5	17 000 Kč	21 000 Kč

Zdroj: Vlastní zpracování

Výše uvedená Tabulka 3 Vzájemné hodnocení členů týmu zobrazuje, jaká celková fiktivní odměna byla přiřazena jednotlivým vývojářům. Uvedené hodnoty jsou součtem hodnocení daného vývojáře od jeho čtyř kolegů. Sloupec Fáze 1 obsahuje hodnoty z prvního hodnocení, které proběhlo po prvním Sprintu. Sloupec Fáze 2 obsahuje hodnoty z druhého hodnocení, které proběhlo dva měsíce po zahájení agilního vývoje metodou Scrum.

**Vývojář 1** obdržel od svých kolegů v podobě fiktivní odměny nejvyšší hodnocení, a to jak v první fázi, tak i ve fázi druhé. V první fázi mu kolegové přidělili 25 000 Kč, v druhé fázi pak 26 000 Kč. Jeho odměna je poměrně stabilní, a to i na úrovni jednotlivých vývojářů v obou fázích hodnocení. Tento vývojář měl v minulosti s metodou Scrum praktickou zkušenost a je tedy možné, že i z tohoto důvodu může být vnímán kolegy jako nejužitečnější člen týmu. **Vývojář 2** získal od svých kolegů druhou nejvyšší fiktivní odměnu v obou fázích hodnocení. V první fázi měl celkovou odměnu 23 000 Kč a v druhé fázi 25 000 Kč. I v tomto případě bylo hodnocení vývojáře v průběhu obou fází poměrně stabilní a jednotlivé částky odměny byly změněny pouze nepatrně. Další stabilní hodnocení a třetí nejvyšší odměna byla udělena **vývojáři 3**. V první fázi to byla částka 22 000 Kč a v druhé fázi 24 000 Kč. Vzhledem k tomu, že vývojář 2 a vývojář 3 neměli žádné praktické zkušenosti s metodou Scrum, je jejich hodnocení vysoké a kolegové jejich práci oceňují. **Vývojář 4** dosáhl v obou fázích na nejnižší fiktivní odměnu od svých kolegů. V první fázi to byla částka 13 000 Kč a ve druhé fázi již pouze částka 4 000 Kč. Již jeho první odměna byla výrazně nižší než odměna u ostatních kolegů, navíc v druhé fázi toto hodnocení ještě výrazně kleslo. Hodnocení u tohoto vývojáře v druhé fázi snížili všichni jeho kolegové. Jeden z jeho kolegů nepřidělil tomuto vývojáři žádnou odměnu. Nízké odměny tohoto vývojáře lze vysvětlit tím, že ho jeho kolegové považují za nejméně užitečného ve svém týmu. Tyto nízké odměny také korespondují s negativním přístupem tohoto vývojáře k agilnímu vývoji a metodě Scrum, který následně činí potíže všem jeho kolegům v týmu, jak uváděli jednotliví vývojáři zejména v druhé fázi rozhovorů.

**Vývojář 5** obdržel čtvrtou nejvyšší fiktivní odměnu od svých kolegů. V první fázi hodnocení to byla částka 17 000 Kč a v druhé fázi částka 21 000 Kč. U tohoto vývojáře dva jeho kolegové hodnocení v daném období nezměnili a dva jeho kolegové naopak své hodnocení shodně navýšili. Zvýšení hodnocení u tohoto vývojáře 5, kterým je junior programátor, může

být jistou formou ohodnocení snahy a pracovitosti tohoto člena týmu, který se dle informací z rozhovorů v druhé fázi, musí potýkat sám bez jejich pomoci s úkoly, kde jeho zkušenosti a znalosti ještě ne vždy dostačují.

### Shrnutí vzájemného hodnocení členů týmu

Prostřednictvím vzájemného přidělení fiktivních odměn mezi členy týmu lze vyhodnotit, že nejvíce oceněným je vývojář 1 a jeho práce má dle jeho kolegů nejvyšší hodnotu. Téměř srovnatelné bylo hodnocení u vývojářů 2 a 3. Jejich hodnocení je také velmi vysoké, takže kolegové považují jejich práci za přínosnou. V pořadí čtvrté nejnižší hodnocení měl vývojář 5, junior programátor, jehož vykonávaná práce není pro ostatní členy tolik přínosná. Ale na druhou stranu kolegové oceňovali pravděpodobně i snahu tohoto junior programátora a jeho celkový přístup. Tomuto členovi týmu tak přidělili výrazně vyšší hodnocení než vývojáři 4. Vývojář 4 dostal od svých kolegů výrazně nižší odměnu, která byla v druhé fázi ještě snížena, což reflektuje jeho negativní přístup. Jeho kolegové ho vnímají jako méně užitečného člena týmu.

### Hodnocení členů týmu Scrum Masterem

Scrum Master provedl v rámci obou rozhovorů také hodnocení jednotlivých vývojářů ve třech oblastech: **aktivita, nasazení a týmovost**. Toto hodnocení probíhalo pomocí stupnice 1 až 10 pro jednotlivé oblasti. Hodnota 1 znamená nejnižší aktivitu, nedostatečné nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší možnou aktivitu, vysoké nasazení a nejlepší týmový přístup. Tabulka níže zobrazuje, jak Scrum Master ohodnotil jednotlivé vývojáře.

**Tabulka 4** Hodnocení členů týmu Scrum Masterem

Hodnocený vývojář	Aktivita		Nasazení		Týmovost	
	Fáze 1	Fáze 2	Fáze 1	Fáze 2	Fáze 1	Fáze 2
Vývojář 1	8	7	9	7	9	8
Vývojář 2	8	7	8	7	8	8
Vývojář 3	5	6	6	7	7	8
Vývojář 4	4	3	4	3	3	2
Vývojář 5	6	7	3	6	7	5

Zdroj: Vlastní zpracování

Výše uvedená **Tabulka 4** Hodnocení členů týmu Scrum Masterem obsahuje tři oblasti, ve kterých Scrum Master hodnotil jednotlivé vývojáře. Sloupce Fáze 1 obsahují hodnoty z prvního hodnocení, které proběhlo po prvním Sprintu. Sloupce Fáze 2 obsahují hodnoty z druhého hodnocení, které proběhlo dva měsíce po zahájení agilního vývoje metodou Scrum.

Scrum Master hodnotil **vývojáře 1** v první fázi v oblasti aktivity osmi body a v druhé fázi sedmi body. Oblast nasazení u tohoto vývojáře hodnotil v první fázi devíti body a v druhé fázi hodnocení snížil na sedm bodů. Týmovost tohoto vývojáře hodnotil Scrum Master nejdříve devíti body a následující hodnocení snížil na osm bodů. Tento vývojář dostal od Scrum Mastera při první fázi nejvyšší hodnocení z celého týmu, a to v oblasti nasazení a týmovosti. Vysoké hodnocení může souviset s tím, že tento vývojář již měl v minulosti praktické zkušenosti s metodou Scrum a mohl tak Masterovi v začátcích pomoci, jak také uvedl při rozhovoru v první fázi. Snížení bodového hodnocení v druhé fázi zároveň také koresponduje se snížením sebehodnocení tohoto vývojáře.

U **vývojáře 2** hodnotil Scrum Master v první fázi aktivitu i nasazení osmi body, v druhé fázi obě hodnocení snížil o jeden bod, tedy na sedm bodů. A oblast týmovosti hodnotil v první i v druhé fázi osmi body, zde bylo hodnocení beze změny. Hodnocení tohoto vývojáře je poměrně stabilní u obou fází, ale v porovnání se sebehodnocením vývojáře se liší v oblasti aktivity, kdy naopak vývojář v druhé fázi své sebehodnocení zvýšil o dva body. I když hodnocení týmovosti Scrum Master ani vývojář u sebehodnocení v daném období nezměnili, Scrum Master hodnotil vývojáře o dva body vyšší hodnotou, než sám sebe posoudil vývojář. Z hodnocení Scrum Mastera lze usuzovat, že práci a přístup tohoto vývojáře lze hodnotit velmi dobře, přestože i zde je prostor ke zlepšení.

**Vývojář 3** obdržel od Scrum Mastera v první fázi v oblasti aktivity hodnocení pěti body a v druhé fázi Scrum Master navýšil hodnocení na bodů šest. Nasazení ohodnotil nejprve šesti body a v druhé fázi také navýšil hodnocení o jeden bod, tedy na sedm bodů. Týmovost u tohoto vývojáře byla v první fázi ohodnocena sedmi body a v druhé fázi bylo hodnocení opět navýšeno o bod, tentokrát na osm bodů. V porovnání s předchozími vývojáři je hodnocení tohoto vývojáře nižší převážně v oblasti aktivity a mírně nižší v oblasti nasazení. Naopak tento vývojář při sebehodnocení zaznamenal výrazný progres v oblasti aktivity, který se ale s hodnocením Scrum Mastera neshoduje.

Hodnocení **vývojáře 4** bylo výrazně nižší než u všech jeho kolegů. Scrum Master hodnotil tohoto vývojáře v oblasti aktivity i v oblasti nasazení během první fáze čtyřmi body, v druhé fázi obě tato hodnocení snížil na tři body. Týmovost tohoto vývojáře vyhodnotil Scrum Master ještě nižším počtem bodů, a to v první fázi třemi body a v druhé fázi pouze dvěma body. Toto nízké hodnocení koresponduje s hodnoceními, které u tohoto vývojáře provedli i jeho kolegové při rozdělování fiktivní odměny. V podstatě se shoduje i s jeho vlastním sebehodnocením. Týmovost však ohodnotil Scrum Master ještě nižším počtem bodů, než ohodnotil sám u sebe tento vývojář. Hodnocení Scrum Mastera rovněž potvrdilo problematickou situaci tohoto vývojáře v týmu, jeho negativní přístup vnímá celý tým.

Scrum Master ohodnotil **vývojáře 5** v první fázi v oblasti aktivity šesti body a v druhé fázi sedmi body. Oblast nasazení zhodnotil v první fázi nízkým počtem bodů – vývojáři přidělil pouze tři body, v druhé fázi navýšil toto hodnocení na šest bodů. Oblast týmovosti naopak hodnotil v první fázi vyšším počtem bodů, a to konkrétně sedmi body, v druhé fázi snížil hodnocení na 5 bodů. Scrum Master při svém hodnocení zaznamenal výrazné zlepšení v oblasti nasazení tohoto vývojáře, zde se může odrážet snaha tohoto junior programátora zvládat složité situace bez pomoci svých kolegů i v případech, kdy na ně nemá dostatek znalostí a zkušeností. Naopak snížení hodnocení u týmovosti mohlo znamenat to, že pro tohoto programátora není současná spolupráce s kolegy dostačující, protože na něj mají málo času. Dané snížení hodnocení se shodně projevilo i při vlastním sebehodnocení. Naopak se při něm neprojevil progres v oblasti nasazení, který zde zaznamenal Scrum Master, ale zde měl sám vývojář poměrně vysoké hodnocení už v první fázi.

### **Shrnutí hodnocení členů týmu Scrum Masterem**

Slovní interpretace hodnocení Scrum Mastera má v rámci této diplomové práce jistá omezení. Po tomto hodnocení by měl následovat rozhovor Scrum Mastera s jednotlivými vývojáři a hodnoty by měly být konzultovány společně. Během rozhovoru by měl Scrum Master jednotlivým vývojářům vysvětlit své hodnocení a také ho porovnat se sebehodnocením daného vývojáře v jednotlivých oblastech. Každý vývojář by také měl objasnit své sebehodnocení a jeho případnou změnu v průběhu daného období. Společně by tak měli nalézat vysvětlení pro nižší hodnocení nebo snížení hodnocení. Bodový pokles by ale neměl být vnímán žádnou ze stran negativně, ale naopak jako společné nalezení prostoru pro další

progres člena týmu. Nyní lze potvrdit, že hodnocení Scrum Mastera se v několika oblastech shodovalo se sebehodnocením jednotlivých členů týmu, ale vyskytly se také případy, kdy se hodnocení lišilo.

### Hodnocení týmu Product Ownerem

Product Owner provedl v rámci obou rozhovorů hodnocení Development týmu jako celku ve třech oblastech: **aktivita, nasazení a týmovost**. Product Owner na rozdíl od Scrum Mastera nehodnotil jednotlivé vývojáře. Hodnotil danou oblast u celého týmu dohromady jednou známkou. K hodnocení měl také k dispozici stupnici 1 až 10. Hodnota 1 znamená nejnižší aktivitu, nedostatečné nasazení a žádný týmový přístup za celý Development tým. Hodnota 10 znamená nejvyšší možnou aktivitu, vysoké nasazení a nejlepší týmový přístup celého Development týmu. Tabulka níže zobrazuje hodnocení týmu Product Ownerem.

**Tabulka 5** Hodnocení týmu Product Ownerem

	Aktivita		Nasazení		Týmovost	
	Fáze 1	Fáze 2	Fáze 1	Fáze 2	Fáze 1	Fáze 2
Development tým	7	8	7	5	5	7

Zdroj: Vlastní zpracování

Výše uvedená **Tabulka 5** Hodnocení týmu Product Ownerem obsahuje tři oblasti, ve kterých Product Owner hodnotil Development tým jako celek. Sloupce Fáze 1 obsahují hodnoty z prvního hodnocení, které proběhlo po prvním Sprintu. Sloupce Fáze 2 obsahují hodnoty z druhého hodnocení, dva měsíce po zahájení agilního vývoje metodou Scrum. Product Owner hodnotil **aktivitu** celého týmu v první fázi sedmi body a v druhé fázi osmi body. Hodnocení aktivity daného týmu je v daném období téměř stabilní a uvedené hodnoty jsou poměrně vysoké. Toto hodnocení se shoduje s informacemi, které Product Owner uvedl při rozhovoru, kde uváděl, že i když vidí ještě prostor pro zlepšování, je s prací vývojářů celkově spokojen. **Nasazení** týmu hodnotil Product Owner v první fázi sedmi body a v druhé fázi snížil hodnocení na 5 bodů. V této oblasti se projevovalo mírné snížení hodnocení, podobně tomu bylo u hodnocení většiny vývojářů Scrum Masterem a rovněž i u některých vývojářů při sebehodnocení. Naopak **týmovost** Product Owner ohodnotil v první fázi nižším počtem bodů, konkrétně pěti body a v druhé fázi navýšil své hodnocení na 7 bodů. Zde byla tendence v hodnocení opačná, než tomu bylo u Scrum Mastera nebo u většiny vývojářů při sebehodnocení, kteří své hodnocení této oblasti v druhé fázi spíše snižovali. Protože Product Owner není stále přítomen v týmu a účastní se pouze vybraných schůzek, je možné, že problémy, které se začaly vyskytovat uvnitř v týmu, nevnímal tak výrazně jako samotní vývojáři a Scrum Master.

### Shrnutí hodnocení týmu Product Ownerem

Hodnocení týmu jako celku v oblastech aktivity a nasazení týmu se z větší části shoduje s hodnocením, které provedl Scrum Master u jednotlivců, nebo se sebehodnocením, které provedli sami vývojáři. Naopak hodnocení týmovosti má u Product Owenera opačnou tendenci. Toto hodnocení by mělo být následně také diskutováno s celým týmem. Při této diskusi by Product Owner mohl vysvětlit podrobnosti k hodnocení jednotlivých oblastí.

## Vyhodnocení vývoje týmu

Scrum tým – vývojáři, Scrum Master a Product Owner – provedli hodnocení vývoje týmu dle Tuckmanova modelu. Toto hodnocení Scrum tým provedl po ukončení prvního Sprintu a následně jej opakovat po dvouměsíční spolupráci. Jednotliví členové týmu přiřadili na vytištěné schéma Tuckmanova modelu bod (značku) k vývojové fázi, která dle jejich názoru nejlépe odpovídá situaci v týmu. Tabulka níže zobrazuje celkové výsledky z první i druhé fáze hodnocení.

**Tabulka 6 Vyhodnocení vývoje týmu dle Tuckmanova modelu**

	Forming (Formování)	Storming (Bouření)	Norming (Normování)	Performing (Optimální výkon)
Fáze 1	7	0	0	0
Fáze 2	0	6	1	0

Zdroj: Vlastní zpracování

Výše uvedená **Tabulka 6 Vyhodnocení vývoje týmu dle Tuckmanova modelu** zobrazuje celkový počet bodů, který byl přiřazen k jednotlivým vývojovým fázím týmu tak, jak situaci vyhodnotili všichni členové Scrum týmu. Řádek Fáze 1 obsahuje počty bodů z prvního hodnocení, které proběhlo po prvním Sprintu. Řádek Fáze 2 obsahuje hodnoty z druhého hodnocení, které proběhlo dva měsíce po zahájení agilního vývoje metodou Scrum. V první fázi hodnocení přiřadili všichni členové Scrum týmu bod k fázi Forming (Formování). V druhé fázi hodnocení přiřadilo šest členů Scrum týmu bod k fázi Storming (Bouření) a jeden člen Scrum týmu přiřadil bod k fázi Norming (Normování).

Přestože se jednotliví členové týmu již znali před zahájením agilní transformace, v první fázi vyhodnotili situaci v týmu všichni respondenti jako Forming (Formování). Z daného hodnocení vyplývá, že i přes tento fakt začaly od samého začátku působit na členy týmu změny ve způsobu spolupráce a stylu vývoje. V této fázi museli členové týmu měnit své zvyky, zavedená pravidla a začít se učit nové věci. Jak vyplynulo také z rozhovorů, toto období bylo pro tým náročné a chaotické, protože si členové teprve začali osvojovat praktiky metody Scrum v praxi. Po dvouměsíční spolupráci nebylo vyhodnocení vývoje týmu zcela jednoznačné. Scrum Master a všichni vývojáři označili, že se tým nachází ve fázi Storming (Bouření). Tato fáze je typická spory v týmu a negativními postoji, dále také například zmatením a frustrací. Že se tyto situace v týmu vyskytují, vyplynulo z druhé fáze rozhovorů, kde bylo také zjištěno, že nejčastější příčinou konfliktů je jeden z vývojářů. Další problémy se projevily u junior programátora (vývojář 5). Pouze Product Owner označil, že se tým již posunul do fáze Norming (Normování). Product Owner pravděpodobně nebyl součástí přímých konfliktů v týmu, protože s týmem tráví čas pouze prostřednictvím schůzek – a jak vyplynulo i z rozhovorů, tato setkání tým již zvládá výrazně lépe, než tomu bylo na začátku. A také vzhledem k tomu že je Product Owner s prací týmu celkově spokojen, mohl nabýt dojem, že tým se již posunul do další fáze.

## Shrnutí vyhodnocení vývoje týmu

Během zkoumaného dvouměsíčního období došlo v týmu k přesunu z vývojové fáze Forming (Formování) do vývojové fáze Storming (Bouření). Také dle získaných informací z rozhovorů situace v týmu odpovídá jak tomuto průběhu, tak jeho vyhodnocení respondenty. V první fázi se na daném hodnocení shodli všichni respondenti a v druhé fázi pouze Product Owner vyhodnotil, že tým je již ve fázi Norming (Normování). Pohled Product Owenera na tým je ale

„vzdálenější“, než je tomu u Scrum Mastera a samotných vývojářů, kteří spolu tráví veškerý pracovní čas.

### 3.2.4 Scrum tabule, Burn-down graf

Vývoj na pilotním projektu byl sledován prostřednictvím softwarového nástroje JIRA, který umožňuje správu Sprintů a jednotlivých úkolů prostřednictvím elektronické Scrum tabule. Tento software zároveň automaticky vyhodnocuje průběh a dokončení jednotlivých Sprintů pomocí vizualizace Burn-down grafu. Tyto elektronické dokumenty byly zachyceny prostřednictvím snímků obrazovky a pomocí obsahové analýzy je níže vysvětlen význam získaných dat.

#### Scrum tabule

Při obsahové analýze elektronického dokumentu bylo zjištěno, že tým používá elektronickou Scrum tabuli rozdělenou na čtyři sloupce: TO DO, IN PROGRESS, FOR TESTING a DONE. Sloupec TO DO obsahuje nové ještě nezačaté úkoly jednotlivých pracovníků. Sloupec IN PROGRESS je určen pro úkoly, na kterých vývojáři aktuálně pracují. Do sloupce FOR TESTING jsou přiřazovány úkoly, na kterých je dokončen vývoj, a jsou určeny pro otestování. A ve sloupci DONE se nacházejí pouze úkoly, na kterých byl dokončen vývoj, byly již otestovány a nebyly u nich nalezeny chyby. Příloha 6 obsahuje snímek elektronické Scrum tabule druhého Sprintu. Tento snímek byl zachycen v polovině tohoto dvoutýdenního Sprintu a v daném okamžiku bylo dokončených osm úkolů, dva úkoly byly připravené pro testování, devět úkolů bylo rozpracovaných a na jedenácti úkolech se ještě nezačalo pracovat. Vzhledem k tomu, že zkoumaný tým má pouze pět vývojářů, tento stav – 9 úkolů ve sloupci IN PROGRESS – odpovídá o vyšší míře rozpracovanosti. Tato větší rozpracovanost se týká pouze dvou programátorů. Jeden z těchto programátorů měl v dané chvíli rozpracované čtyři úkoly a jeho kolega dva úkoly. Ostatní programátoři v daném čase pracovali na jednom úkolu.

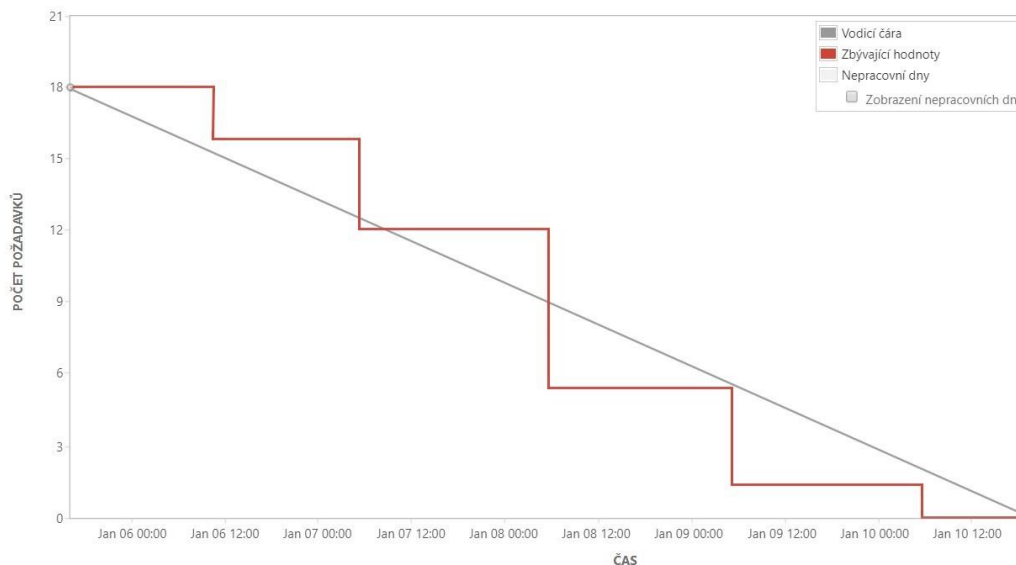
Na základě výše uvedených údajů lze shrnout, že tým používá vhodně rozdělenou elektronickou Scrum tabuli. Rozdělení na 4 sloupce je v souladu s doporučením metody Kanban, která doporučuje rozdělení tabule na minimálně tři sloupce. Naopak s doporučením této metody není v souladu vyšší rozpracovanost, která byla zjištěna zejména u jednoho programátora, který v daném okamžiku, dle údajů uvedených na tabuli, měl rozpracované 4 úkoly. Pravidla v týmu by měla rozpracovanost jednotlivých vývojářů omezovat, aby se snížilo množství právě realizované práce, zvýšila se přehlednost o práci vývojářů a urychlilo se dokončování jednotlivých úkolů. Je potřeba upozornit na to, že pokud na tabuli nedochází k pravidelné aktualizaci úkolů, nemusí se jednat o skutečnou vyšší rozpracovanost, ale pouze o opomenutí správy úkolů ze strany vývojáře.



## Burn-down grafy

Po ukončení každého Sprintu byla provedena obsahová analýza Burn-down grafu, který zachytil průběh Sprintu a to, zda byla plánovaná práce v rámci Sprintu realizována kompletně a bylo dosaženo cíle Sprintu. Tyto grafy ukazují pouze to, jak daný Sprint probíhal a zda tým zvládl svou práci dokončit včas. Ze samotného grafu není možné zjistit příčinu toho, proč nebyla práce dokončena včas. Proto jsou níže uvedené informace vždy ještě dovysvětleny pomocí informací získaných z rozhovorů, při kterých respondenti odpovídali na otázky související s průběhem Sprintů a s plněním jejich cílů. Níže je zobrazen Graf 2 z prvního Sprintu, který trval jeden týden.

Graf 2 Burn-down graf – Sprint 1

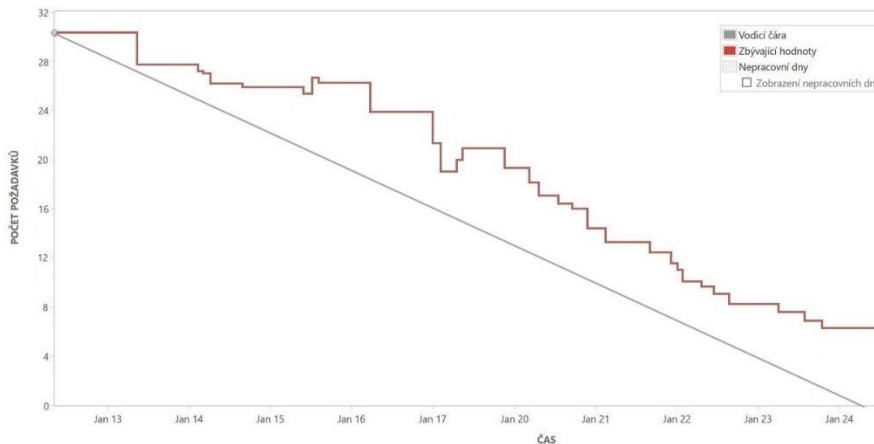


Zdroj: Data z pilotního projektu (firma XY)

Výše uvedený **Graf 2 Burn-down graf – Sprint 1** zobrazuje průběh prvního Sprintu pilotního projektu. Šedá čára je vodítko, kolem kterého by měl průběh Sprintu ideálně mírně oscilovat. Skutečný průběh práce znázorňuje červená čára. Pokud je práce dokončena v rámci Sprintu, červená čára končí na spodní ose dříve než čára šedá. Na výše uvedeném grafu je vidět, že práce během Sprintu více oscilovala kolem vodící čáry. To ale bez dalších potřebných informací nelze konkrétně vyhodnotit. Při rozhovorech však bylo zjištěno, že vývojáři zapominají aktualizovat stav svých úkolů na Scrum tabuli, což může být příčinou této větší oscilace, kdy v určitý den se zdá, že tým práci nestíhá (červená čára nad vodící čárou) a druhý den je naopak dokončeno větší množství práce (červená čára pod vodící čárou). Z výše uvedeného grafu je také patrné, že tým stihl plánovanou práci dokončit včas s určitou rezervou. Toto se shoduje s informacemi, které byly získány během rozhovorů s respondenty, kteří potvrdili, že během prvního Sprintu, který trval pouze jeden týden, byl splněn cíl Sprintu, a také to, že tým měl vývoj naplánovaný s dostatečnou rezervou.

Naopak k nesplnění cíle Sprintu došlo v následujícím Sprintu 2. Tento Sprint 2 i všechny následující Sprints probíhaly již po dobu dvou týdnů, aby měl tým více času na vývoj. Průběh a nesplnění cíle tohoto Sprintu 2 zobrazuje následující Graf 3.

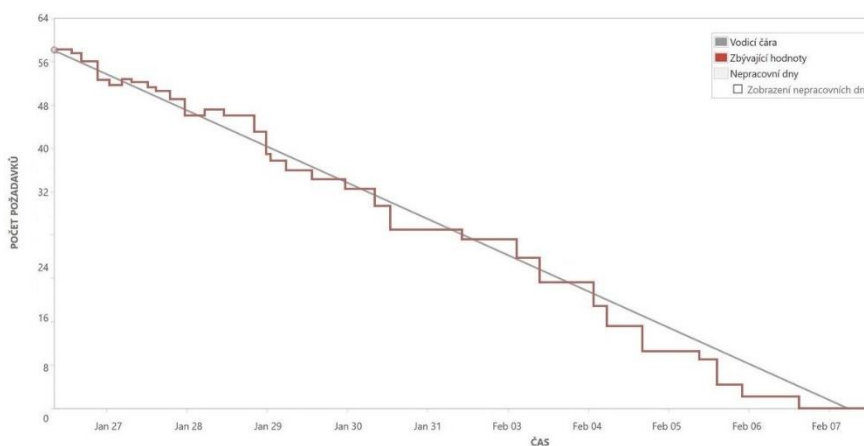
**Graf 3 Burn-down graf – Sprint 2**



Zdroj: Data z pilotního projektu (firma XY)

Výše uvedený **Graf 3 Burn-down graf – Sprint 2** vizualizuje průběh druhého Sprintu. Červená křivka je v tomto grafu po celou dobu záznamu nad vodící šedou přímkou. Z toho je patrné, že během celého Sprintu tým nezvládl své úkoly zpracovávat dostatečnou rychlostí tak, aby je všechny stihl dokončit včas a splnil svůj cíl. V druhé fázi rozhovorů všichni vývojáři i Scrum Master potvrdili, že se při tomto Sprintu potýkali s náročnějším vývojem, než předpokládali dle svých odhadů. Tým tedy v tomto případě udělal chybu již při odhadování své práce a kvůli výrazně vyšší pracnosti nestihl svou práci dokončit včas. Jinou situaci je možné vidět na níže uvedeném Grafu 4, kdy při třetím Sprintu vývojáři dokončili své úkoly včas a svůj cíl splnili.

**Graf 4 Burn-down graf – Sprint 3**

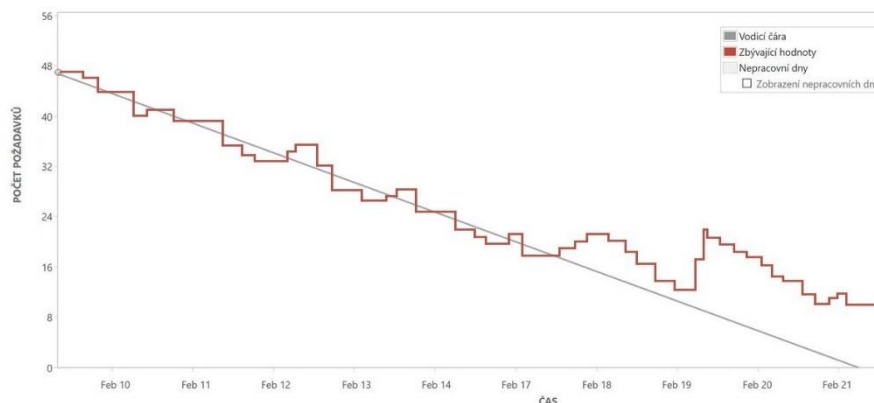


Zdroj: Data z pilotního projektu (firma XY)

Výše zobrazený **Graf 4 Burn-down graf – Sprint 3** je v porovnání s předchozím grafem téměř ideální, jak je vidět na červené křivce, která většinou pouze mírně osciluje kolem vodící přímky. Toto mírné kolísání znamená, že v tomto Sprintu se týmu dařilo úkoly realizovat s potřebnou rychlostí a průběh celého sprintu byl rovnoměrně rozložený. V tomto případě

také původní odhady pracnosti odpovídaly skutečnému vývoji. Všechny úkoly byly dokončeny včas a cíl Sprintu byl v této iteraci splněn. Poslední zkoumanou iterací v rámci pilotního projektu byl Sprint 4. U tohoto sprintu se týmu opět nepodařilo splnit jeho cíl, protože vývojáři nedokončili všechny své úkoly v požadovaném termínu. Tento čtvrtý Sprint je zachycen na níže uvedeném Grafu 5.

**Graf 5 Burn-down graf – Sprint 4**



Zdroj: Data z pilotního projektu (firma XY)

Výše zobrazený **Graf 5 Burn-down graf – Sprint 4** zobrazuje také nesplnění cíle Sprintu a nedokončení všech úkolů. A to přestože větší polovina tohoto grafu je v podstatě ideální, protože křivka opět mírně osciluje kolem vodící přímky. To se ale v druhé polovině Sprintu změnilo a křivka je zobrazena pouze nad vodící přímkou. V této části je také patrné, že byly do Sprintu přidávány nové úkoly – jsou to ty části křivky, které směřují nahoru. Tým pak pracoval na dokončování těchto úkolů – křivka opět klesá – ale již to nestihl v požadovaném termínu. Tyto informace dostupné z grafu je potřeba doplnit informacemi z druhé fáze rozhovorů, kde opět vývojáři a Scrum Master vysvětlili důvody nesplnění cíle tohoto Sprintu. Tentokrát se nejednalo o chybné odhady, ale při vývoji se vyskytlo větší množství neočekávaných chyb a tým je již nestihl v rámci daného Sprintu všechny opravit.

Čtyři výše uvedené Burn-down grafy pomohly vizuálně dokreslit a potvrdit odpovědi vývojářů, Scrum Mastera a Product Ownera, které se týkaly průběhu sledovaných Sprintů a důvodů nesplnění cíle Sprintů. Informace, které tito respondenti uváděli, pak naopak umožnily vysvětlit skutečné důvody podoby křivek u jednotlivých grafů.

### 3.2.5 Agilní smlouva

Další obsahová analýza byla provedena nad tištěným dokumentem, kterým byla současná smlouva mezi firmou XY a jejím klientem, se kterým spolupracují na zkoumaném pilotním projektu. Obsahová analýza byla zaměřena na význam částí smlouvy, které definují rozsah díla, termíny plnění a rozvržení plateb. Při zkoumání tohoto dokumentu bylo cílem zjistit, o jaký typ kontraktu se jedná ve vztahu k současnému stylu vývoje a zda tento dokument odpovídá doporučeným typům smluv pro agilní vývoj.

Předmětem současné smlouvy je vytvoření a předání kompletního díla, kterým je nová softwarová aplikace, jejíž rozsah a kvalita odpovídá popisu dané smlouvy. Smlouva obsahuje stručný popis nosných částí aplikace a popis základních oblastí a funkcionalit, které klient od aplikace očekává. Jsou zde také uvedeny požadované jazykové verze aplikace. Popis aplikace obsahuje programovací jazyky a technologie, které mají být pro vývoj použity. Smlouva

neobsahuje podrobný výčet všech jednotlivých funkcí aplikace ani jejich výslednou vizuální podobu prostřednictvím návrhu. Nejsou zde uvedeny žádné informace, náčrtky nebo diagramy, které by byly dodány z předchozích analýz a které by jasně vymezovaly rozsah vývoje.

Smlouva obsahuje termíny pro dílčí plnění a termín pro finální předání aplikace. První dílčí termín zahrnuje základní uživatelské rozhraní a základní funkcionality aplikace. Druhý dílčí termín plnění zahrnuje předání beta verze aplikace již s plnou funkcionalitou. Finální termín pak zahrnuje nasazení hotové aplikace do produkčního prostředí pro koncové uživatele. Smlouva také definuje rozdělení dílčích plateb u tohoto projektu. První platba byla navázána na první dílčí termín plnění, který zahrnoval dodání základních funkcionalit aplikace. Tento termín byl dva měsíce od zahájení vývoje. Druhá dílčí platba byla navázána na dodání beta verze aplikace, k němuž mělo dojít pět měsíců od zahájení vývoje. Poslední (třetí) platba byla dohodnuta po finálním dokončení a nasazení aplikace do produkčního prostředí, což je šest měsíců od zahájení vývoje. U uvedených termínů a plateb, ale ani v rámci celé smlouvy není zmíněna žádná informace, která by souvisela s agilním vývojem nebo metodou Scrum. Nejsou zde uvedeny ani informace o iteračním vývoji ve Sprints, délce iterací, ceně za vývoj během jedné iterace nebo například popis Definition of Done.

Kontrakt mezi firmou XY a jejím klientem lze zařadit mezi klasické smlouvy s pevným termínem a cenou. Tento typ smlouvy by měl být uveden velmi podrobným popisem celého díla, aby bylo možné správně stanovit cenu a termín dodání. Rozsah funkcionalit aplikace je ale ve zkoumané smlouvě uveden velmi stručně a obsahuje pouze popis základních částí. Pro agilní smlouvu by byl tento popis dostačující, protože rámcově definuje základní funkcionalitu aplikace i to, co od ní klient očekává. Pro klasickou smlouvu může být tento popis problematický, jelikož může znamenat pro vývojový tým možný nárůst pracnosti v průběhu upřesňování nebo změn jednotlivých funkcí aplikace. Vzhledem k fixní ceně a termínu nemusí mít vývojový tým dostatek finančních a časových kapacit, aby vše stihl realizovat. Agilní smlouvy také obsahují finální cenu za dílo, ale má zde být ponechána jistá flexibilita. Dále by měla smlouva umožnit čerpat finance v průběhu vývoje za každou dokončenou iteraci, proto agilní smlouva musí definovat množství hodin vývoje v rámci iterace a cenu za jeden Sprint. Zkoumaná smlouva mezi firmou XY a jejím klientem není vhodná pro současný agilní styl vývoje, protože se jedná o klasický typ smlouvy s pevnou cenou i termínem a zároveň s minimálním popisem funkcionalit aplikace. Tento výsledek se shoduje s informacemi, které byly získány při rozhovorech se Scrum Masterem a Product Ownerem. Tito respondenti uvedli, že na pilotní projekt nebyla uzavřena agilní smlouva.

### **3.2.6 Celkové shrnutí vyhodnocení**

Výše uvedené podkapitoly popsaly vyhodnocení jednotlivých metod výzkumu, které proběhly během praktické části této diplomové práce. Vyhodnocení z jednotlivých metod se v některých případech vzájemně doplňovalo, což zároveň umožnilo ověřit, zda se získaná data z různých zdrojů shodují, či nikoliv. Následující text shrnuje všechny podstatné informace, které byly během výzkumu zjištěny v oblastech, na které byl výzkum zaměřen. Tato podkapitola zmiňuje i zjištěné pozitivní informace a úspěchy, ale převážně se věnuje identifikovaným nedostatkům a chybám, které byly ve sledovaném období nalezeny, aby v další části diplomové práce mohlo být navrženo vhodné doporučení.

## **Příprava na zahájení agilní transformace a sestavování týmu**

Scrum Master a Product Owner přistoupili proaktivně ke své přípravě na své nové role a nový styl vývoje. Před zahájením pilotního projektu se aktivně připravovali a získávali nové znalosti. Tuto přípravu ale zaměřili pouze na své vlastní role a do aktivní přípravy nezapojili jednotlivé vývojáře, přestože v Development týmu několik vývojářů nemělo s metodou Scrum žádnou praktickou zkušenost. Development tým byl vhodně sestaven z pohledu velikosti týmu, ale z pohledu odborných znalostí a zkušeností se během sledovaného období projevily nedostatky, které na začátku nebyly patrné. Důvodem je, že do týmu byl vybrán junior programátor, což se ukázalo jako problematické. Jeho zkušenosti a znalosti nestačují na úkoly, které jsou mu v některých případech zadávány. Zároveň tento junior programátor nemá vždy potřebnou podporu od svých zkušenějších kolegů. Tato situace následně působí problémy, které vedou až ke stavu, kdy programátor není schopen daný úkol sám dokončit, a ohrožuje tak splnění cíle Sprintu. Při výběru vývojářů také nebyl posuzován jejich přístup a postoj k agilnímu stylu vývoje a metodě Scrum. To se v průběhu celého sledovaného období projevilo jako chyba, protože do týmu byl vybrán i vývojář, který pravděpodobně kvůli své negativní zkušenosti z minulosti těmto metodám nevěří. Tato situace pak celému týmu působí problémy, což bylo během výzkumu několikrát potvrzeno při rozhovorech i hodnoceních.

## **Role a vývoj týmu**

Scrum tým se na počátku pilotního projektu nacházel ve fázi Forming (Formování), kdy museli členové týmu procházet novými zkušenostmi, které jim přinesl nový styl vývoje. Během sledovaného období se tým přesunul do vývojové fáze Storming (Bouření), která odpovídá současným problémům, které se v daném týmu vyskytují. Kromě běžných výměn názorů se v týmu největší konflikty projevily v souvislosti s vývojářem 4 a komplikace začal mít také vývojář 5. Problémy a negativní přístup vývojáře 4 se projevily na jeho nízkém bodovém sebehodnocení, které bylo výrazně nižší než u jeho kolegů. Stejně tak problematická situace junior programátora (vývojář 5) v týmu se na sebehodnocení projevila snížením bodů v druhé fázi sebehodnocení. Sebehodnocení ostatních členů bylo výrazně vyšší a stabilnější.

Sami vývojáři při vzájemném hodnocení nejvyšší odměnou ocenili vývojáře 1. Srovnatelně vysoké hodnocení měli také vývojář 2 a vývojář 3. Z těchto výsledků vyplývá, že tito tři vývojáři jsou v týmu považováni za nejpřínosnější. Za nimi následuje vývojář 5, což je Junior programátor, jehož práce sice pro tým není tolik přínosná, ale kolegové pravděpodobně oceňovali jeho proaktivní přístup a snahu. Výrazně nižší hodnocení měl vývojář 4 a toto hodnocení bylo v druhé fázi ještě sníženo. Zde kolegové reagovali na negativní přístup tohoto člena týmu a dle jejich hodnocení je považován za nejméně přínosného. V tomto případě se hodnocení tohoto vývojáře také shodovalo s nízkým hodnocením od Scrum Mastera. I přes identifikované problémy bylo hodnocení Product Ownerem, který ale hodnotil tým jako celek, poměrně vysoké. Také při rozhovorech Product Owner nezmiňoval zásadnější problémy, ale spíše svou spokojenost s vývojem i s týmem jako takovým. Sami vývojáři rovněž neměli žádné závažné výhrady k práci jejich Scrum Mastera ani Product Ownera. Dle jejich názoru tito dva členové týmu odvádí svou práci velmi dobře.

## **Meetingy a aktivity**

Na začátku pilotního projektu se tým potýkal hlavně s problémy, které souvisely s meetingy a aktivitami. Tyto činnosti pro tým byly časově velmi náročné a průběhy schůzek byly v některých případech chaotické. Během sledovaného období se týmu podařilo udělat určitý pokrok ve většině těchto aktivit. Došlo ke zlepšení věcné stránky schůzek, aktivity se podařilo lépe organizovat a zkrátit dobu jejich trvání. Přesto ale některé problémy přetrvávaly i během

dalších Sprintů. U Standup schůzek došlo během daného období ke zlepšení organizace a ke zkrácení doby trvání na 15 až 25 minut. Jde sice o zlepšení, ale pro tento typ schůzky je daný čas stále příliš dlouhý. Také nebyl zcela odstraněn problém s nedochvilností některých vývojářů na tyto schůzky. Bez závažnějších problémů se již zdají být Planning schůzky, na které se dostavují účastníci připraveni. Také Review schůzky, na kterých probíhají prezentace klientovi, jsou ze strany Development týmu zvládnány s jistotou. Jako jediné negativum u Review schůzek bylo zjištěno, že se jich účastní pouze sám Product Owner jako jediný zástupce klienta. Další problematickou oblastí byly retrospektivní schůzky, které z časových důvodů byly vynechány, aby měl tým více času na vývoj. Tým tak tento čas nemohl využít, aby diskutoval a řešil nastalé problémy v uplynulém Sprintu.

### **Praktiky a artefakty**

Při rozhovorech bylo potvrzeno, že formulace User Stories je od Product Ownera dostatečně srozumitelná a tým s těmito podklady může bez problémů pracovat. Scrum Masterovi a týmu se sice podařilo zavést většinu základních technik metody Scrum, ale ani na konci sledovaného období neměl tým žádnou konkrétní formulaci Definition of Done, což je výrazná chyba, protože ta měla být definována před spuštěním prvního Sprintu. Na začátku pilotního projektu tým také nebyl schopen shodnout se na nové metodě odhadů, a tak realizuje odhady své práce stále pouze v hodinách. Sám o sobě tento způsob není chybný, ale tým by se mohl o změnu pokusit, protože některé jejich odhady byly chybné, dokonce i sami někteří členové týmu zájem o jiné způsoby odhadů projevíli.

Samotné Sprints jsou vhodně nastaveny na konstantní délku dva týdny, výjimkou byl pouze první Sprint, který byl jednotýdenní. Tým ale ve sledovaném období 2x nesplnil cíl Sprintu. Průchody čtyřmi Sprints a plnění jejich cílů byly vizualizovány pomocí automaticky generovaných Burn-down grafů, které potvrdily a dokreslily situace, které respondenti popsali při rozhovorech. V prvním případě se jednalo o chybné odhady a následnou výrazně vyšší pracnost vývoje. Z toho důvodu vývojáři nestihli dokončit všechny požadované úkoly v daném termínu. Současně se zde již projevíly také problémy týkající se náročnosti práce pro junior programátora. V druhém případě, kdy nebyl cíl sprintu splněn, se při vývoji vyskytlo větší množství neočekávaných problémů a chyb, jejich oprava trvala déle a zmíněný programátor již neodevzdal svou práci včas. Zbývající dva cíle Sprintů byly v pořádku splněny. Tým pro správu Sprintu využívá elektronickou Scrum tabuli, kterou má vhodně rozdělenou na čtyři sloupce, ale již nemá nastavené omezení rozpracovanosti u jednotlivých členů týmu, takže jeden vývojář může pracovat na několika úkolech zároveň, což zpomaluje dokončení jednotlivých prací.

### **Agilní programování**

Na začátku pilotního projektu tým neměl žádnou konkrétní představu o tom, jaké metody agilního programování bude využívat. Ale na konci sledovaného období byl v této oblasti patrný pokrok. Tým implementoval řešení týkající se Continuous Integration a začal s psaním unit testů, což lze hodnotit pozitivně. Bohužel zatím ale nevyužil žádné z dalších vhodných metod, které by výrazně mohly pomoci junior programátorovi i celému týmu, jako jsou například Code Review nebo párové programování.

## **Agilní smlouva**

Smlouva mezi firmou XY a jejím klientem je klasickým typem kontraktu s pevným termínem a pevnou cenou, který není pro agilní vývoj vhodný. Tato smlouva na pilotní projekt neobsahuje žádné informace o agilním způsobu vývoje nebo využití metody Scrum. Ve smlouvě jsou stanoveny tři pevné milníky, na které jsou navázány části plateb. Firma tedy nedostává zaplacení za práci, kterou vynaložila v jednotlivých sprintech. Dle Product Ownera daná firma i přes zájem dodavatele zatím nesouhlasila se změnou klasické smlouvy na agilní kontrakt.

### **3.2.7 Předpoklady a výsledky výzkumu**

Praktická část diplomové práce se zaměřovala na níže uvedené oblasti, pro které byly v rámci metodiky stanovené předpoklady, se kterými se tým bude potýkat jako s nedostatky a chybami po zavedení metody Scrum na pilotním projektu. Níže v textu je stručně uvedeno, které z těchto předpokladů byly u pilotního projektu skutečně potvrzeny.

#### **Příprava na zahájení agilní transformace a sestavování týmu**

Předpoklad, že Scrum Master nebo Product Owner nebudou mít dostatečnou přípravu na zahájení agilní transformace, se nepotvrdil. Naopak se během první fáze rozhovorů potvrdil předpoklad, že nebude věnována dostatečná pozornost přípravě členů Development týmu. Během druhé fáze rozhovorů se potvrdil předpoklad, že tým nebude vhodně sestaven z pohledu potřebných zkušeností a znalostí. Tento předpoklad se ale potvrdil pouze u junior programátora. Také se při obou fázích rozhovorů potvrdil předpoklad, že tým bude nevhodně sestaven z pohledu přístupu k agilnímu způsobu vývoje. Zde byl tento předpoklad potvrzen také kvůli negativnímu přístupu jednoho vývojáře z celého týmu.

#### **Role a vývoj týmu**

Předpoklad, že bude docházet k problémům s adaptací na nové role, se nepotvrdil u Scrum Mastera ani u Product Ownera. Během obou fází rozhovorů i hodnocení se ale potvrdil u dvou vývojářů, z nichž jeden se není schopen adaptovat kvůli svému negativnímu přístupu k agilnímu vývoji. U druhého vývojáře se jedná spíše o problémy související s jeho menšími zkušenostmi a znalostmi v oblasti programování. Předpoklad, že bude docházet ke konfliktům ve formujícím se týmu, se potvrdil během druhé fáze rozhovorů zejména u výše uvedeného vývojáře, jehož negativní postoj má vliv na zbytek týmu. Žádné výraznější konflikty se nepotvrdily mezi jinými vývojáři ani mezi Scrum Masterem a Product Ownerem.

#### **Meetingy a aktivity**

Dle rozhovorů v první fázi se potvrdil předpoklad, že schůzky budou časově náročné a jejich průběh bude komplikovaný. Jak ale vyšlo z rozhovorů v druhé fázi, tento předpoklad se nadále již tak výrazně nepotvrzoval. Předpokládaná neochota účastnit se schůzek se u vývojářů nepotvrdila. Dle rozhovorů v druhé fázi se potvrdil předpoklad, že budou vynechány některé typy aktivit, zde se konkrétně jednalo o retrospektivní schůzky.

#### **Praktiky a artefakty**

Dle rozhovorů a obsahové analýzy Burn-Down grafů se předpoklad, že nebude dosahováno cílů Sprintů, potvrdil pouze částečně – během čtyř Sprintů nebylo cíle dosaženo dvakrát. Dle rozhovorů se nepotvrdil předpoklad, že User Stories budou pro vývojáře nedostatečně popsány nebo nesrozumitelné. Naopak rozhovory v obou fázích potvrdily předpoklad, že nebude jasně určena Definition of Done. Obsahová analýza Scrum tabule nepotvrdila

předpoklad, že tým bude používat nevhodně rozloženou Scrum tabuli. Naopak ale potvrdila předpoklad, že se v týmu objeví velká rozpracovanost úkolů.

### **Agilní programování**

Rozhovory v první fázi potvrdily předpoklad, že se zahájením pilotního projektu nebudou zahájeny agilní metody programování. Během sledovaného období však došlo k určité změně a dle rozhovorů v druhé fázi se alespoň některé z praktik podařilo uvést do praxe. Stále ale v této oblasti zůstává prostor pro zavedení dalších metod a technik.

### **Agilní smlouva**

Obsahová analýza i rozhovor s Product Ownerem potvrdily předpoklad, že mezi firmou XY a jejím klientem je sepsána smlouva, která neodpovídá agilnímu způsobu spolupráce.

## **3.3 Doporučení**

Níže navržená doporučení jsou rozdělena na oblasti, u kterých se na pilotním projektu dle předpokladů potvrdily určité chyby nebo nedostatky u implementace metody Scrum. Uvedená doporučení se opírají převážně o teoretické části této diplomové práce, konkrétně kapitoly: Metoda Scrum, Metoda Kanban a Implementace agilní metody Scrum. Doporučení jsou určena pro celý tým, protože práce ve Scrumu je primárně o spolupráci členů týmu, nikoliv o práci jednotlivců. U doporučení je pouze pár výjimek, kde je uvedeno, že dané doporučení se týká přímo Scrum Mastera, konkrétního člena týmu nebo případně Product Ownera.

### **3.3.1 Příprava na zahájení agilní transformace a sestavování týmu**

Protože se během výzkumu potvrdil předpoklad, že vývojáři nebudou mít dostatečnou přípravu na zahájení agilní transformace, jsou zde uvedena doporučení, jak by měl Scrum Master dále postupovat, aby tuto chybu korigoval. Tato doporučení lze využít pro nové členy týmu, ale i pro stávající vývojáře. Dále se potvrdily předpoklady, že tým nebude vhodně sestaven z pohledu zkušeností a znalostí, ale také z pohledu přístupu k agilnímu způsobu vývoje. Doporučení se tedy týkají těchto oblastí a popisují způsoby, jak by se Scrum Master mohl těmto chybám do budoucna vyvarovat.

### **Příprava vývojářů na agilní transformaci a metodu Scrum**

Zejména u pilotního projektu, jehož prostřednictvím dochází k agilní transformaci, je důležité, aby všichni vývojáři byli proškoleni o agilních postupech a metodě Scrum. A to i v případech, že vývojáři mají teoretické nebo praktické znalosti o agilním přístupu. Je potřeba, aby informace, které tým bude mít, byly shodné, a bylo možné ověřit, že jim všichni správně rozumí. Scrum Master může samozřejmě využívat pro školení vývojářů externí zdroje. Tato školení jsou veřejně dostupná a jsou vedena zkušenými agilními kouči. Vzhledem k tomu, že firma XY chtěla agilní transformaci realizovat pouze v rámci vlastních zdrojů, tak se doporučení týká nástroje, který může Scrum Master používat v rámci svého týmu. Scrum Master může pro svůj tým využívat agilní nástroj **Coaching Backlog** (Haslinger a Wintersteiger, 2017), pomocí kterého vytvoří plán obsahující položky pro konkrétní školení, koučování nebo poradenství. Tento plán by měl být pouze krátkodobý a měl by se stále přizpůsobovat aktuální situaci v týmu i potřebám vývojářů. Jednotlivá školení a poradenství se mají týkat oblastí, ve kterých potřebují členové týmu doplnit znalosti. Návrh Coaching Backlogu na jeden měsíc pro zkoumaný tým je přiložen v Příloze 7 a obsahuje oblasti, ve kterých má současný tým rezervy, nebo témata, kde by současný tým potřeboval doplnit potřebné znalosti, aby je pak následně mohl využít v praxi. Je zde také uvedeno školení pro



potenciálního nového člena týmu. Témata v Coaching Backlogu jsou seřazena dle priorit. U témat jsou uvedeny také cíle, které by měla školení pomoci naplnit. První čtyři uvedená školení v Coaching Backlogu by měl Scrum Master realizovat se svým týmem během jednoho měsíce a následně tento plán aktualizovat dle aktuálních potřeb týmu.

## Sestavování týmu

Při sestavování týmu je vždy potřeba ověřit, zda jednotliví členové mají potřebné zkušenosti a znalosti pro vývoj daného produktu. Je pochopitelné, že ne vždy se daří sestavit tým, kde všichni členové mají všechny požadované zkušenosti a znalosti na takové úrovni, aby mohli vždy samostatně pracovat na jednotlivých úkolech. Metoda Scrum je ale převážně o spolupráci lidí v týmu, takže případné nedostatky v této oblasti by měli členové týmu řešit společně. Agilní přístup k tomu nabízí také několik metod, které lze využívat k zaškolení a zdokonalování u členů týmů, kteří v dané oblasti tolik zkušeností nemají. U zkoumaného týmu se tato problematika týká junior programátora, kterému pro určité úkoly chybí dostatek znalostí a zkušeností. Jeho kolegové by neměli nechat tohoto programátora, aby se s těmito problémy potýkal sám jako doposud. Naopak by mu měli pomoci se v daných oblastech rozvíjet. Tento progres u jejich kolegy jim následně v dalším období usnadní a urychlí vývoj a také pomůže odstranit některé chyby. V části této kapitoly, která se týká doporučení pro Agilní programování, jsou uvedeny metody, které může tým s junior programátorem využívat. Jedná se o Pair Programming (párové programování) a Code Review (kontrola kódu). Pokud bude do týmu zapojen nový člen, je potřeba vždy ověřit jeho úroveň znalostí a posoudit, zda je pro danou práci dostatečná. Vždy je rovněž potřeba zvážit, zda má tým prostor a možnosti se novému kolegovi věnovat a pomoci mu s jeho rozvojem.

Členové Development týmu kromě svých technických znalostí a dovedností potřebují také další schopnosti. Například schopnost sebeorganizace a spolupráce s ostatními lidmi v týmu, protože dobře fungující tým musí být schopen spolupracovat. Vzhledem k tomu, že ve zkoumaném týmu dochází z pohledu přístupu k agilnímu stylu vývoje ke konfliktům s jedním vývojářem, měl by Scrum Master začít tuto situaci neodkladně řešit, protože disfunkčnost týmu se může prohlubovat a bude tým omezovat v dosahování lepších výsledků. Je důležité, aby si Scrum Master i daný vývojář uvědomili, čeho chtějí dosáhnout, a zda jsou schopni tuto situaci společně řešit. Scrum Master by se měl pokusit odhalit příčiny negativního přístupu tohoto vývojáře a následně se rozhodnout, zda existuje možnost, jak tuto situaci v týmu vyřešit jinak než odchodem pracovníka z týmu. Pokud totiž ve Scrum týmu nefunguje správně jeden z vývojářů, nefunguje správně celý tým. Klíčem může být řešení příčin této disfunkce dle pyramidového konceptu, který popisuje 5 příčin selhávání týmů (Šochová, 2018a, s. 95–96):

- **Nedostatek důvěry:** člen týmu pracuje sám za sebe a omezuje vazby s okolím, považuje se za specialistu na určitou oblast a snaží se udržet svou pozici;
- **Obava z konfliktu:** vyhýbání se konfliktu, udržování umělé harmonie, rozdělování práce do vzájemně nesouvisejících oblastí;
- **Neochota k závazku:** častý problém na začátku agilní transformace, jehož příčiny se nacházejí v nedostatku důvěry;
- **Vyhýbání se zodpovědnosti:** chybí zodpovědnost za chyby;
- **Nezájem o výsledek:** stanovování individuálních cílů místo jednoho společného cíle.

V Příloze 8 jsou uvedeny návrhy pro řešení dle těchto úrovní disfunkcí, které může Scrum Master ve svém týmu využít. Vzhledem k tomu, že se jedná o pyramidový koncept, je vždy potřeba řešit problém od spodní části pyramidy (základny) a teprve potom se posunout o úroveň výše. V návrhu jsou u každé úrovně pyramidy uvedeny oblasti, na které je potřeba se

zaměřit (návrhy) a také konkrétní příklady (praktické použití), které může Scrum Master aplikovat u svého týmu. Odstranění těchto příčin problémů vyžaduje změnu paradigmatu daného člověka směrem k agilnímu způsobu myšlení. Scrum Master by se na tyto oblasti měl zaměřit u celého týmu, ale nyní zejména u daného vývojáře, a diskutovat o těchto tématech a příčinách problémů s jednotlivci i s celým týmem. Dále by měl podporovat vzájemnou důvěru, konstruktivní konflikt, nechat tým podílet se na plánech i rozhodování a nechat ho být zodpovědným za jeho výsledky (Šochová, Kunce, 2019, s. 161). Doporučení v následující podkapitole na tuto problematiku dále navazují z pohledu Tuckmanova modelu a doplňují informace o tom, jak by měl Scrum Master měnit svůj přístup vzhledem k aktuální vývojové fázi týmu.

### 3.3.2 Role a vývoj týmu

Dle očekávání se během výzkumu v daném týmu začaly vyskytovat problémy s adaptací na nové role u některých vývojářů a začalo docházet ke konfliktům. Pro problémy jednotlivých vývojářů jsou navržena doporučení v části 3.3.1 Příprava na zahájení agilní transformace a sestavování týmu a také 3.3.5 Agilní programování. Na úrovni týmu je vhodné přizpůsobit doporučení dle jeho aktuální vývojové fáze. Tým prošel ve sledovaném období z fáze Forming (formování) do fáze Storming (bouření). Scrum Master by měl mít za svůj cíl, aby se tým z těchto fází přesunul přes fázi Norming (normování) až do fáze Performing (optimální výkon). I když tým této fáze dosáhne, neznamená to, že v této fázi již bude fungovat trvale, protože jakákoliv změna tým ovlivňuje a může se stát, že i úspěšný tým se propadne zpět do některé z fází nižších. Změnou může být například odchod člena týmu či příchod nového člena. Scrum Master musí svůj styl vedení přizpůsobovat těmto fázím, aby mohl tým vhodně provést do další fáze.

Tým se v současné době nachází ve fázi **Storming** (bouření) a v této fázi by se měl Scrum Master zaměřit na několik základních oblastí:

- **Komunikace:** povzbuzování jednotlivých členů týmu ke komunikaci;
- **Spolupráce:** s pomocí vývojářů nalézt vhodný způsob spolupráce, aby fungovali jako tým;
- **Konflikty:** konflikty aktivně řešit a pomáhat týmu nacházet řešení;
- **Odpovědnost:** přenechat již část odpovědnosti na samotný tým, delegovat úkoly.

V Příloze 9 je uvedeno jaké konkrétní aktivity může v těchto výše uvedených oblastech Scrum Master začít realizovat a na co se má u svého týmu zaměřovat. V této fázi je stále potřeba, aby se Scrum Master podílel na agilních procesech a praktikách Scrumu a také aby ověřoval, zda členové týmu již pochopili jednotlivé praktiky i principy a jsou schopni naplňovat společný cíl. Po vyřešení zásadních problémů a konfliktů by se zkoumaný tým měl přesunout do další fáze, kterou je **Norming** (normování). V této fázi by měl Scrum Master svou roli vnímat především jako facilitátor a kouč, který podporuje tým k větší odpovědnosti a hledáním slabých stránek vede tým k neustálému zlepšování. V této fázi je také vhodné, aby Scrum Master zapojil širší okruh zúčastněných stran – zákazníky nebo další členy v organizaci – a umožnil tak týmu větší socializaci a podpořil činnost týmu v rámci celé organizace. Může se stát, že i z této fáze se tým vrátí zpět do fáze **Forming** (formování), protože v týmu dojde ke změně, která naruší již zvládnuté postupy a principy. Pokud se tak stane, musí Scrum Master v této situaci opět přizpůsobit svůj styl vedení a v případě nutnosti se vrátit i k vysvětlování základních principů metody Scrum, podporovat pozitivní přístup týmu a spolupráci s dalšími stranami. Naopak pokud se týmu podaří dosáhnout žádoucí fáze **Performing** (optimální výkon), role Scrum Mastera má ustoupit do pozadí, ale přesto zůstane stále potřebná a důležitá. V této fázi by musel Scrum Master zajišťovat vhodné podmínky pro

práci týmu a dohlížet na to, aby tým v této optimální podobě fungoval a nevrátil se do některé z předchozích fází. Úkolem Scrum Mastera v této fázi je také udržovat a podporovat samostatnost týmu i rozvoj a vztahy jednotlivých členů. K tomu, aby Scrum Master ověřil, v jaké fázi se tým nachází, může opět po určitém období (3 měsíce) nebo při nastalé změně použít hodnocení dle Tuckmanova modelu. To znamená, že jednotliví členové týmu provedou opět hodnocení, ve které fázi se tým právě nachází. Na vytištěné schéma, viz Příloha 10, přiřadí každý člen týmu značku do části kruhu, který náleží fázi vývoje, ve které se tým dle jeho názoru nachází. Scrum Master následně body sečte a vyhodnotí aktuální situaci a zda došlo ke změně.

### 3.3.3 Meetingy a aktivity

Při výzkumu se u Standup meetingů potvrdily předpoklady o časově náročných schůzkách a jejich komplikovaném průběhu. Během výzkumu sice došlo k určitému zlepšení a tým si s jistými problémy dokázal poradit, ale délka této schůzky a dochvilnost vývojářů má stále ještě rezervy. U retrospektivní schůzky – Sprint Retrospective – se potvrdil předpoklad, že tým bude některé schůzky vynechávat. Níže uvedená doporučení se týkají těchto dvou typů schůzek a zaměřují se na uvedené problémy.

#### Standup (Daily Scrum)

V případě, že by tým dodržoval základní pravidla stanovená pro Standup meeting, mohla by tato schůzka pětičlenného Development týmu teoreticky trvat pouze pět minut. Jedním z těchto pravidel je formát, v jakém jednotliví členové hovoří o své práci. Jedná se o odpovědi na tři otázky. Tyto otázky ale členům týmu nikdo nepokládá, to by celý proces zbytečně zdržovalo. Vývojáři mají být s touto formou seznámeni a mají ji pak automaticky používat. Jedná se konkrétně o tyto tři otázky (Šochová, Kunc, 2019, s. 107):

- **Co jsem dokončil včera, co pomohlo týmu dosáhnout cíle sprintu?**
- **Co dokončím dnes, co pomůže týmu dosáhnout cíle sprintu?**
- **Existují problémy, které nám stojí v cestě k cíli sprintu?**

Než si všichni členové týmu zvyknou na tento formát a budou se ho automaticky držet, je vhodné, aby byly tyto otázky na viditelném místě přímo při Standupu. Scrum Master může vytisknout Přílohu 11 a toto schéma umístit v místnosti, kde schůzky probíhají, a v případě potřeby členům týmu pouze připomenout a ukázat, čeho se mají držet. Je důležité, aby odpovědi na tyto otázky byly maximálně stručné. Informace o tom, co bylo uděláno, by měla být ze všeho nejstručnější. Vývojář může například na tabuli pouze ukázat, který úkol předešlý den zpracoval, nemusí k němu doplňovat žádné další informace, pokud nejsou užitečné pro jeho kolegy a nemají dopad na splnění cíle Sprintu. Stejným způsobem je možné informovat o tom, co bude vývojář realizovat během aktuálního dne. U třetí odpovědi má vývojář informovat o tom, zda nenastala nějaká skutečnost, která brání v tom, aby bylo dosaženo cíle Sprintu nebo tento cíl jakkoliv ohrožuje. Touto formou je možné, aby každý z vývojářů během jedné minuty předal potřebné informace svým kolegům. Pokud je potřeba konzultovat cokoli dalšího, může si každý z vývojářů domluvit setkání, které může následovat ihned po dokončení Standupu s vybranými kolegy. Ale není žádoucí, aby konkrétní technické záležitosti byly řešeny s celým týmem. Dalším pravidlem, které urychluje průběh schůzky, je také to, že jednotliví členové týmu nesedí na židlích, ale zůstanou po celou dobu schůzky stát. Třetí pravidlo, které by měl tým dodržovat a mělo by postupně eliminovat nedochvilnost vývojářů, se týká začátku schůzky. Standup má začít vždy ve stejnou dobu bez ohledu na to, že některý z členů týmu ještě nedorazil. Tým by neměl na žádného člena týmu

čekat a schůzka by měla být zahájena. Tento přístup vývojáře motivuje, aby byli na schůzce včas a měli tak potřebné informace od svých kolegů.

### **Sprint Retrospective**

Retrospektivní schůzky jsou potřebné proto, aby se tým mohl zlepšovat a adaptovat se na aktuální situaci. Členové Development týmu potřebují mít prostor, kde budou moci sdílet své pozitivní zkušenosti i to, co jim komplikuje práci a chtějí by to změnit. Sami vývojáři by měli tuto schůzku vyžadovat a je jen na nich, jaká metoda nebo postupy jim během této schůzky budou nejvíce vyhovovat. Doporučit lze níže uvedené praktiky (Šochová, Kunc, 2019, s. 113–116):

- **Retrospektiva s lepítky:** používá se nakreslená hvězdice, která je rozdělena na pět částí (začít, víc, pokračovat, méně, přestat), do kterých vývojáři přilepují jednotlivé kartičky s tématy, o kterých chtějí hovořit, a následně o nich hlasují;
- **Časová osa:** již v průběhu Sprintu jsou průběžně zaznamenávány oblasti, které chtějí členové týmu řešit, na časovou osu nakreslenou na zdi a jednotliví vývojáři jim přidělují body, pokud s nimi souhlasí;
- **Fishbone (rybí kost):** efektivní metoda pro nalezení příčiny problému, používají se otázky – co, kde, kdo, kdy a proč;
- **Lod':** analogie časové osy, ale používá obrázek lodě, kde kotvy symbolizují problémy, které tým brzdí, plachty jsou symbolem úspěšných oblastí, jež tým pozitivně motivují;
- **ESVP:** zkratka 4 rolí, do kterých se sami členové rozdělí dle toho, jak se cítí – Explorer, Shopper, Vacationer, Prisoner (průzkumník, nakupující, rekreant, vězeň).

S využitím první výše uvedené praktiky Retrospektiva s lepítky jsou v Příloze 12 zaznamenány oblasti, které jednak samotní vývojáři zmiňovali při rozhovorech, a dále také zjištěné chyby a nedostatky během výzkumu. Tato metoda rozděluje témata, kterými by se tým měl zabývat do pěti skupin dle aktivity, která by v dané oblasti měla následovat: **začít, víc, pokračovat, méně a přestat**. Protože se tým dosud téměř nebo vůbec nezabýval oblastmi jako je – Definition of Done, kontrola kódu, párové programování a praktiky pro retrospektivní schůzky – byla pro aktivitu „**začít**“, doporučena tato čtyři témata, která jsou důležitá z pohledu kvality, spolupráce a adaptace týmu. Naopak vynechávání retrospektivních schůzek, které jsou nedílnou součástí Scrumu, a odhady v hodinách byly doporučeny pro aktivitu „**přestat**“ dělat. Tým by měl „**víc**“ pracovat na školení o Scrumu, Unit testech a na dosahování cíle Sprintů, protože v těchto oblastech určité aktivity již realizuje, ale zároveň zde má ještě prostor pro zlepšování. Také je potřeba pracovat na tom, aby se vyskytovalo „**méně**“ chyb v produktu a konfliktů v týmu a naopak by tým měl dále „**pokračovat**“ v prezentaci svých výsledků při Review schůzkách. Tento návrh může tým použít pro další retrospektivní schůzku a vybrat si z něj témata, která chce aktuálně nejvíce řešit. Tato metoda i ostatní výše uvedené pomáhají týmu identifikovat problémy, nalézat jejich příčiny. Tým však také musí aktivně pracovat na nalezení řešení a jeho uvedení do praxe. Výhradním úkolem Scrum Mastera je zajistit, že se tyto schůzky budou konat pravidelně po ukončení Sprintu a nebudou rušeny. Je potřeba si uvědomit, že tyto schůzky mohou mít pro tým velký přínos. Jejich zrušení z časových důvodů není vhodné, protože právě na této schůzce mohou být řešeny příčiny, proč tým nestihl splnit cíl Sprintu nebo proč má některý člen týmu problémy zvládat svou práci. Čas pro tuto schůzku je potřeba zahrnout do plánu Sprintu a je vhodné stanovit si pevný termín, kdy se schůzka bude pravidelně konat, aby s ní všichni členové týmu počítali.

### 3.3.4 Praktiky a artefakty

Vzhledem k tomu, že se během výzkumu ve dvou případech ze čtyř potvrdil předpoklad, že nebude dosahováno cíle Sprintů, níže popsaná doporučení se týkají praktik, které by měly danému týmu pomoci zvýšit šance na dosahování těchto cílů. Jednou z těchto praktik je zároveň doporučení týkající se rozpracovanosti během Sprintu, protože se během výzkumu potvrdil předpoklad o vyšší rozpracovanosti úkolů u některých členů týmu. Dále byl také potvrzen předpoklad, že nebude jasně určena Definition of Done. V tomto případě tato definice nebyla formálně vyjádřena vůbec, proto níže uvedená doporučení obsahují informace o tom, co zahrnout do základní definice dle aktuální situace na pilotním projektu.

#### Sprint Goal (cíl Sprintu)

Zvolený cíl pro každý Sprint by měl být nastaven realisticky, aby měl tým šanci ho dosáhnout, ale zároveň by měl členy týmu dostatečně motivovat. Cíl také nemá vyjadřovat funkcionalitu jako takovou, ale přímo hodnotu, kterou přinese klientovi. Jako příklady lze uvést tyto správně formulované cíle Sprintů: „zvýšit přehled o financích“, „zrychlit odezvu systému“ nebo „zvýšit počet vracejících se zákazníků“ (Šochová, Kunc, 2019, s. 61). Je důležité, aby každý člen týmu formulaci cíle porozuměl a tým pak byl následně schopen navrhnout správné řešení k jeho naplnění. Vzhledem k tomu, že daný tým nedosáhl dvou cílů Sprintu v jenom případě z důvodu chybných odhadů a v druhém případě kvůli zvýšenému výskytu chyb, níže uvedená doporučení se zaměřují na oblasti, jejichž zlepšení by mělo tyto problémy zmírnit nebo odstranit. Pozitivní dopad na plnění cílů Sprintů mají také doporučení uvedená v podkapitole **3.3.5 Agilní programování**, protože zde doporučené praktiky mají vliv jak na zvýšení rychlosti vývoje, tak umožňují včasné zachycení chyb a jejich rychlejší opravy. Pro zvýšení úspěšnosti při dosahování cílů Sprintů jsou níže uvedena doporučení, která se týkají tří oblastí: **metod odhadů, Standup schůzky a Work in Progress Limits (omezení probíhající práce)**.

#### Metody odhadů

Tým po celou dobu výzkumu prováděl odhady časové náročnosti v hodinách a nevyzkoušel žádnou z doporučovaných metod pro odhady při agilním způsobu vývoje. Přestože metoda Scrum nevyžaduje žádnou konkrétní metodu pro odhady náročnosti, existuje několik doporučovaných metod a některé z nich by mohly potřebám týmu vyhovovat lépe. Důležité je, aby odhady prováděl vždy celý tým společně. Danému týmu lze doporučit, aby pro přechod z odhadů v hodinách na odhady v relativních jednotkách využil metodu **T-Shirt Sizes**, při které jsou odhady prováděny pomocí velikostí triček S, M, L, XL. Tato metoda je jednoduchá, rychlá a umožní týmu odhadovat velké množství položek v Backlogu. T-Shirt Sizes metoda by měla být využívána zejména pro počáteční odhady velkých položek (Epic), aby si tým vytvořil rámcovou představu o pracnosti. Na tyto rámcové odhady pak může tým navázat s další metodou, která umožní zpřesnit náročnost dané položky (User Story). Touto metodou je **Planning Poker**, která pro odhady obvykle využívá stupnici: 0, ½, 1, 3, 5, 8, 13, 20, 40, 100, ?, nekonečno a kartu s obrázkem kávy. Jak používat tyto karty a jak správně postupovat při odhadech je popsáno v teoretické části práce Praktiky a artefakty ve Scrumu v části Ohodnocení a jejich techniky. Tým musí počítat s tím, že tato technika odhadů je náročnější než výše uvedená metoda T-Shirt Sizes. Proto nelze očekávat přínos a vyšší efektivitu na samém počátku zavedení této metody, tým bude muset nejdříve získat praktické zkušenosti a zdokonalit se v této praxi. Následně pak kvalitnější odhady pracnosti pomohou v konečném důsledku i k dosahování cílů Sprintů.

## Standup schůzky

Během Standup schůzky je potřeba každý den ověřit, že se tým skutečně přibližuje k dosažení cíle Sprintu. Tato pravidelná kontrola umožní co nejdříve zachytit potenciálně vznikající problémy. V případě zjištěných překážek se tým musí dohodnout na korekcích a upravit svou činnost tak, aby maximalizoval šanci na dosažení cíle Sprintu. Na toto pravidelné zhodnocení stavu během Standup schůzky by měl minimálně ze začátku dohlédnout Scrum Master a také on by měl následně pomoci tým koordinovat k aktivitám, které povedou k nápravě a k dosažení cíle Sprintu. Jak se bude tým postupně stávat samostatnějším, je potřeba upozornit, že obsah a průběh Standup schůzek je čistě na zodpovědnosti Development týmu. Tuto kontrolu lze provádět pomocí revize odhadů, které jsou uvedeny u jednotlivých úkolů, protože během sprintu dochází ke změnám. Tyto změny nemusí nutně znamenat, že se odhady postupně snižují tím, že je úkol zpracováván, ale vzhledem k zjištěným změnám může dojít i k dodatečnému zvýšení odhadu. To jsou informace, které by měly být týmu známy co nejdříve, protože mohou mít dopad na cíl Sprintu. Každý člen týmu by měl být v tomto směru proaktivní a měl by tyto údaje udržovat aktuální u všech svých úkolů.

## Work in Progress Limits (omezení probíhající práce)

Dalším doporučením pro zlepšení dosahování cílů Sprintu je nastavení omezení pro právě probíhající práci neboli Work in Progress Limits (WIP). Toto omezení snižuje množství „téměř hotové“ práce a umožňuje se soustředit na menší množství úkolů bez přepínání kontextu, které zbytečně ubírá pracovníkům čas. Dále také tyto limity pomohou identifikovat úzká místa a blokátoři práce. Tým tak může tyto problematické oblasti a procesy začít ihned řešit a zajistit plynutí průběhu práce i rychlejší dodání výsledku klientovi. Danému týmu lze doporučit, aby si nastavil tyto limity pro jednotlivé sloupce na Scrum tabuli. Toto omezení se týká zejména sloupce IN PROGRESS, který obsahuje právě zpracováváné úkoly. V tomto sloupci by se mělo nacházet přesně tolik úkolů, kolik je v týmu vývojářů. V daném případě by tedy zkoumaný tým měl mít v tomto sloupci stále 5 úkolů, z nichž každý je přidělen jednomu z vývojářů. Méně úkolů v tomto sloupci znamená, že některý z vývojářů aktuálně nepracuje na žádném z úkolů. Naopak více úkolů značí, že někteří vývojáři ještě nedokončili předchozí úkol, ale již započali další práci. Zkoumaný tým využívá pro správu úkolů software JIRA, který umožňuje nastavení limitů na jednotlivých sloupcích Scrum tabule a automaticky tak dohlíží na tento proces. Tento software ale zatím neumožňuje kontrolovat počet úkolů na úrovni jednotlivých vývojářů. Tato funkcionality je ale plánována a má být do správy elektronických tabulí doplněna (JIRA, 2020). Scrum Master a členové Development týmu musí tedy tuto kontrolu zatím provádět manuálně.

## **Definition of Done**

Protože během zkoumaného období nebyla oficiálně formulována ani základní Definition of Done, doporučení uvádí základní oblasti, které by tato definice měla obsahovat s přihlédnutím na aktuální situaci v daném týmu, protože náročnost definice by měla odpovídat tomu, co je daný tým skutečně schopen splnit. Tento standard není neměnný, je vhodné, aby ho tým rozšiřoval společně s tím, jak bude získávat více zkušeností nebo zavádět praktiky agilního programování, protože tím bude zvyšovat kvalitu produktu. Definition of Done by měl Scrum Master s týmem definovat na úrovni User Stories, které pak budou muset splňovat všechna vymezená kritéria, aby daná User Story mohla být označena za dokončenou (Done). S ohledem na současnou situaci v týmu lze doporučit, aby do Definition of Done byla zahrnuta následující čtyři kritéria:

- funkcionální testy aplikace jsou v pořádku;
- část kódu, která je pokryta Unit testy, je dle výsledků těchto testů v pořádku;

- část kódu, která je pro funkcionalitu aplikace kritická, prošla Code Review;
- Product Owner akceptoval danou User Story.

Funkcionální testy tým provádí v rámci každého Sprintu, takže není nijak problematické je do definice zařadit. Zároveň tým již začal v průběhu výzkumu připravovat Unit testy, takže i ty mohou být součástí standardu, i když zatím ne jako 100% pokrytí kódu aplikace. V následující podkapitole Agilní programování je jednou z doporučených praktik Code Review, kterou by měl tým zavést a v rámci Definition of Done ji vyžadovat alespoň u kritických částí aplikace. Nezbytnou částí definice je samotná akceptace Product Ownerem. Takto sestavená Definition of Done může být rovněž součástí agilní smlouvy mezi firmou XY a jejím klientem. Na tuto základní Definition of Done může tým později navázat postupným zpřísňováním kritérií nebo jejich rozšířením o další pravidla.

### 3.3.5 Agilní programování

Na začátku výzkumu se potvrdil stanovený předpoklad a tým nezačal využívat agilní metody programování. Ale v průběhu výzkumu bylo zjištěno, že Scrum tým již začal s implementací alespoň některých agilních praktik. Následující doporučení se proto týkají dalších metod, které by pro tento tým byly vhodné. Níže uvedené metody jsou doporučeny s ohledem na konkrétní situaci v Development týmu a vyskytující se problémy při vývoji, protože tyto praktiky by měly pomoci při odstraňování chyb nebo by jim měly pomoci předcházet.

#### Pair Programming (párové programování)

První doporučenou technikou pro Development tým je Pair Programming, při kterém dva programátoři pracují společně u jednoho počítače. Jeden z programátorů píše samotný kód a druhý programátor přichází s nápady, návrhy na vylepšení a odstranění chyb. Tato technika je vhodná tam, kde je potřeba rychle zaučit nového člena týmu, vyřešit složitý problém nebo velmi rychle posílit týmovou spolupráci (Šochové a Kunc, 2019, s. 127). A toto jsou také oblasti, které by zkoumaný tým potřeboval řešit, aby mohl lépe fungovat a dosahovat lepších výsledků. Zmíněná technika by měla být uplatněna zejména u současného junior programátora, jehož znalosti a zkušenosti nyní nedostačují na plnění některých úkolů. V týmu jsou zkušení programátoři, kteří by během párového programování mohli svému kolegovi předávat své zkušenosti a rychleji tak posouvat jeho úroveň a znalosti. Zároveň by se tímto způsobem předešlo vzniku některých chyb, které jsou nyní odhalovány až při testování, a při jednom ze Sprintů zabránily i v dosažení naplánovaného cíle. Dalším důvodem pro zavedení této techniky v daném týmu je fakt, že pravděpodobně bude muset být do týmu přijat nový kolega. Ať už se bude jednat o náhradu jednoho z vývojářů, pokud se situaci nepodaří vyřešit jinak, nebo samotné posílení týmu, o kterém hovořil Product Owner při rozhovoru. Párové programování by pak bylo v tomto případě vhodné pro rychlejší zaučení nového člena týmu do již rozběhnutého projektu. V týmu se také během výzkumu vyskytla situace, kdy se tým potýkal s náročnějším vývojem funkcionality a nestihl splnit cíl Sprintu. I v takové situaci může být metoda párového programování pro některé vývojáře užitečná, protože urychlí jak řešení problematických situací, tak i samotný vývoj. Párové programování může tým zavést jen pro určité typy případů (zaučení kolegy, složitý problém) nebo z této praktiky udělat součást vývojového procesu a využívat tuto techniku trvale.

#### Code Review (kontrola kódu)

Pokud budou členové týmu využívat párové programování, tak při něm budou zároveň provádět kontrolu kódu – Code Review, což je další doporučená praktika agilního programování pro zkoumaný Development tým. Kontrolu kódu je ale vhodné provádět

i samostatně bez párového programování. Při Code Review kontroluje jiný vývojář práci svého kolegy, a pokud má výhrady, sdělí je autorovi kódu. Tuto kontrolu může provádět jakýkoliv člen Development týmu, ale je nutné počítat s tím, že například junior programátorovi je nutné nejdříve vysvětlit, jak daný kód funguje (Šochová, Kunc, 2019, s. 128). Zkoumaný tým tak může tuto metodu využít pro kontrolu kódu a odstranění chyb, ale i pro rozšíření znalostí mezi programátory v týmu, protože při kontrole získává programátor nové znalosti o fungování kódu v částech aplikace, které sám nepsal. Primárně by ale měl být takto kontrolován kód u současného junior programátora nebo také u kritických funkcionalit aplikace, jak je doporučeno i pro Definition of Done.

### 3.3.6 Agilní smlouva

Vzhledem k tomu, že byl potvrzen předpoklad, že současná smlouva mezi firmou XY a klientem v žádné části neodpovídá agilní smlouvě, nelze než doporučit, aby firma se svým klientem v této oblasti zahájila jednání a pokusila se pro další období uzavřít smlouvu, která již bude odpovídat agilnímu typu spolupráce. Aby mohla firma XY o této smlouvě s klientem začít vyjednávat, byl v rámci doporučení připraven vzor agilní smlouvy, na kterém může firma svému klientovi vysvětlit základní pravidla a části tohoto typu kontraktu. Cílem tohoto doporučení nebylo připravit finální smlouvu pro firmu XY a jejího klienta, ta musí být sestavena až za účasti obou stran. Smyslem doporučení je usnadnit firmě XY zahájení nového jednání o agilní smlouvě s klientem, který tento typ smlouvy zatím odmítal. Uvedená vzorová smlouva je součástí této diplomové práce jako Příloha 13. V této vzorové smlouvě jsou z pohledu agilního přístupu klíčové zejména níže popsané části.

#### Vzorová agilní smlouva

V rámci **Úvodního ustanovení** agilní smlouvy je potřeba popsat základní atributy aplikace (ve vzorové smlouvě označené jako Software). Tento popis nemá obsahovat žádné detailní výstupy, ale pouze základní znaky dané aplikace. Dále jsou v části Úvodního ustanovení smlouvy již uvedeny informace o Sprintech, které jsou zde definovány jako dílčí smlouvy o dílo, ve kterých probíhají definování požadavků, analýza, vývoj nových funkcionalit aplikace a uvedení do provozu. V části **Předmět smlouvy** jsou ve vzorové smlouvě uvedeny náležitosti, které vymezují povinné atributy definující každý Sprint: obsahová definice Sprintu (v podstatě se jedná o cíl Sprintu), odhad časové náročnosti, termín realizace a informace o dokumentaci. Do této části je také vhodné zahrnout Definition of Done, která by měla být formulována před zahájením prvního Sprintu. Volitelně je možné definovat Sprint také hodinovou odměnou (pokud se liší od částky dohodnuté ve smlouvě) nebo maximální částkou odměny. Část o **Uzavření dílčí smlouvy** musí obsahovat pověřené osoby za obě smluvní strany, které budou mít oprávnění jednotlivé Sprints definovat. **Předání díla** ve smlouvě stanovuje podmínky pro dodání obsahu Sprintu: poskytnutí vzájemné součinnosti, předvedení díla (Review schůzka) a lhůty pro odzkoušení a případné opravy chyb. Agilní smlouvou je definována také **Odměna** za dokončení jednoho Sprintu. Tato paušální částka zahrnuje odměnu za veškerou práci během jednoho Sprintu. Je doporučováno za každého člena týmu účtovat 35 hodin týdně, protože zbývající pracovní čas je ponechán na interní režii pracovníka. Agilní smlouva obvykle neobsahuje fixní částku za daný projekt, protože klient se může kdykoliv rozhodnout, že je daná funkcionalita již dostačující a může vývoj ukončit. Přesto je ale možné, pokud to bude jedna ze stran vyžadovat, do agilní smlouvy doplnit cílovou částku za projekt. Nepřekročení této částky je pak řízeno pomocí prioritizace položek v Product Backlogu, kdy jsou přednostně zpracovávány položky s nejvyšší prioritou. Pokud je již cílová částka vyčerpána, jsou zbývající položky s nízkou prioritou odebrány, což umožní zachovat nezbytnou flexibilitu. **Odstoupení od smlouvy** definuje podmínky, za kterých



mohou jednotlivé strany odstoupit od uzavřené smlouvy. V případě agilní smlouvy je také možné doplnit podmínku pro případ předčasného ukončení projektu, která definuje, kolik Sprintů předem musí být toto ukončení druhé straně oznámeno. Dále agilní smlouva zahrnuje také běžné části, které se nacházejí i v klasických smlouvách, jako je Odpovědnost za škodu, Smluvní pokuta, Autorské právo a další.

Nová agilní smlouva mezi firmou XY a jejím klientem by měla zdůrazňovat obchodní hodnoty a měla by podpořit vzájemný vztah rovnocenného partnerství, proto by v žádném případě neměla zvýhodňovat jednu ze stran. Oba partneři by měli nést rizika i sdílet úspěch společně. Smlouva vhodně podporující agilní styl spolupráce má následně fungovat spíše na pozadí celého procesu a umožnit tak, aby se celý tým mohl zaměřit na samotnou práci a vývoj nového produktu. Product Owner i Scrum Master by měli usilovat o dosažení této změny a vysvětlit zainteresovaným stranám její přínos pro budoucí spolupráci.

## 4 Závěr

Výsledky výzkumu potvrdily několik stanovených předpokladů o chybách a nedostacích, se kterými se zkoumaný tým měl začít potýkat po zavedení metody Scrum na pilotním projektu. Jako první byl potvrzen předpoklad, že nebude věnována dostatečná pozornost přípravě členů Development týmu na agilní styl vývoje a metodu Scrum. U junior programátora byl prokázán předpoklad, že tým nebude vhodně sestaven z pohledu potřebných zkušeností a znalostí. Dále byl také u jednoho ze členů Development týmu ověřen předpoklad, že tým bude nevhodně sestaven z pohledu přístupu k agilnímu způsobu vývoje, což následně vedlo i k potvrzení předpokladu, že bude docházet ke konfliktům ve formující se týmu. U těchto dvou uvedených vývojářů byly také prokázány problémy s adaptací na nové role ve Scrum týmu. Na počátku výzkumu byl také potvrzen předpoklad, že schůzky týmu budou časově náročné a jejich průběh bude komplikovaný. Další fáze výzkumu prokázala určitá zlepšení v této oblasti, ale zároveň potvrdila předpoklad, že tým bude opomíjet některé typy aktivit stanovené metodou Scrum. V průběhu výzkumu byl předpoklad, že nebude dosahováno cílů Sprintů, ověřen pouze částečně – během čtyř Sprintů nebylo cíle dosaženo dvakrát. Naopak zcela se potvrdily předpoklady, že nebude jasně určena Definition of Done, i to, že se v týmu objeví velká rozpracovanost úkolů. První fáze výzkumu také prokázala předpoklad, že nebudou implementovány agilní praktiky programování, ale druhá fáze výzkumu již zaznamenala zavedení alespoň některých metod. Obě fáze výzkumu pak potvrdily, že mezi firmou XY a jejím klientem je stále uzavřena smlouva, která neodpovídá agilnímu způsobu spolupráce.

Naopak výsledky výzkumu neprokázaly předpoklad, že Scrum Master a Product Owner nebudou mít dostatečnou přípravu na zahájení agilní transformace. U Scrum Mastera a Product Owera se také nepotvrdil předpoklad, že bude docházet k problémům s adaptací na nové role. Rovněž předpoklady, že User Stories budou pro vývojáře nedostatečně popsané nebo nesrozumitelné nebo že tým bude používat nevhodně rozloženou Scrum tabuli, výzkum také nepotvrdil. V průběhu výzkumu se u vývojářů také neprokázala předpokládaná neochota účastnit se schůzek definovaných metodou Scrum.

Pro předpoklady, které byly během výzkumu na pilotním projektu pozitivně potvrzeny, byla navržena níže uvedená doporučení pro odstranění zjištěných chyb a nedostatků. Tato doporučení byla, vzhledem k povaze spolupráce u agilního vývoje a metody Scrum, určena pro Scrum tým jako celek. Některá doporučení pak byla primárně určena pro Scrum Mastera, Product Owera nebo konkrétního člena týmu.

Pro **přípravu vývojářů na agilní transformaci a metodu Scrum** byl doporučen agilní nástroj **Coaching Backlog**, který by měl Scrum Master využívat k vytvoření krátkodobého plánu pro školení, koučování nebo poradenství. Do Coaching Backlogu byla vybrána témata pro školení vývojářů, která by měl Scrum Master realizovat během jednoho měsíce: školení pro větší efektivitu Standup meetingů, metody pro odhady časové náročnosti (T-Shirt Sizes, Planning Poker), praktiky pro retrospektivní schůzky. Pro problematiku **sestavování týmu** z pohledu potřebných znalostí a zkušeností s ohledem na současného junior programátora bylo navrženo posílit spolupráci mezi tímto vývojářem a jeho zkušenějšími kolegy. Bylo doporučeno k tomu využít agilních praktik programování **Pair Programming** (párové programování) a **Code Review** (kontrola kódu). Pro nedostatky v oblasti sestavení týmu, z pohledu přístupu k agilnímu stylu vývoje, bylo navrženo zabývat se příčinami zjištěné disfunkce pomocí pyramidového konceptu, který prověřuje **5 příčin selhávání týmů**. Pro jednotlivé úrovně příčin byly uvedeny oblasti, ve kterých lze hledat řešení, včetně příkladů využití u zkoumaného týmu. Další doporučení na tuto problematiku navázalo v oblasti **rolí a vývoje týmu** z pohledu Tuckmanova modelu. Zde byl pro Scrum Mastera doporučen

přístup a zaměření se na vybrané oblasti dle vývojové fáze týmu, ve které se celý tým aktuálně nachází nebo nacházet pravděpodobně bude v budoucnosti. Pro současnou Storming fázi bylo Scrum Masterovi doporučeno několik aktivit v oblasti komunikace, spolupráce, konfliktů a odpovědnosti. Pro sledování budoucího vývoje bylo doporučeno opakovat hodnocení stavu vývoje týmu při změnách v týmu nebo po třech měsících.

Pro odstranění chyb a nedostatků u **meetingů a aktivit**, které stanovuje metoda Scrum, byla navržena doporučení pro **Standup** a **Sprint Retrospective** schůzky. Pro Standup schůzky bylo doporučeno několik drobných opatření, která mají vést k urychlení průběhu této schůzky. Jako základní opatření bylo navrženo dodržování správného formátu této aktivity pomocí tří základních otázek, které jako jediné mají být na této schůzce řešeny. Pro retrospektivní schůzky bylo Scrum Masterovi navrženo, aby tuto aktivitu pravidelně zajišťoval a také aby s týmem využíval jednu z pěti doporučených praktik: **retrospektiva s lepičky, časová osa, fishbone (rybí kost), lod' nebo ESVP**. Pomocí první z těchto metod byl sestaven seznam témat, kterými by se tým měl zabývat na retrospektivní schůzce.

Pro oblast **praktik a artefaktů** byla navržena doporučení týkající se **cíle Sprintu** a **Definition of Done**. Pro odstranění nedostatků, které zabránily v dosažení **cílů sprintů**, byla navržena doporučení týkající se **metody odhadů pracnosti, Standup schůzek a Work in Progress Limits (omezení probíhající práce)**. Pro odhady pracnosti bylo doporučeno využívat dvě metody – **T-Shirt Sizes a Planning Poker** – které by měly týmu pomoci s kvalitou a rychlostí odhadů. U Standup schůzek bylo v této souvislosti doporučeno provádět každodenní ověření, že se tým skutečně přibližuje k dosažení cíle pomocí pravidelné revize zbývajících odhadů času uvedeného u jednotlivých úkolů. Třetím doporučením pro dosahování cílů sprintů bylo nastavení omezení probíhající práce – **Work in Progress Limits** – které pomocí nastavení limitu na Scrum tabuli, kterou tým používá, může snížit rozpracovanost, urychlit dokončování jednotlivých činností a svou práci více zefektivnit. Pro chybějící **Definition od Done** byla doporučena její výchozí formulace obsahující čtyři základní položky, které vychází ze současné kvalitativní úrovně zkoumaného týmu tak, aby byl tým schopen tento standard naplňovat.

V rámci **Agilního programování** bylo zkoumanému týmu doporučeno začít využívat dvě další metody – **Pair Programming (párové programování)** a **Code Review (kontrola kódu)** – které mají pozitivní dopad na kvalitu, rychlost vývoje i sdílení znalostí v týmu. Tyto metody byly doporučeny s ohledem na zjištěné nedostatky v několika oblastech: plnění cílů Sprintů, výskyt chyb, neodpovídající znalosti a zkušenosti člena týmu. Tyto doporučené agilní praktiky programování zkoumanému týmu následně pomohou snížit množství chyb, rychleji dosahovat cílů Sprintů a rozvíjet potřebné znalosti nejen u junior programátora a jeho kolegů, ale v budoucnu i u nových členů týmu. Poslední oblastí, pro kterou bylo navrženo doporučení, byla **Agilní smlouva**. Zde bylo firmě XY doporučeno, aby se na další období pokusila s daným klientem uzavřít agilní typ kontraktu, který by nahradil současnou klasickou smlouvu. Na vzorové agilní smlouvě byl vysvětlen princip i důležité části tohoto typu smlouvy. Tyto informace by měly firmě XY usnadnit zahájení jednání s klientem o tomto novém typu kontraktu, který oficiálně podpoří současný typ spolupráce.

Na pilotním projektu se potvrdila řada předpokladů o chybách a nedostacích, které tým provázely v průběhu implementace agilní metody Scrum. Tento výsledek by ale neměl být vnímán jako negativní. Firma XY zahájila náročnou změnu, která s sebou nese nejen změny procesů, ale hlavně změny principů a přístupů na úrovni jedinců i celé organizace. Zkoumaný tým měl také v určitých oblastech zaznamenaný progres již během sledovaného období. Všechny zjištěné nedostatky a chyby je možné korigovat prostřednictvím uvedených doporučení, takže tým může dále pokračovat ve svém rozvoji a zlepšování. Závěrem lze

doplnit, že celkové výsledky pilotního projektu jsou i přes zjištěné nedostatky celkově pozitivní a rozhodnutí firmy XY o zahájení agilní transformace prostřednictvím pilotního projektu lze považovat za správné.

# Literatura

## Monografie

CARROLL, J. MORRIS, D. *Agile Project Management for Speedy Results!* 2. vydání. Leamington Spa Warwickshire, United Kingdom : In Easy Steps, 2017. ISBN 978-1-84078-641-5.

HENDL, J. *Kvalitativní výzkum: základní teorie, metody a aplikace*. Praha : Portál, 2016. ISBN 978-80-262-0982-9.

MORRIS, D. *Scrum an Ideal Framework for Agile Projects*. Leamington Spa Warwickshire, United Kingdom : In Easy Steps, 2017. ISBN 978-1-84078-731-3

MYSLÍN, J. *Scrum: průvodce agilním vývojem softwaru*. Brno : Computer Press, 2016. ISBN 978-80-251-467.

PAQUETTE, P., FRANKL, M. *Agile Project Management for Business Transformation Success*. New York : Business Expert Press, 2015. ISBN 978-1-63157-323-1

REICHEL, J. *Kapitoly metodologie sociálních výzkumů*. Praha : Grada Publishing, 2009. Sociologie. ISBN 978-80-247-3006-6.

SUTHERLAND, J. *Scrum: The Art of Doing Twice the Work in Half the Time*. London : Random House Business Books, 2014. ISBN 978-1-84794-110-7.

SVOZILOVÁ, Alena. *Projektový management: systémový přístup k řízení projektů*. 3., aktualizované a rozšířené vydání. Praha : Grada publishing, 2016. Expert (Grada). ISBN 978-80-271-0075-0.

ŠOCHOVÁ, Z. *Skvělý ScrumMaster*. Brno : Computer Press, 2018a. ISBN 978-80-251-4927-0.

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

WALDOCK, B. *Being Agile in Business: discover faster, smarter, leaner ways to succeed at work*. Harlow, United Kingdom : Pearson, 2015. ISBN 9781292083704.

## Internetové zdroje

ALFONSO, A. *Make Agile Contracts Work For Your Teams & Clients*. [online]. 2018 [cit. 13-1-2020]. Dostupné z WWW: <https://thedigitalprojectmanager.com/agile-contract/>

ALLES, M. *How to Estimate an Agile/Scrum Story Backlog with Points*. [online]. 2017 [cit. 17-12-2019]. Dostupné z WWW: <https://spin.atomicobject.com/2017/08/21/estimate-agile-scrum-points/>

ALSHAMRANI, A., BAHATTAB, A. A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model. *International Journal of Computer Science Issues (IJCSI)*, 12(1), 106-111. [online]. 2015 [cit. 17-11-2019]. Dostupné z WWW: <https://search.proquest.com/docview/1660801422>

ANDREI, B., at al. A STUDY ON USING WATERFALL AND AGILE METHODS IN SOFTWARE PROJECT MANAGEMENT. *Journal of Information Systems & Operations Management*, 125-135. [online]. 2019 [cit. 17-11-2019]. Dostupné z WWW: <https://search.proquest.com/docview/2237828314?accountid=159230>

ANURINA, O. *Agile SDLC: Skyrocketing Your Project with Agile Principles*. [online]. 2019 [cit. 27-12-2019]. Dostupné z WWW: <https://mlsdev.com/blog/agile-sdlc>

AZANHA, A., at al. *Agile project management with scrum*. *International Journal of Managing Projects in Business*, 10(1), 121-142. [online]. 2017 [cit. 30-12-2019]. Dostupné z WWW: <http://dx.doi.org/10.1108/IJMPB-06-2016-0054>

BATHIA, S. *PSD: Professional Scrum Developer Question Bank and Reference Guide*. [online]. 2019 [cit. 30-12-2019]. ISBN 978-17-326-5797-7 Dostupné z WWW: [https://play.google.com/store/books/details/Sidharth Bathia PSD Professional Scrum Developer Q?id=a4C3DwAAQBAJ](https://play.google.com/store/books/details/Sidharth+Bathia+PSD+Professional+Scrum+Developer+Q?id=a4C3DwAAQBAJ)

BASEER, K. K., at al. *A systematic survey on waterfall vs. agile vs. lean process paradigms*. *I-Manager's Journal on Software Engineering*, 9(3), 34-59. [online]. 2015 [cit. 17-11-2019]. Dostupné z WWW: <https://search.proquest.com/docview/1728276011?accountid=159230>

BLACKBURN, M. *Burndown Chart: What Is It & How Do I Use It?* [online]. 2019 [cit. 19-1-2020]. Dostupné z WWW: <https://www.projectmanager.com/blog/burndown-chart-what-is-it>

BROWN, T. *Scrum vs. kanban: Which agile framework is better?* [online]. 2019 [cit. 1-1-2020]. Dostupné z WWW: <https://opensource.com/article/19/8/scrums-vs-kanban>

CONÇALVES, L. *What Is Sprint Review Meeting and How To Hold Absolutely Interesting One*. [online]. 2019 [cit. 30-12-2019]. Dostupné z WWW: <https://luis-goncalves.com/sprint-review-meeting/>

DHARMALINGAM, N. *An Overview of Agile Contracts*. [online]. 2018 [cit. 13-1-2020]. Dostupné z WWW: <https://www.whizlabs.com/blog/agile-contracts/>

DILEEP, R. *Importance of Continuous Integration and Delivery in Agile Software Development*. [online]. 2019 [cit. 30-12-2019]. Dostupné z WWW: <https://www.greycampus.com/blog/agile-and-scrum/continuous-integration-and-continuous-delivery>

EBY, K. *Comprehensive Guide to the Agile Manifesto*. [online]. 2016 [cit. 25-12-2019]. Dostupné z WWW: <https://www.smartsheet.com/comprehensive-guide-values-principles-agile-manifesto>

FERGUSON, B. *The agile guide to winning at team development*. [online]. 2019 [cit. 11-1-2020]. Dostupné z WWW: <https://www.atlassian.com/blog/teamwork/navigate-tuckman-stages-of-team-development>

GRATIS, B. *Running your first Sprint Planning meeting*. [online]. 2015 [cit. 17-11-2019]. Dostupné z WWW: <https://backlog.com/blog/running-first-sprint-planning-meeting/>

HASLINGER, D., WINTERSTEIGER, A. *Implementing Scrum & Agile Development*. [online]. 2017 [cit. 3-1-2020]. Dostupné z WWW: [https://www.mpirics.com/resources/pdf/consulting/OBJ\\_WP\\_Implementing\\_Scrum.pdf](https://www.mpirics.com/resources/pdf/consulting/OBJ_WP_Implementing_Scrum.pdf)

HEBB, D. *What Agile Methodology is Best For Your Development Teams in 2020?* [online]. 2019 [cit. 25-12-2019]. Dostupné z WWW: <https://www.contentbloom.com/blog/kanban-scrum-complete-guide/>

HOEK, J. *The Ideal Scrum Team Composition for Agile Development*. [online]. 2017 [cit. 27-12-2019]. Dostupné z WWW: <https://www.mendix.com/blog/the-road-to-adopting-scrum-team-composition/>

HUETHER, D. *Definition of Done*. [online]. 2017 [cit. 29-12-2019]. Dostupné z WWW: <https://www.leadingagile.com/2017/02/definition-of-done/>

HYDE, P. *Forming, Storming, Norming and Performing in a Scrum Team*. [online]. 2017 [cit. 11-1-2020]. Dostupné z WWW: <https://www.frontrowagile.com/blog/posts/120-forming-storming-norming-and-performing-in-a-scrum-team>

CHAPMAN, Z. *Is Your Team Using Scrum Properly?* [online]. 2019 [cit. 27-12-2019]. Dostupné z WWW: <https://www.clearlyagileinc.com/agile-blog/is-your-team-using-scrum-properly>

CHOUDARY, A. *Scrum Master: What are the Roles and Responsibilities?* [online]. 2019 [cit. 27-12-2019]. Dostupné z WWW: <https://www.edureka.co/blog/scrum-master-roles-responsibilities/>

JIRA, *Setting Personal WIP or WIP limits per user* [online]. 2020 [cit. 06-06-2020]. Dostupné z WWW: <https://jira.atlassian.com/browse/JRACLOUD-36973>

KIENTZ P., *The pros and cons of Spiral Software Development* [online]. 2017 [cit. 10-12-2019]. Dostupné z WWW: <https://www.dcssoftware.com/pros-cons-spiral-software-development/>

LEHMANN-BENZ, A. *Agile Contract Models – Working with Cross-Corporate Teams.* [online]. 2019 [cit. 12-1-2020]. Dostupné z WWW: <https://www.theprojectgroup.com/blog/en/agile-contract-models/>

LEVISON, M. *Scrum Development Team – Who's In It?* [online]. 2017 [cit. 27-12-2019]. Dostupné z WWW: <https://agilepainrelief.com/notesfromatooluser/2017/06/scrum-development-team-whos-in-it.html>

LUM, M. *5 Scrum Meeting Best Practices: Master the Daily Stand-Up.* [online]. 2016 [cit. 30-12-2019]. Dostupné z WWW: <https://sprint.ly/blog/scrum-meeting-best-practices/>

MANIFESTO FOR AGILE SOFTWARE DEVELOPMENT. *Twelve Principles of Agile Software.* [online]. 2001 [cit. 25-12-2019]. Dostupné z WWW: <https://agilemanifesto.org/principles.html>

MARCIAS, J. A. *5 Steps to Executing a Successful Agile Development Pilot Project.* [online]. 2015 [cit. 3-1-2020]. Dostupné z WWW: <https://blog.softtek.com/en/5-steps-to-executing-a-successful-agile-development-pilot>

MASTROPASQUA, F. *The Ultimate Guide to the Sprint Backlog.* [online]. 2001 [cit. 29-12-2019]. Dostupné z WWW: <https://www.clearlyagileinc.com/agile-blog/the-ultimate-guide-to-the-sprint-backlog>

MISTRY, A. *Agile Story Point Estimation Techniques – T-Shirt Sizing.* [online]. 2017 [cit. 29-12-2019]. Dostupné z WWW: <https://www.c-sharpcorner.com/article/agile-story-point-estimation-techniques-t-shirt-sizing/>



MITCHELL, I. *Walking Through a Definition of Done*. [online]. 2017 [cit. 29-12-2019]. Dostupné z WWW: <https://www.scrum.org/resources/blog/walking-through-definition-done>

MOSKALENKO, S. *What Scrum Says About Estimates*. [online]. 2017 [cit. 29-12-2019]. Dostupné z WWW: <https://www.scrum.org/resources/blog/what-scrum-says-about-estimates>

NAUGHTER, T. *Three Types of Agile Contracts*. [online]. 2018 [cit. 13-1-2020]. Dostupné z WWW: <https://www.contractworks.com/blog/three-types-of-agile-contracts>

NIJLAND, S. *The Development Team*. [online]. 2018 [cit. 27-12-2019]. Dostupné z WWW: <https://medium.com/serious-scrum/the-development-team-575d69054a9b>

OSHIDA, T. *How to get Scrum Right on First Attempt*. [online]. 2019 [cit. 3-1-2020]. Dostupné z WWW: <https://medium.com/@takeshi.yoshida/scrum-sprint-pilot-6287aac19703>

OVEREEM, B. *The 11 Advantages of Using a Sprint Goal*. [online]. 2016 [cit. 28-12-2019]. Dostupné z WWW: <https://www.scrum.org/resources/blog/11-advantages-using-sprint-goal>

PATTON, J. *Dual-track Daily Standup* Jeff Patton & Associates. [online]. 2019 [cit. 14-12-2019]. Dostupné z WWW: <https://www.jpattonassociates.com/dual-track-daily-standup/>

PAVLICHENKO, I. *How To Ensure That Scrum Teams Launch Successfully*. [online]. 2017 [cit. 11-1-2020]. Dostupné z WWW: <https://www.scrum.org/resources/blog/ensure-scrum-teams-launch-successfully>

PICHLER, R. *The Scrum Cycle*. [online]. 2012 [cit. 25-12-2019]. Dostupné z WWW: <https://www.romanpichler.com/blog/the-scrum-cycle/>

PICHLER, R. *10 Tips for Writing Good User Stories*. [online]. 2016 [cit. 29-12-2019]. Dostupné z WWW: <https://www.romanpichler.com/blog/10-tips-writing-good-user-stories/>

POWERS, S. *Coding Standards*. [online]. 2014 [cit. 30-12-2019]. Dostupné z WWW: <https://www.adventureswithagile.com/2014/05/28/coding-standards/>

RADIGAN, D. *4 best practices for sprint planning meetings*. [online]. 2019 [cit. 30-12-2019]. Dostupné z WWW: <https://www.atlassian.com/blog/agile/sprint-planning-atlassian>

REDKA, M. *Kanban vs. Scrum: Which to Use*. [online]. 2018 [cit. 25-12-2019]. Dostupné z WWW: <https://mlsdev.com/blog/kanban-vs-scrum>

REHKOPF, M. *What is a scrum master? Learn what a scrum master is and how they help teams stay on track.* [online]. 2018 [cit. 25-12-2019]. Dostupné z WWW: <https://www.atlassian.com/agile/scrum/scrum-master>

RIBEIRO, R. *How to Run a Sprint Retrospective 101: The Essential Guide.* [online]. 2019 [cit. 31-12-2019]. Dostupné z WWW: <https://www.outsystems.com/blog/posts/sprint-retrospective/>

ROSEKE, B. *How to Use Kanban to Improve Projects.* [online]. 2019 [cit. 1-1-2020]. Dostupné z WWW: <https://www.projectengineer.net/how-to-use-kanban-to-improve-projects/>

ROUS, M. *Sprint (software development)* [online]. 2018 [cit. 28-12-2019]. Dostupné z WWW: <https://searchsoftwarequality.techtarget.com/definition/Scrum-sprint>

SCHWABER, K., SUTHERLAND, J. *The Scrum Guide— The Definitive Guide to Scrum: The Rules of the Game.* [online]. 2013 [cit. 20-12-2019]. Dostupné z WWW: <https://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>

SINHA, D. *Scrum Epic: How to Split an Epic into Chunks in Agile?* [online]. 2019a [cit. 29-12-2019]. Dostupné z WWW: <https://www.knowledgehut.com/blog/agile/what-is-an-epic-agile>

SINHA, D. *A Step-by-step Guide to Implementing Scrum in Organizations* [online]. 2019b [cit. 3-1-2020]. Dostupné z WWW: <https://www.knowledgehut.com/blog/agile/implementing-scrum-in-organizations>

SRIVASTAVA, P., JAIN, S. *A leadership framework for distributed self-organized scrum teams.* *Team Performance Management*, 23(5), 293-314. [online]. 2013 [cit. 26-12-2019]. Dostupné z WWW: <http://dx.doi.org/10.1108/TPM-06-2016-0033>

Stoica, M., at al. *Software development: Agile vs. traditional.* *Informatica Economica*, 17(4), 64-76. [online]. 2013 [cit. 2-1-2020]. Dostupné z WWW: <https://search.proquest.com/docview/1492882301?accountid=159230>

ŠOCHOVÁ, Z. *Agile HR: Jak podporujete zaměstnance?* [online]. 2018b [cit. 19-1-2020]. Dostupné z WWW: <https://soch.cz/blog/management/agile/agile-hr-jak-podporujete-zamestnance/>

ŠOCHOVÁ, Z. *Co je to agile?* [online]. 2010 [cit. 24-12-2019]. Dostupné z WWW: <https://soch.cz/blog/management/agile/co-je-to-agile/>

ŠOCHOVÁ, Z. *Kanban* [online]. 2012 [cit. 24-12-2019]. Dostupné z WWW: <https://soch.cz/blog/management/agile/kanban/>

TRAVNÍČEK, M. *Vytvořte si Smlouvu o agilním vývoji a implementaci softwaru na míru* [online]. 2020 [cit. 30-05-2020]. Dostupné z WWW: <https://www.legito.cz/smlouva-o-agilnim-vyvoji-a-implementaci-softwaru?selectedPreset=11>

UNHELKAR, B. *The Art of Practicing Agile : A Practical Guide for Applying Agile to Real Projects*, Auerbach Publishers, Incorporated, London, ISBN 978-1-4398-5119-7 [online]. 2012 [cit. 25-12-2019]. Dostupné z WWW: <https://ebookcentral.proquest.com/lib/knav/reader.action?docID=1165927>

VERWIJS, CH. *Myth: Having A Sprint Goal Is Optional In Scrum*. [online]. 2019 [cit. 28-12-2019]. Dostupné z WWW: <https://www.scrum.org/resources/blog/myth-having-sprint-goal-optional-scrum>

WOLPERS, S. *15 Sprint Review Anti-Patterns Holding Back Scrum Teams*. [online]. 2019 [cit. 31-12-2019]. Dostupné z WWW: <https://age-of-product.com/sprint-review-anti-patterns/>

ZACK, T. *Agile Transformation. The Basics of Agile Transformation*. [online]. 2019 [cit. 3-1-2020]. Dostupné z WWW: <https://www.leadingagile.com/2018/07/the-basics-of-agile-transformation/>

### **Seminární práce**

BUSCHOVÁ, L. *Agilní transformace v malé IT společnosti*. Praha, 2019. Seminární práce. VŠEM.

### **Zdroje firmy XY**

XY. *Vnitropodnikové pokyny a informace*. 2020. Dostupné interně.

XY. *About*. [online]. 2020. Dostupné z WWW: <https://www.xy.cz/about/>

## Seznam příloh

Příloha 1 Otázky pro rozhovory – fáze 1 .....	I
Příloha 2 Otázky pro rozhovory – fáze 2 .....	II
Příloha 3 Odpovědi z rozhovorů – fáze 1 .....	VI
Příloha 4 Odpovědi z rozhovorů – fáze 2 .....	XVII
Příloha 5 Vzájemné hodnocení členů Development týmu .....	XXVII
Příloha 6 Scrum tabule Sprint 2 .....	XXX
Příloha 7 Coaching Backlog .....	XXIX
Příloha 8 Pět příčin selhávání týmů – návrhy pro překonání disfunkce .....	XXX
Příloha 9 Podpora týmu ve Storming fázi .....	XXXI
Příloha 10 Tuckmanův model .....	XXXII
Příloha 11 Standup – 3 otázky .....	XXXIII
Příloha 12 Retrospektiva s lepítky .....	XXXIV
Příloha 13 Vzor agilní smlouvy.....	XXXV

# Přílohy

## Příloha 1 Otázky pro rozhovory – fáze 1

### Otázky pro Scrum Mastera

1. Jak jste se připravoval na zahájení přechodu na Scrum?
2. Popište prosím roli Scrum Mastera, jak ji vnímáte Vy osobně. Jaká by byla Vaše definice této role?
3. Co patří mezi odpovědnosti Scrum Mastera?
4. Jak jste sestavovali Scrum tým? Jak probíhal výběr členů?
5. Popište, prosím, jak probíhal první Planning a jakou metodou byly prováděny odhady práce.
6. Jak jste vypočítali kapacitu týmu pro první sprint?
7. Popište, prosím, jak probíhají první Standup meetingy týmu. Jak dlouho trvají a co je během nich řešeno.
8. Popište, prosím, jak proběhl první Sprint. Co se podařilo a co naopak dělalo týmu problémy?
9. Popište, prosím, jak proběhlo první Review a zda byl splněn cíl Sprintu?
10. Popište, prosím, oblasti, které byly řešeny během první Retrospektivy.
11. Jak je nastavena Definition of Done?
12. Kolik času trávíte s Development týmem?

### Otázky pro Product Ownera

1. Jak jste se připravoval na zahájení přechodu na Scrum?
2. Popište prosím roli Product Ownera, jak ji vnímáte Vy osobně. Jaká by byla Vaše definice této role?
3. Jaké jsou povinnosti a úkoly Product Ownera?
4. Jak by měla vypadat User Story? Jaká je její správná struktura?
5. Jak probíhala příprava User Stories?
6. Jak probíhal první Planning?
7. Popište, prosím, jak dle Vás proběhl první Sprint?
8. Popište, prosím, jak dle Vás probíhalo první Review?
9. Kolik času trávíte s Development týmem a jak s nimi spolupracujete?

10. Řešili jste novu formu spolupráce a úpravu kontraktu na agilní typ smlouvy?

### **Otázky pro členy Development týmu**

1. Jak jste se připravoval na zahájení přechodu na Scrum?
2. Popište, prosím, jak se dle Vás liší role vývojáře ve Scrumu v porovnání s klasickým přístupem řízení projektů, ze kterého nyní přecházíte na agilní metodu Scrum.
3. Jsou pro Vás User Stories srozumitelné a dostatečně konkrétní?
4. Jak probíhal Planning na první Sprint?
5. Popište, prosím, jak probíhají denní Standup schůzky.
6. Popište, prosím, jak dle Vás proběhl první Sprint? Jak se Vám pracovalo, měl jste s něčím problémy nebo se naopak něco zlepšilo?
7. Popište, prosím, jak probíhalo Review po prvním Sprintu?
8. Jaké oblasti jste probírali na první Retrospektivě?
9. Využíváte nebo jste nově začali využívat některé metody agilního programování? Např.: Pair Programming, Coding Standard, Continuous Integration a Code Refactoring nebo jiné? Případně, které byste chtěli začít používat.

Zdroj: vlastní zpracování (2020)

## **Příloha 2 Otázky pro rozhovory – fáze 2**

### **Otázky pro Scrum Mastera**

1. Jak byste popsal dosavadní dvou měsíční agilní vývoj na pilotním projektu ze svého pohledu?
2. Jak po dvou měsících agilního vývoje probíhá spolupráce mezi členy týmu? Změnil se jejich přístup k agilnímu vývoji a ke Scrumu?
3. Jak dlouho nyní Standup meetingy trvají a jaký je jejich průběh?
4. Popište prosím, jak probíhají nyní Planning meetingy a jakou metodu odhadu nakonec tým používá?
5. Jak dlouhé jsou sprinty a jaký je jejich průběh z vašeho pohledu? Je dosahováno cíle sprintů?
6. Jak probíhají v současnosti Review? Kdo se těchto schůzek účastní a jak probíhá samotná prezentace?
7. Popište prosím témata, která řešíte na retrospektivních schůzkách?
8. Jak je nyní nastavená Definition of Done oproti počátku Scrumu?
9. Jak se mění rychlost týmu?
10. Uveďte prosím, jaká máte očekávání na další měsíce.

### **Otázky pro Product Ownera**

1. Jak na vás během dvou měsíců agilního vývoje působila spolupráce s vývojáři a Scrum Masterem?
2. Jakých schůzek se účastníte a jak dle vás tyto schůzky probíhají?
3. Kteří zástupci z vaší firmy se účastní Review?
4. Jak jste spokojen s plněním cílů jednotlivých Sprintů?
5. Jak hodnotíte kvalitu práce Development týmu?
6. Jsou nějaké nové změny nebo návrhy ohledně typu kontraktu mezi vaší firmou a dodavatelem?
7. Uveďte prosím, jaká máte očekávání na další měsíce.

### **Otázky pro členy Development týmu**

1. Jak byste popsal dosavadní dvou měsíční agilní vývoj na pilotním projektu ze svého pohledu?
2. Jak se vám nyní tímto agilním způsobem spolupracuje s kolegy vývojáři?
3. Jak hodnotíte práci Vašeho Scrum Mastera?

4. Jak hodnotíte práci Vašeho Product Ownera?
5. Došlo k nějakým změnám v tom, jaký mají nyní průběh Standup schůzky?
6. Při prvním sprintu byl pro vás Planning velmi náročný časově i obsahově. Jak je tomu nyní?
7. Od Scrum Mastera a Product Ownera vím, že během dvou Sprintů se nepodařilo dosáhnout cíle. Co bylo dle vás příčinou?
8. Začali jste již se zaváděním alespoň některých agilních metod programování?

Zdroj: vlastní zpracování (2020)



## Příloha 3 Odpovědi z rozhovorů – fáze 1

### Rozhovor se Scrum Masterem – fáze 1

Ráda bych s Vámi nyní udělala rozhovor na téma zahájení agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

#### 1. Jak jste se připravoval na zahájení přechodu na Scrum?

*V minulých měsících jsem se snažil získat co nejvíce informací o agilním přístupu a metodě Scrum. Zaměřoval jsem se hlavně na svou roli Scrum Mastera, na jeho náplň a vhodný přístup k týmu. Nastudoval jsem některé knihy o Scrumu a doporučené postupy na webových stránkách. V prosinci jsem se účastnil dvou denního kurzu, který byl zaměřen na implementaci agilních metod. Nejednalo se ale o certifikační kurz Scrum Mastera, ale pouze o úvod do problematiky agilu a Scrumu. Kurz by přínosný i když jsem se přímo nedozvěděl nové informace, tak zajímavé bylo propojení s praxí a možnost ptát se zkušeného lektora a moci porovnat svůj přístup s kolegy v kurzu. Rád bych v dohledné době navázal certifikačním kurzem pro Scrum Mastera.*

#### 2. Popište prosím roli Scrum Mastera, jak ji vnímáte Vy osobně. Jaká by byla Vaše definice této role?

*Myslím, že Scrum Master v sobě zahrnuje několik rolí, které by měl používat dle potřeby. Jako hlavní vnímám roli lídra, které tým vede a motivuje. Scrum Master je také ochránitelem týmu a má pro tým vytvářet vhodné podmínky k práci. Zároveň je zde úzká spolupráce s Product Ownerem, kde vnímám Scrum Mastera jako pravou ruku Product Ownera. Scrum Master by měl být také koučem, měl by umět řešit spory v týmu. Neměl by ale situace řešit jako z nadřazené pozice nebo dávat členům týmu něco příkazem. Pro mě je to něco jak průvodce Scrumem, který předává své znalosti o Scrumu na členy týmu nebo i dalším lidem ve firmě.*

#### 3. Co patří mezi odpovědnosti Scrum Mastera?

*Scrum Master se má postarat o všechny potřebné akce a schůzky, které se ve Scrumu vyskytují. Má dohlížet na průběh schůzek, dodržování času a náplně schůzek. Měl by být co nejvíce s vývojáři a dle potřeby k dispozici Product Ownerovi. Měl by dohlížet na to, aby byly dodržovány aktivity, které jsou Scrumem předepsané. Vývojářům by měl pomáhat s odstraňováním problémů a překážek ať už jsou jakékoliv povahy.*

#### 4. Jak jste sestavovali Scrum tým? Jak probíhal výběr členů?

*Členy týmu jsme vybírali dle jejich odborných znalostí a zaměření tak, abychom byli schopni pokrýt všechny činnosti, které potřebujeme realizovat. Také jsme zohledňovali senioritu vývojářů, abychom měli v týmu zkušené pracovníky, kteří budou schopni pomáhat těm méně zkušeným kolegům. A také jsme chtěli do týmu alespoň nějakého vývojáře, který již má praktickou zkušenost s agilním vývojem. Tuto sestavu se podařilo vytvořit ze stávajících pracovníků, takže se všichni již znají. Tým vývojářů má nyní 5 členů. Ale předpokládáme, že se ještě možná rozroste o dalšího jednoho nebo dva členy. Ale to až asi za dva nebo tři měsíce. Bude záležet, jak bude probíhat vývoj a jaké budou požadavky ze strany klienta na rozvoj produktu, tedy kolik práce bude potřeba realizovat a jak rychle.*

#### 5. Popište, prosím, jak probíhal první Planning a jakou metodou byly prováděny odhady práce.

*První Sprint jsme naplánovali pouze na jeden týden, abychom si vyzkoušeli celý průběh Sprintu co nejdříve. Plánovací schůzka nám trvala asi dvě a půl hodiny. Backlogu bylo připraveno několik User Stories, které*

*Product Owner připravil s mojí pomocí. Vývojáři již měli rámcově představu o obsahu User Stories z předešlé schůzky. Na začátku schůzky jsme s týmem řešili, jak budeme dělat odhady, jestli použijeme nějakou doporučenou metodu a jaké jednotky budeme používat. V tomto ohledu jsou mezi členy týmu názorové rozpory a jediné na čem se nakonec byli schopni dohodnout bylo to, že alespoň pro zatím budou používat jako jednotku času hodiny nebo dny, protože vše ostatní bylo pro některé členy týmu příliš abstraktní. Domluvili jsme se na tom, že za pár Sprintů zkusíme udělat změnu v používaných odhadech a uvidíme co nakonec bude lépe vyhovovat. Takže na prvním Planningu se odhadovalo v hodinách a každý člen týmu si svou práci rozdělil na zhruba 3 až 4 úkoly, ke kterým přiřadil svůj odhad.*

6. Jak jste vypočítali kapacitu týmu pro první Sprint?

*Počítali jsme, že tým bude ve vývoji pomalejší než obvykle, takže jsme nechali dle odhadů zhruba jeden den rezervu pro každého vývojáře.*

7. Popište, prosím, jak probíhají první Standup meetingy týmu. Kde a kdy probíhají. Jak dlouho trvají a co je během nich řešeno.

*Scházíme se v zasedací místnosti v 10:00 a Product Owner se připojuje online, pokud se účastní. Doba se trvání se pohybovala přibližně 20 až 30 minut. Vývojáři jeden po druhém řeknou, co zvládli udělat den předem a na čem pracují daný den. A pak se většinou řeší dotazy nebo problémy, na které někdo z nich narazil nebo se upřesňují informace k jednotlivým úkolům. Také aktualizujeme stav úkolů na Scrum Boardu. Zatím je to moc dlouhé a někdy trochu chaotické.*

8. Popište, prosím, jak proběhl první Sprint. Co se podařilo a co naopak dělalo týmu problémy?

*Pro první Sprint jsme si zvolili opravu jednoduchou User Story a vývoj funkcionality byl poměrně snadný a objem práce byl malý. Udělali jsme to tak, abychom se vyvarovali tomu, že se to hned na poprvé nepovede a tým to bude demotivovat v další práci nebo klient začne pochybovat, zda je to dobrá cesta. Takže vývoj jako takový byl bez větších problémů. Poměrně náročné bylo pro tým plánování na Sprint, protože schůzka byla celkem dlouhá. Někomu dělá problém být včas na Standupu, protože teprve na desátou hodinou přichází do práce, ostatní na něj pak čekají. Standupy jsou také poměrně dlouhé, ale to se snad časem podaří redukovat. Většina členů týmu má problém s aktualizací tasků na tabuli a zapominají je přesouvat, když už je mají hotové. Takže stav tabule neodpovídá tomu, na čem se pracuje.*

9. Popište, prosím, jak proběhlo první Review a zda byl splněn cíl Sprintu?

*Cíl Sprintu se podařilo splnit, protože plánovanou funkcionalitu tým dokončil a bylo tedy možné výsledek představit na Review. Na schůzku dorazil ze strany klienta pouze Product Owner a vývojáři se účastnili všichni. Předem se ještě dohodli mezi sebou, kdo bude výsledek prezentovat i když se do toho nikomu z nich moc nechtělo. Samotná prezentace trvala pár minut a následovala ještě diskuse nad doladěním výsledku. Celkově si myslím, že to probíhalo dobře i přes to, že pro vývojáře to byla spíše nekomfortní situace, protože nejsou zvyklí na prezentování.*

10. Popište, prosím, oblasti, které byly řešeny během první Retrospektivy.

*Mluvili jsme o dochvilnosti některých členů na Standupu a o délce těchto schůzek. Uvažovali jsme o posunutí Standupu na pozdější čas, ale některým kolegům se to zdálo nevyhovující kvůli přerušování práce. Takže jsme se domluvili, že zatím čas ponecháme na současných 10 hodin dopoledne a po dalším to vyhodnotíme znovu. Také jsme se pobavili o jednotlivých ceremoniích ve Scrumu a jejich významu. Hlavně o Standupu, protože jej vývojáři vnímají spíše jako obtěžující. Snažil jsem se vysvětlit, že je to čas pro ně, kdy si mají sdělit potřebné informace, že to není report pro mě nebo Product Ownera.*

## 11. Jak je nastavena Definition of Done?

*Zatím jsme se k tomu bohužel vůbec nedostali. Víím, že je to chyba, ale je těch nových věcí, které nyní zavádíme, mnoho. Zatím jsme to tedy nijak neformulovali a postupujeme tak, že funkcionality, která je otestovaná je hotová a může se předat Product Ownerovi k akceptaci.*

## 12. Kolik času trávíte s Development týmem?

*Jsem tu s nimi celý den. Všichni jsme v jednom Open Space i s kolegy z dalších projektů. Takže v kontaktu jsme stále a kdykoliv spolu můžeme mluvit.*

Na závěr rozhovoru bych Vás požádala o hodnocení jednotlivých členů týmu. Hodnoťte na stupnici 1 až 10 týmovost, nasazení a aktivitu u jednotlivých členů Development týmu na tomto pilotním projektu. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup daného člena týmu. Na toto hodnocení máte nyní 30 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## Rozhovor s Product Ownerem – fáze 1

Ráda bych s Vámi nyní udělala rozhovor na téma zahájení agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

### 1. Jak jste se připravoval na zahájení přechodu na Scrum?

*S přípravou mi hodně pomáhal nový Scrum Master. A i když i pro něho, jako projektového manažera, je tato role nová, tak má už mnohem více znalostí o Scrumu a o tom co bych měl já jako Product Owner vědět a umět. Sám si také nastudoval informace o tom, co vše se ve Scrumu odehrává a jak by jednotlivé schůzky měly probíhat a jak se na ně připravit. Učit jsem se, jak napsat User Story nebo jak by měl vypadat Product Backlog. Víím, je to teprve naprostý základ a mám se toho ještě moc co učit, a hlavně v praxi bude vidět co z toho už funguje a kde mám rezervy.*

### 2. Popište prosím roli Product Ownera, jak ji vnímáte Vy osobně. Jaká by byla Vaše definice této role?

*Je to role, která ve Scrumu rozhoduje o tom, co a kdy se bude realizovat. Chápu roli Product Ownera jako někoho, kdo ví, jak má nebo jak bude nový produkt vypadat. Proč a pro koho má být vyvinut. Co od něj zákazníci i naše firma očekává. Také je potřeba, aby Product Owner byl někdo, kdo dokáže tyto informace předat směrem k vývojářům i k dalším lidem ve firmě.*

### 3. Jaké jsou povinnosti a úkoly Product Ownera?

*Příprava Product Backlogu a User Stories. Být k dispozici pro vývojáře a komunikovat s nimi. Účastnit se Review a plánování.*

### 4. Jak by měla vypadat User Story? Jaká je její správná struktura?

*User Story musí obsahovat hodnotu pro uživatele. Popis by měl obsahovat co daný uživatel chce a také proč co chce. Pokud je to vhodné tak, lze doplnit například nákres pro vysvětlení.*

5. Jak probíhala příprava User Stories?

*Pro evidenci jsem použil tabulku, ve které eviduji seznam User Stories. Samotná příprava pak probíhala buď se Scrum Masterem nebo jsem je připravoval sám a pak jsem je spolu konzultovali.*

6. Jak probíhal první Planning?

*Schůzka byla celkem náročná a trvala přes dvě hodiny. Zvolili jsme si User Story, která nebyla moc složitá na realizaci a s týmem jsme pak k nim řešili detaily a vývojáři si každý za sebe připravili úkoly a určili odhad času, který budou zhruba potřebovat pro realizaci. Na začátku se ještě řešila forma, ve které se budou uvádět odhady.*

7. Popište, prosím, jak dle Vás proběhl první Sprint?

*Myslím, že jsme všichni byli rádi, že jsme to celé zvládli i když první Sprint jsme si dali jako zkušební s malou pracností pro vývoj, aby zbyl čas na vše nové kolem. Takže až se pojedeme v normálním režimu, tak to bude o dost jiné a náročnější pro nás všechny. Planning byl pro vývojáře i pro mě docela náročný. Pak jsem se účastnil dvou Standupů, ale jen online, pro případ, kdyby od mě někdo něco potřeboval. Jinak jsem se snažil být k dispozici a během týdne jsme řešili co bylo potřeba.*

8. Popište, prosím, jak dle Vás probíhalo první Review?

*Někteří vývojáři byli dost nervózní a bylo vidět, že byli rádi, když prezentace skončila. Ale myslím, že vůbec nemuseli mít obavy, protože to zvládli a svoji práci odprezentovali dobře. Myslím, že časem si zvyknou a nebude jim to dělat žádné problémy.*

9. Kolik času trávíte s Development týmem a jak s nimi spolupracujete?

*Zatím jsem se účastnil potřebných schůzek jako je Planning, Review a vzdáleně dvou Standupů. Jinak jsem na telefonu a pokud je potřeba, tak není problém, abych se za nimi zastavil v průběhu Sprintu.*

10. Řešili jste novu formu spolupráce a úpravu kontraktu na agilní typ smlouvy?

*Tento návrh zde byl od dodavatele zmíněn, ale vedení naší firmy chtělo zachovat současný typ kontraktu, takže máme stále původní smlouvu i na pilotní projekt, kde je uvedena záloha, celková částka a termín dokončení projektu. Tato část smlouvy zahrnuje první etapu projektu, která má trvat 5 měsíců a následně bude určitě probíhat další rozvoj, ale ten již není součástí této smlouvy. Takže je otázka, jak se to bude smluvně upravovat dále. Ale touto otázkou se nyní nikdo nezabývá.*

Na závěr rozhovoru bych Vás požádala o hodnocení Development týmu jako celku. Do tohoto hodnocení prosím nezahrnujte Scrum Mastera. Hodnoťte na stupnici 1 až 10 týmovost, nasazení a aktivitu týmu na tomto pilotním projektu. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup daného člena týmu. Na toto hodnocení máte nyní 30 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

## Rozhovor se členem Development týmu – fáze 1

### Vývojář 1

Ráda bych s Vámi nyní udělala rozhovor na téma zahájení agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak jste se připravoval na zahájení přechodu na Scrum?

*Já již s touto metodou zkušenosti mám z minulosti, takže jsem nyní neměl potřebu se nějak konkrétně připravovat nebo něco zjišťovat. Spíše jsem nějaké věci konzultovali společně se Scrum Masterem, který se zajímal o mé zkušenosti a agilem a se Scumem.*

2. Popište, prosím, jak se dle Vás liší role vývojáře ve Scrumu v porovnání s klasickým přístupem řízení projektů, ze kterého nyní přecházíte na agilní metodu Scrum.

*Pro mě to znamená být více součástí celého řešení. To znamená podílet se na návrzích řešení, a to s celým týmem. Také je zde mnohem větší spolupráce s klientem prostřednictvím role Product Ownera, ale obvykle je možné se setkat i s jinými lidmi ze strany klienta. Jednotliví členové týmu musí začít fungovat společně, ne jako jednotlivci. A mnohem více spolupracovat a vědět o tom, co kdo dělá.*

3. Jsou pro Vás User Stories srozumitelné a dostatečně konkrétní?

*Zatím jsme řešili jednu User Story, která byla celkem jednoduchá a srozumitelná, ale uvidíme, jak se to bude vyvíjet dále.*

4. Jak probíhal Planning na první Sprint?

*Bylo to celkem náročné časově i obsahem. Nějak jsme se nemohli shodnout na tom, v čem ty odhady vlastně budeme dělat, a nakonec jsme to nechali v hodinách. Je i to celkem jedno, ale moc mi to ke Scrumu nejde. Ale chápu, že změnit vše najednou je možná příliš a není to vhodné někomu nařídít, když s tím má problém.*

5. Popište, prosím, jak probíhají denní Standup schůzky.

*Zatím celkem zmateně a hlavně dlouze. Kluci mají obvykle pocit, že tam musí popsat co, a hlavně jak včera dělali. A pak se také zasekneme na řešení nějakého problému, který se týká jen dvou lidí. Co mi asi vadí nejvíce je to, když se musí na někoho čekat, protože nestíhá přijít do práce na desátou hodinu.*

6. Popište, prosím, jak dle Vás proběhl první Sprint? Jak se Vám pracovalo, měl jste s něčím problémy nebo se naopak něco zlepšilo?

*O zlepšení se mluvit nedá vůbec, zatím je to spíše o tom, že jsme to odstartovali a teď se v tom budeme učit plavat. Já s tím osobně nemám problém a změnu vítám i když to nějakou dobu bude náročnější, ale důležité je, aby se to postupně zlepšovalo a byly vidět výsledky. Pro začátek je fajn, že jsme byli schopni za jeden týden něco funkčního vyvinout a nějak se organizovat.*

7. Popište, prosím, jak probíhalo Review po prvním Sprintu?

*Prezentoval jsem většinu toho, co jsme dělali a kolegové mě doplňovali. Za klienta byl na Review pouze Product Owner, což mě trošku překvapilo, že nedorazil alespoň ještě někdo. Kolegové měli před schůzkou celkem trému, ale nakonec byli snad také spokojeni, že to zvládli. Prezentovali jsme*

*výsledek naší týdenní práce. Obraz jsme sdíleli na připojenou velkou televizní obrazovku, takže Product Owner mohl vidět celý výsledek. Pak jsme ještě probírali, co bude potřeba vylepšit v rámci dalšího Sprintu nebo později.*

8. Jaké oblasti jste probírali na první Retrospektivě?

*Vysvětlovali se převážně pravidla Scrum schůzek a řešily se převážně Standupy, které jsou zatím celkem problematické.*

9. Využíváte nebo jste nově začali využívat některé metody agilního programování? Např.: Pair Programming, Coding Standard, Continuous Integration a Code Refactoring nebo jiné? Případně, které byste chtěli začít používat.

*Zatím jsme neřešili, ale měli bychom se určitě co nejdříve začít zabývat automatickými testy, a tedy celkově Continuous Integration. Máme tu také méně zkušené vývojáře a zatím je kontrola jejich kódu minimální, takže definice standardů a kontrola kódu zkušenějším programátorem je určitě minimálně vhodná.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## **Rozhovor se členem Development týmu – fáze 1**

### **Vývojář 2**

Ráda bych s Vámi nyní udělala rozhovor na téma zahájení agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak jste se připravoval na zahájení přechodu na Scrum?

*O Scrumu mám všeobecné informace a agilnímu přístupu rozumím, ale nepracoval jsem tímto způsobem v žádné firmě. Takže představu mám, ale určitě vítám pomoc od kolegů nebo od Scrum Mastera, které nás povede a bude nám říkat co a jak.*

2. Popište, prosím, jak se dle Vás liší role vývojáře ve Scrumu v porovnání s klasickým přístupem řízení projektů, ze kterého nyní přecházíte na agilní metodu Scrum.

*Zatím tam vnímám výraznější kontakt se zákazníkem, kdy máme možnost se ho ptát a mít přímo jeho zpětnou vazbu na naši práci. Určitě se asi bude postupně dost vyvíjet celý tým a naše spolupráce, ale neumím si to ještě přesněji představit.*

3. Jsou pro Vás User Stories srozumitelné a dostatečně konkrétní?

*Ano, během prvního Sprintu to pro mě bylo dostačující.*

4. Jak probíhal Planning na první Sprint?

*Meeting trval celkem dlouho, protože jsme řešili, jak budou odhady probíhat a pak jsme si vyjasňovali podrobnosti pro User Story a pak jsme teprve odhadovali. Je nezvyklé to řešit před klientem a s klientem.*

5. Popište, prosím, jak probíhají denní Standup schůzky.

*Doufám, že se nám to podaří nějak zkrátit, protože skoro půl hodina mi přijde jako ztráta času pro pět vývojářů. Ale líbí se mi mít přehled o tom co kdo dělá a vidět tak celý obraz projektu.*

6. Popište, prosím, jak dle Vás proběhl první Sprint? Jak se Vám pracovalo, měl jste s něčím problémy nebo se naopak něco zlepšilo?

*Jsem rád, že jsme ho zvládli dokončit a mohli jsme práci odprezentovat. Bylo to občas náročné, zmatené a nejisté, ale asi jsme ani nic jiného nečekali. Horší by bylo nebo bude, pokud se nám nepodaří dokončit to k čemu se zavážeme.*

7. Popište, prosím, jak probíhalo Review po prvním Sprintu?

*Velkou část prezentoval kolega. Má z nás s prezentacemi největší zkušenosti, tak jsme mu to rádi přenechali. Průběh byl dobrý, Product Owner byl spokojený.*

8. Jaké oblasti jste probírali na první Retrospektivě?

*Standupy. Někdy nám to zabere snad půl hodiny, a to je dost.*

9. Využíváte nebo jste nově začali využívat některé metody agilního programování? Např.: Pair Programming, Coding Standard, Continuous Integration a Code Refactoring nebo jiné? Případně, které byste chtěli začít používat.

*Určitě bych navrhoval automatické testy, ale jak to znám z praxe, tak na to obvykle není nikde čas nebo se to tak alespoň říká.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## Rozhovor se členem Development týmu – fáze 1

### Vývojář 3

Ráda bych s Vámi nyní udělala rozhovor na téma zahájení agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak jste se připravoval na zahájení přechodu na Scrum?

*Scrum znám, ale nemám s touto metodou v praxi osobní zkušenost. Řešíme vše se Scrum Masterem a on nám s tím pomáhá, když si nejsem něčím jistý tak se ptám, ale žádnou jinou přípravu jsme neměli.*

2. Popište, prosím, jak se dle Vás liší role vývojáře ve Scrumu v porovnání s klasickým přístupem řízení projektů, ze kterého nyní přecházíte na agilní metodu Scrum.

*Myslím, že je to o větší spolupráci mezi lidmi v týmu a také s klientem, se kterým jsme dříve nebyli v kontaktu téměř vůbec.*

3. Jsou pro Vás User Stories srozumitelné a dostatečně konkrétní?

*Product Owner to popsal srozumitelně a potřebné detaily jsme si vyjasnili. A pak už je to spíše o tom shodnout se na detailech řešení.*

4. Jak probíhal Planning na první Sprint?

*Byl to trochu zmatek ohledně samotného odhadování. Máme na to různé názory, tak jsme se nemohli dohodnout. Na to, že jsme toho vlastně moc nenaplánovali, nám to trvalo docela dlouho a pak jsme stejně na nějaké věci zapomněli, že se musí udělat.*

5. Popište, prosím, jak probíhají denní Standup schůzky.

*Já jsem za ně rád i když se v tom teď trochu ztrácíme, tak myslím, že za nějaký čas bude rychlejší. Teď nám to prostě trvá, ale pokud budeme lépe organizovaní, tak to bude mít přínos. Je super vědět na čem dělá kolega a co už má hotové a co řeší za problém, protože mi to může pomoci při mé práci.*

6. Popište, prosím, jak dle Vás proběhl první Sprint? Jak se Vám pracovalo, měl jste s něčím problémy nebo se naopak něco zlepšilo?

*Někdy jsem se celkem bavil, jak nám to nejde a jak jsou nám některé věci nepříjemné i když je to třeba jen Standup schůzka s týmem a jak se nám některé věci nechce říkat před klientem. Práce pro mě byla bez problémů, ale byl to testovací Sprint a ostatní Sprints budou náročnější. Tak uvidím, jak se bude dařit.*

7. Popište, prosím, jak probíhalo Review po prvním Sprintu?

*Byl jsem celkem nervózní, nemám moc rád tyto typy akcí. Nevěděl jsem kdo všechno od klienta tam bude, ale nakonec tam byl jen Product Owner, tak to nebylo tak formální, jak jsem se obával. Naši práci předváděl kolega a vše šlo bez problému, nic se nepokazilo. Product Owner se měl pár otázek a připomínek, ale celkově byl s výsledkem ok.*

8. Jaké oblasti jste probírali na první Retrospektivě?

*Tak nějak obecně Scrum a pak téma byl Standup, který nás moc nebaví, trvá nám dlouho, než to všechno vyřešíme. Tak někdy někdo přijde pozdě, protože zapomene, že v deset hodin už má být připraven.*



9. Využíváte nebo jste nově začali využívat některé metody agilního programování? Např.: Pair Programming, Coding Standard, Continuous Integration a Code Refactoring nebo jiné? Případně, které byste chtěli začít používat.

*Tak to se asi ještě neřešilo nebo alespoň nově nic zatím nezavádíme. A jestli, kdy a co budeme tak to netuším.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## Rozhovor se členem Development týmu – fáze 1

### Vývojář 4

Ráda bych s Vámi nyní udělala rozhovor na téma zahájení agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak jste se připravoval na zahájení přechodu na Scrum?

*Scrum jsme měli ve firmě, kde jsem dříve pracoval, tak nemám pocit, že bych se na to měl připravovat.*

2. Popište, prosím, jak se dle Vás liší role vývojáře ve Scrumu v porovnání s klasickým přístupem řízení projektů, ze kterého nyní přecházíte na agilní metodu Scrum.

*Ve Scrumu jsem zažil v bývalé firmě obrovský tlak na vývojáře, kteří pracovali stále pod velkým stresem, aby rychle něco dodali. Product Owner neuměl pořádně vysvětlit co vlastně chce a pak se na Review řešilo proč to funguje, jak to funguje, a ne tak jak to chtěl klient. Nasazení Scrumu tam tenkrát nic nevyřešilo, spíše naopak se zvýšil stres a množství chyb oproti klasickému vývoji. Oba přístupy mají své plusy a minusy, ale je to v konkrétních lidech. Myslím, že většina vývojářů stejně upřednostňuje klasický vývoj oproti agilu a Scrum. Už jen kvůli častým schůzkám a neustálým změnám.*

3. Jsou pro Vás User Stories srozumitelné a dostatečně konkrétní?

*Zatím nemůžu moc hodnotit, řešili jsme jednu User Story a ty další by prý měly být připravené, ale zatím jsme je neprobírali.*

4. Jak probíhal Planning na první Sprint?

*Dlouhá chaotická schůzka, jak bývá u začínajících Planningů obvykle zvykem. Většinou se vývojáři těchto odhadů také bojí, protože nemají moc čas si to promyslet a zavazují se, že to dodají příští týden. Takže někdo odhady nadhodnotí a někdo zase něco nedomyslí a dá malý odhad, pak se shodnout trvá dost dlouho a zde to bylo podobné.*

5. Popište, prosím, jak probíhají denní Standup schůzky.

*Asi jak se dalo čekat, že to nebude úplně efektivní. Takže schůzky jsou dlouhé, protože se rozebírají moc velké detaily. Většinou každý popisuje, co už udělal a jak to udělal moc podrobně. Nebo se rovnou řeší nějaký problém, který nastal.*

6. Popište, prosím, jak dle Vás proběhl první Sprint? Jak se Vám pracovalo, měl jste s něčím problémy nebo se naopak něco zlepšilo?

*Na vývoj byl Sprint velmi nenáročný, což je dobře, jinak bychom mohli mít problém s realizací. Takže jsme si to vlastně vyzkoušeli, abychom snad v dalším Sprintu fungovali o něco lépe. Problémy byly snad se všemi ceremoniemi, které ve Scrumu jsou, takže schůzky dlouhé, zmatené, ale vývoj jako takový i přesto celkem fungoval.*

7. Popište, prosím, jak probíhalo Review po prvním Sprintu?

*Review probíhalo dobře. Prezentace byla dobrá a Product Owner se zdál být spokojený.*

8. Jaké oblasti jste probírali na první Retrospektivě?

*Hlavně účel a pravidla Standupů, protože s tím mají kolegové problém. Zapomínají na ně, chodí pozdě a pak každý potřebuje poměrně dost času na to, aby řekl, co dělat, co teď dělá a jestli nemá nějaké překážky. Tak uvidíme teď další týden, jestli se něco změní.*

9. Využíváte nebo jste nově začali využívat některé metody agilního programování? Např.: Pair Programming, Coding Standard, Continuous Integration a Code Refactoring nebo jiné? Případně, které byste chtěli začít používat.

*Nevím, jak to má Scrum Master v plánu řešit. Určitě je nutné řešit Continuous Integration jinak budeme narážet na velké problémy. V minulé firmě jsme to neřešily, protože na to vývojáři nedostali prostor a pak se stále řešili chyby v produkčním prostředí.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## Rozhovor se členem Development týmu – fáze 1

### Vývojář 5

Ráda bych s Vámi nyní udělala rozhovor na téma zahájení agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak jste se připravoval na zahájení přechodu na Scrum?

*Já nijak, Scrum Master nám říká, co a jak, takže se tím řídím.*

2. Popište, prosím, jak se dle Vás liší role vývojáře ve Scrumu v porovnání s klasickým přístupem řízení projektů, ze kterého nyní přecházíte na agilní metodu Scrum.

*Moc nevím, jestli se moje role bude nějak lišit. Asi je to spíše o stylu spolupráce s klientem.*

3. Jsou pro Vás User Stories srozumitelné a dostatečně konkrétní?

*Pochopil jsem z toho, co má být výsledkem a proč se to má realizovat, takže snad v pořádku.*

4. Jak probíhal Planning na první Sprint?

*Pro mě bylo plánování dost náročné. V podstatě nejsem zvyklý odhady dělat téměř vůbec. A pak je dělat před klientem, a ještě když mají kolegové jiný názor je náročné.*

5. Popište, prosím, jak probíhají denní Standup schůzky.

*Někdy jsem přišel pozdě, takže na mě kolegové čekali. Taky se mi to občas zdálo zbytečné, protože tam kluci řeší věci, které mě nezajímají.*

6. Popište, prosím, jak dle Vás proběhl první Sprint? Jak se Vám pracovalo, měl jste s něčím problémy nebo se naopak něco zlepšilo?

*Bylo hodně schůzek. Bude mi asi trvat, než si zvyknu na Standup a prezentace klientovi. Ale jinak s prací jsem problém neměl.*

7. Popište, prosím, jak probíhalo Review po prvním Sprintu?

*Bylo to ok, naše práce se líbila, tak jsme byli rádi.*

8. Jaké oblasti jste probírali na první Retrospektivě?

*Řešilo se, že chodíme pozdě na Standup a že tam řešíme věci, které zdržují o statní.*

9. Využíváte nebo jste nově začali využívat některé metody agilního programování? Např.: Pair Programming, Coding Standard, Continuous Integration a Code Refactoring nebo jiné? Případně, které byste chtěli začít používat.

*Nevím, jestli to někdo řeší, ale v týmu jsme o tom nemluvili. Párové programování bych si rád vyzkoušel s nějakým zkušenějším kolegou, abych se tak naučil něco nového.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

## **Příloha 4 Odpovědi z rozhovorů – fáze 2**

### **Rozhovor se Scrum Masterem – fáze 2**

Ráda bych s Vámi nyní udělala druhý rozhovor na téma agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak byste popsal dosavadní dvou měsíční agilní vývoj na pilotním projektu ze svého pohledu?

*Pro mě osobně to byly velmi náročné dva měsíce. A myslím že nejen pro mě. Určitě jsem počítal s tím, že to nebude jednoduché, ale realita byla asi ještě složitější. Na druhou stranu po velmi náročném začátku se mnoho věcí začalo rychle zlepšovat. Během těchto dvou měsíců jsem získal hodně nových zkušeností, ale asi i proto mám nyní ještě větší potřebu se v této oblasti dále rozvíjet. Protože jsem si vědom, že jsme teprve na začátku i když základní procesy a přístupy nám již celkem fungují a situace se již zdá být stabilnější ať se jedná o samotný vývoj nebo jednotlivé praktiky Scrumu.*

2. Jak po dvou měsících agilního vývoje probíhá spolupráce mezi členy týmu? Změnil se jejich přístup k agilnímu vývoji a ke Scrumu?

*Řekl bych, že k zásadním změnám v přístupu nedošlo. Stále se projevují problémy přístupu jednoho z vývojářů, který se otevřeně staví proti agilnímu vývoji. U něj bohužel k žádné změně nedošlo a ostatním vývojářům jeho negativní postoje působí někdy problémy, protože je těžší se s ním dohodnout na spolupráci. V týmu máme také junior programátora, pro kterého je tento způsob práce nový a společně s náročností vývoje se občas také projevují problematické situace, které je potřeba řešit. Tři další vývojáři jsou téměř bez problémů, jak pracovně, tak i komunikačně a lidsky.*

2. Jak dlouho nyní Standup meetingy trvají a jaký je jejich průběh?

*Standupy se lehce zkrátily. Obvykle schůzka trvá 15 minut, maximálně 25 minut. Vývojáři si již na tyto schůzky celkem zvykli a schůzky jsou také méně chaotické. Stále ještě řešíme dochvilnost některých vývojářů, ale není to tak problematické jako na začátku.*

3. Popište prosím, jak probíhají nyní Planning meetingy a jakou metodu odhadu nakonec tým používá?

*I tyto schůzky se pomalu zlepšují a nejsou tak chaotické a dlouhé jako na samém začátku. V průběhu Sprintu se snažíme s Product Ownerem průběžně seznamovat vývojáře s novými funkcionalitami, aby na plánování byly již trochu připraveni a odhadovalo se jim lépe. Odhadujeme stále v hodinách. K zavedení změny v odhadech jsme se bohužel ještě nedostali.*

4. Jak dlouhé jsou sprinty a jaký je jejich průběh z vašeho pohledu? Je dosahováno cíle sprintů?

*Sprinty máme dvoutýdenní, abychom měli více času na vývoj. Ne vždy se nám podařilo dosáhnout cíle sprintu. Při jednom sprintu byl vývoj náročnější, než jsme očekávali a v druhém případě jsme řešili větší množství chyb. U zbývajících sprintů se nám podařilo cíle dosáhnout i když některé funkce ještě zůstávaly neotestované a testovalo se ještě průběhu dalšího sprintu.*

5. Jak probíhají v současnosti Review? Kdo se těchto schůzek účastní a jak probíhá samotná prezentace?

*Schůzky se účastní všichni vývojáři, já a Product Owner. Pouze na jedné schůzce byli přítomni další zástupci klienta. Prezentace je obvykle bez problémů. Myslím, že vývojáři si již celkem zvykli na tuto situaci a při prezentaci jsou si jistější.*

6. Popište prosím témata, která řešíte na retrospektivních schůzkách?

*Některé retrospektivy jsme bohužel vynechali, protože jsme byli pod časovým tlakem a dokončovali jsme některé funkcionality nebo testování z předešlého Sprintu. Takže to bylo na úkor těchto schůzek. V podstatě to vychází, že byla jedna Retrospektiva za měsíc místo dvou. Při těchto schůzkách jsme se obvykle zabývali problémy ve spolupráci mezi jednotlivci v týmu nebo problémy, které tým řeší při odhadování časové náročnosti.*

7. Jak je nyní nastavená Definition of Done oproti počátku Scrumu?

*Zde k žádné změně zatím nedošlo.*

8. Jak je vypočítána kapacita týmu na Sprint?

*Na vývojáře počítáme 36 hodin týdně. Zbývající 4 hodiny je interní režie, opravy chyb, aktualizace softwaru atd. Z těch 36 hodin ještě odečítáme schůzky a ceremonie, tak abychom dostali čistý čas na vývoj za týden. Tento výsledek pak vynásobíme dvěma, protože máme dvoutýdenní sprinty. Na jednoho vývojáře je kapacita na jeden sprint asi 67 hodin.*

9. Uveďte prosím, jaká máte očekávání na další měsíce.

*Chtěli bychom se určitě zlepšit v dosahování cílů Sprintu a abychom v rámci sprintu vždy zvládali i testování. Také bude pravděpodobně potřeba řešit personální obsazení v týmu vývojářů, protože to současně není zcela vyhovující. Dále bychom se chtěli věnovat agilním technikám programování.*

Na závěr rozhovoru bych Vás opět požádala o hodnocení jednotlivých členů týmu. Princip hodnocení je stejný jako při prvním rozhovoru. Hodnoťte na stupnici 1 až 10 týmovost, nasazení a aktivitu u jednotlivých členů Development týmu na tomto pilotním projektu. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup daného člena týmu. Na toto hodnocení máte nyní 30 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## **Rozhovor s Product Ownerem – fáze 2**

Ráda bych s Vámi nyní udělala druhý rozhovor na téma agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak na vás během dvou měsíců agilního vývoje působila spolupráce s vývojáři a Scrum Masterem?

*Dva měsíce nám utekly velmi rychle. Myslím, že jsme udělali velký kus práce i když jsme asi doufali, že toho zvládneme více. A to nejen v samotném vývoji, ale také v procesech samotných. Celkově se mi s týmem i se Scrum Masterem spolupracuje dobře. Máme ještě všichni rezervy, učíme za pochodu*

*a máme dost prostoru pro zlepšení, ale s každým sprintem se posouváme dopředu. Je to také velmi cenná zkušenost pro nás všechny. I když se může stát, že ne každému tento styl práce nakonec vyhovuje, tak je určitě dobré si ho vyzkoušet. Ohledně našeho týmu si myslím, že většina lidí bude s tímto přístupem spokojenější a bude podávat lepší výsledky práce, než tomu bylo dříve.*

2. Jakých schůzek se účastníte a jak dle vás tyto schůzky probíhají?

*Stále se účastním Planningu a Review. Až na výjimky se alespoň vzdáleně připojím i na Standupy. Celkově si myslím, že schůzky pomalu přestávají být tak chaotické a vyčerpávající. Také většina vývojářů s nimi přestává mít takové problémy jako na začátku. I například při Review jsou si více jistější.*

3. Kteří zástupci z vaší firmy se účastní Review?

*Této schůzky se účastním většinou sám. Pouze na jedné schůzce se mnou byl můj nadřízený, a ještě jeden projektový manažer. Oba byli s prezentací velmi spokojeni, ale bohužel z časových důvodů se již dalších prezentací neúčastnili.*

4. Jak jste spokojeni s plněním cílů jednotlivých Sprintů?

*Ve dvou případech se sice nepodařilo cíl sprintu dodržet, ale vím že tým dělá maximum, aby cíle sprintu dosáhl. Ve zbylých dvou případech bylo cíle dosaženo. Na začátek agilního vývoje to asi není špatné. V případech, kdy se nepodařilo dosáhnout cíle se jednalo pouze o malou část, kterou team nestihl realizovat anebo v druhém případě ještě opravoval chyby, na které se narazilo při testování.*

5. Jak hodnotíte kvalitu práce Development týmu?

*Vývojáři odvádějí velmi dobrou práci a kvalita produktu je vysoká. Chyby se vyskytují, ale tým je obratem opravuje. Ke mně se zásadní chyby moc nedostávají. Společně ladíme převážně jen drobné úpravy a detaily.*

6. Jsou nějaké nové změny nebo návrhy ohledně typu kontraktu mezi vaší firmou a dodavatelem?

*Ne, změnu smlouvy dosud nikdo neřešil. Ale vím, že dodavatel by o to měl zájem. Takže uvidíme v následujících měsících, jestli se tento typ spolupráce promítne do nové smlouvy.*

7. Uveďte prosím, jaká máte očekávání na další měsíce.

*Věřím, že i nadále bude naše spolupráce dobrá a výsledky se budou dále zlepšovat. Rád bych, aby se tým ještě rozšířil o dalšího vývojáře, protože naše plány s produktem narůstají a rádi bychom vývoj urychlili.*

Na závěr rozhovoru bych Vás požádala o hodnocení Development týmu jako celku. Do tohoto hodnocení prosím nezahrnujte Scrum Mastera. Princip hodnocení je stejný jako při prvním rozhovoru. Hodnoťte na stupnici 1 až 10 týmovost, nasazení a aktivitu týmu na tomto pilotním projektu. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup daného člena týmu. Na toto hodnocení máte nyní 30 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

## Rozhovor se členem Development týmu – fáze 2

### Vývojář 1

Ráda bych s Vámi nyní udělala druhý rozhovor na téma agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak byste popsal dosavadní dvou měsíční agilní vývoj na pilotním projektu ze svého pohledu?

*Tak začátek byl dost náročný, než jsme se nějak lépe zorganizovali a začali si zvykat na nová pravidla. Postupně se to celkem zlepšuje a některé věci, které nám na začátku dělali problémy, se již celkem začínají automatizovat. Ještě máme mnoho co zlepšovat, ale to tak bude stále. Vývoj na projektu jako takový, si myslím, že jde správným směrem bez zásadních problémů, což je pro nás velmi důležité.*

2. Jak se vám nyní tímto agilním způsobem spolupracuje s kolegy vývojáři?

*Ve spolupráci jsou určitě mezi námi rozdíly, a to celkem podstatné. Lze již velmi dobře odlišit komu tento styl práce vyhovuje a komu ne. Myslím si, že většina z nás je spokojená i když je to občas náročné. Ale pracuje se nám dobře, tak bychom rádi tímto směrem pokračovali. Ale asi se nám k tomu nepodaří přesvědčit jednoho našeho kolegu, který svůj negativní postoj dává najevo od samého začátku pilotního projektu. Myslím, že bychom to měli už nějak vyřešit. Také náš junior programátor to má někdy asi dost náročné, protože práce, která mu je někdy přidělena přesahuje jeho zkušenosti a znalosti.*

3. Jak hodnotíte práci Vašeho Scrum Mastera?

*Myslím, že svojí práci dělá dobře a že ho práce baví. Také se za běhu učí, takže si není samozřejmě tak jistý jako někdo, kdo těch zkušeností má na této pozici více. Zatím jsme nezvládli některé praktiky ze Scrumu úplně nejlépe, ale netvrdím, že je to vina pouze Scrum Mastera i když to je primárně on kdo se o to má starat. Ale myslím, že se mu celkem daří přesunutí do této nové role. O svůj tým se stará a snaží se zajistit vše potřebné, abychom fungovaly co nejlépe. A snaží se i podporovat tým v samostatnosti.*

4. Jak hodnotíte práci Vašeho Product Ownera?

*Řekl bych, že velmi dobře rozumí jejich byznysu. Umí vysvětlit potřeby, které jeho firma má a co od toho produktu očekávají. Je ochotný se pobavit o návrzích řešení a zajímají ho možnosti, které mu můžeme při realizaci nabídnout. Na schůzky se vždy pečlivě připravuje, takže dokáže vysvětlit požadavky a doplnit potřebné informace k User Stories.*

5. Došlo k nějakým změnám v tom, jaký mají nyní průběh Standup schůzky?

*Už jsme si na ně v celku zvykli. Takže to snad už nikdo nebereme jako velký problém. Myslím, že bychom je mohli, ale ještě zkrátit. Občas se nám to stejně protáhne a prostě se za povídáme nebo se začne řešit nějaký problém.*

6. Při prvním sprintu byl pro vás Planning velmi náročný časově i obsahově. Jak je tomu nyní?

*Oproti té první plánovací schůzce jsme na tom určitě již lépe. Nyní již Planning probíhá rychleji a je lépe organizovaný. Odhady děláme stále v hodinách a řekl bych že to už tak asi zůstane.*

7. Od Scrum Mastera a Product Ownera vím, že během dvou Sprintů se nepodařilo dosáhnout cíle. Co bylo dle vás příčinou?

*V jednom případě jsme podcenili náročnost vývoje některých částí a vývoj nám trval déle. Takže odhady neodpovídaly realitě. Měli jsme si vzít na odhadování více času a lépe si to celé promyslet, protože daná funkcionalita byla složitější. Také jsme na našeho Junior programátora zadali poměrně složitý úkol, se kterým si nevěděl moc rady a my jsme neměli čas s tím pomoci. Nakonec jsme to za něj museli dodělat. V druhém případě jsme měli hodně chyb na frontendové části a jejich oprava byla časově náročná. Stav byl takový, že to nebylo vhodné prezentovat klientovi a bylo potřeba nejdříve tyto věci vyladit. Tomu prvnímu problému se dalo asi předejít mnohem snadněji než u toho druhého nesplnění cíle.*

8. Začali jste již se zaváděním alespoň některých agilních metod programování?

*Zde jsme udělali pokrok a podařilo se nám posunout se dál s Continuous Integration. Nyní jsou zdrojové kódy sdílené ve verzovacím systému Git a každé odeslání nového kódu startuje automaticky kompilaci na serveru. Následně se spustí unit testy a výsledky se pak pošlou vývojářům. Nemáme zdaleka vše pokryté unit testy, ale začali jsme je psát, což je posun dopředu. Také bychom chtěli zahrnout i integrační testy a případně další analýzy.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Princip obou hodnocení je stejný jako při prvním rozhovoru. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## **Rozhovor se členem Development týmu – fáze 2**

### **Vývojář 2**

Ráda bych s Vámi nyní udělala druhý rozhovor na téma agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak byste popsal dosavadní dvou měsíční agilní vývoj na pilotním projektu ze svého pohledu?

*Myslím, že se nám práce tímto způsobem daří. Samozřejmě to není nic ideálního, ale za ty dva měsíce jsme udělali pokrok a stále se zlepšujeme. Mě osobně práce tímto způsobem vyhovuje a počítám s tím, že budeme takto pokračovat. Máme ještě rezervy v některých oblastech vývoje nebo i ve spolupráci, ale myslím, že po těch dvou měsících ty výsledky jsou pozitivní. A když bychom se dále snažili zlepšovat, tak se to ještě odrazí na naší práci. Jsou zde také nějaké věci, které se nepodařilo realizovat, například jsme nesplnili některé cíle Sprintu.*

2. Jak se vám nyní tímto agilním způsobem spolupracuje s kolegy vývojáři?

*Kolegy znám všechny již delší dobu, takže jsem na ně zvyklý. I když k nějaké změně ve spolupráci určitě došlo. Řešíme situace, které dříve nebyly a jsme určitě ve větším kontaktu než dříve. Občas se nějaké problémy samozřejmě objevují, protože se samozřejmě na všem neshodneme a máme jiné názory na určité postupy. Vliv na naši spolupráci má určitě i to, jak si postupně zvykáme na*



*nové postupy a pravidla ve Scrumu. Což se u každého z nás trošku liší. Ne každému z nás tento způsob spolupráce vyhovuje, takže je to pak na tom také vidět. Ale myslím že celkově je to dobré.*

3. Jak hodnotíte práci Vašeho Scrum Mastera?

*Má kolem týmu spoustu práce, a ne vše se nám vždy daří podle očekávání, takže určitě musí řešit složité situace. Snaží se nám pomáhat a také poslouchá naše názory. I když je toho na něj někdy příliš a chybí mu určitě ještě praktické zkušenosti, tak si myslím, že tuto práci zvládá dobře.*

4. Jak hodnotíte práci Vašeho Product Ownera?

*Ze spolupráce s klientem jsem měl trošku obavy, ale nyní vše funguje dobře. Dává nám dobrou zpětnou vazbu na naši práci, takže poměrně rychle dokážeme udělat ještě nějaké úpravy tak, aby vše fungovalo dle jeho představ.*

5. Došlo k nějakým změnám v tom, jaký mají nyní průběh Standup schůzky?

*Už to naštěstí není tak zmatené. Už ty schůzky ani netrvají tak dlouho. Kolem čtvrt hodiny většinou. Občas na někoho čekáme, ale už se to nestává tak často.*

6. Při prvním sprintu byl pro vás Planning velmi náročný časově i obsahově. Jak je tomu nyní?

*Je to nyní lepší, už to pro nás není tak náročné jako byly ty první Planning schůzky. Už víme dopředu, co nás čeká a co budeme řešit, tak se nám už plánuje lépe.*

7. Od Scrum Mastera a Product Ownera vím, že během dvou Sprintů se nepodařilo dosáhnout cíle.

Co bylo dle vás příčinou?

*Teď naposledy se nám stalo, že jsme se potýkali s velkým množstvím chyb. Opravy a uvedení do potřebného stavu trvalo skoro dva dny, se kterými jsme ale nepočítali. Takže jsme nebyli schopni na Review výsledek odprezentovat. V jednom Sprintu se nám stalo, že jsme práci špatně odhadli a ve finále nám vývoj zabral více času.*

8. Začali jste již se zaváděním alespoň některých agilních metod programování?

*Ano, a je super, že jsme začali psát automatické testy i když někdy není čas a těch testů bych chtěl psát více. Ale jsem rád za jakoukoliv automatizaci.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Princip obou hodnocení je stejný jako při prvním rozhovoru. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## **Rozhovor se členem Development týmu – fáze 2**

### **Vývojář 3**

Ráda bych s Vámi nyní udělala druhý rozhovor na téma agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak byste popsal dosavadní dvou měsíční agilní vývoj na pilotním projektu ze svého pohledu?

*Myslím, že dosahujeme dobrých výsledků na projektu a že jsme se za ty dva měsíce naučili a zvykli si na základní praktiky Scrumu. Na začátku ta změna byla poměrně velká. Měl jsem obavy abychom neohrozili projekt jako takový, ale nyní si myslím, že to bylo dobré rozhodnutí a že bychom měli tak to dál pokračovat.*

2. Jak se vám nyní tímto agilním způsobem spolupracuje s kolegy vývojáři?

*Já jsem se spoluprací s kolegy spokojen. Většina jsou velmi zkušené programátory a pracuje se s nimi velmi dobře. Víím, že jeden z našich kolegů zrovna nesdílí náš pozitivní přístup k agilnímu vývoji, ale nevím, jestli to lze nějak změnit. Jinak jsem za tento tým velmi rád a spolupráce mě s kolegy baví.*

3. Jak hodnotíte práci Vašeho Scrum Mastera?

*Mně se s ním spolupracuje perfektně. Vzhledem k tomu, že jsem dříve praktickou zkušenost se Scrumem neměl, tak mi pomohl se zorientovat. Je stále k dispozici, když něco potřebujeme řešit a snaží se nám pomoci nebo poradit.*

4. Jak hodnotíte práci Vašeho Product Ownera?

*Jeho práce je pro nás určitě přínosná, protože jsme s ním přímo v kontaktu a můžeme se ho rovnou ptát na nějaké nejasnosti a doladit s ním zadání. Má o produktu velmi dobrý přehled, takže dokáže i vysvětlit důvody proč nějaké věci chtějí a jiné naopak ne. Takže myslím, že svojí práci dělá určitě velmi dobře.*

5. Došlo k nějakým změnám v tom, jaký mají nyní průběh Standup schůzky?

*Já jsem za tyto schůzky od začátku velmi rád a nyní nám již nezabírají tolik času. Celkem jsme se naučili stručně shrnout základní informace o tom co jsme udělali a co dělat budeme. Někdy se spíše zasekneme na tom, když někdo od někoho něco potřebuje nebo v případě řešení nějakého problému.*

6. Při prvním sprintu byl pro vás Planning velmi náročný časově i obsahově. Jak je tomu nyní?

*Už jsme se o hodně zlepšili. Nyní plánování není tak zmatené jako bylo na začátku. Trochu mě mrzí, že jsme zůstali u odhadování v hodinách a nezkusili jsme nějaké jiné způsoby.*

7. Od Scrum Mastera a Product Ownera víím, že během dvou Sprintů se nepodařilo dosáhnout cíle. Co bylo dle vás příčinou?

*Ano, cíl jsme nesplnili dvakrát. Jednou kvůli špatným odhadům a podruhé pak kvůli opravám chyb.*

8. Začali jste již se zaváděním alespoň některých agilních metod programování?

*To primárně řeší moji kolegové, ale už máme na některé části kódu unit testy, které se automaticky spouštějí poté, co odešleme nový kód na server. Chci se do psaní testu také zapojit, abychom jich měli co nejvíce.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Princip obou hodnocení je stejný jako při prvním rozhovoru. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto

hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## Rozhovor se členem Development týmu – fáze 2

### Vývojář 4

Ráda bych s Vámi nyní udělala druhý rozhovor na téma agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak byste popsal dosavadní dvou měsíční agilní vývoj na pilotním projektu ze svého pohledu?

*Začátek byl opravdu hodně chaotický, postupně s to vcelku zlepšilo a některé aktivity už nyní nedělají v týmu takové problémy. Pro mě je to stále hodně schůzek a administrativy kolem a pak zbytečný tlak na splnění cílů Sprintu, abychom byli schopní předvést, že jsme něco dokončili.*

2. Jak se vám nyní tímto agilním způsobem spolupracuje s kolegy vývojáři?

*Svých kolegů si vážím, dělají dobrou práci. Vypadá to, že jim tento styl práce vyhovuje lépe, než mě osobně. Možná spíše já naopak někdy způsobuji problémy jim, protože si myslím, že bych byl efektivnější při klasickém stylu vývoje.*

3. Jak hodnotíte práci Vašeho Scrum Mastera?

*Snaží se dělat svoji práci, jak nejlépe asi v tuto chvíli umí. Má bohaté zkušenosti jako projektový manažer, takže to mu možná v něčem pomáhá. Základní věci o Scrumu zná a snaží se je používat i když všechno za ty dva měsíce prostě ještě nefunguje, jak by mělo a je otázka, jestli nebo kdy bude. Zatím ale celkem opomenul i základní věci jako Definition of Done.*

4. Jak hodnotíte práci Vašeho Product Ownera?

*Role Product Ownera na tomto projektu funguje dobře. Dokáže srozumitelně vysvětlit co za výsledek od nás očekává a jaké má představy. Zatím se nestalo, že by během Sprintu byly požadavky změnil a ohrozil tak vývoj.*

5. Došlo k nějakým změnám v tom, jaký mají nyní průběh Standup schůzky?

*Standupu jsou asi o něco kratší než na samém začátku. Myslím, že by měly být ale ještě o něco rychlejší, protože i 15 nebo 20 minut je stále hodně.*

6. Při prvním sprintu byl pro vás Planning velmi náročný časově i obsahově. Jak je tomu nyní?

*V porovnání s prvním plánováním jsme na tom určitě výrazně lépe. Jak časově, tak organizačně během schůzky. Otázka sama je potom přesnost těch odhadů, které se tam stanoví.*

7. Od Scrum Mastera a Product Ownera vím, že během dvou Sprintů se nepodařilo dosáhnout cíle.

Co bylo dle vás příčinou?

*Příčina byla určitě v tom, že jsme měli hodně problematický vývoj, ve kterém se vyskytly chyby a ty jsme nestihli opravit v čas. A pak také ve sprintu předtím nebyl čas na to si vše lépe promyslet a rozvrhnout. Proto potom bylo překvapení kolik času vlastně potřebujeme.*

8. Začali jste již se zaváděním alespoň některých agilních metod programování?

*Je dobře, že se začalo řešit Continuous Integration a psaní unit testů. To je pro začátek určitě krok dobrým směrem a snad to vydrží i dále a automatické testy budou přibývat.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Princip obou hodnocení je stejný jako při prvním rozhovoru. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

---

## Rozhovor se členem Development týmu – fáze 2

### Vývojář 5

Ráda bych s Vámi nyní udělala druhý rozhovor na téma agilní transformace na pilotním projektu. Chtěla bych Vás požádat o souhlas s nahráváním celého rozhovoru s tím, že uvedené informace budou použity v rámci diplomové práce. Název firmy, Vaše jméno a příjmení i název projektu bude pro tyto účely anonymizováno.

1. Jak byste popsal dosavadní dvou měsíční agilní vývoj na pilotním projektu ze svého pohledu?

*Pro mě je tento typ vývoje velmi rychlý a tím, že nemám tolik zkušeností s programováním jako kolegové, tak je to někdy trochu náročné, protože pokud se na něčem zdržím více než bych očekával, tak mohu způsobit problém celému týmu. Protože pak to za mě musí někdo dodělat nebo opravit, a přitom kolegové mají své práce také dost. Na druhou stranu se toho zase mohu poměrně dost a rychle naučit. Takže jsem určitě za tuto zkušenost rád, ale potřeboval bych občas více pomoci od svých kolegů. Jinak je to pro mě občas trochu stresující, určitě více než práce před pilotním projektem.*

2. Jak se vám nyní tímto agilním způsobem spolupracuje s kolegy vývojáři?

*Jsem rád, že jsem v týmu velmi dobrých programátorů, od kterých se toho mohu dost naučit. Když se jich na něco zeptám, tak jsou schopni mi poradit. Což mi hodně usnadní práci. Bylo by skvělé, kdyby na mě měli občas více času a mohli jsme pracovat více společně. Uvítal bych i větší kontrolu nad mojí prací, aby pak nedocházelo zbytečně k chybám, protože něco neudělám dobře.*

3. Jak hodnotíte práci Vašeho Scrum Mastera?

*Asi dělá svoji práci dobře. Neumím to úplně porovnat, protože s jiným Scrum Masterem jsem nepracoval.*

4. Jak hodnotíte práci Vašeho Product Ownera?

*Zadává nám práci a vysvětluje své požadavky. Máme co dělat a víme co budeme dělat v následujících Sprintech, takže asi svoji práci dělá, jak by měl.*

5. Došlo k nějakým změnám v tom, jaký mají nyní průběh Standup schůzky?

*Určitě jsme už nyní rychlejší a schůzka netrvá obvykle tak dlouho. Už jsem si na ně docela i zvykl a opravdu jen výjimečně se opozdím.*

6. Při prvním sprintu byl pro vás Planning velmi náročný časově i obsahově. Jak je tomu nyní?

*Myslím, že tady jsme se zlepšili opravdu hodně, a i já už si na odhadování také zvykám. Rád bych si vyzkoušel i odhady v jiných jednotkách, než jsou hodiny. To jsme zatím ale nepoužili a nevyzkoušeli.*

7. Od Scrum Mastera a Product Ownera vím, že během dvou Sprintů se nepodařilo dosáhnout cíle.

Co bylo dle vás příčinou?

*Mysleli jsme si, že vývoj bude jednodušší, tak ty odhady byly poměrně nízké. Pak při samotném vývoji se ukázalo, že je tam vícepráce. Já jsem tam bohužel měl dost náročný úkol, se kterým jsem si sám nevěděl úplně rady a potřeboval jsem pomoci od kolegů. Ale měli na mě opravdu jen málo času. Nakonec to dopadlo tak, že to kolega po mně převzal a práci za mě dokončil.*

8. Začali jste již se zaváděním alespoň některých agilních metod programování?

*Bohužel jsme se zatím vůbec nedostali k párovému programování a ani kolegové moc nemají čas se podívat na můj kód a poradit mi co je potřeba opravit anebo napsat lépe. Ale máme už nějaké unit testy, které kolegové připravují, takže snad se dostaneme brzo i k dalším metodám.*

Na závěr rozhovoru bych Vás požádala o sebehodnocení a hodnocení členů Development týmu. Princip obou hodnocení je stejný jako při prvním rozhovoru. Sebehodnocení se týká Vaší aktivity, nasazení a Vašeho týmového přístupu na tomto pilotním projektu. Ohodnoťte se na stupnici 1 až 10 v těchto třech oblastech. Hodnota 1 znamená nejnižší aktivitu, nasazení a žádný týmový přístup. Hodnota 10 znamená nejvyšší aktivitu, nasazení a nejlepší týmový přístup. Poté prosím rozdělte fiktivní odměnu 20 000 Kč mezi své kolegy dle svého uvážení. Odměna se netýká Scrum Mastera ani Product Ownera, ani Vás samotného, takže částku dělíte pouze mezi své čtyři kolegy vývojáře. Na toto hodnocení máte nyní 60 minut času. Pokud nemáte nyní otázky k hodnocení, tak Vám v tuto chvíli děkuji za rozhovor.

## Příloha 5 Vzájemné hodnocení členů Development týmu

### Hodnocení vývojáře 1

Hodnotitel	Fiktivní odměna	
	Fáze 1	Fáze 2
Vývojář 2	8 000 Kč	7 000 Kč
Vývojář 3	6 000 Kč	7 000 Kč
Vývojář 4	5 000 Kč	5 000 Kč
Vývojář 5	6 000 Kč	7 000 Kč
Celkem	25 000 Kč	26 000 Kč

Zdroj: Vlastní zpracování

### Hodnocení vývojáře 2

Hodnotitel	Fiktivní odměna	
	Fáze 1	Fáze 2
Vývojář 1	8 000 Kč	7 000 Kč
Vývojář 3	5 000 Kč	7 000 Kč
Vývojář 4	5 000 Kč	5 000 Kč
Vývojář 5	5 000 Kč	6 000 Kč
Celkem	23 000 Kč	25 000 Kč

Zdroj: Vlastní zpracování

### Hodnocení vývojáře 3

Hodnotitel	Fiktivní odměna	
	Fáze 1	Fáze 2
Vývojář 1	7 000 Kč	7 000 Kč
Vývojář 2	6 000 Kč	7 000 Kč
Vývojář 4	5 000 Kč	5 000 Kč
Vývojář 5	4 000 Kč	5 000 Kč
Celkem	22 000 Kč	24 000 Kč

Zdroj: Vlastní zpracování

#### Hodnocení vývojáře 4

Hodnotitel	Fiktivní odměna	
	Fáze 1	Fáze 2
Vývojář 1	2 000 Kč	1 000 Kč
Vývojář 2	3 000 Kč	1 000 Kč
Vývojář 3	3 000 Kč	0 Kč
Vývojář 5	5 000 Kč	2 000 Kč
Celkem	13 000 Kč	4 000 Kč

Zdroj: Vlastní zpracování

#### Hodnocení vývojáře 5

Hodnotitel	Fiktivní odměna	
	Fáze 1	Fáze 2
Vývojář 1	3 000 Kč	5 000 Kč
Vývojář 2	3 000 Kč	5 000 Kč
Vývojář 3	6 000 Kč	6 000 Kč
Vývojář 4	5 000 Kč	5 000 Kč
Celkem	17 000 Kč	21 000 Kč

Zdroj: Vlastní zpracování





## Příloha 7 Coaching backlog

Coaching backlog – měsíční plán				
Priorita	Téma	Obsah	Účastníci	Cíl
1	Standup	Školení o praktikách a pravidlech Standup schůzek dle doporučení v kapitole 3.3.3 Meetingy aktivity	Development tým	Zkrátit dobu trvání schůzek na nutné minimum.
2	T-Shirt Sizes	Školení o metodě odhadů určené pro odhady pracnosti větších množství velkých položek (Epic).	Development tým	Začít využívat metodu T-Shirt Sizes místo současného odhadování v hodinách.
3	Planning Poker	Školení o metodě odhadů určené pro menší položky v Backlogu.	Development tým	Začít využívat metodu Planning Poker místo současného odhadování v hodinách.
4	Retrospective	Školení o metodách pro pořádání retrospektivních schůzek (Retrospektiva s lepitky, Časová osa, Fishbone (rybí kost), Loď, ESVP). Na závěr tým vybere jednu z praktik, kterou začne využívat.	Development tým	Výběr praktiky pro retrospektivní schůzky
5	Úvod do Scrumu	Obecný úvod do agilního vývoje a metody Scrum přizpůsobený znalostem a zkušenostem potenciálního nového člena týmu. Zaškolení včetně praktik a pravidel, které jsou na pilotním projektu využívány současným týmem.	Pro nové členy týmu	V případě, že nastoupí do týmu nový člen, doplnit jeho znalosti a informace o metodě Scrum na potřebnou úroveň a seznámit jej s procesy a pravidly, které současný tým již využívá.

Zdroj: Vlastní zpracování

## Příloha 8 Pět příčin selhávání týmů – návrhy pro překonání disfunkce

**Nezájem o výsledek:** členové týmu jsou primárně zaměřeni na své vlastní výsledky, upřednostňují před společným cílem

- **Návrhy:** zveřejnit v rámci firmy i u klienta výsledky týmu. Navrhnout odměny dle dosažených výsledků.
- **Praktické použití:** využít opět vzájemné hodnocení členů týmu s fiktivní odměnou, ale tentokrát již tuto odměnu členům týmu skutečně vyplátit.

**Vyhýbání se zodpovědnosti:** členové týmu se vyhýbají odpovědnosti nebo činí odpovědné druhé.

- **Návrhy:** cíle a závazky týmu zveřejnit v rámci firmy. Kontrolovat progres. Přenechat primární odpovědnost na týmu.
- **Praktické použití:** cíl Sprintu vždy umístit na viditelné místo (tabule, nástěnka), aby i ostatní týmy viděli závazek týmu.

**Neochota k závazku:** protože se členové týmu obávají konfliktu nejsou ochotní přijmout závazky.

- **Návrhy:** Nastavit jasné termíny a očekávání. Dodržovat plány, ale počítat i s neočekávanými situacemi.
- **Praktické použití:** srozumitelné definice cílů Sprintů, nastavení Definition of Done, která stanovuje očekávaný standard.

**Obava z konfliktu:** členové týmu nejsou ochotni zdravě debatovat o problémech, drží se raději stranou a raději volí umělou harmonii.

- **Návrhy:** Podporovat konstruktivní konflikt a jeho řešení. Zaměřovat se na problémy a myšlenky, nikoliv na konkrétní členy týmu.
- **Praktické použití:** zahájit otevřenou komunikaci s týmem o negativních postojích, které se v týmu objevují, nechat každého člena týmu, aby se k situaci vyjádřil a navrhl řešení.

**Nedostatek důvěry:** členové týmu mají obavy ze zranitelnosti vůči sobě navzájem a nechtějí, aby ostatní věděli, že chybují nebo jim chybí znalosti.

- **Návrhy:** Vytvářet bezpečné prostředí, ve kterém nikdo z týmu nebude mít obavu přiznat svou chybu nebo neznalost.
- **Praktické použití:** Scrum Master může tyto obavy zmírnit, když bude sám přiznávat své vlastní chyby a bude například otevřeně hovořit o tom, že ještě nemá všechny potřebné zkušenosti nebo znalosti metodou Scrum.

Zdroj: Vlastní zpracování

## Příloha 9 Podpora týmu ve Storming fázi

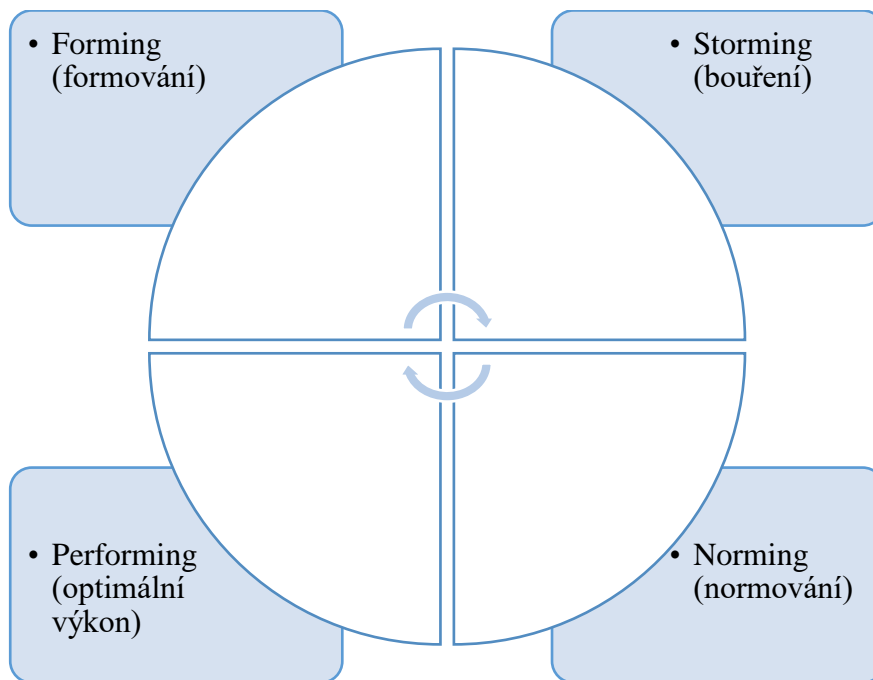
<b>Podpora týmu ve Storming fázi</b>			
<b>Komunikace</b>	<b>Spolupráce</b>	<b>Konflikty</b>	<b>Odpovědnost</b>
Povzbudit komunikaci mezi junior programátorem (vývojář 5) a ostatními členy týmu. Zapojit tohoto člena týmu do všech aktivit, hovořit s ním o náročnosti jeho práce a o zvládnání stresu.	Podpořit zejména spolupráci Junior programátora se seniorními členy týmu. Například pomocí párového programování nebo při kontrole kódu.	Zaměřit se na konfliktní situace mezi vývojářem 4 a zbývajícími členy týmu a sledovat další vývoj situace.	Po realizaci školení pro Standup schůzky a uvedení doporučených praktik do praxe, přenechat odpovědnost za průběh těchto meetingů na samotném týmu. Pouze sledovat jak se týmu další tyto schůzky zefektivnit a poskytovat případné poradenství.
Zvážit zda uspořádání stolů netvoří bariéry pro komunikaci mezi některými členy týmu, případně realizovat změny.	Využívat agilní praktiku párového programování nejen pro junior programátora ale i mezi ostatními členy týmu.	Hledat příčiny konfliktů s vývojářem 4 a aktivně je s tímto členem týmu řešit.	Vzhledem k bezproblémovým Review schůzkám, přenechat přípravu těchto schůzek na samotném týmu. Ale stále sledovat a dohlížet na průběh těchto setkání.
Zajistit také občasné mimopracovní aktivity pro celý tým (sport, kulturní akce), aby byla podpořena komunikace v mimopracovní rovině.	Diskutovat s týmem o dalších zlepšeních spolupráce, ale přenechat aktivitu v nápadech a řešeních na samotných vývojářích a poskytovat pouze potřebnou podporu.	Pokud není potřeba tak do nových konfliktů mezi členy týmu ihned nezasahovat a nechat na samotných členech týmu, jestli jsou schopni některé situace vyřešit.	Nechat tým rozhodnout o tom jakou praktiku budou využívat u retrospektivní schůzky. Pouze vysvětlit doporučené principy těchto praktik.

Zdroj: Vlastní zpracování

## Příloha 10 Tuckmanův model

### Ve které fázi se nachází váš tým?

Prosím každého člena týmu, aby přiřadil značku do té části kruhu, která dle jeho názoru odpovídá současné fázi vývoje týmu dle Tuckmanova modelu.



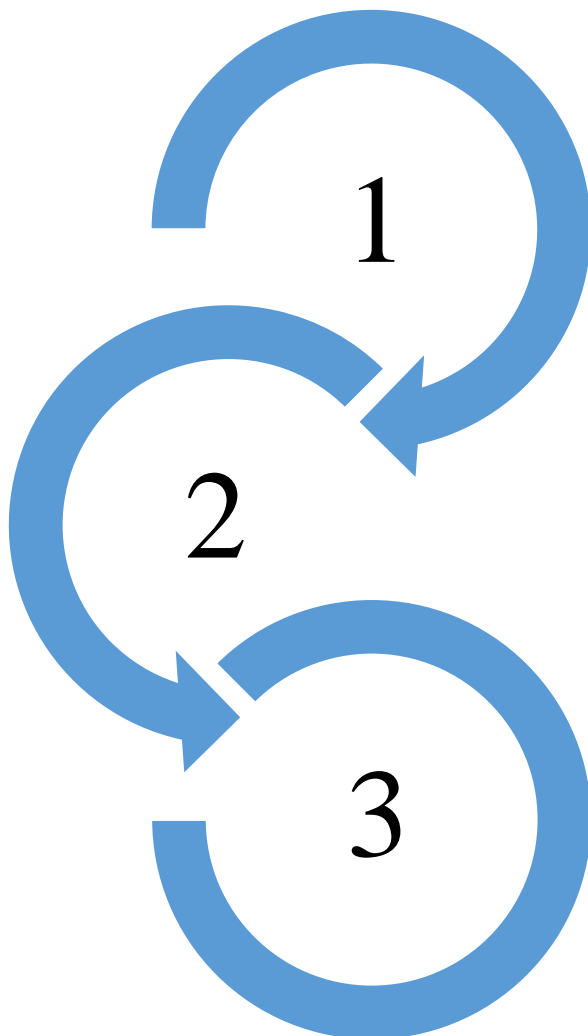
Zdroj: Vlastní zpracování

## Standup Meeting – 3 otázky



Pondělí až pátek v **10:00**

Doba trvání schůzky **5 minut**

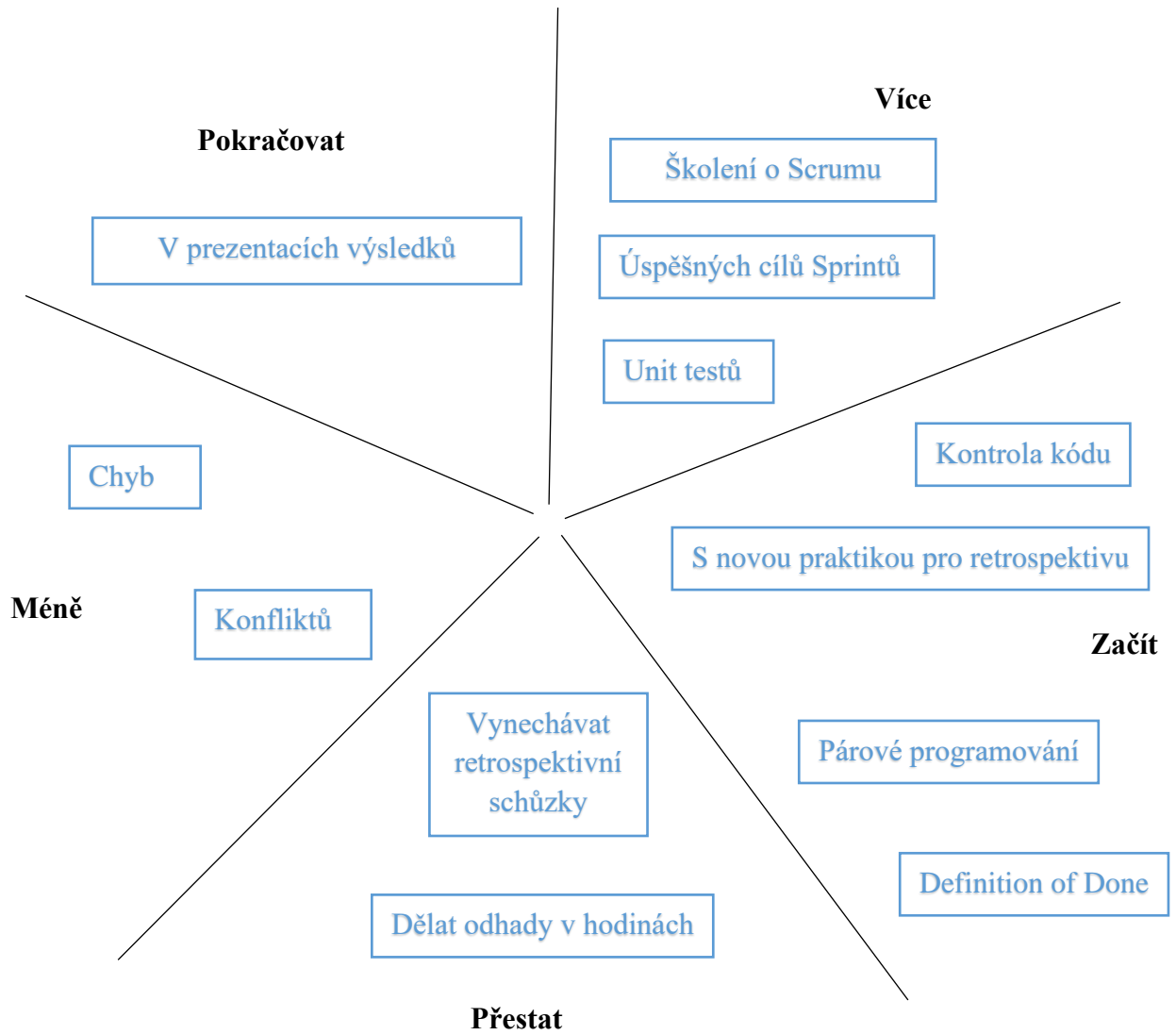


Co jsem dokončil včera, co pomohlo týmu dosáhnout cíle sprintu?

Co dokončím dnes, co pomůže týmu dosáhnout cíle sprintu?

Existují problémy, které nám stojí v cestě k cíli sprintu?

## Příloha 12 Retrospektiva s lepičky



Zdroj: Vlastní zpracování

## Příloha 13 Vzor agilní smlouvy

**TATO SMLOUVA O AGILNÍM VÝVOJI A IMPLEMENTACI SOFTWARE (DÁLE JEN „SMLOUVA“) BYLA UZAVŘENA NÍŽE UVEDENÉHO DNE, MĚSÍCE A ROKU MEZI TĚMITO SMLUVNÍMI STRANAMI**

Jméno: .....

IČO: .....

Místo podnikání: .....

Telefon: .....

E-mail: .....

(dále jako „**Objednatel**“)

a

Firma: .....

IČO: .....

Sídlo: .....

Za kterou jedná: .....

Telefon: .....

E-mail: .....

(dále jako „**Zhotovitel**“)

(Objednatel a Zhotovitel dále též společně jako „**Smluvní strany**“ a každý jednotlivě jako „**Smluvní strana**“)

### **SMLUVNÍ STRANY UJEDNÁVAJÍ NÁSLEDUJÍCÍ:**

#### **Preambule**

Smluvní strany uzavírají tuto Smlouvu za účelem úpravy vzájemných práv a povinností při takzvaném **agilním vývoji softwaru**.

Mezi způsoby vývoje softwaru patří tradiční způsob vývoje a agilní vývoj.

**Tradiční způsob vývoje softwaru** je založen na předem detailně stanoveném rozsahu dodávky softwaru, kdy je rozsáhle definována přesná funkčnost budoucího produktu a zejména cena. Tento způsob vývoje má ovšem své nevýhody v tom, že často nejsou požadavky objednatele dost dobře předem známy, dochází ke změnám v zadání, což prodlužuje a prodražuje celý projekt.

**Agilní vývoj softwaru** je naopak založen na co nejrychlejším dodání funkčního základu projektu s tím, že ke zdokonalení a přidání dalších funkcí dochází v průběhu, tedy za ostrého provozu. Vývoj projektu probíhá v takzvaných sprintech, které zahrnují analýzu požadavků, vývoj a implementaci. Tyto sprinty se opakují a v rámci nich dochází k postupnému upřesňování požadavků na funkčnost projektu.

Tato Smlouva tedy upravuje práva a povinnosti stran v rámci takzvaného **agilního vývoje softwaru**.

## 1. Úvodní ustanovení

- 1.1. Tato Smlouva má povahu rámcové smlouvy, která upravuje sjednávání dílčích smluv o dílo.
- 1.2. „**Software**“ pro účely této Smlouvy znamená počítačový program vyvíjený Zhotovitelem podle této Smlouvy.
- 1.3. „**Hardware**“ pro účely této Smlouvy znamená nezbytné hmotné elektronické vybavení, které je potřebné k fungování Softwaru.
- 1.4. Software lze popsat a určit podstatnými znaky tak, že:
  - 1.4.1. Software má sloužit k:  
.....
  - 1.4.2. Hlavními funkcemi a podstatnou náležitostí Softwaru bude: .....  
.....  
.....  
.....  
.....
  - 1.4.3. Software bude funkční na následující platformě:  
.....
  - 1.4.4. Software bude Zhotovitelem programován v jazyce:  
.....  
.....
  - 1.4.5. Minimální požadavky na Hardware jsou:.....  
.....  
.....  
.....
- 1.5. „**Dokumentace**“ pro účely této Smlouvy znamená jakékoliv dokumenty v českém či anglickém jazyce, které mají být dodány Zhotovitelem a jsou určeny k použití společně se Softwarem, včetně všech manuálů, návodů, licenčních dokumentů. V této Smlouvě tam, kde se hovoří o Softwaru, platí, že je tím myšlena i Dokumentace.
- 1.6. „**Sprint**“ pro účely této Smlouvy znamená jednu jednotlivou fázi vývoje Softwaru, při kterém dochází k analýze, vývoji a implementaci Softwaru. Sprint je zahájen definováním požadavků na nové funkčnosti Softwaru; obvykle následuje vytvoření technické specifikace, návrh funkčního modelu, na který navazuje samotný vývoj a testování. Sprint je ukončen předáním a uvedením Sprintu do provozu. Každý Sprint je tak vymezen dílčí smlouvou o dílo; tam, kde se v této Smlouvě hovoří o Sprintu, je tím myšlena dílčí smlouva o dílo a naopak.

## 2. Předmět Smlouvy

- 2.1. Zhotovitel se zavazuje postupně vyvíjet a implementovat Software ve Sprintech za podmínek a za cenu stanovenou v této Smlouvě. Objednatel se zavazuje Sprints včetně



případné Dokumentace převzít a uhradit za jejich provedení Zhotoviteli odměnu za podmínek a ve výši dohodnuté v této Smlouvě.

2.2. Smluvní strany se zavazují dohodnout obsah Sprintu v průběhu vývoje Softwaru. Nebude-li Smluvními stranami dohodnuto jinak, budou se Sprints a vztahy z nich řídit touto Smlouvou a podmínkami stanovenými v této Smlouvě.

2.3. Sprint bude Smluvními stranami definován alespoň těmito údaji:

2.3.1. **obsahovou definicí Sprintu**, tedy požadavky Objednatele na novou funkcionalitu či na změnu předcházející funkcionality Softwaru, opravu chyb a podobně.

2.3.2. **odhad časové náročnosti** na realizaci Sprintu, tedy hodinová doba práce potřebná k tomu, aby došlo k dokončení Sprintu.

2.3.3. **termín realizace**, respektive datum dokončení a předání Sprintu.

2.3.4. zda bude Sprint obsahovat **Dokumentaci** a v jakém rozsahu.

2.4. Sprint může být dále definován těmito údaji:

2.4.1. **hodinovou odměnou** Zhotovitele za realizaci Sprintu. V takovém případě se za učení odměny považuje takto dohodnutá odměna namísto odměny dohodnuté v této Smlouvě.

2.4.2. **maximální částkou odměny** Zhotovitele za realizaci Sprintu.

### 3. Uzavření dílčí smlouvy

3.1. Za Objednatele je oprávněna dílčí smlouvu uzavřít, tedy Sprint definovat, zaslat návrh Sprintu Zhotoviteli a jednat o podmínkách Sprintu, pouze pověřená osoba (dále jen „**Pověřená osoba Objednatele**“).

3.2. Za Pověřenou osobu Objednatele se považuje:

3.2.1. ....

3.3. Dílčí smlouva vzniká potvrzením návrhu Sprintu pověřenou osobou Zhotovitele (dále jen „**Pověřená osoba Zhotovitele**“).

3.4. Za Pověřenou osobu Zhotovitele se považuje:

3.4.1. ....

3.5. Potvrzení Sprintu s uvedením jiné ceny je považováno za protinávrh. Potvrzení Sprintu s jinou nepodstatnou změnou je považováno za protinávrh.

### 4. Místo plnění

4.1.1. Software, respektive jednotlivé Sprints, budou dodány a implementovány na adrese

.....  
.....  
.....

## **5. Práva a povinnosti Objednatele**

- 5.1. Objednatel se zavazuje poskytnout Zhotoviteli a jeho zaměstnancům veškerou potřebnou součinnost pro to, aby Zhotovitel mohl splnit své povinnosti podle této Smlouvy.
- 5.2. Objednatel se zavazuje zajistit zaměstnancům Zhotovitele všechny potřebné přístupy, zejména vstup do svých prostor a k technickým prostředkům v rozsahu, který bude nutný ke splnění povinnosti Zhotovitele dle této Smlouvy.
- 5.3. Objednatel se zavazuje zajistit zaměstnancům Zhotovitele vhodné, hygienické a bezpečné podmínky pro činnosti, které budou vykonávány v sídle Objednatele.
- 5.4. Dokumenty vytvořené Zhotovitelem za účelem splnění této Smlouvy (například analytické dokumenty, návrh softwarové architektury, zdrojové kódy) patří k know-how Zhotovitele. Objednatel se zavazuje vynaložit veškeré možné úsilí k tomu, aby se tyto dokumenty nedostaly do dispozice třetích osob.
- 5.5. Objednatel se zavazuje poskytnout Zhotoviteli informace a dokumenty nezbytné pro plnění závazků Zhotovitele podle této Smlouvy. Smluvní strany se dohodly, že je výlučně na posouzení Objednatele, zda je k poskytnutí informací či dokumentů třeba uzavření zvláštní smlouvy k převzetí závazku mlčenlivosti. Odmítne-li Objednatel poskytnout informace či dokumenty, které budou dle Zhotovitele potřebné k řádnému splnění předmětu této Smlouvy či k dokončení Sprintu, má Zhotovitel právo od této Smlouvy či od dílčí smlouvy uzavřené na základě této Smlouvy odstoupit.
- 5.6. Objednatel se zavazuje během vývoje Softwaru za účelem předejití škodám pravidelně denně provádět zálohu dat.

## **6. Práva a povinnosti Zhotovitele**

- 6.1. Zhotovitel je povinen při plnění povinností dle této Smlouvy postupovat s veškerou odbornou péčí a v souladu s právními předpisy.
- 6.2. Zhotovitel a jeho zaměstnanci jsou povinni dodržovat bezpečnostní opatření, o kterých budou Objednatelem informováni. Zhotovitel odpovídá za to, že o těchto opatřeních informuje své zaměstnance.
- 6.3. Zhotovitel se zavazuje dbát pokynů Objednatele, které mu Objednatel sdělí. Obdrží-li Zhotovitel od Objednatele pokyn zřejmě nesprávný, upozorní ho na to a splní takový pokyn jen tehdy, když na něm Objednatel trvá. V takovém případě Zhotovitel neodpovídá za škodu vzniklou v důsledku takového pokynu. Není-li možné takový pokyn splnit, má Zhotovitel právo od této Smlouvy či od dílčí smlouvy uzavřené na základě této Smlouvy odstoupit.

## **7. Předání díla**

- 7.1. Zhotovitel se zavazuje Objednatele informovat o připravenosti dodat Sprint. Smluvní strany se zavazují poskytnout si potřebnou součinnost k tomu, aby mohlo dojít k jeho předání.
- 7.2. Zhotovitel před předáním předvede Objednateli, že předávaný Sprint je funkční, bez vad bránících jeho užití k dohodnutým účelům.
- 7.3. Objednatel má 10 pracovních dnů k tomu, aby Software odzkoušel a vytkl Zhotoviteli vady. Nevytkne-li Objednatel v této lhůtě vady, má se zato, že je dílo (Sprint) bez vad.

7.4. Má-li dílo (Sprint) vady, zavazuje se Zhotovitel vady odstranit podle své volby buď ve lhůtě 5 kalendářních dnů ode dne vytknutí vad, nebo v rámci následujícího Sprintu.

## 8. Autorské právo

8.1. Smluvní strany se dohodly, že předáním Sprintu nabývá Objednatel oprávnění užívat Software jakožto předmět autorskoprávní ochrany jakýmkoli zákonem povoleným způsobem, v neomezeném množství i teritoriálním rozsahu a po neomezenou dobu, ledaže je v této Smlouvě uvedeno jinak (dále též jen „Licence“).

8.2. Objednatel není je oprávněn upravovat či jinak měnit Software a spojovat jej s jinými programy.

8.3. Objednatel není oprávněn zasahovat do Softwaru a dalších předaných materiálů, není oprávněn jej rozmnožovat neomezeným způsobem, není oprávněn jej předávat třetím osobám a umožnit jeho použití třetími osobami.

## 9. Odměna

9.1. Objednatel se zavazuje zaplatit Zhotoviteli za dokončení jednoho Sprintu zaplatit ..... Kč (dále jako „Odměna“).

9.2. Zhotovitel je plátcem DPH. Všechny částky uvedené v této Smlouvě jsou uvedeny bez DPH. Ke všem cenám bude připočteno DPH ve výši dle platných právních předpisů platných k datu zdanitelného plnění. Dnem zdanitelného plnění je den akceptace plnění.

## 10. Odpovědnost za škodu

10.1. Zhotovitel plně odpovídá Objednateli za jakoukoliv škodu způsobenou Zhotovitelem Objednateli jakýmkoliv porušením povinnosti Zhotovitele uvedené v této Smlouvě.

10.2. Smluvní strany tímto ujednávají limitaci náhrady škody neúmyslně způsobené Zhotovitelem Objednateli porušením jakékoliv povinnosti Zhotovitele uvedené v této Smlouvě do celkové výše .....Kč.

10.3. Objednatel se tímto výslovně vzdává práva na náhradu škody od Zhotovitele neúmyslně nebo nikoli z hrubé nedbalosti způsobené Zhotovitelem porušením jakékoliv povinnosti Zhotovitele uvedené v této Smlouvě v souvislosti s plněním této Smlouvy.

## 11. Smluvní pokuta

11.1. V případě, že se Objednatel dostane do prodlení se zaplacením Odměny nebo její části podle odstavce 9 této Smlouvy, zavazuje se zaplatit Zhotoviteli smluvní pokutu ve výši 0,5 % z dlužné částky za každý den prodlení až do úplného zaplacení Odměny.

11.2. V případě, že Zhotovitel poruší svou povinnost mlčenlivosti uvedenou v článku 14 této Smlouvy, zavazuje se zaplatit Objednateli smluvní pokutu ve výši .....Kč za každé jedno porušení uvedené povinnosti.

11.3. Zaplacení smluvní pokuty nemá vliv na nárok druhé Smluvní strany na náhradu škody v plném rozsahu.

## 12. Odstoupení od Smlouvy

12.1. Objednatel má právo od této Smlouvy nebo od dílčích smluv o dílo odstoupit v případě, že:

- 12.1.1. Zhotovitel je v prodlení s dokončením Sprintu delším než 10 dní;
- 12.1.2. Zhotovitel vstoupí do likvidace nebo bude na jeho majetek prohlášen soudem konkurz nebo bude zamítnut návrh na vyhlášení konkurzu pro nedostatek majetku nebo zanikne bez likvidace a/nebo bude soudem prohlášen úpadek Zhotovitele a/nebo Zhotovitel vstoupí do insolvence;
- 12.1.3. Zhotovitelem oznámené okolnosti vyšší moci trvají déle než 30 dní a Smluvní strany se nedohodnou jinak;
- 12.2. Zhotovitel má právo od této Smlouvy nebo od dílčích smluv o dílo odstoupit v případě, že:
  - 12.2.1. Objednatel je v prodlení se zaplacením Odměny nebo její části delším než 30 dní;
  - 12.2.2. Objednatel vstoupí do likvidace nebo bude na jeho majetek prohlášen soudem konkurz nebo bude zamítnut návrh na vyhlášení konkurzu pro nedostatek majetku nebo zanikne bez likvidace a/nebo bude soudem prohlášen úpadek Objednatele a/nebo Klient vstoupí do insolvence;
  - 12.2.3. Objednatelem oznámené okolnosti vyšší moci trvají déle než 30 dní a Smluvní strany se nedohodnou jinak;
- 12.3. Kterákoliv Smluvní strana má právo odstoupit od této Smlouvy i z kteréhokoliv zákonného důvodu.
- 12.4. Odstoupení je účinné doručením písemného oznámení o odstoupení druhé Smluvní straně.

### **13. Vypořádání při odstoupení od Smlouvy**

- 13.1. Odstoupení od Smlouvy se nedotýká udělené Licence k předmětu této Smlouvy, Licence tak v případě odstoupení od Smlouvy nadále trvá. Pokud by se ukázalo, že je toto ustanovení Smlouvy z jakéhokoliv důvodu neplatné, zavazuje se Zhotovitel udělit licenci k dodanému Softwaru a Dokumentaci podle této Smlouvy ve stejném rozsahu, v jakém byla Objednateli na základě této Smlouvy udělena.
- 13.2. Jestli některá ze stran odstoupí od Smlouvy, jsou strany povinny vzájemně se vypořádat takto:
  - 13.2.1. Zhotovitel se zavazuje předat veškeré dokončené nebo rozpracované Sprints, včetně veškeré dohodnuté Dokumentace ve stavu, v jakém se ke dni odstoupení nacházejí.
  - 13.2.2. Zhotovitel se zavazuje nejpozději do 30 dnů ode dne odstoupení od Smlouvy provést vyúčtování a sdělit Objednateli část dosud neuhrazené Odměny. Současně je Zhotovitel povinen doložit Objednateli oprávněnost Odměny.
  - 13.2.3. Je-li Odměna stanovena za dokončení Sprintu či Softwaru, provede Zhotovitel odhad míry dokončení Sprintu či Softwaru a vypočítá odpovídající podíl Odměny. Nesouhlasí-li Objednatel s částkou určenou odhadem sdělenou Zhotovitelem, sdělí Objednatel do 30 dnů ode dne obdržení odhadu Zhotovitele svůj odhad Zhotoviteli. Nedojde-li tímto způsobem k dohodě, zavazuje se Objednatel uhradit tuto část Odměny na základě vyčíslení (ocenění) třetí osobou (znalcem), kterého strany společně vyberou. Náklady ocenění hradí strany rovným dílem.

## 14. Důvěrnost

- 14.1. Zhotovitel se zavazuje, že nezpřístupní ani nepoužije žádnou informaci obchodní a/nebo výrobní povahy, se kterou se seznámí v souvislosti s plněním této Smlouvy, zejména nezpřístupní ani nepoužije:
- 14.1.1. žádnou takovou informaci obsaženou v této Smlouvě,
  - 14.1.2. databázi zákazníků Objednatele ani kontakty na ně,
  - 14.1.3. cenovou politiku Objednatele,
  - 14.1.4. marketingovou strategii Objednatele,
  - 14.1.5. informace o uzavřených smlouvách a dodavatelích Objednatele,
  - 14.1.6. způsob fungování podniku Objednatele,
  - 14.1.7. strategická rozhodnutí a podnikatelské záměry Objednatele. (dále jen „**Důvěrná informace**“).
- 14.2. Povinnost mlčenlivosti podle odstavce 14 této Smlouvy platí s výjimkou případů, kdy:
- 14.2.1. Objednatel udělil předchozí písemný souhlas s takovým zpřístupněním nebo použitím Důvěrné informace,
  - 14.2.2. právní předpis nebo veřejnoprávní orgán stanoví povinnost zpřístupnit nebo použít Důvěrnou informaci,
  - 14.2.3. takové zpřístupnění nebo použití Důvěrné informace je nezbytné pro realizaci této Smlouvy,
  - 14.2.4. je to podle jakékoliv smlouvy nebo dohody uzavřené mezi Smluvními stranami dovoleno.
- 14.3. Mezi Důvěrné informace nepatří žádné informace, které jsou v době jejich zpřístupnění nebo použití běžně dostupné veřejnosti.
- 14.4. Objednatel tímto dává Zhotoviteli souhlas k tomu, aby jej Zhotovitel uváděl jako svého zákazníka.
- 14.5. Zhotovitel bere na vědomí, že Důvěrné informace tvoří obchodní tajemství Objednatele.

## 15. Rozhodné právo

- 15.1. Tato Smlouva se řídí právním řádem České republiky, zejména zák. č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů.

## 16. Závěrečná ustanovení

- 16.1. V této Smlouvě, pokud z kontextu jasně nevyplývá jinak, zahrnuje význam slova v jednotném čísle rovněž význam daného slova v množném čísle a naopak, význam slova vyjadřujícího určitý rod zahrnuje rovněž ostatní rody. Nadpisy jsou uváděny pouze pro přehlednost a nemají vliv na výklad této Smlouvy.
- 16.2. Nevymahatelnost či neplatnost kteréhokoliv ustanovení této Smlouvy nemá vliv na vymahatelnost či platnost zbývajících ustanovení této Smlouvy, pokud z povahy nebo obsahu takového ustanovení nevyplývá, že nemůže být odděleno od ostatního obsahu této Smlouvy.
- 16.3. Tato Smlouva představuje úplné ujednání mezi Smluvními stranami ve vztahu k předmětu této Smlouvy a nahrazuje veškerá předchozí ujednání ohledně předmětu této Smlouvy.

- 16.4. Tato Smlouva může být změněna písemnými dodatky podepsanými všemi Smluvními stranami.
- 16.5. Tato Smlouva je vyhotovena v 2 stejnopisech. Každá Smluvní strana obdrží 1 stejnopis této Smlouvy.
- 16.6. Každá ze Smluvních stran nese své vlastní náklady vzniklé v důsledku uzavírání této Smlouvy.
- 16.7. Tato Smlouva nabývá platnosti a účinnosti v okamžiku jejího podpisu všemi Smluvními stranami.
- 16.8. Smluvní strany si tuto Smlouvu přečetly, souhlasí s jejím obsahem a prohlašují, že je ujednána svobodně.

### **NA DŮKAZ ČEHOŽ SMLUVNÍ STRANY PŘIPOJUJÍ SVÉ PODPISY**

V Praze dne

---

V Praze dne

---

Zdroj: Trávníček (2020), upraveno.